

## **Primeiros Elogios ao Use a Cabeça! PHP & MySQL**

“PHP e MySQL são duas das tecnologias de desenvolvimento web mais populares de hoje em dia, e este livro mostra aos leitores o porquê. Criar um site sem elas, atualmente, é tão inimaginável quanto fazer web design sem CSS. Este livro é uma ótima introdução e é também uma boa fonte de risadas. É o livro que eu queria ter tido quando estava aprendendo.”

— **Harvey Quamen, Professor Associado de Inglês e Humanities Computing, Universidade de Alberta**

“Tudo o que nos acostumamos a aceitar como trabalho pesado no processo de aprendizado técnico foi abandonado, e em seu lugar foi criado um incomum método de aprendizado divertido. Eu tenho plena confiança de que a série Use a Cabeça! irá revolucionar a indústria de livros técnicos, e que estes métodos acabarão se tornando o padrão. Aposto que até a minha avó tecnofóbica poderia aprender técnicas de PHP e MySQL após uma única leitura. Ela provavelmente ainda se divertiria ao fazê-lo!”

— **Will Harris, Administrador de Bancos de Dados, Powered By Geek**

“Ler Use a Cabeça! PHP & MySQL é como assistir a uma aula do professor ‘legal’. Faz você ficar com vontade de aprender.”

— **Stephanie Liese, Desenvolvedora Web**

“Usando imagens e humor, o livro é fácil de digerir, e ainda por cima apresenta um sólido conhecimento técnico.”

— **Jereme Allen, Desenvolvedor Web**

“Após uma desafiadora e ultrarrápida primeira leitura, e vários projetos práticos divertidos, como por exemplo “Meu cachorro foi abduzido por alienígenas” e a “Agência de Encontros Mismatch”, não vejo a hora de adicionar um pouco de PHP aos meus sites.”

— **David Briggs, Engenheiro de Software e Escritor de Livros Técnicos**

## **Elogios ao Use a Cabeça! HTML com CSS & XHTML**

“Eric e Elisabeth Freeman claramente entendem do assunto. À medida que a Internet vai se tornando mais complexa, fica cada vez mais importante criar páginas web atraentes. O design elegante é o cerne de todos os capítulos aqui, e cada conceito é apresentado com doses iguais de pragmatismo e inteligência.”

— **Ken Goldstein, Vice-Presidente Executivo & Diretor-Gerente, Disney Online**

“A web seria um lugar muito melhor se todo designer HTML começasse lendo este livro.”

— **L. David Baron, Diretor Técnico de Layout & CSS, Mozilla Corporation,**  
<http://dbaron.org/>

“Eu escrevo HTML e CSS há dez anos, e o que antes era um longo processo de aprendizado por tentativa e erro agora acaba de ser reduzido a um único e atraente livro. HTML costumava ser algo que você simplesmente ficava fuçando até que as coisas ficassem bem na tela, mas com o advento de padrões web e a exigência de melhor acessibilidade, práticas de programação preguiçosas já não são mais aceitáveis... nem do ponto de vista empresarial, nem do ponto de vista da responsabilidade social. Use a Cabeça! HTML com CSS & XHTML lhe ensina como fazer as coisas corretamente desde o princípio, sem fazer com que o processo pareça impossível de aprender. A linguagem HTML, quando corretamente ensinada, não é mais complicada do que aprender inglês, e os Freemans fizeram um excelente trabalho em manter todos os conceitos comprehensíveis.”

— **Mike Davidson, Presidente & CEO, Newsvine Inc.**

“Ah, ótimo. Vocês fizeram um livro de XHTML simples o suficiente para que um CEO consiga entendê-lo. O que farão a seguir? Um livro de contabilidade simples o suficiente para que o meu desenvolvedor consiga entender? Se continuar assim acabaremos tendo de fazer coisas como colaborar e trabalhar em equipe.”

— **Janice Fraser, CEO, Adaptive Path**

“Este livro tem humor, charme, mas o mais importante: tem coração. Eu sei que parece algo ridículo de se dizer sobre um livro técnico, mas eu realmente percebo que, no seu cerne, este livro (ou pelo menos os seus autores) realmente está preocupado em fazer os leitores aprenderem o material. Isso se faz notar no estilo, na linguagem e nas técnicas. O aprendizado – realmente entender e compreender – pelo leitor é claramente o objetivo número um na mente dos Freemans. E obrigado, obrigado, obrigado pela forte e sensível insistência que o livro prega pelo respeito aos padrões. É ótimo ver um livro para iniciantes, o qual acredito, que vai ser amplamente lido e estudado, fazer uma campanha tão eloquente e persuasiva pelo valor do respeito aos padrões na programação de páginas web. Encontrei aqui até mesmo alguns argumentos que ainda não tinham-me ocorrido – argumentos que eu posso me lembrar e usar quando as pessoas me perguntam, e elas perguntam, ‘qual a vantagem de respeitar os padrões, e por que eu devo fazê-lo?’ Agora eu tenho mais munição! Também gostei do fato de o livro apresentar alguns fundamentos sobre a mecânica de como realmente colocar a página no ar – FTP, fundamentos do servidor web, estruturas de arquivos, etc.”

— **Robert Neer, Diretor de Desenvolvimento de Produto, Movies.com**

## **Elogios ao Use a Cabeça! JavaScript**

“Tão prático e útil, e tão bem-explicado. Este livro faz um excelente trabalho de apresentar um iniciante total ao JavaScript, e é mais uma prova da eficiência do método de ensino Use a Cabeça! De todos os outros livros de JavaScript, Use a Cabeça! JavaScript é ótimo para aprender, comparado com outros livros de referência do tamanho de listas telefônicas.”

— **Alex Lee, Estudante, Universidade de Houston**

“Uma excelente opção para o desenvolvedor JavaScript iniciante.”

— **Fletcher Moore, Desenvolvedor e Designer Web, Georgia Institute of Technology**

“Mais um ótimo livro no clássico estilo ‘Use a Cabeça!’”

— **TW Scannell**

“O JavaScript vem sendo há bastante tempo o sistema por trás das páginas web, no lado do cliente, mas vem sendo também mal entendido e mal utilizado. Com Use a Cabeça! JavaScript, Michael Morrison apresenta uma introdução direta e fácil de entender sobre a linguagem, removendo quaisquer mal entendidos que possam ter existido e mostrando como usá-la da forma mais eficiente para melhorar suas páginas web.”

— **Anthony T. Holdener III, Desenvolvedor de aplicações web e autor de Ajax: O Guia Definitivo**

“Uma página web tem três partes – conteúdo (HTML), aparência (CSS) e comportamento (JavaScript). Use a Cabeça! HTML apresentou os dois primeiros, e este livro usa a mesma abordagem divertida e prática para introduzir o JavaScript. A forma divertida com que o livro apresenta o JavaScript, e as muitas formas como ele reforça as informações para que você não as esqueça, fazem deste um livro perfeito para iniciantes usarem para começar a aprender a tornar suas páginas web interativas.”

— **Stephen Chapman, Dono da Felgall Pty Ltd., editor de JavaScript de *about.com***

“Este é o livro que eu estive procurando para recomendar aos meus leitores. É simples o suficiente para totais iniciantes, mas inclui aprofundamentos suficientes para ser útil aos usuários mais avançados. E torna divertido o processo de aprendizado. Este pode acabar sendo o único livro de JavaScript de que você precisará.”

— **Julie L. Baumler, Editora de JavaScript de *BellaOnline.com***

**Outros livros da série Use a Cabeça!**

Use a Cabeça! Java

Use a Cabeça! Análise & Projeto Orientado a Objetos (A&POO)

Use a Cabeça! Ajax Iniciação Rápida

Use a Cabeça! HTML com CSS e XHTML

Use a Cabeça! Padrões de Projeto

Use a Cabeça! Servlets e JSP

Use a Cabeça! PMP

Use a Cabeça! SQL

Use a Cabeça! Desenvolvimento de Software

Use a Cabeça! JavaScript

Use a Cabeça! C#

Use a Cabeça! PHP & MySQL

Use a Cabeça! Física

Use a Cabeça! Álgebra

Use a Cabeça! Ajax Profissional

Use a Cabeça! Estatística

Use a Cabeça! Ruby on Rails

# Use a Cabeça! PHP & MySQL



Não seria fabuloso se houvesse um livro de PHP e MySQL que fizesse os bancos de dados e a programação web no lado do servidor parecerem uma combinação feita no céu? Mas provavelmente é só uma fantasia...

Lynn Beighley  
Michael Morrison



**ALTA BOOKS**  
EDITORIA  
Rio de Janeiro 2010

## **Use a Cabeça! PHP & MySQL**

Copyright © 2010 da Starlin Alta Con. Com. Ltda.

ISBN: 978-85-7608-502-7

### **Produção Editorial:**

Starlin Alta Con. Com. Ltda.

### **Gerência de Produção:**

Maristela Almeida

### **Coordenação Administrativa:**

Anderson Câmara

### **Supervisão de Produção:**

Angel Cabeza

### **Tradução:**

Marcelo Santos

### **Revisão Gramatical:**

Fátima Regina Félix

### **Revisão Técnica:**

Giuliana Cirelli

*Formada em Engenharia Elétrica  
com ênfase em eletrônica.*

*Especializada em Programação  
em linguagem C++ e nos  
ambientes de desenvolvimento  
Eclipse e Visual Studio.*

### **Diagramação:**

Haroldo Sodré

### **Fechamento:**

Luis Rodrigues

*Translated From Original: Head First PHP & MySQL ISBN: 978-0-596-00630-3*

*Use a Cabeça! PHP & MySQL © 2010 Starlin Alta Con. Com. Ltda.  
Authorized translation of the English edition of Head First PHP & MySQL  
© 2009 Lynn Beighley and Michael Morrison. This translation is published  
and sold by permission of O'Reilly Media, Inc., the owner of all rights to  
publish and sell the same. PORTUGUESE language edition published by  
Editora Starlin Alta Con. Com. Ltda. Copyright © 2010 by Editora Starlin  
Alta Con. Com. Ltda.*

Todos os direitos reservados e protegidos pela Lei nº 9.610/98. Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida, sejam quais forem os meios empregados: eletrônico, mecânico, fotográfico, gravação ou quaisquer outros. Todo o esforço foi feito para fornecer a mais completa e adequada informação, contudo a editora e o(s) autor(es) não assumem responsabilidade pelos resultados e usos da informação fornecida. Recomendamos aos leitores testar a informação, bem como tomar todos os cuidados necessários (como o backup), antes da efetiva utilização. Este livro não contém CD-ROM, disquete ou qualquer outra mídia.

**Erratas e atualizações:** Sempre nos esforçamos para entregar a você, leitor, um livro livre de erros técnicos ou de conteúdo; porém, nem sempre isso é conseguido, seja por motivo de mudança de software, interpretação ou mesmo quando alguns deslizes constam na versão original de alguns livros que traduzimos. Sendo assim, criamos em nosso site, [www.altabooks.com.br](http://www.altabooks.com.br), a seção Erratas, onde relataremos, com a devida correção, qualquer erro encontrado em nossos livros.

**Avisos e Renúncia de Direitos:** Este livro é vendido como está, sem garantia de qualquer tipo, seja expressa ou implícita.

**Marcas Registradas:** Todos os termos mencionados e reconhecidos como Marca Registrada e/ou comercial são de responsabilidade de seus proprietários. A Editora informa não estar associada a nenhum produto e/ou fornecedor apresentado no livro. No decorrer da obra, imagens, nomes de produtos e fabricantes podem ter sido utilizados e, desde já, a Editora informa que o uso é apenas ilustrativo e/ou educativo, não visando ao lucro, favorecimento ou desmerecimento do produto/fabricante.

### **Impresso no Brasil**

O código de propriedade intelectual de 1º de julho de 1992 proíbe expressamente o uso coletivo sem autorização dos detentores do direito autoral da obra, bem como a cópia ilegal do original. Esta prática generalizada, nos estabelecimentos de ensino, provoca uma brutal baixa nas vendas dos livros a ponto de impossibilitar os autores de criarem novas obras.



Para os meus pais, que frequentemente usam aplicações web e  
estão sempre me apoiando

- Lynn Beighley

Para Rasmus Lerdorf, que iniciou sozinho a linguagem que  
acabaria se tornando o PHP como nós o conhecemos. Prova  
duradoura de que realmente basta uma pessoa para liderar a  
todos nós em direção a um caminho novo e mais esclarecido.

- Michael Morrison

o(s) autor(es)

## Os Autores de Use a Cabeça! PHP & MySQL

Lynn Beighley



**Lynn Beighley** é uma escritora de ficção presa no corpo de uma escritora de livros técnicos. Após descobrir que escrever livros técnicos realmente dava dinheiro, ela aprendeu a aceitar e desfrutar a profissão. Depois de voltar à escola para obter um Mestrado em Ciência da Computação, trabalhou para os acrônimos NRL e LANL. Depois, descobriu o Flash e escreveu seu primeiro best-seller. Vítima de um timing infeliz, mudou-se para Silicon Valley pouco antes da grande crise. Passou alguns anos trabalhando para o Yahoo! e escrevendo outros livros e cursos de treinamento. Finalmente rendendo-se à sua vocação de escritora, mudou-se para a área de Nova Iorque para obter um MFA em Escrita Criativa. Sua tese, escrita no estilo *Use a Cabeça!*, foi defendida diante de uma sala lotada de professores e colegas estudantes. Foi extremamente bem-recebida, e ela terminou seu curso, terminou *Use a Cabeça! SQL*, e acabou de terminar *Use a Cabeça! PHP & MySQL*. Ufa!

Lynn adora viajar, escrever e inventar histórias detalhadas sobre totais estranhos. Ela tem um pouco de medo de OVNIs.



Michael Morrison

**Michael Morrison** tem sido um contribuinte entusiasmado ao mundo online desde quando mantinha um BBS no seu Commodore 64, na época em que ser nerd era bem menos maneiro do que hoje em dia. Alguns milhares de bauds depois, ele ainda fica fascinado com o progresso que fizemos, e com que rapidez. Michael não tem mais um BBS, mas ainda se mantém bastante envolvido com os equivalentes modernos e com as ferramentas que usamos para criá-los. Ele passa a maior parte do seu tempo “oficial” escrevendo sobre tecnologias relacionadas com a web, tendo escrito ou co-escrito mais de cinquenta livros, a respeito desde programação de jogos para celulares até XML. Entrou no universo *Use a Cabeça!* com *Use a Cabeça! JavaScript* e não parou mais.

Michael é também o fundador da Stalefish Labs ([www.stalefishlabs.com](http://www.stalefishlabs.com)), uma empresa de entretenimento especializada em jogos, brinquedos e mídia interativa. E já foi visto passando tempo fora da Internet (o quê?!) andando de skate, jogando hóquei no gelo e cuidando do seu lago de carpas com sua esposa Masheed. Ele até mesmo dorme, de vez em quando.

# Conteúdo (Sumário)

Introdução	xxvii
1      Está Vivo: Adicionando Vida às Suas Páginas Estáticas	1
2      Como Tudo se Combina: Conectando-se ao MySQL	59
3      Criando os Seus Próprios Dados: Crie e Preencha um Banco de Dados	103
4      Sua Aplicação na Web: Aplicações Realistas e Práticas	159
5      Quando um Banco de Dados Não é o Suficiente: Trabalhando com Dados Armazenados em Arquivos	223
6      Presuma que Estão Todos Querendo Te Pegar: Tornando a Sua Aplicação Segura	295
7      Lembra de Mim?: Criando Aplicações Web Personalizadas	345
7½      Compartilhar é Cuidar: Elimine Código Duplicado	417
8      Colhendo Dados: Controle Seus Dados, Controle Seu Mundo	427
9      Vivendo Melhor Através das Funções: Funções String e Personalizadas	501
10      Regras Para Substituição: Expressões Regulares	561
11      Desenhando Gráficos Dinâmicos: Visualizando seus Dados...e Mais!	605
12      Interfaces com o Mundo: Republicação e Serviços Web	657
i      Os Dez Principais Tópicos (Que Não Abordamos): Sobras	713
ii      Um Lugar para Brincar: Configure um Ambiente de Desenvolvimento	731
iii      Obtenha Ainda Mais Proveito: Amplie Seu PHP	749
Índice Remissivo	755

# Conteúdo (a coisa real)

## Introdução

**Seu cérebro ligado no PHP e MySQL.** Você está tentando aprender alguma coisa, enquanto o seu cérebro está lhe fazendo um favor ao certificar-se de que você não aprenda. Seu cérebro está pensando “é melhor guardar espaço para coisas mais importantes, como por exemplo, quais animais selvagens evitar e se fazer yoga em baixo d’água é uma boa ideia”. Então, como enganar o seu cérebro, fazendo-o pensar que a sua vida depende de aprender PHP e MySQL?

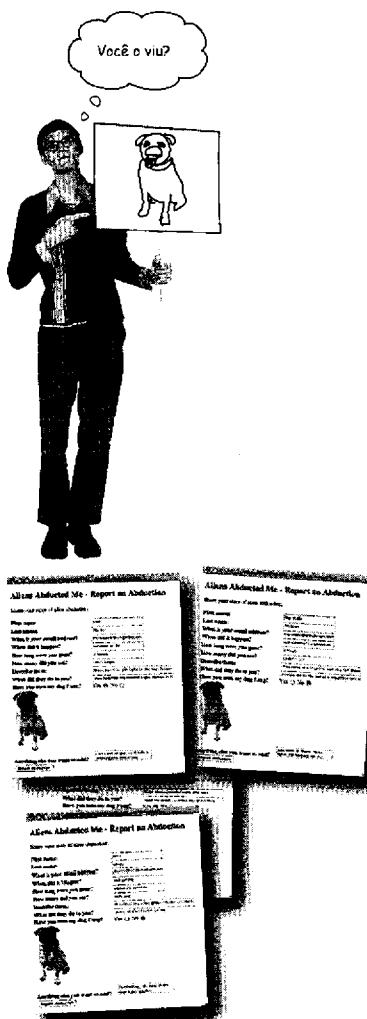
A quem se destina este livro?	xxviii
Sabemos o que você está pensando	xxix
Metacognição	xxxii
Faça seu cérebro lhe obedecer	xxxiii
Leia-me	xxxiv
Revisão técnica	xxxvi
Agradecimentos	xxxvii

# Adicionando Vida às Suas Páginas Estáticas

## 1

### Está Vivo

**Você tem criado ótimas páginas web com HTML, com toques de CSS.** Mas tem percebido que os visitantes do seu site não podem fazer muita coisa além de olhar passivamente o conteúdo das páginas. A comunicação está unidirecional, e você gostaria de mudar isso. Na verdade, o que você realmente gostaria é de saber o que o seu público está pensando. Mas para isso é preciso permitir que os usuários digitem informações em um formulário web, para que possa saber o que eles têm em mente. E também precisa ser capaz de processar essas informações, para que elas lhe sejam entregues. Está começando a parecer que vai precisar de mais do que HTML para levar seu site para o próximo nível.



HTML é estático e chato	2
PHP dá vida às páginas web	3
Um formulário ajuda Owen a descobrir toda a história	5
Formulários são feitos de HTML	6
O formulário HTML apresenta problemas	8
HTML roda no cliente	10
PHP roda no servidor	11
Os scripts PHP rodam no servidor	12
Use PHP para acessar os dados do formulário	16
Os scripts PHP precisam ficar em um servidor!	18
O servidor transforma PHP em HTML	22
Algumas regras PHP para o seu código	25
Encontrando o nome perfeito para a sua variável	26
Variáveis servem para armazenar dados de scripts	31
\$ . POST é uma variável especial que armazena dados do formulário	33
\$ . POST transporta os dados do formulário para o seu script	34
Criando o corpo da mensagem de email com PHP	44
Até mesmo texto simples pode ser formatado... um pouco	46
Newlines precisam de aspas duplas	47
Crie uma mensagem de email para Owen	48
As variáveis armazenam as partes do email	49
Enviando uma mensagem de email com PHP	50
Owen começa a receber emails	53
Owen começa a perder emails	54

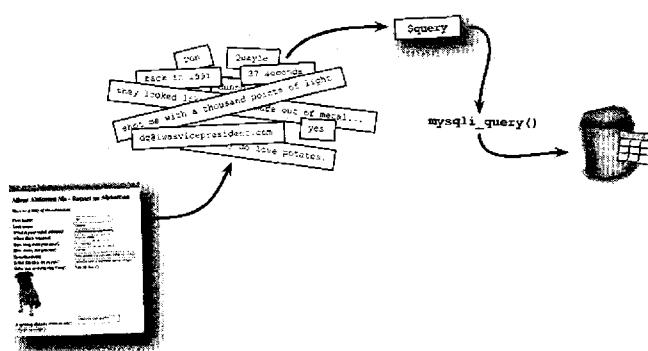
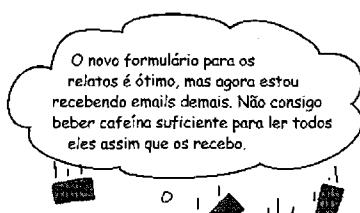
# Conectando-se ao MySQL

2

## Como Tudo se Combina

**Saber como as coisas se encaixam, antes de começar a construir, é uma boa ideia.** Você criou seu primeiro script PHP, e ele está funcionando bem. Mas obter os seus resultados em um email já não é bom o suficiente. Agora precisa de uma forma de guardar os resultados do seu formulário, para que possa mantê-los pelo tempo que precisar, e ter acesso a eles quando quiser. Um banco de dados MySQL pode armazenar seus dados para você. Mas é preciso conectar seu script PHP ao banco de dados MySQL para fazer isso acontecer.

O formulário de Owen funciona bem. Bem até demais...	60
MySQL é excelente para armazenar dados	61
Owen precisa de um banco de dados MySQL	62
Crie um banco de dados e uma tabela MySQL	64
A instrução INSERT em ação	67
Use SELECT para obter dados da tabela	70
Deixe o PHP lidar com as coisas tediosas do SQL	73
O PHP permite usar os dados do formulário web de Owen	74
Conecte-se ao seu banco de dados a partir do PHP	76
Insira dados com um script PHP	77
Use funções PHP para falar com o banco de dados	78
Conecte-se com mysqli_connect()	80
Construa a query INSERT no PHP	85
Consulte o banco de dados MySQL com PHP	86
Encerre sua conexão com mysqli_close()	87
\$_POST fornece os dados do formulário	91
Owen precisa de ajuda para peneirar seus dados	96
Owen sai em busca de Fang	98



## Crie e Preencha um Banco de Dados

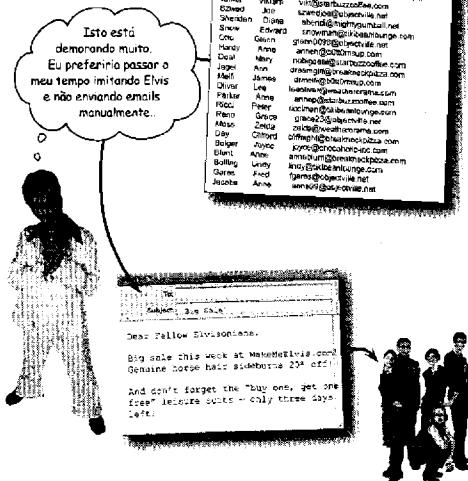
**3**

### Criando os Seus Próprios Dados

#### Nem sempre você tem os dados de que precisa.

Às vezes você mesmo tem que criar os dados que pretende usar, às vezes precisa criar as tabelas que irão armazenar esses dados E às vezes precisa criar o banco que armazenará os dados que precisa criar. Confuso? Você não ficará. Prepare-se para aprender como criar seus próprios bancos de dados e tabelas. E se isso não for o suficiente, ainda criará, no processo, a sua primeira aplicação combinando PHP e MySQL.

Elmer's customer mailing list:	
Anderson	Jillian
W.	Kevin
McDonald	Amanda
Swanson	John
Raudenbush	Johanna
Burgess	Chris
Tobin	Tom
Wells	Andrew
McGinnis	Jo
Ryan	Ashley
McKinney	Clay
McGuire	David
Powers	Brian
Marsden	Annie
Tecchio	Delia
Taylor	Vikram
Cooper	Jim
Snow	Edward
Cox	Gwen
McDowell	Anna
Deacon	Mark
Hightower	Ann
Oliver	James
Foster	Alma
Mac	Peter
Abbot	Paul
Wyatt	Zoe
Ortiz	Christopher
Elmer	John
Bullring	Levi
Flannery	Andy
Jacobs	Anna



A loja Elvis está aberta	104
Elmer precisa de uma aplicação	105
Visualize o projeto da aplicação de Elmer	106
Tudo começa com uma tabela	109
Faça contato com o servidor MySQL	110
Crie um banco de dados para os emails de Elmer	111
Crie uma tabela dentro do banco de dados	112
Precisamos definir os nossos dados	113
Conheça alguns tipos de dados MySQL	114
Crie sua tabela com uma consulta	117
<b>SELECIONE</b> o banco de dados antes de usá-lo	120
<b>DESCRIBE</b> revela a estrutura das tabelas	123
Elmer está pronto para armazenar dados	125
Crie o script "Adicionar Email"	126
O outro lado da aplicação de Elmer	133
O funcionamento interno do script "Enviar Email"	134
Em primeiro lugar, obtenha os dados	135
<code>mysqli_fetch_array()</code> obtém os resultados da consulta	136
Loop para um WHILE	139
Loop através dos dados com while	140
Você tem email... de Elmer!	145
Às vezes, as pessoas querem sair	146
Removendo dados com DELETE	147
Use WHERE e DELETE para apagar dados específicos	148
Minimize o risco de apagamentos acidentais	149
QueroSerElvis.com é uma aplicação web	154

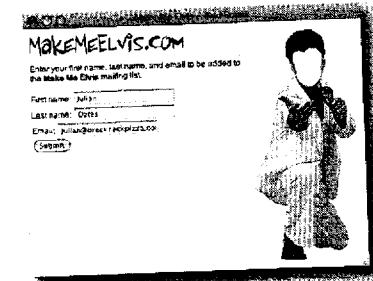
# 4

## Aplicações Realistas e Práticas

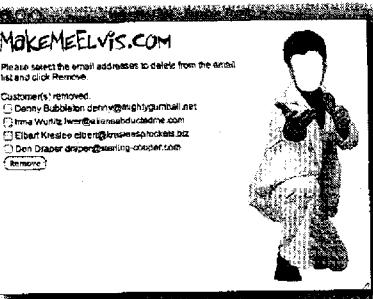
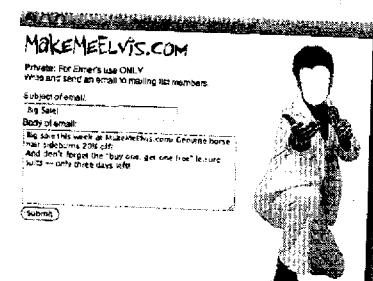
### Sua Aplicação na Web

#### Às vezes você precisa ser realista e repensar seus planos.

Ou então planejar com mais cuidado logo no começo. Uma vez lançada a sua aplicação na Web, você poderá descobrir que não planejou suficientemente bem. Coisas que pensou que funcionariam podem não ser boas o suficiente no mundo real. Este capítulo dá uma olhada em alguns problemas do mundo real que podem ocorrer quando você transfere sua aplicação do ambiente de testes para um site real. E enquanto isso, lhe mostraremos exemplos de códigos PHP e SQL importantes.



Elmer tem alguns clientes irritados	160
Protegendo Elmer de... Elmer	163
Exija bons dados do formulário	164
A lógica por trás da validação de Enviar Email	165
Seu código pode tomar decisões com IF	166
Testando em busca da verdade	167
IF verifica mais do que apenas igualdade	168
A lógica por trás da validação de Enviar Email	171
Funções PHP para verificar variáveis	172
Teste múltiplas condições com AND e OR	179
Os usuários do formulário precisam de feedback	183
Facilite a entrada e a saída do PHP	193
Use um flag para evitar código duplicado	194
Codifique o formulário HTML apenas uma vez	195
Um formulário que referencia a si mesmo	199
Aponte a ação do formulário para o script	200
Verifique se o formulário foi submetido	202
Alguns usuários ainda estão insatisfeitos	206
As linhas da tabela precisam ser identificáveis individualmente	208
As chaves primárias possibilitam a identificação individual	210
Das caixas de verificação até as IDs dos clientes	215
Faça loop através de um array com foreach	216



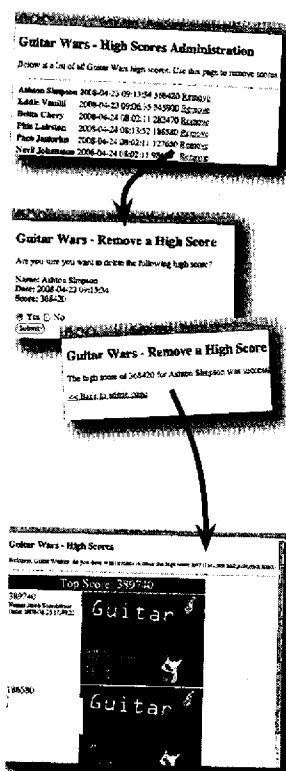
# Trabalhando com Dados Armazenados em Arquivos

5

## Quando um Banco de Dados Não é o Suficiente

### Não acredite no hype... pelo menos não naquele sobre os bancos de dados.

Certamente os bancos são maravilhosos para se armazenar todo tipo de dados que envolvam texto, mas e quanto aos dados binários? Coisas como imagens JPEG e documentos PDF? Faz sentido armazenar todas essas figuras da sua coleção de palhetas raras de guitarra em uma tabela de banco de dados? Geralmente não. Esses tipos de dados normalmente são armazenados em arquivos, e nós os deixaremos nos arquivos. Mas é inteiramente possível ter o melhor de dois mundos – este capítulo revela que você pode usar arquivos e bancos de dados juntos para criar aplicações PHP cheias de dados binários.



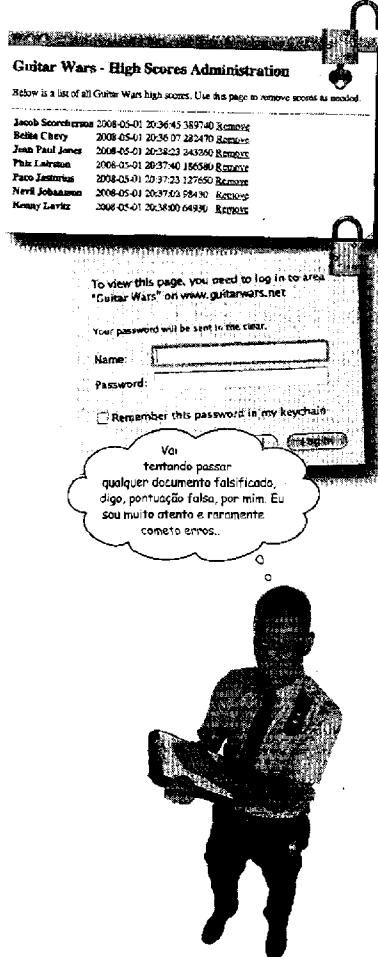
Guitarristas virtuais gostam de competir	224
A imagem é a prova	225
A aplicação precisa armazenar imagens	226
Planejando os uploads de arquivos gráficos no Guitar Wars	231
O banco de dados das pontuações máximas precisa ser ALTERado	232
Como fazer para o usuário enviar a imagem?	236
Insira o (nome do) arquivo gráfico no banco de dados	238
Descubra o nome do arquivo enviado	239
Para onde foi o arquivo enviado?	244
Crie um lar para os arquivos gráficos enviados	248
Dados compartilhados precisam ser compartilhados	254
Serão necessários dados do script compartilhado	255
Pense em require_once como um "inserir"	256
O mais importante para as pontuações máximas é a ordem	258
Homenageando o Guitar Warrior número um	261
Formate a maior pontuação com HTML e CSS	262
São permitidas apenas imagens pequenas	267
A validação de arquivos torna a aplicação mais robusta	268
Planeje uma página Admin	272
Gere links para remoção de pontuações na página Admin	275
Os scripts são capazes de comunicarem-se uns com os outros	276
Sobre GETs e POSTs	278
GET, POST e a remoção de pontuações	280
Isole a maior pontuação para ser removida	283
Controle o quanto você pode remover com LIMIT	284

## 6

# Tornando a Sua Aplicação Segura

## Presuma que Estão Todos Querendo Te Pegar

**Seus pais estavam certos: não fale com estranhos.** Ou pelo menos não confie neles. Na pior das situações, não dê a eles as chaves para acessar os dados da sua aplicação, presumindo que não vão fazer nada de errado. Vivemos num mundo cruel, e você não pode partir do princípio que todos são confiáveis. Na verdade, como desenvolvedor de aplicações web você precisa ser metade incrédulo e metade adepto de teorias de conspiração. Sim, as pessoas em geral são más e elas definitivamente querem te pegar! OK, talvez isso seja um pouco de exagero, mas é muito importante levar a segurança a sério e elaborar as suas aplicações de forma que elas fiquem protegidas contra qualquer um que pretenda causar danos.

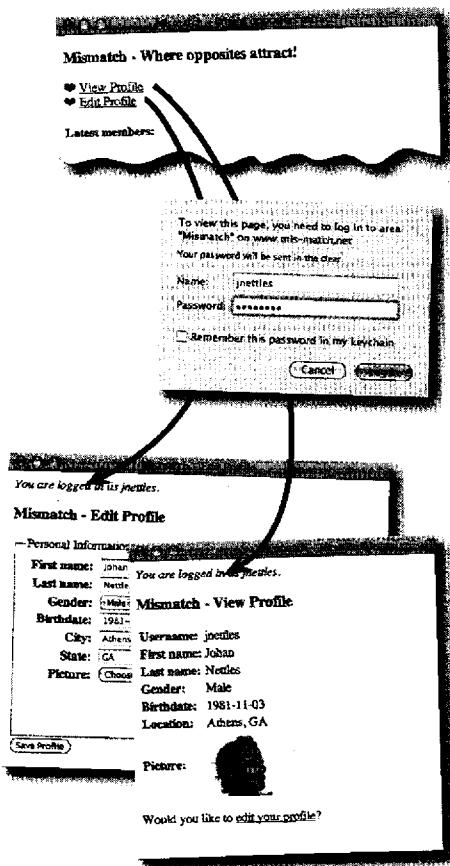


O dia em que a música morreu	296
Para onde foram as pontuações?	297
Segurança contra os bárbaros	299
Protegendo a página Guitar Wars Admin	300
A autenticação HTTP exige cabeçalhos	302
Tudo Sobre o Cabeçalho	304
Controle os cabeçalhos com PHP	305
Autenticando com cabeçalhos	306
Crie um script Autorizar	314
Guitar Wars Episódio II: Ataque dos Clones de Pontuação	318
Subtração por adição	319
A segurança requer intervenção humana	320
Planeje moderação no Guitar Wars	321
Abra espaço para aprovações com ALTER	322
Pontuações não-aprovadas não são dignas	327
O hack de um milhão de pontos	330
Tudo em moderação...?	331
Como exatamente ela fez isso?	333
Enganando o MySQL com comentários	334
O formulário Add Score recebeu uma injeção de SQL	335
Proteja seus dados contra injeções de SQL	336
Um INSERT mais seguro (com parâmetros)	337
A validação de formulários nunca é inteligente demais	339
Cessar fogo!	341

# Criando Aplicações Web Personalizadas

## Lembra de Mim?

**Ninguém gosta de ser esquecido, especialmente usuários de aplicações web.** Se a aplicação tiver que trabalhar a noção de “comunidade”, ou seja, se for planejado que os usuários interajam com a aplicação de uma forma pessoal, então ela terá de se lembrar dos usuários. Você detestaria ter que se apresentar novamente à sua família a cada vez que entrasse em casa. Não precisa fazê-lo porque seus parentes possuem uma coisa maravilhosa chamada memória. Mas aplicações web não se lembram das pessoas automaticamente – é preciso que um desenvolvedor web esperto use as ferramentas disponíveis (PHP e MySQL, talvez?) para criar aplicações web personalizadas que realmente sejam capazes de se lembrar dos usuários.



Dizem que os opositos se atraem	346
O negócio do Mismatch são os dados pessoais	347
O Mismatch precisa dos logins dos usuários	348
Preparando o banco de dados para os logins	351
Construindo uma interface de usuário para o login	353
Criptografe as senhas com SHA()	354
Comparando senhas	355
Autorizando usuários com HTTP	358
Fazendo login dos usuários com autenticação HTTP	361
Um formulário para novos usuários se cadastrarem	365
Do que é feito o cookie?	375
Use cookies com PHP	376
Repensando o fluxo dos logins	379
Um login baseado em cookies	380
Fazer logout significa apagar cookies	385
As sessões não dependem do cliente	389
Mantendo-se atualizado com os dados das sessões	391
Renove o Mismatch com as sessões	392
Faça Logout com sessões	393
Complete a sessão de transformações	398
Os usuários não estão se sentindo bem-vindos	404
As sessões tem vida curta...	406
...mas os cookies podem durar para sempre!	407
Sessões + Cookies = Persistência de login superior	409

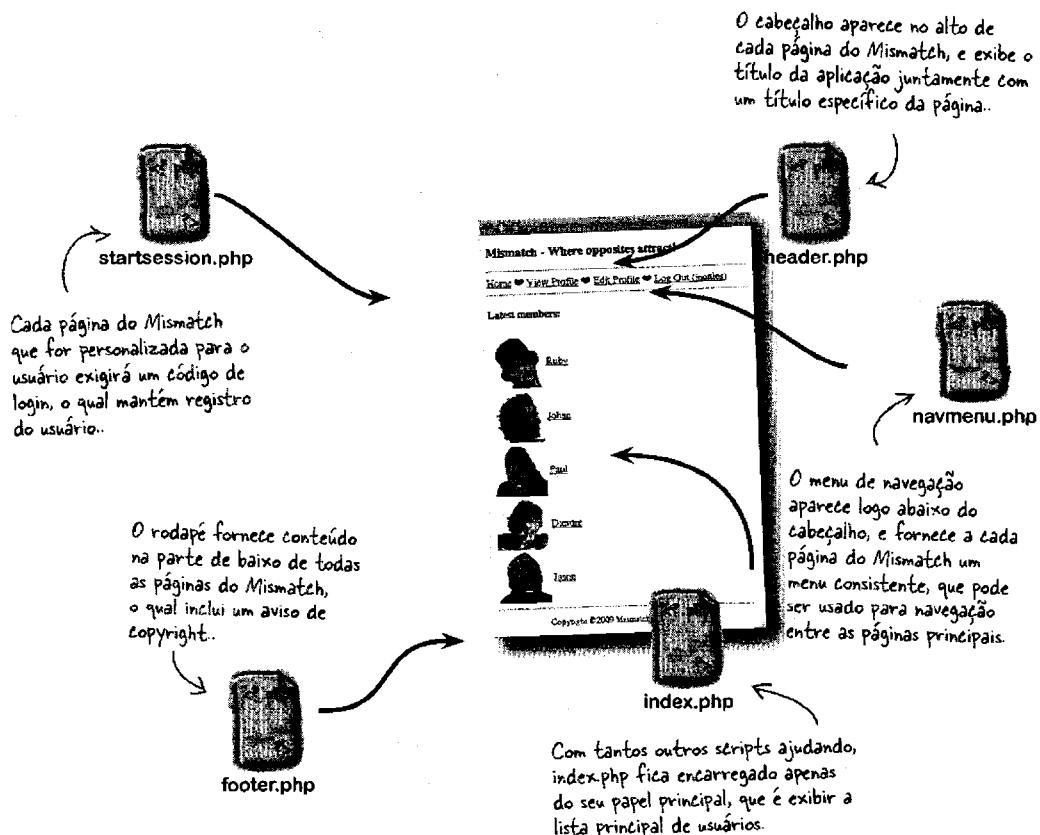
# Elimine Código Duplicado



## Compartilhar é Cuidar

**Guarda-chuvas não são a única coisa que pode ser compartilhada.** Em qualquer aplicação web, você poderá encontrar situações em que um mesmo código encontra-se duplicado em mais de um lugar. Isso não só é um desperdício de recursos, como também pode levar a problemas de manutenção, uma vez que você inevitavelmente acabará fazendo modificações e estas terão de ser efetuadas em mais de um lugar. A solução é eliminar o código duplicado, compartilhando-o. Em outras palavras, você mantém o código duplicado em apenas um lugar, e depois, apenas referencia esse código sempre que precisar dele. A eliminação de código duplicado resulta em aplicativos mais eficientes, de manutenção mais fácil, e finalmente, mais robustos..

O Mismatch está em pedaços	421
Reconstruindo o Mismatch a partir de um template	422
Reconstrua o Mismatch com templates	424
O Mismatch está novamente em pé... e muito melhor organizado	426



# Controle Seus Dados, Controle Seu Mundo

# 8

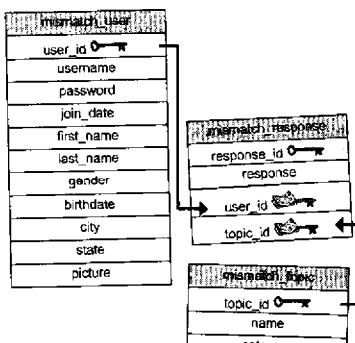
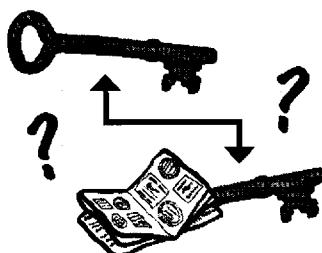
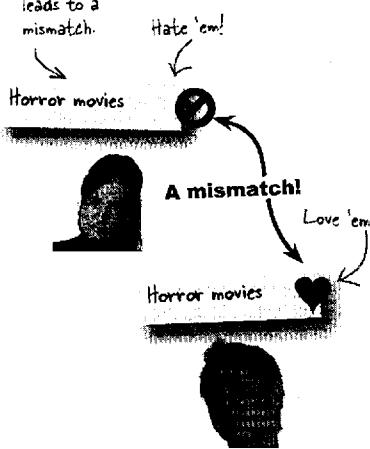
## Colhendo Dados

**Nada como uma boa colheita de dados no outono.** Um sem-fim de informações prontas para serem examinadas, classificadas, comparadas, combinadas, enfim, qualquer coisa que a sua excelente aplicação web precisar que seja feito.

Compensador? Sim. Mas assim como as colheitas na vida real, é preciso muito trabalho duro e uma boa dose de conhecimento para se obter controle sobre os dados em um banco MySQL. Os usuários da web exigem mais do que dados estáticos e enfadonhos.

Eles querem dados enriquecedores... dados compensadores... dados relevantes. Então, o que você está esperando? Dê a partida no seu trator MySQL e mãos à obra!

Sidney's  
dislike of  
horror  
movies  
leads to a  
mismatch.



Fazendo o desencontro perfeito	428
Os desencontros se referem aos dados	429
Modele o banco de dados com um schema	431
Junte várias tabelas	436
Chaves estrangeiras em ação	437
As tabelas podem combinar linha por linha	438
Uma linha leva a muitas outras	439
Checando linhas de muitas-para-muitas	440
Crie um questionário Mismatch	445
Coloque as respostas no banco de dados	446
É possível comandar um formulário com dados	450
Gere o formulário do questionário Mismatch	456
Procure por um pouco de normalidade	462
Ao normalizar, pense em termos de átomos	463
Três passos para se criar um banco de dados normalizado	465
Alterando o banco de dados do Mismatch	469
Então, o Mismatch realmente é normal?	470
Uma consulta dentro de uma consulta dentro de uma consulta...	472
Vamos todos juntar as tabelas	473
Ligue os pontos	474
Certamente podemos fazer mais com os inner joins	475
Apelidos para tabelas e colunas	477
Joins, ao trabalho!	478
Cinco passos para um desencontro com sucesso	485
Compare usuários pela “desencontrabilidade”	487
Tudo o que precisamos é um loop FOR	488

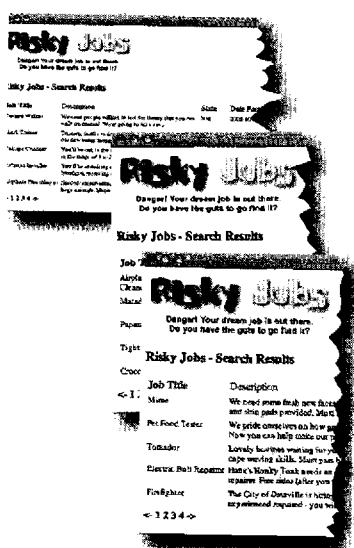
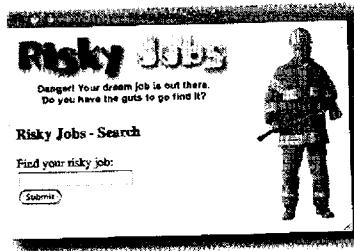
## 9

# Funções String e Personalizadas

## Vivendo Melhor Através das Funções

### As funções levam as suas aplicações para um nível superior.

Você já vem usando as funções internas do PHP para realizar as coisas. Agora é hora de dar uma olhada em mais algumas funções internas realmente úteis. E depois você aprenderá a criar suas próprias funções personalizadas para levá-lo mais além do que sequer imaginava que fosse possível. Bem, talvez não tão longe que você comece a usar espadas laser, mas as funções personalizadas certamente simplificam o seu código e o tornam reutilizável.



Strings e funções personalizadas	501
Um bom trabalho arriscado é difícil de encontrar	502
A busca não deixa margem para erros	504
As consultas SQL podem ser flexíveis com LIKE	505
Transforme uma string em palavras individuais	510
Implode() cria uma string a partir de substrings	513
Pré-processe a string de busca	519
Substitua caracteres de busca indesejados	520
A consulta precisa de termos de busca legítimos	524
Copie elementos não-vazios para um novo array	525
Às vezes você só precisa de parte de uma string	528
Extraia substrings do início ou do fim	529
Podemos classificar nossos resultados com múltiplas consultas	532
As funções lhe permitem reutilizar código	536
Construa uma consulta com uma função personalizada	537
Funções personalizadas: o quanto personalizadas elas realmente são?	538
SWITCH toma muito mais decisões do que IF	542
Dê a build_query() a capacidade de classificar	545
Podemos paginar os nossos resultados	548
Obtenha apenas as linhas de que você precisa com LIMIT	549
Controle os links das páginas com LIMIT	550
Mantenha registro dos dados da paginação	551
Defina as variáveis da paginação	552
Revise a consulta para ter resultados paginados	553
Gere os links de navegação da página	554
Montando o script Search completo	557
O script Search completo, continua...	558

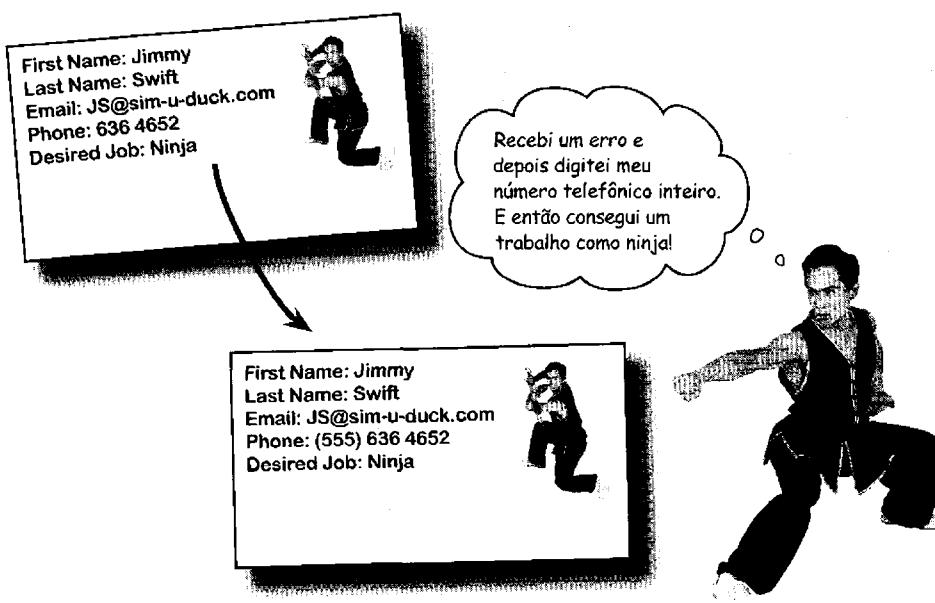
# Expressões Regulares

# 10

## Regras Para Substituição

**As funções string são adoráveis. Mas, ao mesmo tempo, são limitadas.** Certamente elas podem realizar tarefas como lhe dizer a extensão da sua string, truncar a string, trocar certos caracteres por outros. Mas às vezes você precisa de liberdade para lidar com manipulações de texto mais complexas. É aqui que as expressões regulares podem ajudar. Elas podem modificar strings de forma precisa, com base em um conjunto de regras, em vez de um só critério.

A RiskyJobs permite que os usuários submetam currículos	562
Decida como deverá ser a aparência dos seus dados	566
Formule um padrão para números telefônicos	569
Confira padrões com relação a expressões regulares	570
Crie padrões usando metacaracteres	572
Ajuste os padrões com classes de caracteres	579
Verifique os padrões com preg_match()	584
Padronize os dados dos números de telefone	591
Livre-se dos caracteres indesejados	592
Conferir endereços de email pode ser complicado	596
Sufixos de domínios estão por toda a parte	598
Use PHP para verificar o domínio	599
Validação de email: montando o quadro	600



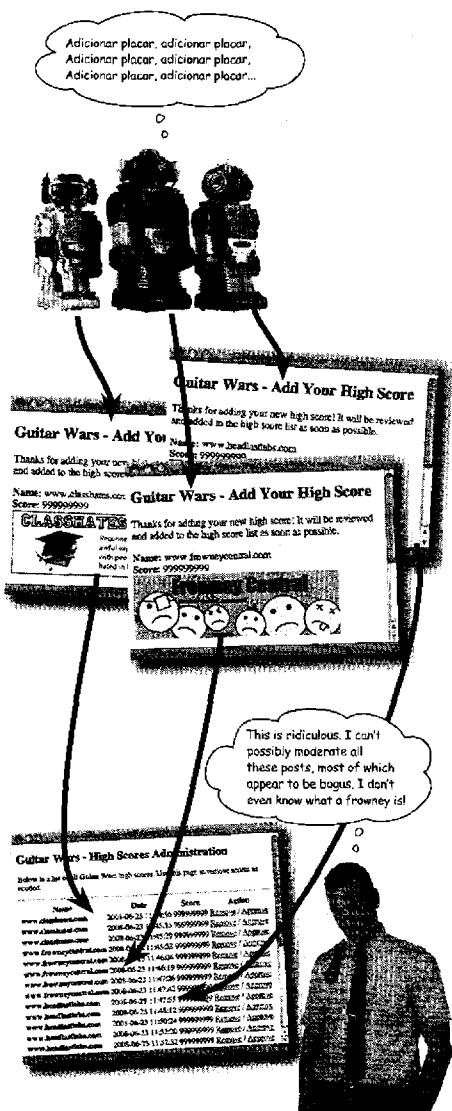
# Visualizando seus Dados... e Mais!

11

## Desenhando Gráficos Dinâmicos

**É claro que todos nós conhecemos o poder de uma boa consulta e os consequentes resultados satisfatórios.**

Mas os resultados de consultas nem sempre falam por si mesmos. Às vezes é útil apresentar os dados de uma forma diferente, uma forma mais visual. O PHP possibilita o fornecimento de uma representação gráfica dos dados: gráficos pizza, gráficos em barra, diagramas de Venn, desenhos Rorschach, qualquer coisa. Vale de tudo para ajudar os usuários a compreender os dados que fluem através da sua aplicação. Mas nem todos os gráficos úteis, em aplicações PHP, se originam do seu banco de dados. Por exemplo, você sabia que é possível evitar ataques de bots de spam com imagens geradas dinamicamente?



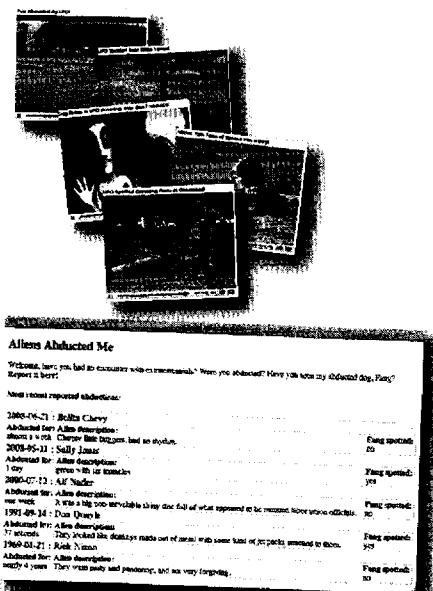
Guitar Wars Reloaded: A Vingança das Máquinas	606
Nenhum formulário de input está a salvo	607
Precisamos separar os humanos das máquinas	608
Podemos vencer a automação usando a automação	611
Gere o texto da senha de CAPTCHA	613
Visualizando a imagem CAPTCHA	614
Por dentro das funções gráficas GD	616
Desenhando texto com uma fonte	620
Gere uma imagem CAPTCHA aleatória	623
A sanidade retorna ao Guitar Wars	625
Adicione CAPTCHA ao script Adicionar Pontuação	627
Cinco graus de oposição	630
Pondo a desencontrabilidade em tabela	631
Armazenando dados do gráfico com barras	632
Lendo entre as linhas com o senhor dos gráficos	635
De um array para outro	636
Crie um array de tópicos desencontrados	638
Formulando um plano para execução de gráficos de barras	639
Amassando as categorias	640
A matemática das categorias	641
Fundamentos dos gráficos de barras	644
Desenhe e mostre a imagem do gráfico de barras	647
Imagens individuais do gráfico de barras para todos	650
Os usuários do Mismatch estão gostando dos gráficos de barras	653

# 12

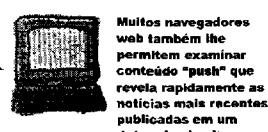
## Repúblicaçāo e Serviços Web

### Interfaces com o Mundo

**Existe um grande mundo lá fora, mundo esse que a sua aplicação web não pode se dar ao luxo de ignorar.** Ou o que talvez seja mais, você é que não gostaria que o mundo ignorasse a sua aplicação. Uma excelente maneira de fazer o mundo notá-la é tornar os seus dados disponíveis para repúblicaçāo, o que significa que os usuários poderão assinar o conteúdo do seu site, em vez de ter de visitá-lo diretamente para encontrar novas informações. Não apenas isso, a sua aplicação pode fazer interfaces com outras aplicações, através dos serviços web, e pode tirar proveito dos dados de outras pessoas para fornecer uma experiência mais rica.



A alguns clientes de email têm suporte a conteúdo "push", permitindo que você receba atualizações do site da mesma forma como recebe mensagens de email.



Até mesmo dispositivos móveis provêm acesso a conteúdo "push", o qual é enviado automaticamente quando algo se modifica no

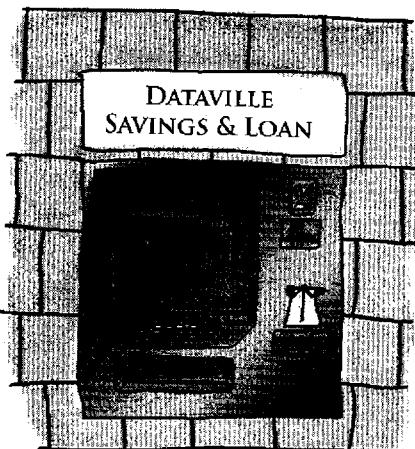
Owen precisa avisar ao mundo sobre Fang	658
Envie dados sobre abduções alienígenas para as pessoas	659
O RSS envia conteúdo web para as pessoas	660
O RSS é na verdade XML	651
Do banco de dados para o news reader	666
Visualizando RSS	669
O que interessa a um comunicador	671
Gere um envio RSS dinamicamente	672
Crie um link para o envio RSS	676
Um vídeo vale mais que mil palavras	678
Obtendo conteúdo web de terceiros	680
Republicando vídeos do You Tube	681
Fazendo uma requisição de vídeo no You Tube	682
Owen está pronto para criar uma requisição REST	686
O You Tube fala XML	690
Deconstruindo uma resposta XML do You Tube	694
Visualize os dados de vídeo XML	695
Acesse os dados XML com objetos	696
De elementos XML para objetos PHP	697
Penetre nos dados XML com objetos	698
Não sem um namespace!	699
Cada vez mais avistamentos de Fang	701
Disponibilize vídeos para visualização	702
Formate os dados de vídeo para exibição	703

## Sobras

### **Os Dez Principais Tópicos (Que Não Abordamos)**

**Mesmo depois de tudo isso, ainda tem mais um pouco** Ainda temos mais algumas coisas que você precisa conhecer. Não seria certo nós as ignorarmos, mesmo que só precisem de uma breve menção. Portanto, antes de guardar o livro, dê uma lida nestes pequenos, mas importantes parágrafos sobre PHP e MySQL. Além disso, uma vez concluído este aqui, só faltarão mais dois pequenos apêndices... e o índice... e talvez algumas propagandas... e aí o livro realmente acaba. Promessa!

#1. Adapte este livro para funções PHP4 e mysql	714
#2. Permissões do usuário no MySQL	716
#3. Relato de erros para o MySQL	718
#4. Erros PHP com tratamento de exceções	719
#5. PHP orientado a objetos	721
#6. Tornando segura a sua aplicação PHP	723
#7. Proteja sua aplicação de scripts cross-site	725
#8. Precedência de operadores	727
#9. Qual é a diferença entre PHP 5 e PHP 6	728
#10. Reutilizando o PHP de terceiros	730



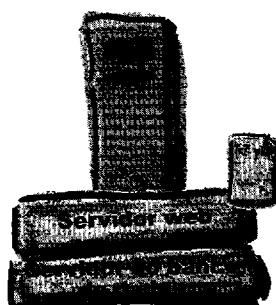
## Configure um Ambiente de Desenvolvimento



### Um Lugar para Brincar

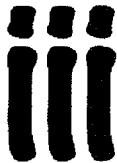
**Você precisa de um lugar para praticar suas recém-adquiridas habilidades em PHP e MySQL, porém sem tornar os seus dados vulneráveis na web.** É sempre uma boa ideia ter um local seguro para desenvolver sua aplicação PHP antes de lançá-la no mundo (ou seja, na web). Este apêndice contém instruções para a instalação de um servidor web, do MySQL e do PHP, para lhe fornecer um lugar seguro onde possa trabalhar e praticar.

#### Servidor



Crie um ambiente de desenvolvimento PHP	732
Descubra o que você tem	732
Você tem um servidor web?	733
Você tem o PHP? Qual versão?	733
Você tem o MySQL? Qual versão?	734
Comece pelo Servidor Web	735
Passos para a instalação do PHP	737
Instalando o MySQL	738
Passos para a instalação do MySQL no Windows	739
Habilitando o PHP no Mac OS X	742
Passos para a instalação do MySQL no Mac OS X	742
Passando do ambiente de produção para um site no ar	744
Dump seus dados (e as suas tabelas)	745
Prepare-se para usar seus dados	745
Mova os dados consolidados para o servidor ativo	746
Conecte-se ao servidor ativo	747

# Amplie Seu PHP



## Obtenha Ainda Mais Proveito

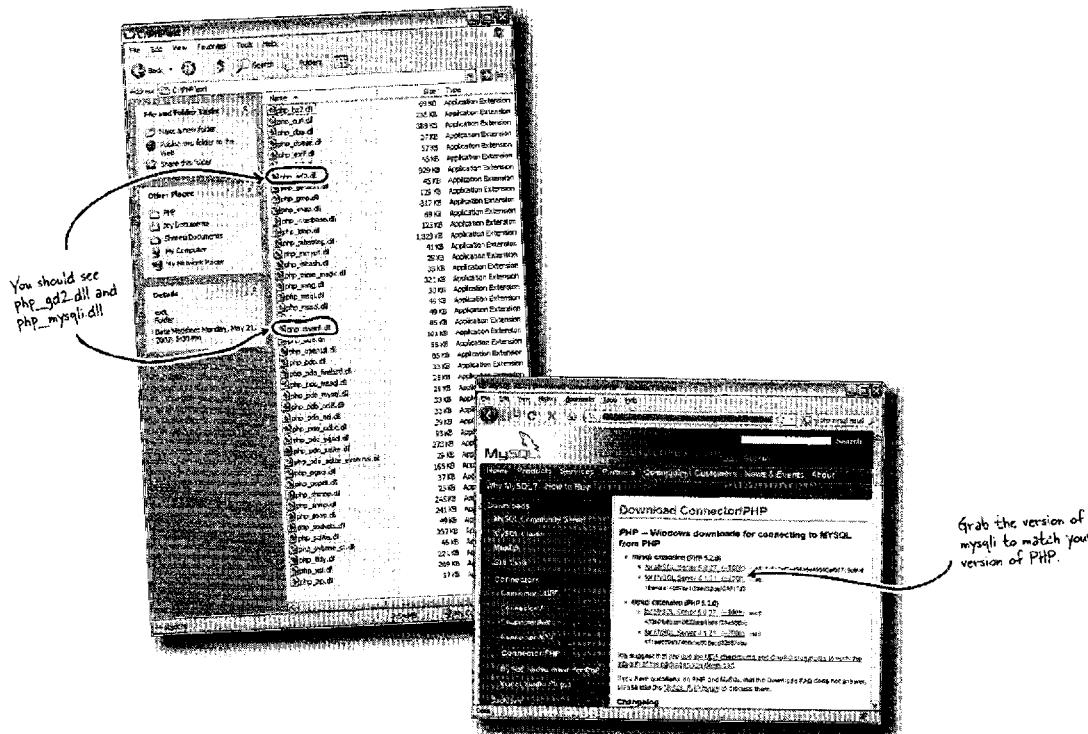
**Sim, você pode programar com PHP e MySQL e criar ótimas aplicações web.** Mas você sabe que ainda deve haver mais a se explorar. E há. Este pequeno apêndice irá lhe mostrar como instalar a extensão mysqli e a extensão biblioteca gráfica GD. Em seguida, mencionaremos mais algumas extensões do PHP que você poderá querer baixar. Porque às vezes não tem problema em querer sempre mais.

Estendendo o seu PHP

750

E no Mac..

753





# Como Usar Este Livro

## Introdução



Nesta seção, respondemos à importante pergunta:  
"Então, por que foi que eles colocaram isso em um  
livro de PHP e MySQL?"

## A quem se destina este livro?

Se você puder responder “sim” a todas estas perguntas:

- ① Você é um web designer com experiência em HTML ou XHTML, e deseja melhorar as suas páginas?
- ② Você deseja ir além do simples HTML e aprender, entender e se lembrar como usar PHP e MySQL para criar aplicações web?
- ③ Você prefere conversas informais e estimulantes, em vez de aulas enfadonhas e acadêmicas?

então este livro é para você.

## Quem provavelmente deve fugir deste livro?

Se você puder responder “sim” a qualquer uma destas perguntas:

- ④ Você não tem conhecimento nenhum sobre conceitos básicos de programação, tais como variáveis e loops?  
(Mas mesmo que jamais tenha programado antes, você provavelmente será capaz de absorver os conceitos-chaves de que precisa, neste livro.)
- ⑤ Você é um super desenvolvedor web PHP procurando por um livro de referência?
- ⑥ Você tem medo de tentar algo diferente? Você prefere arrancar um dente do que misturar listras com xadrez? Você acha que um livro técnico não pode ser sério se um dos exemplos é um banco de dados sobre abduções alienígenas?

então este livro não é para você.



*Nota do departamento do dept. de marketing:  
este livro é para qualquer um que possua um  
cartão de crédito.*

## Sabemos o que você está pensando

"Como isso aqui pode ser um livro de PHP e MySQL sério?"

"Por que tantas figuras?"

"É realmente possível aprender desta forma?"

Seu cérebro  
acha que ISTO é  
importante.

## Sabemos o que o seu cérebro está pensando

Seu cérebro pede novidade. Ele está sempre buscando, vasculhando, esperando por algo fora do comum. Ele foi construído dessa forma, e isso lhe ajuda a se manter vivo..

Assim, o que o seu cérebro faz com todas as coisas rotineiras, comuns, normais, que você encontra? Faz todo o possível para evitar que elas interfiram no trabalho real do cérebro – registrar coisas que sejam importantes. Ele não se dá ao trabalho de armazenar as coisas chatas; elas nunca conseguem passar pelo filtro "isto aqui obviamente não é importante".

Como o seu cérebro sabe o que é ou não é importante? Suponha que você esteja fazendo uma caminhada na selva e um tigre pule na sua frente; o que acontece com a sua mente e o seu corpo?

Os neurônios se acendem. As emoções fluem. A *química dispara*.

E é assim que o seu cérebro sabe que...

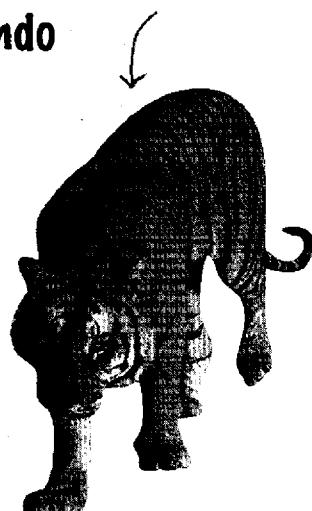
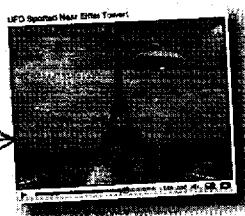
### **Isto deve ser importante! Não se esqueça!**

Mas imagine que você esteja em casa ou em uma biblioteca. É uma área segura, aconchegante, livre de tigres. Você está estudando. Se preparando para uma prova. Ou tentando aprender algum difícil assunto técnico que o seu chefe acha que só vai levar uma semana, dez dias no máximo.

Só um problema. Seu cérebro está tentando lhe fazer um grande favor. Ele está tentando se certificar de que este conteúdo obviamente desimportante não ocupe recursos que são escassos. Recursos esses que são mais bem empregados armazenando-se as coisas realmente importantes. Como tigres. Como o perigo do fogo. Como esconder rapidamente a janela do navegador com o vídeo de discos voadores no You Tube, antes que o seu chefe apareça.

E não há uma forma simples de dizer ao seu cérebro "ei, cérebro, muito obrigado, mas não importa o quanto este livro seja chato, e quão poucas emoções eu esteja sentindo neste exato momento, eu realmente quero que você se lembre destes assuntos."

Vídeos de OVNIs no You  
Tube são obviamente mais  
interessantes para o seu  
cérebro do que um livro  
de TI.



Ótimo. Só mais  
750 difíceis e  
enfadonhas páginas...



Entendemos que o leitor de um livro da série "Use a Cabeça! PHP & MySQL" é um aprendiz.

Então, o que é preciso para se aprender algo? Primeiramente, você precisa entender o assunto, e depois se certificar de que não vai esquecê-lo. Não adianta tentar entulhar fatos dentro da sua cabeça. Com base nas pesquisas mais recentes em ciência cognitiva, neurobiologia e psicologia da educação, é necessário muito mais do que texto em uma página para um verdadeiro aprendizado. Nós sabemos como chamar a atenção do seu cérebro

**Alguns dos princípios de aprendizado Use a Cabeça:**

**Use o recurso visual.** Imagens são muito mais memoráveis do que palavras sozinhas, e tornam o aprendizado muito mais eficiente (até 89% de melhoria em estudos sobre lembrança e transferência). Elas também tornam as coisas mais compreensíveis. **Coloque as palavras dentro ou perto das imagens** às quais elas se referem, em vez de na parte de baixo da página ou na página seguinte, e os estudantes terão duas vezes mais probabilidade de conseguir resolver problemas relacionados ao conteúdo.

**Use um estilo conversacional e personalizado.**

Em estudos recentes, os estudantes tiveram um desempenho até 40% melhor, em testes pós-aprendizado, quando o conteúdo falava diretamente ao leitor, usando um estilo conversacional, em primeira pessoa, do que quando se usava um tom mais formal. Conte histórias, em vez de lecionar. Use linguagem casual. Não se leve muito a sério. Em que você prestaria mais atenção: uma estimulante conversa informal ou uma aula?

Erro!  
Senha desconhecida.

**Faca o estudante pensar um pouco mais profundamente.**

Em outras palavras, a não ser que você ativamente exerçite os seus neurônios, não vai acontecer muita coisa dentro da sua cabeça. O leitor precisa estar motivado, engajado, curioso e inspirado para resolver problemas, chegar a conclusões e gerar conhecimento novo. E, para isso, você precisa de desafios, exercícios e perguntas que provoquem o raciocínio, bem como de atividades que envolvam ambos os hemisférios do cérebro e vários sentidos.



**Obtenha - e mantenha - a atenção do leitor.** Todos nós já passamos por uma situação de "eu realmente quero aprender isto, mas não consigo passar da página um acordado". O seu cérebro presta atenção em coisas que sejam fora do comum, interessantes, estranhas, inesperadas. Aprender um assunto técnico novo e difícil não precisa ser enfadonho. O seu cérebro aprenderá muito mais rapidamente se não for.

**Provoque as emoções.** Nós sabemos que a sua capacidade de se lembrar de algo depende largamente do conteúdo emocional. Você se lembra das coisas com que se importa. Você se lembra de algo quando sente algo. Não, não estamos falando de histórias de partir o coração sobre um garoto e seu cachorro. Estamos falando de emoções como surpresa, curiosidade, diversão, "que ... é essa?", e a sensação de "eu sou fera!" que vem quando você resolve um quebra-cabeças, aprende algo que todo mundo acha que é difícil ou percebe que você sabe algo que o Bob "eu sou mais técnico que vós", do departamento de engenharia, não sabe.

Uma pequena correção. Na verdade, nós temos uma história de partir o coração sobre um garoto e seu cachorro - o cachorro foi abduzido por alienígenas, e você ajudará o menino a encontrá-lo!

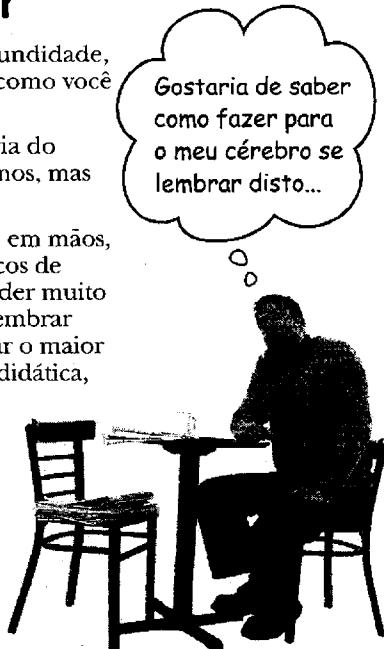
## Metacognição: pensando sobre pensar

Se você realmente quer aprender com mais rapidez e mais profundidade, preste atenção ao modo como prestamos atenção. Pense sobre como você pensa. Aprenda sobre como você aprende.

A maioria das pessoas não faz aulas de metacognição ou de teoria do aprendizado na escola. Sempre se esperou que nós aprendêssemos, mas raramente nos ensinaram a aprender.

Mas nós partimos do princípio de que, se você está com este livro em mãos, é porque realmente quer aprender a criar sites baseados em bancos de dados, usando PHP e MySQL. E provavelmente não quer despende muito tempo. Para poder usar o que você irá ler neste livro, terá de se lembrar do que leu. E para isso, você precisa entender o assunto. Para tirar o maior proveito possível deste livro ou de qualquer livro ou experiência didática, assuma a responsabilidade sobre o seu cérebro. Focalize o seu cérebro neste conteúdo.

O segredo é fazer o seu cérebro ver o novo material que você está aprendendo como sendo Realmente Importante. Crucial para o seu bem-estar. Tão importante quanto um tigre. Caso contrário, você se verá numa batalha constante, com o seu cérebro fazendo o possível para que o novo conteúdo não seja gravado..



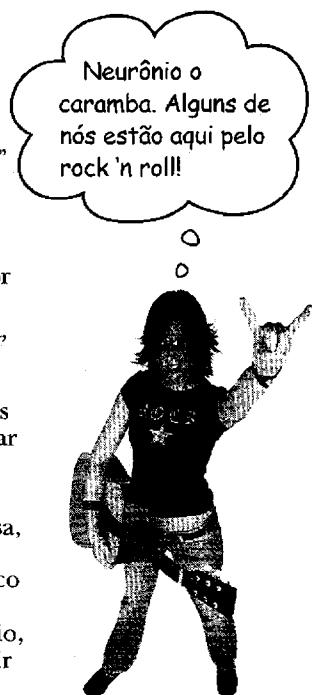
### **Então, como é que você faz com que o seu cérebro trate o “PHP e MySQL” como se ele fosse um tigre faminto?**

Existe a forma lenta e tediosa, e existe a forma mais rápida e mais eficiente. A forma lenta é pura repetição. Você obviamente sabe que é possível aprender e se lembrar até dos assuntos mais chatos, se você continuar empurrando repetidamente a mesma coisa para o seu cérebro. Com suficiente repetição, seu cérebro diz “ele não sente que isto é realmente importante, mas continua olhando as mesmas coisas repetidas e repetidas vezes, então suponho que deve ser importante....”

A forma mais rápida é fazer qualquer coisa que aumente a atividade cerebral, e de preferência, diferentes tipos de atividade cerebral. As coisas listadas na página anterior são uma grande parte da solução, e são coisas que comprovadamente ajudam o seu cérebro a trabalhar em seu favor. Por exemplo, alguns estudos mostram que colocar palavras dentro das figuras que elas descrevem (em vez de colocá-las em algum outro lugar na página, como, por exemplo, em uma legenda ou no corpo do texto) obriga o seu cérebro a tentar entender a relação entre as palavras e a figura, e isto faz mais neurônios se ativarem. Quando mais neurônios se ativam, há maiores chances de o seu cérebro entender que isto é algo a que vale a pena prestar atenção, e possivelmente registrar.

Um estilo conversacional ajuda porque as pessoas tendem a prestar mais atenção quando percebem que estão envolvidas em uma conversa, uma vez que, nessa situação, espera-se que elas acompanhem o que está sendo dito para saber quando e como responder. O mais fantástico é que o seu cérebro não necessariamente liga se a “conversa” é entre você e um livro! Por outro lado, se a escrita tiver um estilo formal e frio, seu cérebro o perceberá da mesma forma como a experiência de ouvir uma lição sentado em uma sala de aula cheia de estudantes passivos. Não é preciso ficar acordado.

Mas as figuras e o estilo conversacional são apenas o início...



## como usar este livro

# Aqui está o que NÓS fizemos:

Nós usamos figuras, porque o seu cérebro está ajustado para prestar atenção em sinais visuais, e não em texto. Pelo menos no que concerne ao seu cérebro, uma imagem realmente vale por mil palavras. E quando texto e imagens precisam trabalhar juntos, nós colocamos o texto dentro das figuras, porque o seu cérebro funciona de forma mais eficiente quando o texto se encontra dentro da coisa a que ele se refere, em vez de em uma legenda ou escondido em algum lugar dentro do texto.

Nós usamos redundância, dizendo a mesma coisa de formas diferentes e com diferentes tipos de mídia, e visando a múltiplos sentidos, para aumentar a chance de o conteúdo ser registrado em mais de uma área do seu cérebro.

Nós usamos conceitos e figuras de formas inesperadas, porque o seu cérebro está ajustado para prestar atenção em novidades, e usamos figuras e ideias com pelo menos algum conteúdo emocional, porque o cérebro está ajustado para prestar atenção à bioquímica das emoções. Aquilo que nos faz sentir algo é mais provável de ser lembrado, mesmo que esse sentimento não seja nada mais que um pouco de humor, surpresa ou interesse.

Nós usamos um estilo personalizado, conversacional, porque o seu cérebro está ajustado para prestar mais atenção quando ele acredita que você está envolvido em uma conversação do que quando pensa que está passivamente ouvindo uma apresentação. Seu cérebro faz isso mesmo quando você está lendo.

Nós incluímos mais de 80 atividades, porque o seu cérebro está ajustado para aprender e se lembrar mais quando você faz algo do que quando lê sobre algo. E fizemos os exercícios desafiadores, porém resolvíveis, porque isso é o que a maioria das pessoas prefere.

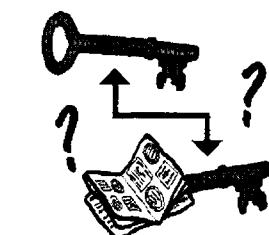
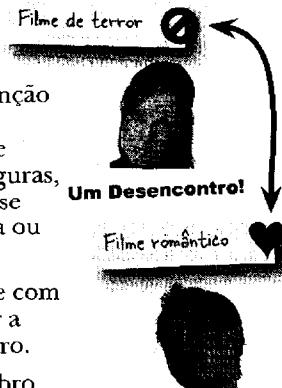
Nós usamos vários estilos de aprendizado, porque você poderá preferir procedimentos passo-a-passo, enquanto que outras pessoas poderão querer entender o quadro geral primeiro, e outras ainda só querem ver um exemplo. Mas independentemente do seu método preferido de aprendizado, todo mundo se beneficia em ver o mesmo conteúdo representado de diversas formas.

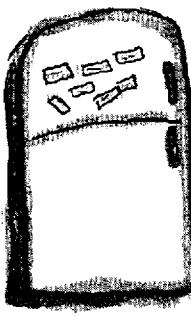
Nós incluímos conteúdo para ambos os hemisférios do seu cérebro, porque quanto mais partes do seu cérebro você utiliza, maior a probabilidade de você aprender e se lembrar, e maior o tempo durante o qual conseguirá se manter focado. Uma vez que trabalhar um lado do cérebro frequentemente significa que o outro lado terá um tempo para descansar, você poderá ser mais produtivo nos estudos, por um período de tempo maior.

Nós incluímos estórias e exercícios que apresentam mais de um ponto de vista, porque o seu cérebro está ajustado para aprender com mais profundidade quando é forçado a fazer avaliações e julgamentos.

Nós incluímos desafios, com exercícios e fazendo perguntas que nem sempre tem uma resposta direta, porque o seu cérebro está ajustado para aprender e se lembrar quando precisa trabalhar em algo. Pense no seguinte: não é possível fazer seu corpo ficar em forma apenas olhando as pessoas malhando na academia. Mas nós fizemos nosso melhor para garantir que, quando estiver trabalhando duro, será sempre nas coisas certas, para que não tenha que gastar um neurônio sequer a mais do que o necessário processando um exemplo difícil de entender ou tentando decifrar um texto difícil, cheio de termos técnicos obscuros ou sisudo demais.

Nós usamos pessoas. Nas estórias, exemplos, figuras, etc., porque, bem, porque você é uma pessoa. E o seu cérebro presta mais atenção nas pessoas do que nas coisas.





# Veja o que fazer para que o seu cérebro se curve em sinal de submissão

Ok, fizemos a nossa parte. O resto é com você. Estas dicas funcionam como um ponto de partida; ouça o seu cérebro e descubra o que funciona e o que não funciona para você. Experimente coisas diferentes.

*Recorte esta página e cole na porta da sua geladeira.*

## ● Vá devagar. Quanto mais você entende, menos você tem que memorizar.

Não leia, simplesmente. Pare e pense. Quando o livro lhe apresentar uma pergunta, não pule direto para a resposta. Imagine que uma pessoa real está lhe fazendo a pergunta. Quanto mais aprofundadamente você forçar o seu cérebro a pensar, maior a sua chance de realmente aprender e se lembrar do que aprendeu.

## ● Faça os exercícios. Faça suas próprias anotações.

Nós elaboramos os exercícios, mas se os fizéssemos para você, isso seria como ir à academia e ter alguém para malhar por você. E não basta só olhar os exercícios. Use um lápis. Há muita evidência de que a atividade física durante o processo de aprendizado pode melhorar este último.

## ● Leia as seções “Não Existem Perguntas Idiotas”.

Todas elas. Elas não são comentários opcionais fazem parte do conteúdo central! Não as deixe de lado.

## ● Que isso seja a última coisa que você lê antes de dormir. Ou pelo menos a última coisa desafiante

Parte do aprendizado (especialmente a transferência para a memória de longo prazo) ocorre depois que você fecha o livro. Seu cérebro precisa de algum tempo para realizar todo o processamento. Se você tentar absorver novas informações durante esse tempo, uma parte do que você acabou de aprender será perdida.

## ● Beba água. Em grande quantidade.

Seu cérebro funciona melhor quando envolvido em um bom banho. A desidratação (que pode acontecer antes mesmo de você sentir sede) prejudica as funções cognitivas.

## ● converse sobre o que está lendo. Em voz alta

Falar ativa uma parte diferente do cérebro. Se você quer entender algo ou aumentar as suas chances de se lembrar do assunto depois, pronuncie-o em voz alta. Melhor ainda, tente explicá-lo, em voz alta, a outra pessoa. Você aprenderá mais rapidamente e poderá descobrir ideias que não tinha percebido ainda durante a leitura.

## ● Ouça o seu cérebro.

Preste atenção para saber se o seu cérebro está ficando sobrecarregado. Se você perceber que está começando a ler superficialmente ou esquecer o que acabou de ler, é hora de fazer uma pausa. Uma vez que você passa de certo ponto, não é possível aprender mais rápido tentando absorver mais e mais conteúdo e poderá até prejudicar o processo.

## ● Sinta algo.

Seu cérebro precisa saber que isto é importante. Envolve-se com as estórias. Crie as suas próprias legendas para as fotos. Resmungar sobre uma piada ruim ainda é melhor do que não sentir nada.

## ● Escreva bastante código.

Só existe uma forma de aprender a programar: escrevendo um monte de código. E é isso que você irá fazer ao longo deste livro. Programar é uma habilidade, e a única forma de se ficar bom nela é praticando. Nós iremos lhe oferecer muitas oportunidades de praticar: cada capítulo terá exercícios que apresentam problemas para você resolver. Não deixe de fazê-los – uma boa parte do aprendizado acontece quando você resolve os exercícios. Nós incluímos soluções – não tenha medo de dar uma olhada na solução caso não consiga resolver algum exercício! (é comum um pequeno detalhe atrapalhar toda a solução) Mas tente resolver o problema antes de olhar as soluções. E, definitivamente, faça o exercício funcionar antes de passar para a parte seguinte do livro.

O PHP e o MySQL lhe permitem criar aplicações web reais – não se esqueça de enviá-las e experimentá-las em um servidor web verdadeiro..

## **Leia-me**

Isto é uma experiência de aprendizado, e não um livro de referência. Nós deliberadamente retiramos tudo que pudesse atrapalhar o aprendizado de qualquer que seja o assunto tratado em um determinado ponto do livro. E na primeira leitura, você precisa começar pelo começo, porque cada capítulo assume que você aprendeu o que viu no anterior.

### **Nós começamos ensinando alguns conceitos simples de programação e fundamentos de conexão a banco de dados, em seguida apresentamos funções PHP e instruções MySQL mais complexas, e finalmente mostramos alguns conceitos mais complexos sobre aplicações.**

Embora seja importante permitir aos usuários adicionar dados e a retirar dados de uma aplicação web, antes disso, é preciso primeiramente, entender a sintaxe tanto do PHP quanto do MySQL. Assim, nós começamos apresentando instruções PHP e MySQL que você pode experimentar na prática. Assim, você poderá imediatamente fazer algo com PHP e MySQL, e começará a se sentir estimulado em usar essas tecnologias. Depois, mais adiante no livro, lhe mostraremos as boas práticas de projeto de aplicações e de bancos de dados. Nesse ponto, já terá um sólido domínio da sintaxe de que vai precisar e poderá se concentrar em aprender os conceitos.

### **Nós não abordaremos todas as instruções, funções ou palavras-chaves do PHP ou MySQL**

Teria sido possível colocar cada instrução, função e palavra-chave do PHP e do MySQL neste livro, mas nós achamos que iria preferir um livro que não fosse do tamanho de um catálogo telefônico, e lhe ensinasse as instruções, funções e palavras-chaves mais importantes. Apresentaremos aquelas que você precisa conhecer, e que usará em 95% do tempo. E após terminar este livro, terá a confiança para procurar sozinho informações sobre aquela função de que precisa para terminar a aplicação que acabou de escrever.

*Na verdade, você pode usar o PHP 4 com este livro, fazendo algumas modificações ao código. Consulte o item I do Apêndice i.*

### **Nós estamos abordando o PHP 5 e o MySQL 5.0.**

Devido ao fato de muita gente ainda usar o PHP 4 ou 5, nós evitamos usar códigos específicos para o PHP 4, 5 ou 6 sempre que possível. Sugerimos que você use o PHP 5 ou 6 e o MySQL 5 ou 6 ao estudar os conceitos deste livro. No desenvolvimento deste livro, nos concentramos no PHP 5 e no MySQL 5, ao mesmo tempo nos certificando de que o código fosse compatível com versões posteriores.

### **Você precisa de um servidor web com suporte a PHP.**

O PHP precisa ser executado através de um servidor web para funcionar corretamente. Você precisa do Apache ou de algum outro servidor web instalado na sua máquina local, ou em alguma máquina à qual você tenha algum acesso, para que possa executar comandos MySQL sobre os dados. Consulte os Apêndices ii e iii para instruções sobre como instalar e estender o PHP e o MySQL.

## Nós usamos o MySQL.

Embora exista uma linguagem SQL Padrão, neste livro nós usaremos a sintaxe particular do MySQL. Com apenas algumas poucas modificações na sintaxe, o código deste livro deverá funcionar no Oracle, MS SQL Server, PostgreSQL, DB2 e em mais alguns outros Sistemas de Gerenciamento de Bancos de Dados Relacionais (Relational Database Management Systems ou RDBMSs) existentes. Você terá de pesquisar as funções e a sintaxe particulares do PHP, para se conectar a esses outros RDBMSs. Se fôssemos abordar todas as variações de sintaxe para cada comando, este livro teria muito mais páginas do que tem. Nós gostamos das árvores, por isso nos limitamos ao MySQL.

## As atividades NÃO são opcionais.

Os exercícios e atividades não são opcionais; fazem parte do conteúdo central do livro. Alguns deles servem para ajudar a memorização, outros para o entendimento e alguns irão lhe ajudar a pôr em prática o que acabou de aprender. Não deixe de fazê-los. As palavras cruzadas são a única coisa que não precisa fazer, mas são boas para dar ao seu cérebro uma chance de pensar de uma forma diferente nas palavras e termos que você está aprendendo.

## A redundância é intencional e importante.

Uma diferença importante de um livro Use a Cabeça! é que nós queremos que você realmente entenda o assunto. E queremos que termine este livro lembrando-se do que aprendeu. A maioria dos livros de referência não tem a retenção e a recuperação como alvo, mas este é um livro de aprendizado, e portanto, verá alguns conceitos aparecendo mais de uma vez.

## Os exemplos são os mais simplificados possíveis.

Nossos leitores nos dizem que é frustrante ter que navegar por 200 linhas de um exemplo, procurando pelas únicas duas linhas que eles precisam entender. A maioria dos exemplos deste livro é mostrada dentro do menor contexto possível, para que a parte que você está tentando aprender fique clara e simples. Não espere que todos os exemplos sejam super robustos ou sempre completos – eles foram escritos especificamente para o propósito didático, e não são necessariamente 100% funcionais.

Nós colocamos todos os códigos de exemplo e aplicações na Web, para que você possa copiar e colar partes deles no seu editor de texto ou Terminal MySQL ou então enviá-los como estão para o seu próprio servidor web, para realizar testes. Você os encontrará em

<http://www.altabooks.com.br>

Vários dos exemplos, porém, são aplicações web completas, capazes de fazer coisas bem poderosas..

Os endereços de websites de terceiros podem ser alterados ou desatualizados a qualquer momento pelos seus mantenedores, sendo assim, a Alta Books não controla ou se responsabiliza por qualquer conteúdo de websites.

## Os exercícios Poder do Cérebro não tem respostas.

Para alguns deles, não existe uma única resposta; para outros, parte da experiência didática das atividades Poder do Cérebro é você decidir se e em que contexto as suas respostas estão corretas. Em alguns dos exercícios Poder do Cérebro, você encontrará dicas para lhe guiar na direção correta.

*equipe de revisão*

## Revisão técnica

**Jereme Allen**



**David Briggs**



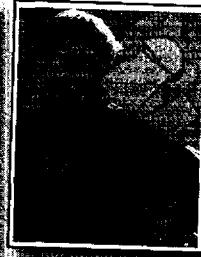
**Will Harris**



**Stephanie Liese**



**Steve Milano**



### *Revisores Técnicos:*

**Jereme Allen** é um desenvolvedor web sênior com experiência no uso das tecnologias mais avançadas para a criação de aplicações web. Ele tem mais de nove anos de experiência no uso de PHP, MySQL, bem como vários outros frameworks, sistemas operacionais, linguagens de programação e software de desenvolvimento.

**David Briggs** é um autor de livros técnicos e engenheiro de localização de software que vive em Birmingham, Inglaterra. Quando não está sendo detalhista sobre como guiar os usuários através de algum software particularmente difícil, não tem nada que ele goste mais do que ir ao parque local com sua esposa, Paulette e Cleo, o cachorro da família.

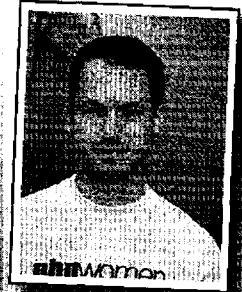
**Will Harris** é o responsável por um departamento de TI que fornece serviços para 11 empresas em 4 continentes, e é o Vice-Presidente da filial em Las Vegas do PASS (Professional Association for SQL Server). À noite, ele entra numa cabine telefônica, coloca o seu uniforme web 2.0, e sai por aí ajudando os designers e desenvolvedores do Powered By Geek a garantir que as suas plataformas de dados permaneçam flexíveis, portáteis, de fácil manutenção e RÁPIDAS, usando MySQL e Rails. Ele também gosta de passar seu tempo com sua esposa, Heather, suas lindas filhas Mara e Ellie, e seu cachorro Swiper.

**Stephanie Liese** é uma instrutora técnica e desenvolvedora web em Sacramento, Califórnia. Quando ela não está pregando as virtudes do código que obedece aos padrões ou removendo bugs de algum layout CSS, você a encontrará suando em uma pesada aula de yoga.

**Harvey Quamen**



**Chris Shiflett**



Quando **Steve Milano** não está escrevendo código para o The Day Job™ ou tocando punk rock com sua banda, a Onion Flavored Rings, em algum porão mal ventilado, ele provavelmente está em casa com seu laptop, negligenciando a companhia felina de Ralph ou a companhia humana de Bianca.

**Harvey Quamen** desistiu de uma carreira em programação de computadores para se juntar ao chique e glamouroso mundo universitário. Atualmente, é Professor Associado de Inglês e Humanities Computing na Universidade de Alberta, onde dá aulas sobre cybercultura, literatura do século XX e desenvolvimento web – incluindo PHP e MySQL.

**Chris Shiflett** é o Chefe de Tecnologia da OmniTI, onde lidera a prática de segurança de aplicações web e guia a iniciativa de desenvolvimento web. Chris é um reconhecido líder nas comunidades de segurança do PHP e de aplicações web – um blogueiro bastante lido em shiflett.org, um popular palestrante em conferências no mundo todo, e o fundador do PHP Security Consortium. Seus livros incluem Essential PHP Security (O'Reilly) e HTTP Developer's Handbook (Sams).

# Agradecimentos

## Nossos editores:

Muito obrigado a Brett McLaughlin pela incrível sessão de storyboarding que nos colocou no caminho certo, e pelo seu compromisso inquebrantável com o aprendizado cognitivo.

Este livro não existiria se não fosse pelo esforço, paciência e persistência heróicos de Sanders Kleinfeld. Ele sempre conseguia apanhar os pinos com os quais estávamos fazendo malabarismo, toda vez que inevitavelmente deixávamos cair um (ou três!), e somos gratos por isso. Esperamos que ele tenha a chance de descansar por alguns dias antes de encarar um outro projeto tão difícil quanto este.

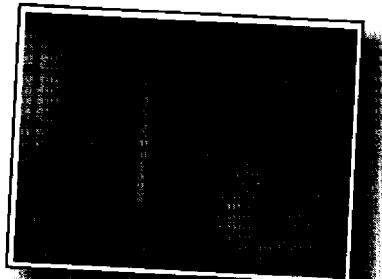
## A equipe O'Reilly:



Lou Barr

Obrigado a Lou Barr pelas suas fenomenais habilidades em design, que fizeram este livro ter uma aparência tão boa.

Obrigado também a Brittany Smith por todo o seu trabalho em cima da hora e a Caitrin McCollough por colocar no ar os sites de exemplo. E a Laurie Petrycki por acreditar que nós poderíamos escrever mais um ótimo livro da série Use a Cabeça!



Brett McLaughlin



Sanders Kleinfeld

## E também:



Finalmente, muitíssimo obrigado a Elvis Wilson por elaborar os vídeos de alienígenas no You Tube para o Capítulo 12. Ótimo trabalho! Especialmente considerando que ele é um simples diretor de arte do tempo das cavernas.



# 1 Adicionando Vida às Suas Páginas Estáticas

Está Vivo

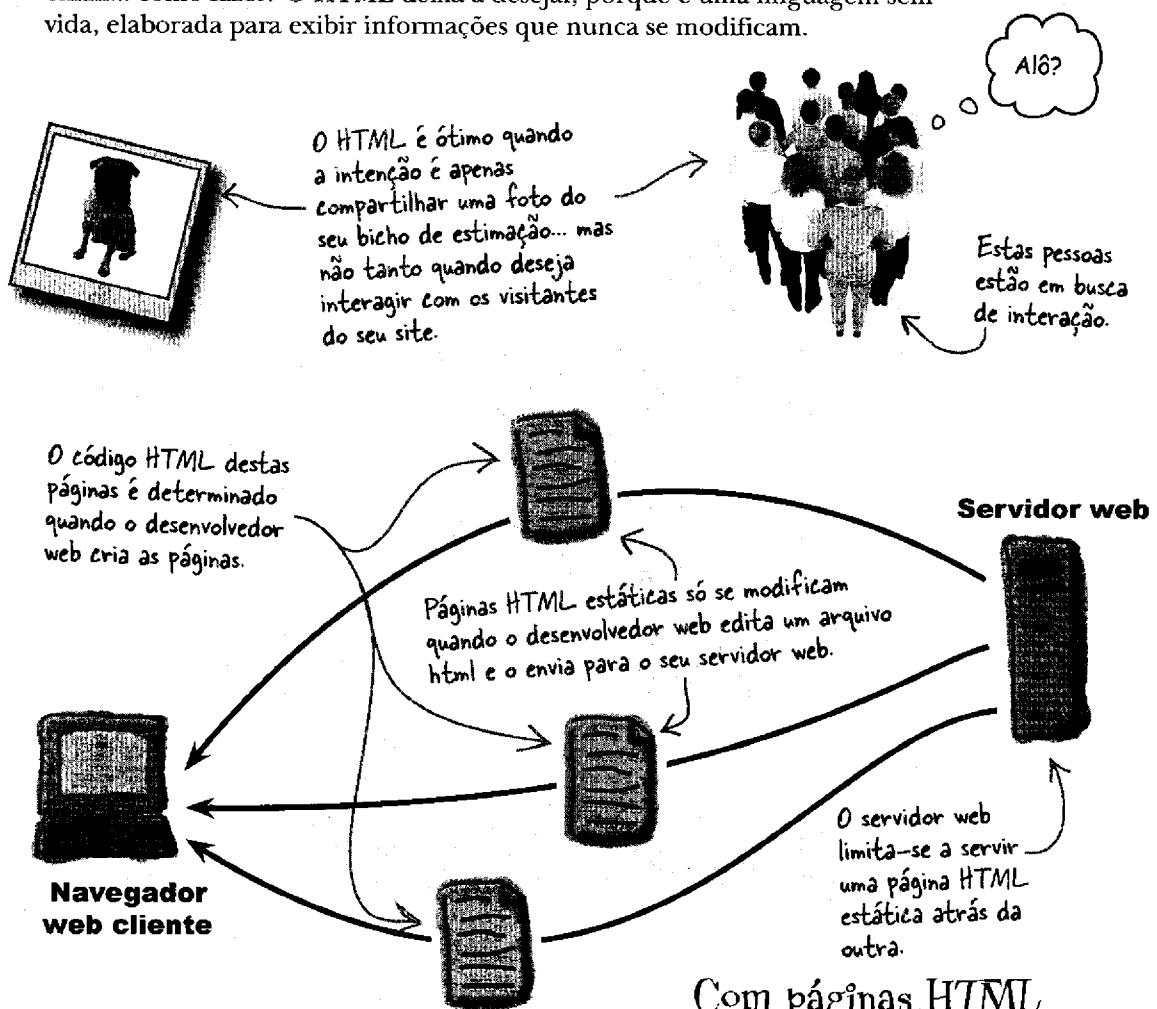


Você tem criado ótimas páginas web com HTML, com recursos de CSS mas tem percebido que os visitantes do seu site não podem fazer muita coisa além de olhar passivamente o conteúdo das páginas. A comunicação apresenta-se de forma unidirecional e você gostaria de mudar isso. Na verdade, o que você realmente gostaria é de saber **o que o seu público está pensando**. Mas para isso precisa permitir que os usuários digitem informações em um formulário web, para que possa saber o que eles têm em mente. E seu site precisa ser capaz de **processar essas informações**, para que elas lhe sejam entregues. Está começando a parecer que você irá precisar de mais do que HTML para levar seu site para o próximo nível.

as vezes somente HTML não é suficiente

## HTML é estático e chato

O HTML é ótimo para se criar páginas web, até aí você já sabe. Mas, e quando você precisa que as páginas realizem alguma ação? Suponha que seja necessário fazer uma busca em um banco de dados ou enviar um email... como fazer? O HTML deixa a desejar, porque é uma linguagem sem vida, elaborada para exibir informações que nunca se modificam.



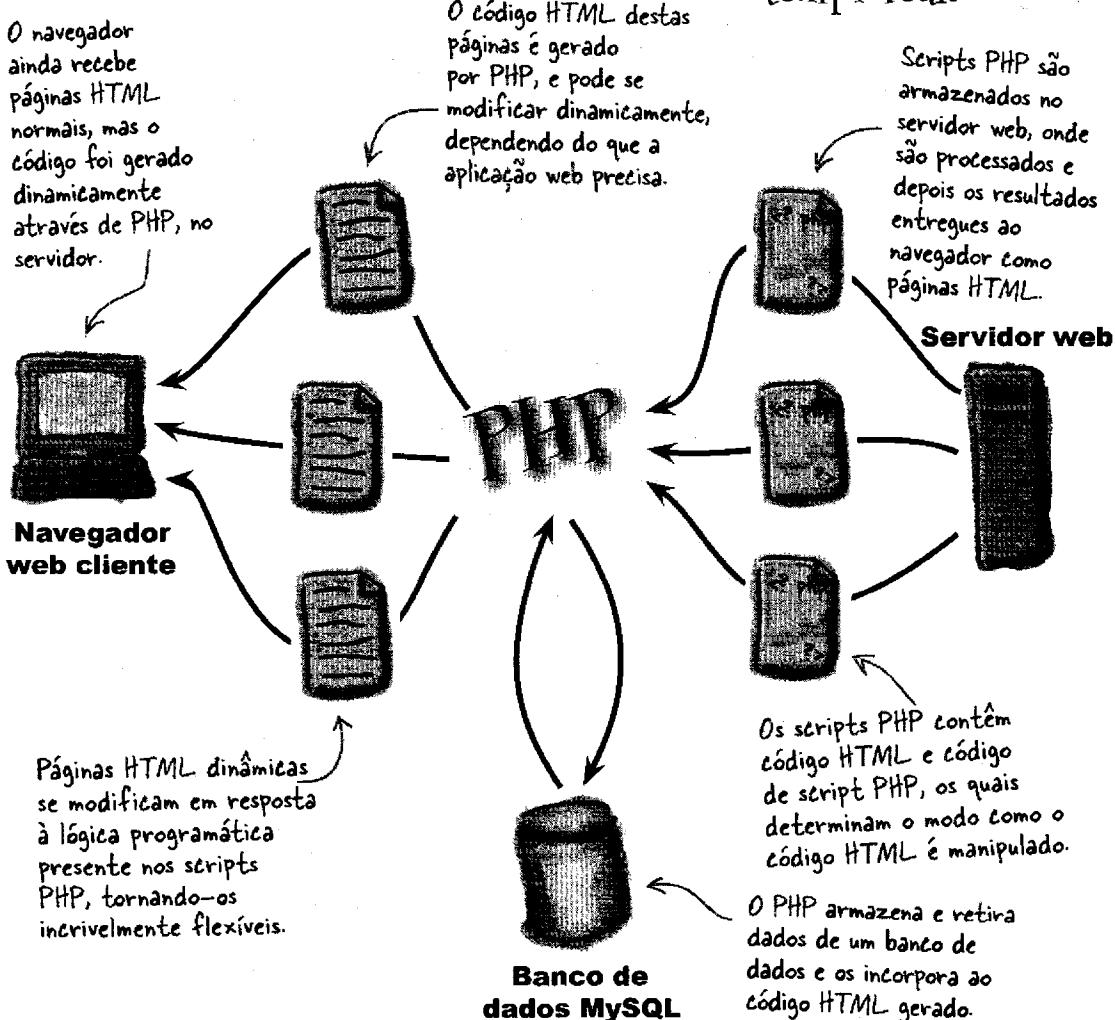
O servidor web representa uma grande parte do problema com o HTML estático, uma vez que ele funciona como nada mais que um mecanismo de entrega sem graça. O navegador requisita uma página, o servidor responde com HTML, fim da história. Para transformar sites em **aplicações** web interativas, o servidor precisa assumir um papel novo, mais dinâmico... um papel possibilitado pelo **PHP**.

Com páginas HTML puras, o servidor simplesmente serve HTML estático, que só é capaz de exibir conteúdo.

Com uma pequena  
ajuda do servidor!

## PHP dá vida às páginas web

O PHP lhe permite manipular o conteúdo das páginas web *no servidor*, imediatamente antes de a página ser enviada ao navegador cliente. Funciona da seguinte maneira: um script PHP é executado no servidor, e pode alterar ou gerar código HTML à vontade. Uma página HTML então é entregue ao navegador, que não sabe e nem liga se há PHP envolvido na produção ou ajuste do HTML, no lado do servidor.

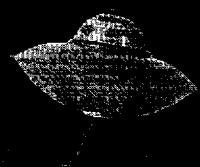


Com PHP na parada,  
o servidor web  
é capaz de gerar  
páginas HTML de  
forma dinâmica, em  
tempo real.

## Cães no espaço

Este é o Owen. Ele perdeu seu cachorro, Fang. Mas encontrar seu cão não se restringe a uma simples tarefa de procurar pelo bairro. Acontece que Fang foi abduzido por alienígenas, o que expande a busca de Owen por todo o universo. Owen sabe um pouco de HTML e CSS e acha que um site personalizado poderia ajudá-lo a resolver seu problema, ao permitir que outras pessoas compartilhem suas próprias experiências de abdução alienígena.

Mas, para obter informações das pessoas, Owen precisará de um formulário web capaz de receber input (entrada de dados) dos usuários e notificá-lo sobre esse input. Sem problema – o HTML possui várias tags para a criação de formulários web.



Você o viu?



Os detalhes são confusos, mas o que nós sabemos é que Fang foi arrebatado em direção ao céu por um raião de luz

Owen sabe um pouco de HTML e CSS e acha que poderá usar a web como forma de ajudá-lo a descobrir o paradeiro de seu cão, Fang.

## Um formulário ajuda Owen a descobrir toda a história

O novo site de Owen, AliensMeAbducteme.com, tem por objetivo conectá-lo com abduzidos que possam ajudar a esclarecer um pouco sobre o desaparecimento de Fang. Owen sabe que precisa de um formulário HTML para requisitar histórias de abdução dos visitantes e que precisa descobrir se por acaso, eles viram Fang durante as suas jornadas interestelares. Mas Owen precisa de ajuda para conseguir montar o site e colocá-lo no ar. Eis o que ele tem em mente para o formulário.

**Aliens Abducted Me - Report an Abduction**

Share your story of alien abduction:

First name:

Last name:

What is your email address?

When did it happen?

How long were you gone?

How many did you see?

Describe them:

What did they do to you?

Have you seen my dog Fang?  Yes  No



Anything else you want to add?

Este formulário é 100% puro HTML!

Owen deseja receber uma mensagem de email quando o usuário submeter o formulário.

Quaisquer comentários adicionais são colocados aqui.

Este é o campo para o endereço de email do visitante.

Owen quer uma descrição física dos aliens.

Owen espera que alguém responda sim, que eles viram Fang na espaçonave alienígena.

Quaisquer comentários adicionais são colocados aqui.

### O que você achou do formulário HTML de Owen?

Você consegue pensar em quaisquer problemas com os quais Owen poderá se deparar quando tentar obter dados sobre abduções usando este formulário? Vá em frente, ponha suas ideias no papel.

em vez de interagir, faltou usar interação no BD.

# Formulários são feitos de HTML

O formulário "Relate uma Abdução de Owen" (baixe este código através do site da altabooks.com.br) se compõe, inteiramente, de tags e atributos HTML. Há campos de texto para a maioria das perguntas, botões de rádio para descobrir se o visitante viu Fang e uma área de texto para comentários adicionais. E o formulário está configurado para enviar os dados recebidos para o endereço de email de Owen.

"mailto" é um protocolo que permite que dados de formulários sejam enviados por email.

```
<p>Share your story of alien abduction:</p>
<form method="post" action="mailto:owen@aliensabductedme.com">
  <label for="firstname">First name:</label>
  <input type="text" id="firstname" name="firstname" /><br />
  <label for="lastname">Last name:</label>
  <input type="text" id="lastname" name="lastname" /><br />
  <label for="email">What is your email address?</label>
  <input type="text" id="email" name="email" /><br />
  <label for="whenithappened">When did it happen?</label>
  <input type="text" id="whenithappened" name="whenithappened" /><br />
  <label for="howlong">How long were you gone?</label>
  <input type="text" id="howlong" name="howlong" /><br />
  <label for="howmany">How many did you see?</label>
  <input type="text" id="howmany" name="howmany" /><br />
  <label for="aliendescription">Describe them:</label>
  <input type="text" id="aliendescription" name="aliendescription" size="32" /><br />
  <label for="whattheydid">What did they do to you?</label>
  <input type="text" id="whattheydid" name="whattheydid" size="32" /><br />
  <label for="fangspotted">Have you seen my dog Fang?</label>
  Yes <input id="fangspotted" name="fangspotted" type="radio" value="yes" />
  No <input id="fangspotted" name="fangspotted" type="radio" value="no" /><br />
  <br />
  <label for="other">Anything else you want to add?</label>
  <textarea id="other" name="other"></textarea><br />
  <input type="submit" value="Report Abduction" name="submit" />
</form>
```

O formulário fica contido dentro de tags <form> de abertura e de fechamento.

Nenhuma surpresa aqui - o formulário é 100% puro código HTML!

Se precisar de um refresh de memória sobre como criar formulários HTML, dê uma olhada no Capítulo 14 de Use a Cabeça! HTML com CSS & XHTML

Owen irá receber o conteúdo deste formulário enviado para ele neste endereço de email - ponha aqui o seu próprio endereço para testar o formulário.

Este valor diz ao servidor como enviar os dados. Será "post" ou "get". Nós explicaremos a diferença mais à frente.

As tags input dizem ao formulário para esperar informações, aqui.

O atributo type diz à ação do formulário para esperar texto.

O botão "submeter" diz ao formulário para executar a ação exposta ali.

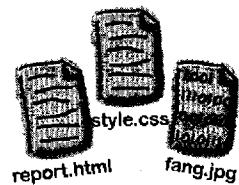


## TEST DRIVE

### Teste o formulário “Relatar uma Abdução”.

Baixe o código da página “Relatar uma Abdução” no site da Alta Books, em [www.altabooks.com.br](http://www.altabooks.com.br). O código estará no **cadastro deste Livro**. O folder ch1 contém o formulário de Owen em **report.html**, bem como uma folha de estilo (**style.css**) e uma imagem de Fang (**fang.jpg**).

Abra a página **report.html** em um editor de texto e troque o endereço de email de Owen pelo seu. Em seguida, abra a página em um navegador web, digite algumas informações no formulário e clique no botão “Relatar Abdução” (Report Abduction).



Share your story of alien abduction:

First name: Alf  
Last name: Nader  
What is your email address? alfn@myrealgreen.com  
When did it happen? last November  
How long were you gone? 11 hours  
How many did you see? dozens  
Describe them: little green men  
What did they do to you? asked me about UFO regulations & Fang spotted me  
Have you seen my dog Fang? Yes  No

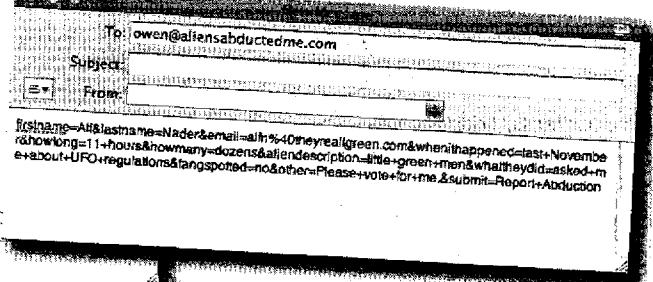
Anything else you want to add?

Please vote for me.

**Report Abduction**

Quando você submete o formulário, isso resulta nos dados a serem enviados... ou quase isso.

O formulário HTML não sabe, realmente, como enviar uma mensagem de email, então ele delega a tarefa para o programa de email do próprio usuário.



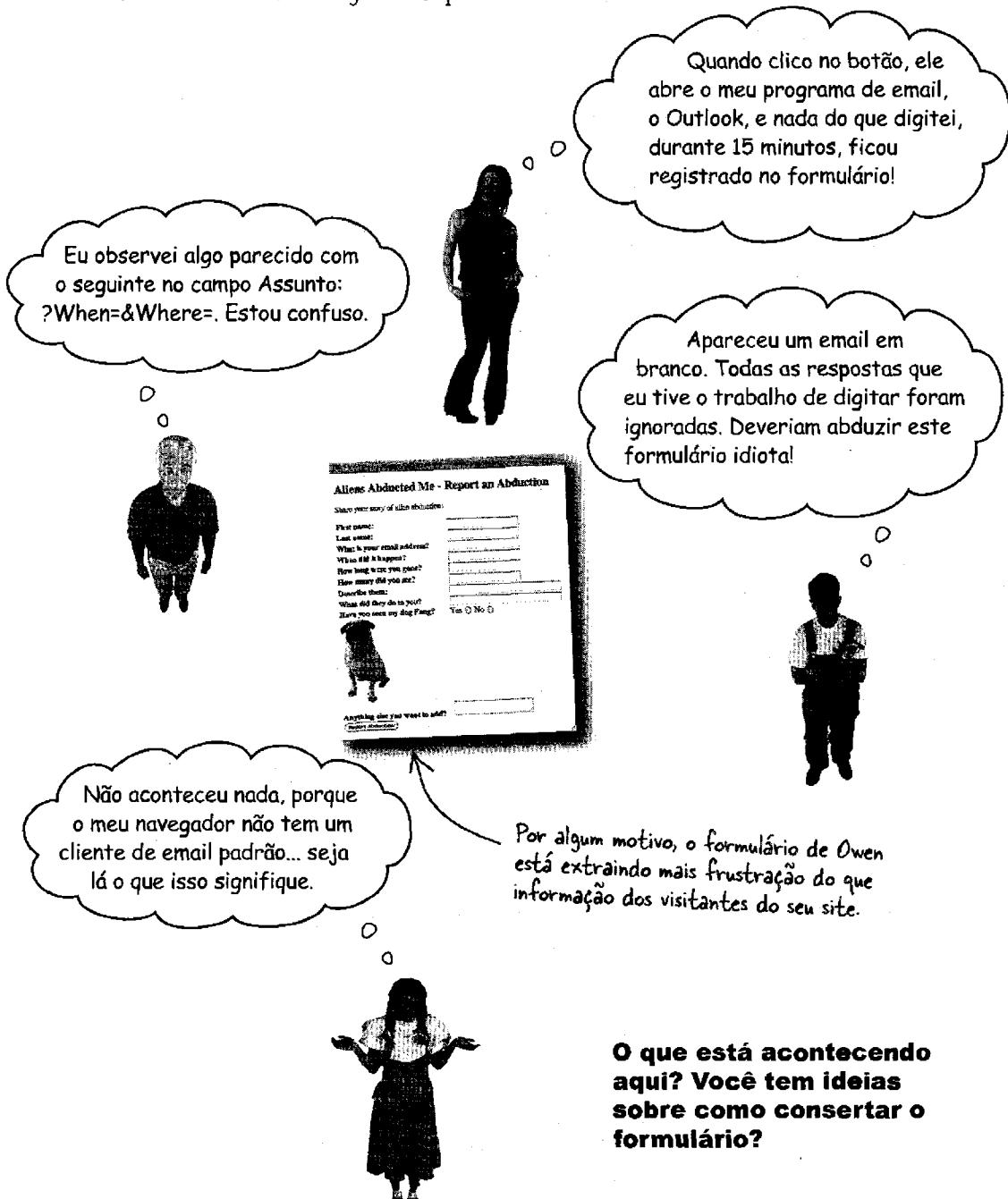
Os dados do formulário não são enviados para Owen, a não ser que o usuário manualmente envie esta estranha mensagem de email.

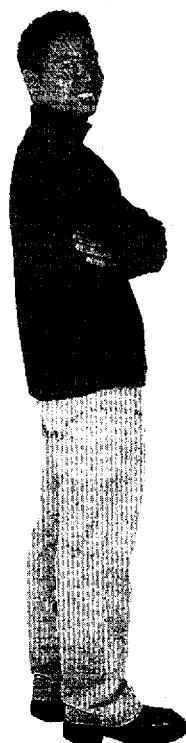
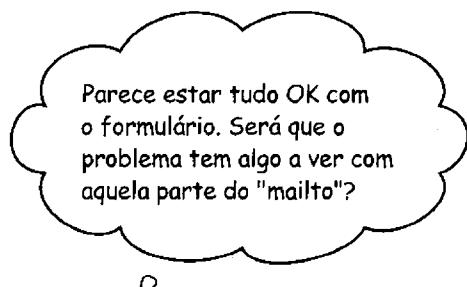
**Então, o que você achou? Você recebeu os dados do formulário como uma mensagem de email na sua caixa de entrada?**

*mailto = má idéia*

## O formulário HTML apresenta problemas

O formulário “Relatar uma Abdução” está no ar, operante, mas ele não consegue obter muitas informações dos usuários. Será que a abdução de Fang corresponde a um incidente isolado... ou tem algo de errado com o formulário. Vejamos o que os usuários têm a dizer.





### Sim. O código HTML do formulário está correto, mas “mailto” não é uma boa maneira de enviar dados

O formulário de Owen funciona bem até o usuário clicar no botão “Relatar Abdução”. Nesse ponto, você passa a depender do recurso de `mailto` para empacotar os dados do formulário em uma mensagem de email. Mas este email não é enviado automaticamente – ele é criado no programa de mail padrão do computador do usuário. E o pior: o usuário precisa **enviar ele mesmo o email**, para que os dados sejam realmente enviados para você! Você não tem nenhum controle sobre a entrega do email, significando que ele poderá – ou não – fazer a viagem do seu formulário web, passando pelo navegador do usuário e pelo seu cliente de email, até chegar a você como uma mensagem de email. Nada bom.

Você precisa de uma forma para ter o controle sobre a entrega do formulário web. Mais especificamente, você precisa que o PHP empacote os dados em uma mensagem de email, e depois se certifique de que essa mensagem será enviada. Isto envolve desviar a sua atenção do **cliente** (HTML, `mailto`, etc.) para o **servidor** (PHP).

O formulário está ótimo, até que você clica no botão “Relatar Abdução” – então tudo desmorona!

Aliens Abducted Me - Report an Abduction

Share your story of alien abduction:

First name: \_\_\_\_\_  
Last name: \_\_\_\_\_  
What is your email address? \_\_\_\_\_  
When did it happen?  
How long were you gone?  
How many did you see?  
Describe them:  
What did they do to you?  
Have you seen my dog Fang? Yes  No

Anything else you want to add? \_\_\_\_\_

A small illustration of a black dog is shown next to the "Describe them:" section.

## HTML roda no cliente

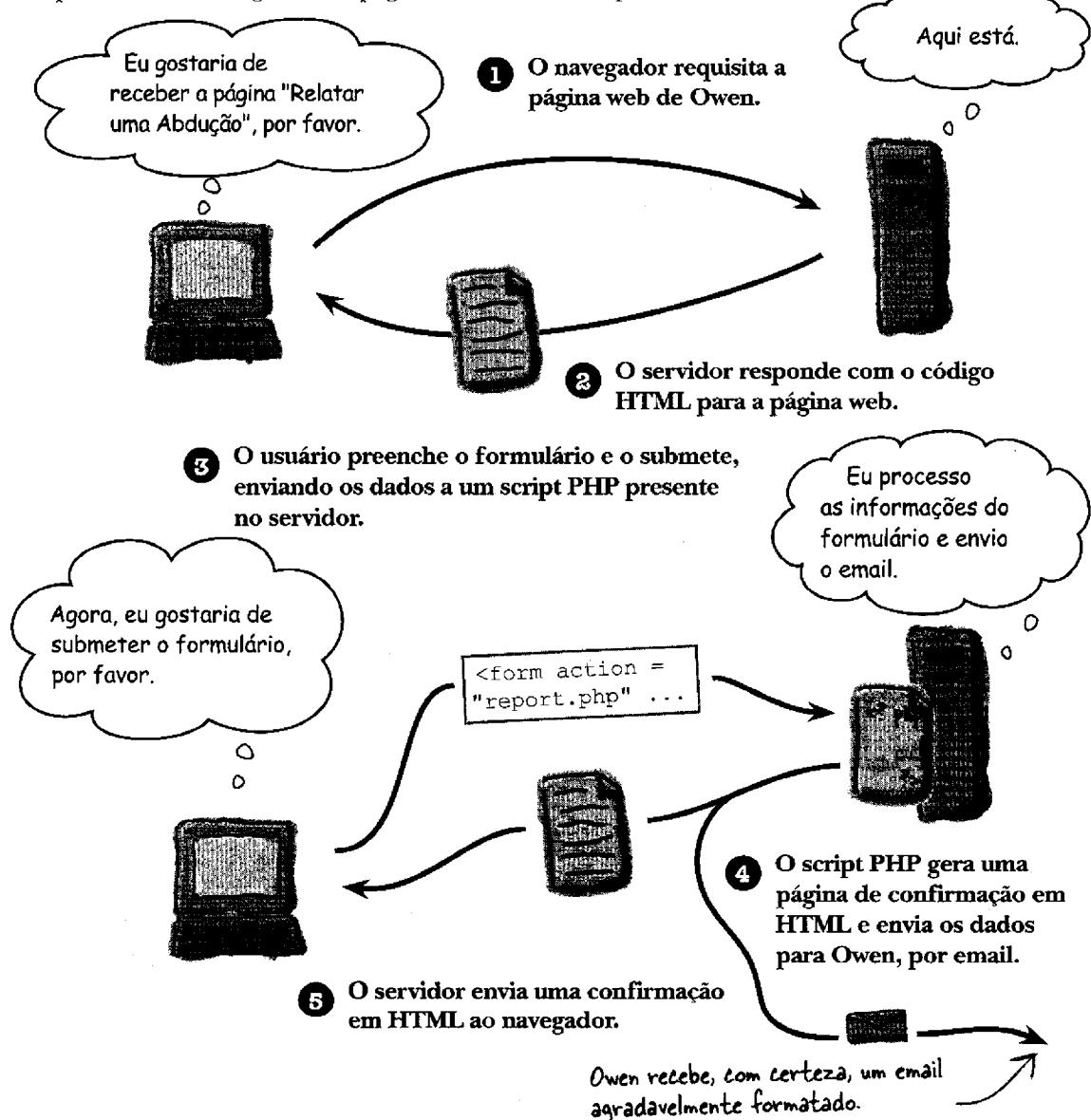
O formulário de Owen foi escrito em HTML puro, com uma ação mailto que tenta enviar os dados via email. Embora a página report.html venha de um servidor web, ela é preenchida e processada inteiramente no navegador web do usuário.



O papel do servidor, aqui, limita-se a apenas entregar a página web ao navegador. Quando o usuário submete o formulário, o navegador (cliente!) precisa resolver sozinho a questão de como fazer os dados serem enviados via email. O cliente não está equipado para entregar dados do formulário – esse é um trabalho para o servidor.

## PHP roda no servidor

O PHP lhe permite controlar os dados que o usuário digita no formulário, enviando-os para você de forma **transparente**. O usuário digita o seu caso de abdução no formulário, clica no botão “Relatar Abdução” e pronto! O código PHP cria a mensagem de email, a envia para você e então gera uma página de confirmação para o usuário.



Marque os lugares onde você acha que os scripts PHP devem residir:

- no cliente  no servidor  em ambos  em nenhum dos dois

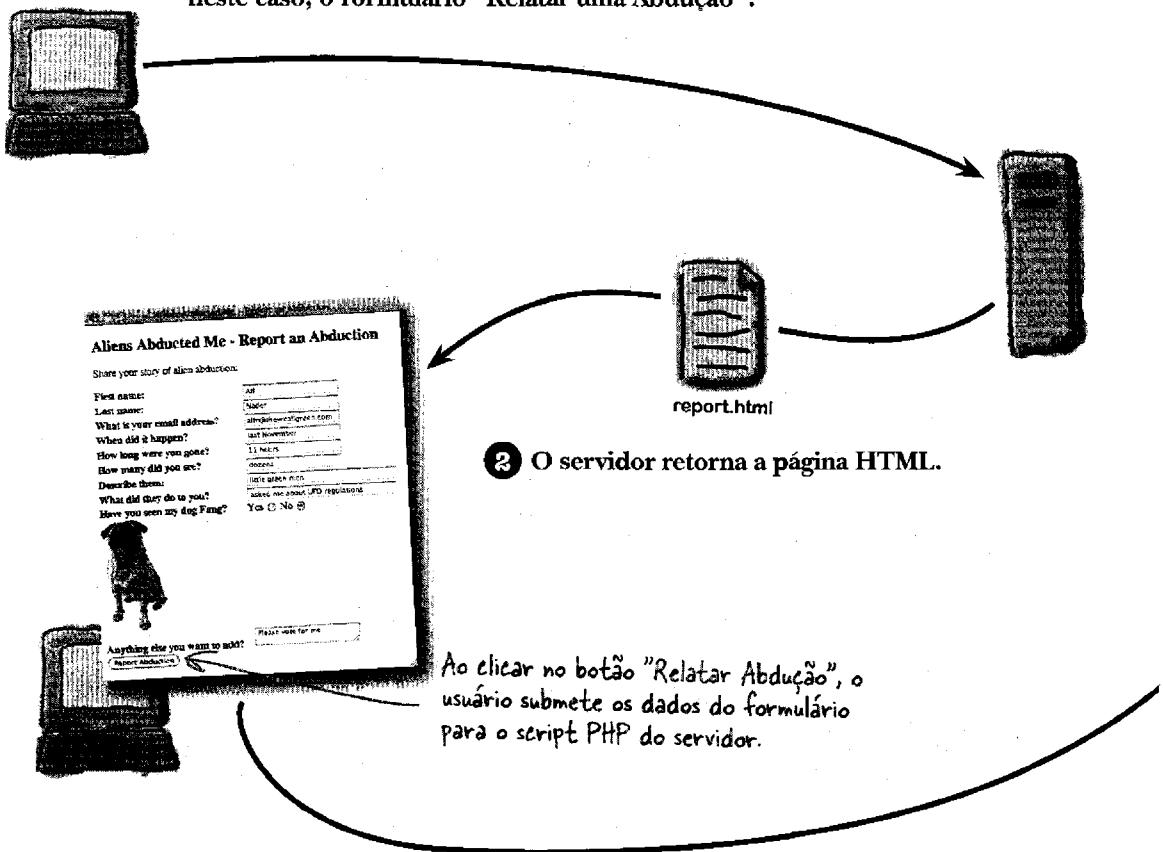
*php é uma linguagem do lado-servidor*

## Os scripts PHP rodam no servidor

O código PHP roda no servidor e é armazenado em **scripts PHP** que geralmente tem a extensão **.php**. Os scripts PHP frequentemente se parecem bastante com páginas HTML normais, porque contêm tanto código HTML quanto CSS. De fato, quando o servidor executa um script PHP, o resultado final é sempre HTML e CSS puros. Assim, todo script PHP acaba se transformando em HTML e CSS, uma vez terminada a sua execução no servidor.

Vamos dar uma olhada mais detalhada no modo como o script PHP modifica o fluxo do formulário web de Owen.

- 1 O navegador do cliente requisita uma página HTML – neste caso, o formulário “Relatar uma Abdução”.



- 3 O usuário preenche o formulário e o submete, fazendo o navegador repassar os dados para um script PHP no servidor.

O PHP é uma linguagem de programação do lado do servidor (serve-side) - pois ele roda em um servidor Web.

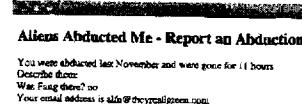
- 5 O servidor retorna uma página de HTML puro, gerada pelo script PHP.

O script PHP roda no servidor!



Embora o nome da página apareça com uma extensão .php no navegador, ela é puro HTML, neste ponto.

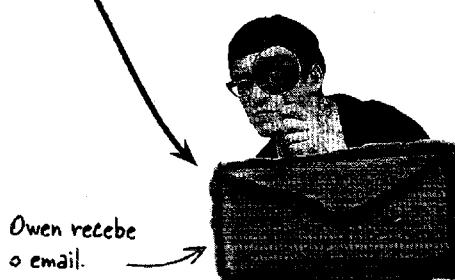
- 6 O navegador exibe a página de confirmação.



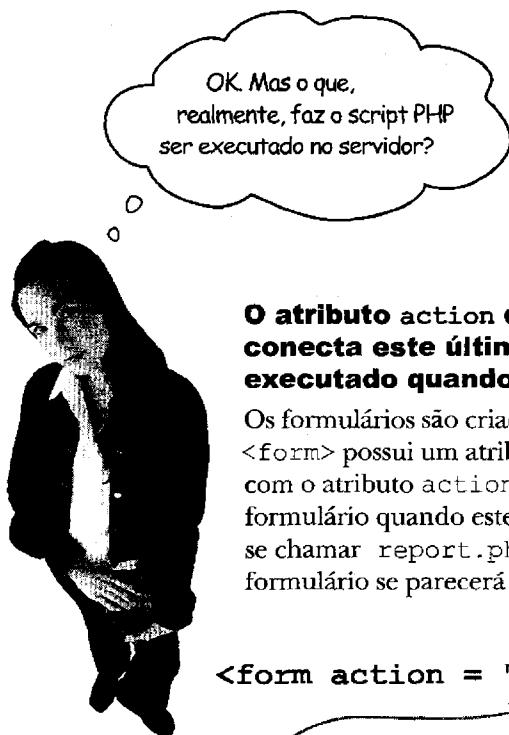
O usuário vê uma página de confirmação.

- 4 O servidor executa o script PHP, o qual envia um email e gera uma página de confirmação em HTML.

- 7 O email é enviado para a Caixa de Entrada de Owen.



## **o atributo action do formulário**



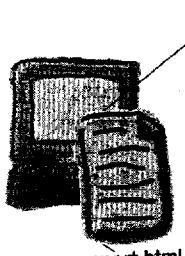
**O atributo action dos elementos do formulário é o que conecta este último ao script PHP, fazendo o script ser executado quando o formulário é submetido.**

Os formulários são criados usando-se a tag HTML <form> e cada tag <form> possui um atributo action. O arquivo cujo nome você inclui com o atributo action é usado pelo servidor web para processar o formulário quando este é submetido. Assim, se o script PHP de Owen se chamar report.php, então a tag <form> para conectá-lo ao formulário se parecerá com o seguinte:

```
<form action = "report.php" method = "post">
```

Este é o nome do seu script PHP.

Quando o usuário clica no botão “Relatar Abdução”, a ação do formulário faz o script report.php ser executado **no servidor**, para processamento dos dados.



report.html

```
<html>
<head>
  <title>Aliens Abducted Me - Report an Abduction</title>
</head>
<body>
  <h2>Aliens Abducted Me - Report an Abduction</h2>
  <link rel="stylesheet" type="text/css" href="style.css" />
</body>
<h2>Aliens Abducted Me - Report an Abduction</h2>

<p>Share your story of alien abduction:</p>
<form method="post" action="report.php">
  <label for="firstname">First Name:</label>
  <input type="text" id="firstname" name="firstname" /><br />
```

O atributo action da tag <form> é o que faz o script PHP ser executado no servidor, quando o formulário é submetido.



report.php

não existem  
**Perguntas Ídiotas**

**P:** O que significa PHP?

**R:** PHP é um acrônimo que originalmente significava Personal Home Pages (Páginas Pessoais). De alguma forma, o acrônimo teve o seu significado modificado para PHP: Hypertext Processor (Processador de Hipertexto PHP). Este último é considerado um acrônimo recursivo, porque ele referencia a si mesmo – um acrônimo (PHP) dentro do acrônimo. Inteligente? Confuso? Você decide!

**P:** Mesmo que o meu navegador esteja mostrando que o nome da página termina com .php, ela ainda é puro HTML? Como é possível?

**R:** Isso é possível porque a página **começa** a sua vida como código PHP no servidor, mas é **transformada** em código HTML antes de ser enviada para o navegador. Assim, o servidor executa o código PHP e o converte em HTML antes de enviá-lo para ser visualizado no navegador. Isso significa que, mesmo que o arquivo .php contenha código PHP, o navegador nunca vê esse código – ele vê apenas o código HTML, resultado da execução do código PHP no servidor.

**P:** Mas não é verdade que toda página web se origina no servidor, inclusive as páginas HTML puras, em arquivos .html?

**R:** Sim. Todos os arquivos que compõem um site ficam armazenados no servidor – .html, .css, .php, etc. Mas nem todos eles são **processados** pelo servidor. Os arquivos HTML e CSS, bem como arquivos gráficos, são enviados diretamente para o navegador do cliente, sem preocupações sobre o conteúdo deles. Os arquivos PHP são diferentes porque contêm códigos que são processados e **executados** no servidor web. Não é o código PHP que é enviado ao navegador, e sim os **resultados** obtidos ao se executar o código PHP, – esses resultados são HTML e CSS puros.

seu 1º script php

## Use PHP para acessar os dados do formulário

Então, Owen precisa de um script PHP capaz de fazer com que as informações do formulário cheguem a ele de forma mais confiável do que através do e-mail. Vamos criar esse script.

Não se preocupe em entender tudo agora – nós chegaremos lá:

```
<html>
  <head>
    <title>Aliens Abducted Me - Report an Abduction</title>
  </head>
  <body>
    <h2>Aliens Abducted Me - Report an Abduction</h2>
    <?php
      $when_it_happened = $_POST['when_it_happened'];
      $how_long = $_POST['howlong'];
      $alien_description = $_POST['description'];
      $fang_spotted = $_POST['fangspotted'];
      $email = $_POST['email'];

      echo "Thanks for submitting the form.<br />";
      echo "You were abducted " . $when_it_happened;
      echo " and were gone for " . $how_long . "<br />" ;
      echo "Describe them: " . $alien_description . "<br />" ;
      echo "Was Fang there? " . $fang_spotted . "<br />" ;
      echo "Your email address is " . $email;

    ?>
  </body>
</html>
```

Os scripts PHP frequentemente começam parecendo-se bastante com uma página HTML normal.

Ah, é aqui que as coisas ficam interessantes – este é o começo do código PHP propriamente dito.

É perfeitamente normal um script PHP incluir tags e atributos HTML normais.

O bloco de código inteiro é PHP... o restante do script é HTML normal.

Este pedaço de código PHP obtém os dados do formulário, para que eles possam ser exibidos como parte de uma página de confirmação.

Aqui, nós usamos PHP para gerar código HTML a partir dos dados do formulário.

Assim como uma página web normal, este script PHP termina fechando quaisquer tags HTML abertas.



## TEST DRIVE

**Modifique o formulário de Owen de forma que ele use um script PHP para processar os dados do formulário.**

Crie um novo arquivo de texto chamado report.php e digite nele todo o código da página anterior. Esse é o script que irá processar o formulário web de Owen.

O script PHP ainda não está conectado ao formulário; abra a página report.html em um editor de textos e modifique a ação do formulário para report.php em vez de mailto.

```
<form action = "report.php" method = "post">
```



Abra a página report.html em um navegador web, digite algumas informações do seu caso de abdução no formulário e clique no botão "Relatar Abdução".

Aliens Abducted Me - Report an Abduction

Share your story of alien abduction:

First name: Ari  
Last name: Lacer  
What is your email address: ari@greyrealty.com  
When did it happen? last November  
How long were you gone? 11 hours  
How many did you see? dozens  
Describe them: Little green men  
What did they do to you? Asked me about Fang  
Have you seen my dog Fang? Yes  No   
Your email address is: ari@mail.com

You were abducted ' . \$when\_it\_happened; echo ' and were gone for ' .  
Show long; echo '  
Describe them: ' . \$alien\_description; echo '  
Was Fang there? ' . \$fang\_spotted; echo '  
Your email address is ' . \$email; ?>

Please vote for me.  
Report Abduction

Dependendo do seu navegador, você poderá ver uma página web com um texto esquisito ou possivelmente apenas o código-fonte PHP do script report.php.

### Aliens Abducted Me - Report an Abduction

```
Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8"/>
<title>Aliens Abducted Me - Report an Abduction</title>
</head>
<body>
<h2>Aliens Abducted Me - Report an Abduction</h2>
```

Anything else you want to add?

Report Abduction

Você acha que é assim mesmo que o script PHP deveria funcionar?  
Escreva aqui a sua opinião, e o que acha que está acontecendo.

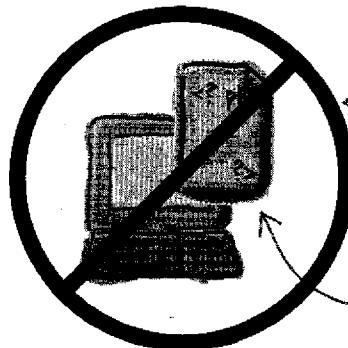
*Não, o servidor não deve estar fazendo  
além de ser PHP corretamente.*

## Os scripts PHP precisam ficar em um servidor!

A não ser que você, por acaso, tenha um servidor web rodando no seu computador local, o script `report.php` não poderá ser executado quando você submeter o formulário "Relatar uma Abdução". Lembre-se, o PHP é uma linguagem de programação e precisa de um ambiente onde possa rodar. Esse ambiente é um servidor web com suporte a PHP. Scripts PHP e páginas web que dependem deles **precisam ser colocadas em um servidor web real**, não basta apenas abrir um script diretamente a partir de um sistema de arquivos local.

E se você tiver um servidor web instalado localmente, e ele tiver suporte a PHP, então poderá testar os scripts PHP diretamente no seu computador.

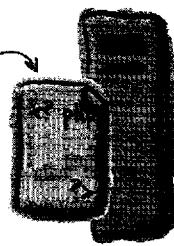
Os navegadores web  
não sabem nada sobre  
PHP e, portanto, não  
são capazes de rodar  
scripts PHP.



Ao contrário de páginas HTML, que podem ser abertas localmente em um navegador web, os scripts PHP precisam sempre ser "abertos" através de uma URL, a partir de um servidor web.

Este script PHP representa  
apenas um monte de  
código sem sentido para o  
navegador web.

O servidor web  
entende este  
código PHP e  
executa o script!



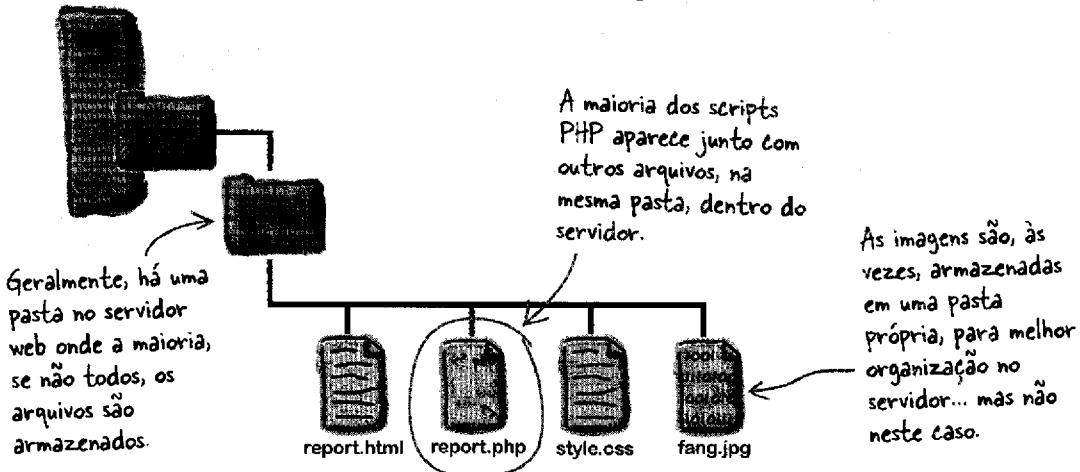
Os servidores web com suporte  
a PHP são equipados para  
rodar scripts PHP e transformá-  
los em páginas HTML que os  
navegadores possam entender.

Uma forma rápida de saber se uma  
página web está sendo entregue por  
um servidor web é olhar se a URL  
começa com "http:". Páginas web  
abertas como arquivos locais sempre  
começam com "file:"

Os scripts PHP devem ser  
executados em um servidor  
web, ou então não funcionarão.

## Coloque os seus scripts PHP no servidor

É perfeitamente válido criar e editar scripts PHP no seu computador local. Porém, você precisa colocar os arquivos em um servidor web antes de executá-los. Os arquivos PHP, frequentemente, são colocados junto com arquivos HTML em um servidor. Não há nada fora do comum em colocar scripts PHP no servidor web – basta colocá-los em um lugar onde as suas páginas web possam acessá-los. Para fazer o upload (envio) dos arquivos para o servidor, você precisará da ajuda de um utilitário, como por exemplo, um programa de FTP (File Transfer Protocol ou Protocolo de Transferência de Arquivos).



Fazer o upload dos seus scripts PHP para um servidor web não é o suficiente – esse servidor também precisa ter o PHP instalado nele. Alguns servidores incluem o PHP por padrão, alguns não.

### não existem Perguntas Idiotas

**P:** Como eu posso saber se o meu servidor web tem o PHP instalado?

**R:** Você pode perguntar ao seu administrador, ou à sua empresa de hospedagem de sites, ou então pode realizar um pequeno teste, você mesmo. Crie um arquivo de texto chamado `teste.php` e coloque o seguinte código nele:

```
<?php
    phpinfo();
?>
```

Este código pede para que informações sobre o PHP sejam mostradas no

Agora, envie (upload) o `teste.php` para o servidor, e então digite a sua URL em um navegador web. Se o PHP estiver instalado no seu servidor, aparecerão várias informações detalhadas sobre o PHP, incluindo a sua versão. Bingo!



**Relaxe**

Se você não tiver o PHP instalado no seu servidor web, consulte

**o Apêndice ii.**

Nele, encontrará instruções para ter o PHP instalado e operante no seu servidor.

Lembre-se de deletar o script `phpinfo` quando estiver terminado, desta forma, ninguém mais consegue ver isto

você está aqui >

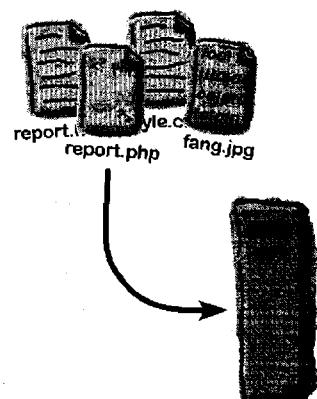
*test drive no seu script php*



## TEST DRIVE

**Faça o upload dos arquivos da página  
“Relatar uma Abdução” e teste o formulário...  
novamente.**

Envie os arquivos report.html, report.php, style.css e fang.jpg para um servidor web que tenha o PHP instalado. Digite a URL da página report.html no seu navegador, preencha o formulário com informações do seu caso de abdução, e clique no botão “Relatar Abdução”.



**Aliens Abducted Me - Report an Abduction**

Share your story of alien abduction:

First name: Al  
Last name: Nader  
What is your email address? aln@theyrealgreen.com  
When did it happen? last November  
How long were you gone? 11 hours  
How many did you see? dozens  
Describe them: little green men  
What did they do to you? asked me about UFO regulations  
Have you seen my dog Fang? Yes  No

Please vote for me.

O script PHP funcional! Ele  
exibe os dados do formulário em  
uma página de confirmação.

**Aliens Abducted Me - Report an Abduction**

You were abducted last November and were gone for 11 hours  
Describe them:  
Was Fang there? no  
Your email address is aln@theyrealgreen.com



Legal. Agora só é preciso adicionar um pouco de código PHP para enviar os dados do formulário por email.

**Isso mesmo. O script report.php ainda precisa de algum código para enviar os dados para Owen por email.**

Mas isso não é problema, porque o PHP oferece uma função, que é um pedaço de código pronto reutilizável, que você pode usar para enviar mensagens por email. Você só precisa determinar o que a mensagem deve informar, e então pode usar o PHP para criá-la e enviá-la.



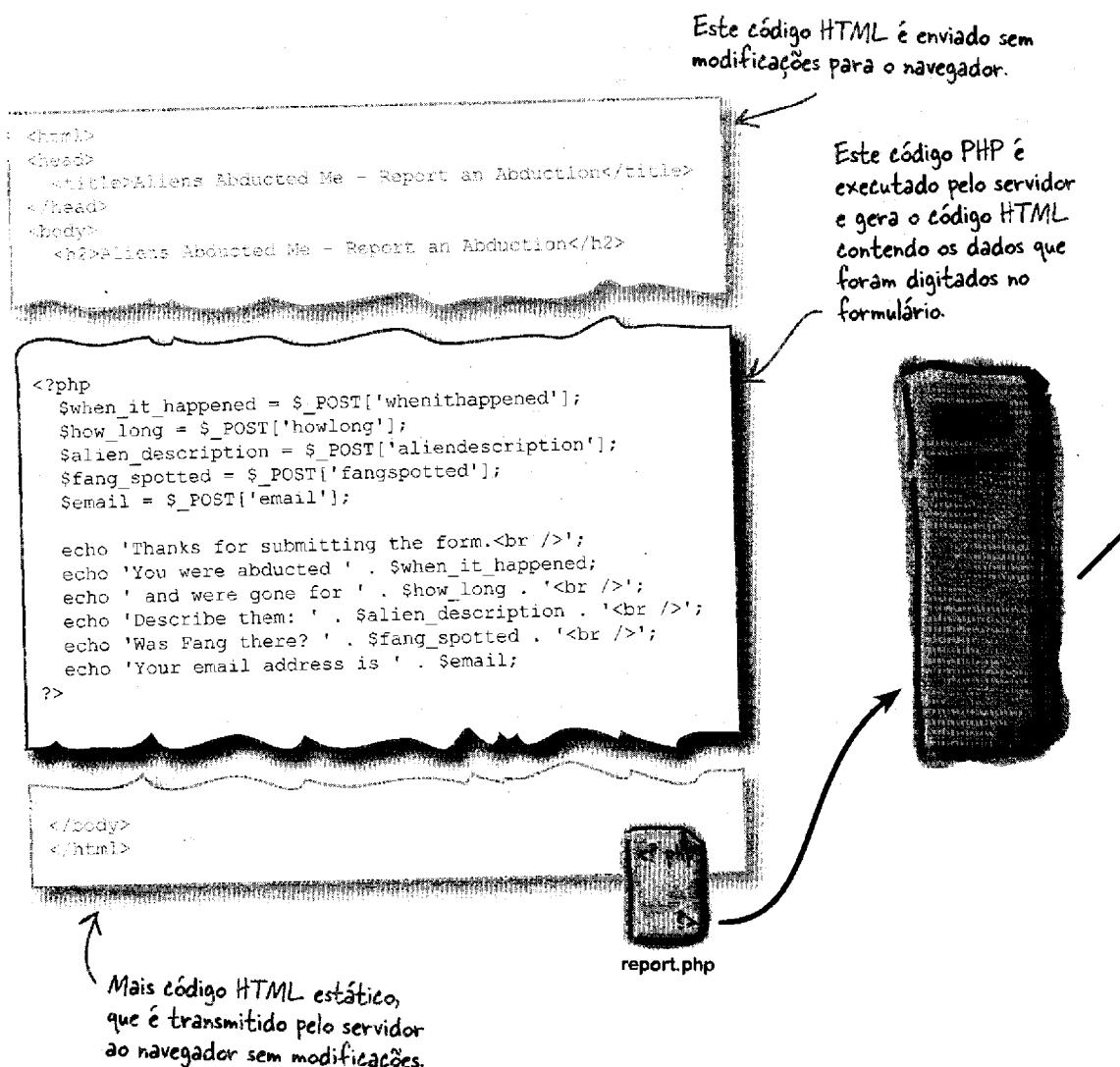
Um momentinho! Nós nem sabemos como o script report.php original funciona, e agora ainda temos de torná-lo capaz de enviar emails. Isso, é tipo, muito difícil... o que vocês estão pensando!?

### **É verdade. Para fazer mais coisas com o PHP, é preciso saber mais sobre o PHP**

Assim, para adicionar a funcionalidade de email ao script `report.php` de Owen, você terá de se aprofundar um pouco mais no PHP e precisará ter um entendimento sólido de como o script funciona – pelo menos até aqui.

# O servidor transforma PHP em HTML

Para que se entenda como um script PHP funciona, o principal é entender o que acontece com o script quando ele é executado no servidor. A maioria dos scripts PHP contém tanto código PHP quanto HTML, e o PHP é executado e transformado em HTML antes de o servidor enviar tudo – na forma de HTML – para o navegador web do cliente. No script report.php de Owen, o código PHP gera a maior parte do conteúdo HTML no corpo da página de confirmação. O código HTML em torno do PHP é enviado sem modificações.



Este código HTML é criado em tempo real pelo script PHP, o que permite ao código fazer coisas interessantes, como por exemplo, incluir dados que acabaram de ser digitados em um formulário.

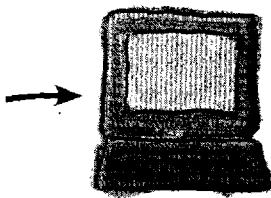
Estático - isto não se modifica.

Dinâmico - modifica-se cada vez que alguém submete o formulário!

```
<head>
<title>Aliens Abducted Me - Report an Abduction</title>
</head>
<body>
<h2>Aliens Abducted Me - Report an Abduction</h2>

Thanks for submitting the form.<br />
You were abducted last November and were gone for 11 hours<br />
Describe them: <br />
Was Fang there? no<br />
Your email address is alfn@theyreallgreen.com

</body>
</html>
```



O resultado final do script PHP é uma página web em HTML puro, que foi gerada dinamicamente no servidor.

### Aliens Abducted Me - Report an Abduction

```
You were abducted last November and were gone for 11 hours
Describe them:
Was Fang there? no
Your email address is alfn@theyreallgreen.com
```

## Desconstruindo o script PHP de Owen

O script report.php é acionado pelo formulário “Relatar uma Abdução” e o seu trabalho (no momento) é acessar os dados digitados e gerar uma página de confirmação. Vejamos como.

O primeiro pedaço de código corresponde a HTML puro. Ele simplesmente configura a página que estamos criando, incluindo algumas tags HTML requeridas em toda a página web.

```
<html>
<head>
    <title>Aliens Abducted Me - Report an Abduction</title>
</head>
<body>
    <h2>Aliens Abducted Me - Report an Abduction</h2>
```

Aqui as coisas começam a ficar interessantes. Estamos prontos para sair do código HTML e entrar no PHP. A tag <?php abre uma seção de código PHP – tudo o que vier depois desta tag é puro PHP.

```
<?php
```

Sim, este código HTML é mínimo – normalmente você teria um DOCTYPE, tags <meta>, etc., mas estamos mantendo as coisas simples, aqui.

Este código reúne os dados do formulário e os armazena em variáveis individuais, para que possamos acessá-los posteriormente. As variáveis do PHP permitem que você armazene valores, sejam eles números, textos ou outros tipos de dados.

```
$when_it_happened = $_POST['whenithappened'];
$how_long = $_POST['howlong'];
$alien_description = $_POST['description'];
$fang_spotted = $_POST['fangspotted'];
$email = $_POST['email'];
```

Cada linha do código PHP reúne os dados do field do formulário para uma nova variável!

Agora sim! Aqui, as variáveis que acabamos de criar são colocadas em ação, sendo inseridas no código HTML, gerado dinamicamente. O comando echo ocasiona o output (saída) de código HTML, que é retornado diretamente para o navegador web.

```
echo 'Thanks for submitting the form.<br />';
echo 'You were abducted ' . $when_it_happened;
echo ' and were gone for ' . $how_long . '<br />';
echo 'Describe them: ' . $alien_description . '<br />';
echo 'Was Fang there? ' . $fang_spotted . '<br />';
echo 'Your email address is ' . $email;
```

Produz a saída de código html para o navegador

A tag ?> correspondente a <? encerra a seção de código PHP. Daqui em diante, estamos de volta ao código HTML normal.

Isto fecha o código php

```
?>
```

Isto finaliza o código PHP – depois desta tag, estamos de volta ao HTML normal.

```
</body>
</html>
```

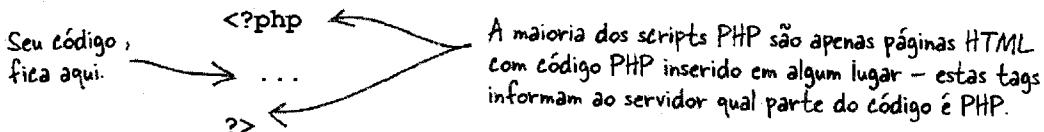
Agora, termine a página fechando as tags HTML que havíamos aberto.

## O seu código

### Algumas regras PHP para a sua vida

O script `report.php` de Owen revela algumas regras fundamentais da linguagem PHP, que se aplicam a todo script PHP. Vamos dar uma olhada nelas.

- O código PHP sempre fica entre tags `<?php` e `?>`.



- Toda instrução PHP precisa terminar com ponto e vírgula (;)

`echo 'Thanks for submitting the form.<br />' ;`

Se por acaso seu código falhar, verifique se você não esqueceu de colocar o "ponto e vírgula". Isso acontece com mais frequência do que se pode imaginar.

O "ponto e vírgula" informa ao PHP que isto corresponde ao final de uma instrução.

- Caso haja qualquer código PHP em uma página web, é uma boa ideia nomear o arquivo com `.php`, e não com `.html`, no servidor.



Não é essencial, mas é uma boa ideia nomear scripts PHP com a extensão `.php`.

- Os nomes de variáveis PHP precisam começar com um cifrão (\$).

`$email = $_POST['email'];`

O cifrão identifica claramente a variável PHP, responsável por armazenar informações dentro de um script PHP.

Observando as variáveis usadas no script `report.php`, você consegue ver quaisquer outras regras do PHP referentes a variáveis? Escreva-as aqui!

...  
...  
...

## nomeando as variáveis

### Encontrando o nome perfeito para a sua variável

Além de começar com \$, os nomes de variáveis PHP são também “case-sensitive”. Mas isso não é tudo – existem outras regras importantes para os nomes que você pode dar às suas variáveis. Algumas delas se referem à sintaxe, significando que o seu código falhará se você as ignorar, enquanto outras outras obras correspondem a apenas boas ideias transmitidas de uma geração de programadores PHP para outra.

Vamos começar com as regras oficiais que irão definitivamente causar problemas se você ignorá-las ao nomear suas variáveis. Siga estas regras para criar nomes de variáveis válidas.

A variável corresponde a um container onde você pode armazenar dados, e cada variável tem um nome único.

O primeiro caractere tem de ser um cifrão (\$). Entendido!

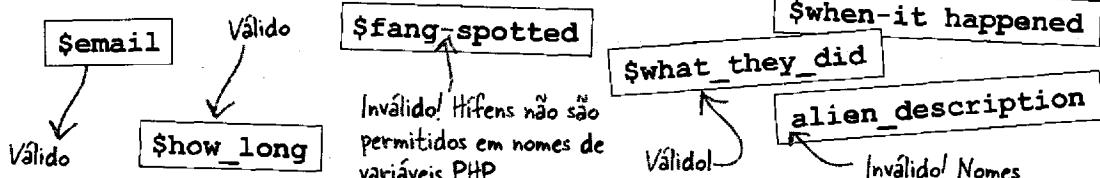
O nome da variável precisa ter no mínimo um caractere.

Sem contar o \$, que é requerido no nome de toda variável.

O primeiro caractere depois do cifrão pode ser uma letra ou um “underscore” (\_), e os caracteres seguintes podem ser uma letra, um underscore ou um número.

Inválido! Nomes de variáveis PHP não podem conter hifens nem espaços.

Espaços e caracteres especiais, fora \_ e \$, não são permitidos em nenhuma parte do nome de uma variável



Se você não seguir estas regras, o seu código não funcionará. Existem ainda, outras regras que podem ser seguidas – por uma questão de “convenções” de programação. Estas são regras que ajudam a tornar o código PHP um pouco mais consistente e fácil de ser lido.

Use minúsculas nos nomes das variáveis.

Em nomes de variáveis com mais de uma palavra, separe-as com “underscores”.

Se você ignorar estas duas últimas regras, não serão geradas falhas no seu código. Certamente, você encontrará códigos PHP que não seguem estas regras e, mesmo assim, funcionam perfeitamente. Isso ocorre porque estas regras – sobre as quais estamos nos referindo – tratam-se de uma convenção estilística. Elas tornam-se úteis quando começam a criar e nomear suas próprias variáveis.

Os nomes das variáveis PHP devem começar com um cifrão(\$), e não podem conter espaços.

não existem

## Perguntas Ídiotas

**P:** Faz diferença escrever os comandos PHP em maiúsculas ou minúsculas?

**R:** Sim e não. Em geral, o PHP não é "case-sensitive", portanto, você pode misturar maiúsculas e minúsculas na maioria dos comandos. Isso significa que você pode usar echo, ECHO ou EchO. Entretanto, por uma questão de convenção, é uma ótima ideia ser consistente nos seus scripts. A maioria dos programadores PHP prefere usar minúsculas na maior parte dos seus códigos, e é por isso que verá a gráfic echo sendo usada nos exemplos de códigos deste livro.

**P:** Então, mesmo sendo uma má prática de programação, posso misturar maiúsculas e minúsculas no meu código PHP?

**R:** Não completamente. A exceção corresponde aos nomes das variáveis, o que também se aplica às localidades de armazenamento de dados que você cria. Vejamos a variável \$email usada no script "Relatar uma Abdução", como exemplo. O nome desta variável é "case-sensitive", portanto, você não pode se referir a ela como \$EMAIL ou \$eMail. Todos os nomes de variáveis, em PHP, são "case-sensitive", por isso, é importante nomear as variáveis com cuidado e referenciá-las de forma consistente, em todo o código. Falaremos mais sobre nomes de variáveis em outro momento.

**P:** Realmente não tem problema colocar código PHP e HTML no mesmo arquivo?

**R:** Nenhum problema. Na verdade, em muitos casos, é absolutamente necessário fazer isso.

**P:** E por que colocar PHP e HTML no mesmo arquivo?

**R:** Porque o verdadeiro propósito um servidor web é servir páginas web HTML

para os navegadores. O PHP não muda esse fato. O que o PHP lhe permite fazer é modificar o conteúdo HTML em tempo real, inserindo dados como a data de hoje, informações retiradas de um banco de dados, ou até mesmo valores calculados, como o valor total do pedido, em um site de compras. Assim, o PHP permite que você manipule o HTML que compõe as páginas web, em vez de tê-las criadas estaticamente, apenas uma vez. É bastante comum ter uma página HTML com código PHP aqui e ali para inserir dados importantes, ou alterar programaticamente o HTML de alguma forma.

**P:** O código PHP inserido em um arquivo HTML precisa estar na sua própria linha ou posso colocá-lo em uma linha HTML, como parte do atributo de uma tag HTML, por exemplo?

**R:** Fora a necessidade de colocar o seu código PHP dentro das tags <?php e ?, não há restrições sobre como inserir esse código no HTML. Na verdade, é necessário colocar, frequentemente, um bloco de código PHP no meio do código HTML, por exemplo, quando está sendo configurado o atributo de uma tag HTML. Esse é um uso perfeitamente legítimo do PHP.

**P:** Já vi códigos PHP iniciados por <? como tag de abertura, em vez de <?php. Isso está certo?

**R:** Na verdade, não. Tecnicamente isso é permitido, mas não recomendado. É preciso que uma configuração do servidor esteja habilitada para que a tag abreviada (<?) funcione. A tag <? php usualmente funciona, então é melhor simplesmente usá-la, sabendo que o seu código vai funcionar.

**P:** Se o servidor web sempre retorna HTML puro para o navegador do cliente, por que as URLs mostram o nome do script PHP, como, por exemplo, paginaweb.php?

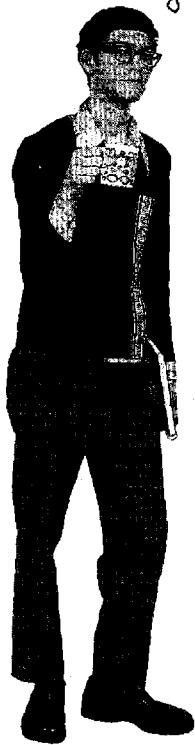
**R:** Lembre-se de que toda página web corresponde ao resultado de uma comunicação de duas vias, envolvendo uma requisição feita pelo navegador cliente e uma resposta do servidor web. A URL é a base da requisição, enquanto conteúdo retornado pelo servidor é a resposta. Os scripts PHP são requisitados da mesma forma que páginas HTML normais, através de URLs digitadas no navegador ou linkadas a partir de outras páginas, ou como ações de formulários. Isso explica porque a URL de uma "página" PHP mostra o nome do script.

A outra metade da equação é a resposta do servidor, que é o código resultante gerado pelo script PHP. Uma vez que a maioria dos scripts PHP gera HTML, faz sentido que esse código seja HTML e não PHP. Assim, não é por acidente que a URL referece um arquivo .php no servidor, o que faz o código PHP ser executado no servidor, resultando em um conteúdo puramente HTML, que é retornado ao navegador.

**P:** As variáveis PHP podem armazenar outros tipos de dados?

**R:** Certamente. Você pode usá-las para armazenar dados Booleanos (verdadeiro/falso). E dados numéricos podem ser tanto números inteiros quanto de ponto flutuante (decimais). Há também os "arrays", os quais armazenam um conjunto de dados, bem como objetos, os quais associam um conjunto de dados ao código usado para manipulá-los. Os "arrays" serão abordados um pouco mais adiante ainda neste capítulo, enquanto os "objetos" são assunto do Capítulo 12. Existe também um tipo de dado especial chamado NULL, que representa "nenhum valor". Por exemplo, uma variável que não tenha nenhum valor atribuído a ela é considerada NULL.

*adicionando dados perdidos no owen*



Ou a memória do PHP não é tão  
boa assim ou há algo de errado  
com o script... há alguns dados  
que não aparecem.

**Aliens Abducted Me - Report an Abduction**

Share your story of alien abduction:

First name: Alfin  
Last name: Nader  
What is your email address? alfin@theyrealgreen.com  
When did it happen? last November  
How long were you gone? 11 hours  
How many did you see? dozens  
Describe them: little green men  
What did they do to you? asked me about UFO regulations  
Have you seen my dog Fang? Yes  No



Please vote for me.

Claramente, foi  
digitada uma descrição  
(describe) dos aliens no  
formulário...

...mas a descrição não  
aparece na página de  
confirmação.

**Aliens Abducted Me - Report an Abduction**

You were abducted last November and were gone for 11 hours

Describe them:

Was Fang there? no

Your email address is alfin@theyrealgreen.com



## Aponte seu lápis

Há um problema com os dados referentes à descrição dos aliens, no script report.php de Owen. Circule as linhas do código que você acha que tem a ver com o problema e escreva o que elas estão causando. Alguma ideia do que há de errado?

```

<html>
<head>
    <title>Aliens Abducted Me - Report an Abduction</title>
</head>
<body>
    <h2>Aliens Abducted Me - Report an Abduction</h2>

    <?php
        $when_it_happened = $_POST['whenithappened'];
        $how_long = $_POST['howlong'];
        $alien_description = $_POST['description'];
        $fang_spotted = $_POST['fangspotted'];
        $email = $_POST['email'];

        echo 'Thanks for submitting the form.<br />';
        echo 'You were abducted ' . $when_it_happened;
        echo ' and were gone for ' . $how_long . '<br />';
        echo 'Describe them: ' . $alien_description . '<br />';
        echo 'Was Fang there? ' . $fang_spotted . '<br />';
        echo 'Your email address is ' . $email;
    ?>

    </body>
</html>

```



report.php

## Aponte seu lápis Solução

Há um problema com os dados referentes à descrição dos aliens, no script report.php de Owen. Circule as linhas do código que você acha que tem a ver com o problema e escreva o que elas estão causando. Alguma ideia do que há de errado?

```
<html>
<head>
    <title>Aliens Abducted Me - Report an Abduction</title>
</head>
<body>
    <h2>Aliens Abducted Me - Report an Abduction</h2>

    <?php
        $when_it_happened = $_POST['whenithappened'];
        $how_long = $_POST['howlong'];
        $alien_description = $_POST['description'];
        $fang_spotted = $_POST['fangspotted'];
        $email = $_POST['email'];

        echo 'Thanks for submitting the form.<br />';
        echo 'You were abducted ' . $when_it_happened;
        echo ' and were gone for ' . Show_long . '<br />';
        echo 'Describe them: ' . $alien_description . '<br />';
        echo 'Was Fang there? ' . $fang_spotted . '<br />';
        echo 'Your email address is ' . $email;
    ?>

</body>
</html>
```

Esta linha do código obtém a descrição dos aliens do respectivo campo no formulário HTML, e os dados armazena em uma variável PHP chamada \$alien\_description.

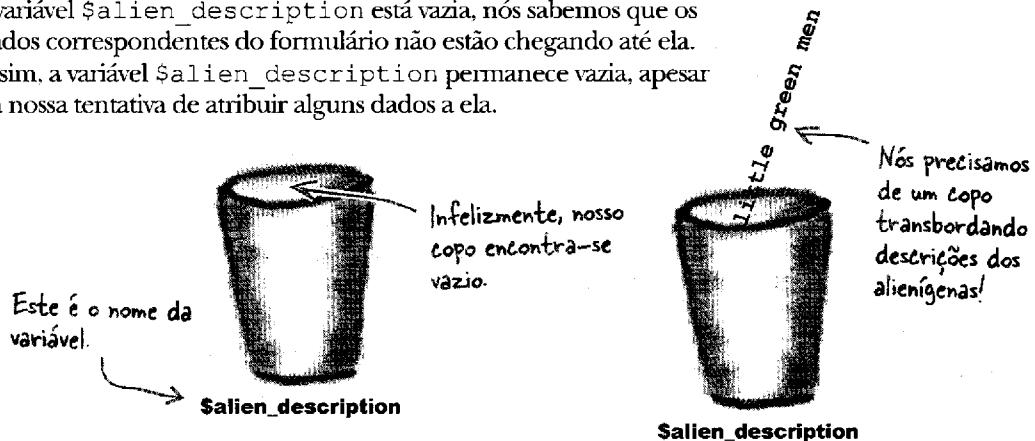
Este código combina a descrição dos aliens com alguns outros textos e códigos HTML e envia tudo isso para o navegador.

Por algum motivo, a variável \$alien\_description parece estar vazia... nada bom.



# Variáveis servem para armazenar dados de scripts

As variáveis PHP são como containers que armazenam informações, da mesma forma que um copo armazena uma bebida. Uma vez que a variável `$alien_description` está vazia, nós sabemos que os dados correspondentes do formulário não estão chegando até ela. Assim, a variável `$alien_description` permanece vazia, apesar da nossa tentativa de atribuir alguns dados a ela.



Uma forma de consertar o script seria atribuir a string exata que estamos esperando para a variável `$alien_description`, desta forma:

```
$alien_description = 'Homenzinhos verdes';
```

O sinal de igual instrui o PHP a atribuir o valor da direita à variável da esquerda.

Em PHP, os "pedaços" de texto, também chamados de strings, devem sempre ser colocados dentro de aspas, que podem ser simples ou duplas.

Este código armazena, definitivamente, o texto 'homenzinhos verdes' na variável `$alien_description`. Mas nós resolvemos um problema criando outro – este código faz a descrição ser sempre a mesma, independentemente do que o usuário digitar no formulário.

## PODER DO CÉREBRO

Por algum motivo, a atribuição dos dados do formulário referentes à descrição dos alienígenas à variável `$alien_description` não está funcionando.

```
$alien_description = $_POST['description'];
```

O que você acha que está errado com este código?



O problema obviamente tem algo a ver com aquele negócio de \$\_POST.  
Mas não tenho ideia do que possa ser.

**O problema, de fato, tem algo a ver com \$\_POST, que é um mecanismo usado para que os dados do formulário sejam transferidos para um script.**

O cifrão no início de \$\_POST é uma dica... \$\_POST é um container! Mais especificamente, \$\_POST é um conjunto de locais de armazenamento usados para que os dados de um formulário web fiquem armazenados. No caso de Owen, todos os dados enviados para script report.php são armazenados quando alguém preenche o formulário e clica no botão de "Relatar Abdução". Assim, para acessar os dados do formulário e fazer qualquer coisa com eles, nós temos que usar \$\_POST. Lembre-se deste código?

```
$when_it_happened = $_POST['whenithappened'];  
$how_long = $_POST['howlong']; ←  
$alien_description = $_POST['description'];  
$fang_spotted = $_POST['fangspotted'];  
$email = $_POST['email'];
```

↑  
Os dados do formulário  
referentes à duração da  
abdução são atribuídos à  
variável \$how\_long.

O mesmo acontece aqui,  
exceto pelo fato de que os  
dados referentes ao email  
estão sendo armazenados  
na variável \$email.

Assim, os dados de cada campo do formulário "Relatar uma Abdução" são acessados usando-se \$\_POST. Mas o que exatamente é \$\_POST... uma variável?

## \$\_POST é uma variável especial que armazena dados do formulário

`$_POST` é uma variável especial, conhecida também como **superglobal**, porque ela é interna ao PHP e fica disponível para o script inteiro. `$_POST` já existe quando o seu script é executado – você não a cria, como as outras variáveis no PHP.

**Aliens Abducted Me - Report an Abduction**

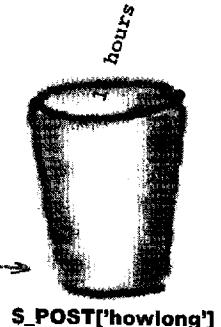
Share your story of alien abduction:

First name: Alf  
Last name: Nader  
What is your email address? alinghttheyreallgreen.com  
When did it happen? last November  
How long were you gone? 13 hours  
How many did you see? dozens  
Describe them: little green men  
What did they do to you? asked me about UFO regulations  
Have you seen my dog Fang? Yes  No



Please vote for me.  
 Report Abduction

A superglobal `$_POST` armazena todos os dados digitados no formulário.



`$_POST['howlong']`

```
<html>
<head>
<title>Aliens Abducted Me - Report an Abduction</title>
</head>
<body>
<h2>Aliens Abducted Me - Report an Abduction</h2>
<?php
$when_it_happened = $_POST['whenithappened'];
>Show_long = $_POST['howlong'];
$alien_description = $_POST['description'];
$fang_spotted = $_POST['fangspotted'];
$email = $_POST['email'];

echo 'Thanks for submitting the form.<br />';
echo 'You were abducted ' . $when_it_happened;
echo ' and were gone for ' . Show_long . '<br />';
echo 'Describe them: ' . $alien_description . '<br />';
echo 'Was Fang there? ' . $fang_spotted . '<br />';
echo 'Your email address is ' . $email;
?>

</body>
</html>
```



A superglobal `$_POST` é vinculada diretamente ao método de submissão usado pelo formulário HTML. Se o método tiver sido definido como `post`, então todos os dados são empacotados dentro da superglobal `$_POST`, onde cada informação pode ser retirada e usada conforme necessário.

...

`<form method="post" action="report.php">`

...



O método de submissão do formulário determina o modo como os dados são fornecidos ao script PHP.

O nome “`howlong`” (“por quanto tempo”) vem do atributo `name` usado na tag `<input>` correspondente a este campo do formulário.

### ~~PODER DO CÉREBRO~~

*e Amor*  
Como você acha que a superglobal `$_POST` funciona? Como ela é capaz de armazenar múltiplos valores originários de todas aquelas caixas de texto do formulário de Owen?

`$_POST` é como um array

## `$_POST` transporta os dados do formulário para o seu script

`$_POST` é um tipo especial de container PHP, conhecido como **array**, que armazena um conjunto de variáveis sob um único nome. Quando o usuário submete o formulário de Owen, os dados que ele digitou nos campos são armazenados no array `$_POST`, cujo trabalho é repassar esses dados para o script.

Cada elemento do array `$_POST` corresponde a uma determinada informação, digitada em um determinado campo do formulário. Para acessar os dados referentes a um campo específico, você usa o nome desse campo com `$_POST`. Por exemplo, a duração da abdução é armazenada em `$_POST['howlong']`. O código HTML do formulário de Owen revela o modo como os nomes do formulário se relacionam com os dados armazenados em `$_POST`.

```
<p>Share your story of alien abduction:</p>
<form method="post" action="report.php">
  <label for="firstname">First name:</label>
  <input type="text" id="firstname" name="firstname" /><br />
  <label for="lastname">Last name:</label>
  <input type="text" id="lastname" name="lastname" /><br />
  <label for="email">What is your email address?</label>
  <input type="text" id="email" name="email" /><br />
  <label for="whenithappened">When did it happen?</label>
  <input type="text" id="whenithappened" name="whenithappened" /><br />
  <label for="howlong">How long were you gone?</label>
  <input type="text" id="howlong" name="howlong" /><br />
  <label for="howmany">How many did you see?</label>
  <input type="text" id="howmany" name="howmany" /><br />
  <label for="aliendescription">Describe them:</label>
  <input type="text" id="aliendescription" name="aliendescription" size="32" /><br />
  <label for="whattheydid">What did they do to you?</label>
  <input type="text" id="whattheydid" name="whattheydid" size="32" /><br />
  <label for="fangspotted">Have you seen my dog Fang?</label>
  Yes <input id="fangspotted" name="fangspotted" type="radio" value="yes" />
  No <input id="fangspotted" name="fangspotted" type="radio" value="no" /><br />
  <br />
  <label for="other">Anything else you want to add?</label>
  <textarea name="other"></textarea><br />
  <input type="submit" value="Report Abduction" name="submit" />
</form>
```

### Aliens Abducted Me - Report an Abduction

Share your story of alien abduction:

First name:  
Last name:  
What is your email address?  
When did it happen?  
How long were you gone?  
How many did you see?  
Describe them:  
What did they do to you?  
Have you seen my dog Fang?

Ali  
Name  
info@theyreal.org.com  
last November  
11 hours  
dotted  
little green men  
told me about UFO regulations  
Yes  No



## Aponte seu lápis

Examine o código de report.php que está fazendo a descrição dos aliens ficar em branco e depois escreva a solução para o problema. Dica: use o código HTML do formulário exposto na página anterior para ajudar a isolar o problema.

```

<html>
<head>
<title>Aliens Abducted Me - Report an Abduction</title>
</head>
<body>
<h2>Aliens Abducted Me - Report an Abduction</h2>

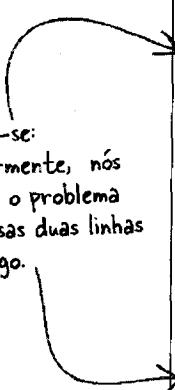
<?php
$when_it_happened = $_POST['whenithappened'];
$how_long = $_POST['howlong'];
$alien_description = $_POST['description'];
$fang_spotted = $_POST['fangspotted'];
$email = $_POST['email'];

echo 'Thanks for submitting the form.<br />';
echo 'You were abducted ' . $when_it_happened;
echo ' and were gone for ' . $how_long . '<br />';
echo 'Describe them: ' . $alien_description . '<br />';
echo 'Was Fang there? ' . $fang_spotted . '<br />';
echo 'Your email address is ' . $email;
?>

</body>
</html>

```

Lembre-se:  
anteriormente, nós  
isolamos o problema  
para essas duas linhas  
de código.




report.php



## Aponte seu lápis Solução

Examine o código de report.php que está fazendo a descrição dos aliens ficar em branco e depois escreva a solução para o problema. Dica: use o código HTML do formulário exposto na página anterior para ajudar a isolar o problema.

```
...<input type="text" id="aliendescription" name="aliendescription" size="32" />...</head><body><h2>Aliens Abducted Me - Report an Abduction</h2><?php$when_it_happened = $_POST['whenithappened'];$show_long = $_POST['howlong'];$alien_description = $_POST['aliendescription'];$fang_spotted = $_POST['fangspotted'];$email = $_POST['email'];echo 'Thanks for submitting the form.<br />';echo 'You were abducted ' . $when_it_happened;echo ' and were gone for ' . $show_long . '<br />';echo 'Describe them: ' . $alien_description . '<br />';echo 'Was Fang there? ' . $fang_spotted . '<br />';echo 'Your email address is ' . $email;?></body></html>
```

O nome do campo em report.html é "aliendescription", diferente do nome "description" usado com \$\_POST.

Nós precisamos modificar \$\_POST para corrigir o nome do campo: "aliendescription".

report.html

report.php



## TEST DRIVE

### Conserte e teste o script.

Modifique a linha de código problemática em report.php e depois a envie para o seu servidor web. Abra a página report.html no seu navegador, preencha o formulário com as informações do seu caso de abdução e clique no botão “Relatar Abdução” para submeter o formulário agora corrigido script.

**Aliens Abducted Me - Report an Abduction**

Share your story of alien abduction:

First name: Alf  
Last name: Nader  
What is your email address? alfr@theyreallgreen.com  
When did it happen? last November  
How long were you gone? 11 hours  
How many did you see? dozens  
Describe them: little green men  
What did they do to you? asked me about UFO regulations  
Have you seen my dog Fang? Yes  No

**Aliens Abducted Me - Report an Abduction**

Anything else you want to add?

You were abducted last November and were gone for 11 hours.  
Describe them: little green men  
Was Fang there? no  
Your email address is alfr@theyreallgreen.com

Agora, página de confirmação exibe corretamente os dados referentes à descrição dos aliens!

Excelente. Mas acontece que ainda estão faltando alguns dados do formulário...



## Aponte seu lápis

Há alguns dados digitados no formulário "Relatar uma Abdução" que ainda não estamos usando. Lembre-se, esses dados incluem informações vitais sobre um caso de abdução, as quais poderiam ajudar Owen a encontrar seu cão Fang. Portanto, precisamos acessar ou filtrar todos os dados do caso e armazená-los em variáveis PHP.

Atualmente o script report.php está ignorando cinco campos do formulário. Inacreditável!

```
...
<form method="post" action="report.php">
    <label for="firstname">First name:</label>
    <input type="text" id="firstname" name="firstname" />
    <label for="lastname">Last name:</label>
    <input type="text" id="lastname" name="lastname" />
    <label for="email">What is your email address?</label>
    <input type="text" id="email" name="email" /><br />
    <label for="whenithappened">When did it happen?</label>
    <input type="text" id="whenithappened" name="whenithappened" value="last November" />
    <label for="howlong">How long were you gone?</label>
    <input type="text" id="howlong" name="howlong" /><br />
    <label for="howmany">How many did you see?</label>
    <input type="text" id="howmany" name="howmany" /><br />
    <label for="aliendescription">Describe them:</label>
    <input type="text" id="aliendescription" name="aliendescription" size="32" /><br />
    <label for="whattheydid">What did they do to you?</label>
    <input type="text" id="whattheydid" name="whattheydid" size="32" /><br />
    <label for="fangspotted">Have you seen my dog Fang?</label>
    Yes <input id="fangspotted" name="fangspotted" type="radio" value="yes" />
    No <input id="fangspotted" name="fangspotted" type="radio" value="no" /><br />
    
    <label for="other">Anything else you want to add?</label>
    <textarea id="other" name="other"></textarea><br />
    <input type="submit" value="Report Abduction" name="submit" />
</form>
</body>
</html>
```



Escreva o código PHP a fim de criar quatro novas variáveis que armazenem os dados do formulário que estão faltando: \$name, \$how\_many, \$what\_they\_did e \$other. Dica: Crie a variável \$name de forma que ela armazene o nome completo do usuário.

\$name = \$\_POST['firstname'] . ' ' . \$\_POST['lastname'];  
 \$how\_many = \$\_POST['howmany'];  
 \$how\_long = \$\_POST['howlong'];  
 \$other = \$\_POST['other'];

A tag `<input>` de cada campo do formulário é a chave para acesso aos dados, a partir do PHP.

**Aliens Abducted Me - Report an Abduction**

Share your story of alien abduction:

First name:	Alf
Last name:	Nader
What is your email address?	alf@theyrallgreen.com
When did it happen?	last November
How long were you gone?	21 hours
How many did you see?	dozens
Describe them:	little green men
What did they do to you?	asked me about UFO regulations
Have you seen my dog Fang?	Yes <input checked="" type="radio"/> No <input type="radio"/>

Anything else you want to add?

Please vote for me.

Seu trabalho ainda não terminou. A página de confirmação gerada pelo script PHP precisa usar essas novas variáveis para exibir mais informações sobre a abdução.

Precisamos partir disto...

Aliens Abducted Me - Report an Abduction

You were abducted last November and were gone for 11 hours  
Describe them: little green men  
Was Fang there? no  
Your email address is alfn@theyreallgreen.com

...para isto! Repare na quantidade de novas informações exibidas.

Aliens Abducted Me - Report an Abduction

Thanks for submitting the form.  
You were abducted last November and were gone for 11 hours  
Number of aliens: dozens  
Describe them: little green men  
The aliens did this: asked me about UFO regulations  
Was Fang there? no  
Other comments: Please vote for me.  
Your email address is alfn@theyreallgreen.com

O nome do usuário não é essencial para a página de confirmação, mas você precisará dele posteriormente, quando enviarmos o email com os dados para Owen.

Usando todas as variáveis que acabou de criar, exceto \$name, complete o código abaixo, o qual irá gerar uma página de confirmação mais informativa.

```
echo 'Thanks for submitting the form.<br />';  
echo 'You were abducted ' . $when_it_happened;  
echo ' and were gone for ' . $how_long . '<br />';  
echo 'Describe them: ' . $alien_description . '<br />';  
echo 'What they did? ' . $what_they_did . '<br />';  
echo 'Was Fang there? ' . $fang_spotted . '<br />';  
echo 'Anything else? ' . $other . '<br />';  
echo 'Your email address is ' . $email;
```



# Aponte seu lápis Solução

Há alguns dados digitados no formulário "Relatar uma Abdução" que ainda não estamos usando. Lembre-se: esses dados incluem informações vitais sobre um caso de abdução, as quais poderiam ajudar Owen a encontrar seu cão Fang. Portanto, precisamos acessar ou filtrar todos os dados do caso e armazená-los em variáveis PHP.

Atualmente o script report.php está ignorando cinco campos do formulário. Inacreditável!

```
...
<form method="post" action="report.php">
    <label for="firstname">First name:</label>
    <input type="text" id="firstname" name="firstname" />
    <label for="lastname">Last name:</label>
    <input type="text" id="lastname" name="lastname" />
    <label for="email">What is your email address?</label>
    <input type="text" id="email" name="email" /><br />
    <label for="whenithappened">When did it happen?</label>
    <input type="text" id="whenithappened" name="whenithappened" />
    <label for="howlong">How long were you gone?</label>
    <input type="text" id="howlong" name="howlong" /><br />
    <label for="howmany">How many did you see?</label>
    <input type="text" id="howmany" name="howmany" /><br />
    <label for="aliendescription">Describe them:</label>
    <input type="text" id="aliendescription" name="aliendescription" size="32" /><br />
    <label for="whattheydid">What did they do to you?</label>
    <input type="text" id="whattheydid" name="whattheydid" size="32" /><br />
    <label for="fangspotted">Have you seen my dog Fang?</label>
    Yes <input id="fangspotted" name="fangspotted" type="radio" value="yes" />
    No <input id="fangspotted" name="fangspotted" type="radio" value="no" /><br />
    <br />
    <label for="other">Anything else you want to add?</label>
    <textarea id="other" name="other"></textarea><br />
    <input type="submit" value="Report Abduction" name="submit" />
</form>
</body>
</html>
```

report.html

O ponto final lhe permite juntar várias strings de texto em uma só - um processo conhecido como concatenação.

## Aliens Abducted Me - Report an Abduction

Share your story of alien abduction:

First name:	<input style="width: 100px;" type="text" value="Ali"/>
Last name:	<input style="width: 100px;" type="text" value="Nader"/>
What is your email address?	<input style="width: 100px;" type="text" value="ali@theyreallgreen.com"/>
When did it happen?	<input style="width: 100px;" type="text" value="last November"/>
How long were you gone?	<input style="width: 100px;" type="text" value="21 hours"/>
How many did you see?	<input style="width: 100px;" type="text" value="dozens"/>
Describe them:	<input style="width: 100px;" type="text" value="little green men"/>
What did they do to you?	<input style="width: 100px;" type="text" value="asked me about UFO regulations"/>
Have you seen my dog Fang?	<input checked="" type="radio" value="Yes"/> Yes <input type="radio" value="No"/> No



Anything else you want to add?

Please vote for me.

A tag `<input>` de cada campo do formulário é a chave para acesso aos dados, a partir do PHP

Escreva o código PHP a fim de criar quatro novas variáveis que armazenem os dados do formulário que estão faltando: \$name, \$show\_many, \$what\_they\_did e \$other. Dica: Crie a variável \$name de forma que ela armazene o **nome completo** do usuário.

```
$name = $_POST['firstname'] . ' ' . $_POST['lastname'];
$show_many = $_POST['howmany'];
$what_they_did = $_POST['whattheydid'];
$other = $_POST['other'];
```

Este espaço separa o primeiro nome do sobrenome.

Seu trabalho ainda não terminou. A página de confirmação gerada pelo script PHP precisa usar essas novas variáveis para exibir mais informações sobre a abdução.

Precisamos partir disto...

...para isto! Repare na quantidade de novas informações exibidas.

### Aliens Abducted Me - Report an Abduction

You were abducted last November and were gone for 11 hours  
 Describe them: little green men  
 Was Fang there? no  
 Your email address is alfn@theyreallgreen.com

### Aliens Abducted Me - Report an Abduction

Thanks for submitting the form.  
 You were abducted last November and were gone for 11 hours  
 Number of aliens: dozens  
 Describe them: little green men  
 The aliens did this: asked me about UFO regulations  
 Was Fang there? no  
 Other comments: Please vote for me.  
 Your email address is alfn@theyreallgreen.com

O nome do usuário não é essencial para a página de confirmação, mas você precisará dele posteriormente, quando enviarmos o email com os dados para Owen.

O comando echo é usado para enviar as informações ao navegador, na forma de conteúdo HTML.

Usando todas as variáveis que acabou de criar, exceto \$name, complete o código abaixo, o qual irá gerar uma página de confirmação mais informativa.

As tags <br /> ajudam a formatar as informações – não se esqueça de que estamos usando PHP para criar HTML.

```
echo 'Thanks for submitting the form.<br />';
echo 'You were abducted ' . $when_it_happened;
echo ' and were gone for ' . $how_long . '<br />';
echo 'Number of aliens: ' . $how_many . '<br />';
```

Novamente, pontos finais são usados para concatenar strings e variáveis.

```
echo 'Describe them: ' . $alien_description . '<br />';
echo 'The aliens did this: ' . $what_they_did . '<br />';
echo 'Was Fang there? ' . $fang_spotted . '<br />';
echo 'Other comments: ' . $other . '<br />';

echo 'Your email address is ' . $email;
```



## **TEST DRIVE**

### **Ajuste o script de Owen e teste as modificações.**

Adicione o código das novas variáveis a `report.php`, bem como o código que envia as variáveis para o navegador, com o HTML formatado. Em seguida, envie o script para o seu servidor web, abra a página `report.html` no seu navegador e preencha o formulário com as informações do seu caso de abdução. Finalmente, clique no botão "Relatar Abdução" para submeter o formulário e ver os resultados.

— não existem —

### **Perguntas Ídiotas**

**P:** O que realmente acontece quando eu faço a concatenação de várias strings, usando pontos finais?

**R:** Concatenar, neste caso, significa juntar diversas strings para formar outra, totalmente diferente. O resultado final é sempre uma única string, não importa quantas tenham existido originalmente. Assim, quando você concatena strings como parte de um comando `echo`, o PHP, primeiramente, as combina em uma só string e depois envia esta última para o navegador.

**P:** Quando eu concateno uma variável com uma string, a variável precisa conter texto?

**R:** Não. Embora a concatenação sempre resulte em uma string, as variáveis não precisam conter strings para que você possa usá-las em um processo de concatenação. Por exemplo, digamos que uma variável contenha um número; o PHP primeiramente converte o número em uma string e depois a inclui na concatenação.

**P:** O que acontece com o código PHP no navegador?

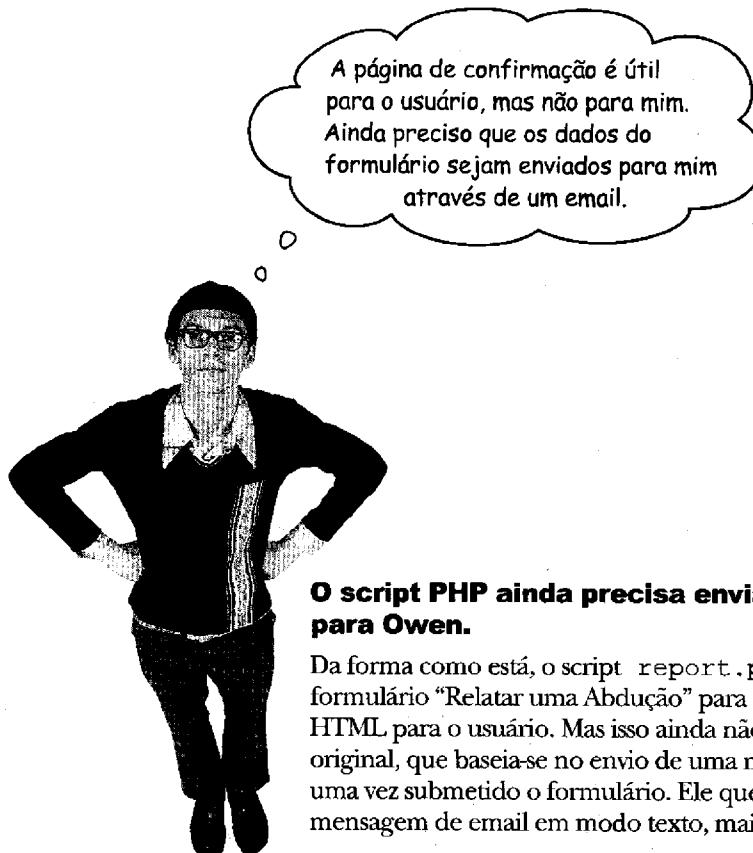
**R:** Nada. E isso porque o código PHP nunca é visto pelo navegador. O código PHP é executado no servidor e se transforma em código HTML, este sim sendo enviado para o navegador. Assim, o navegador nem sequer fica sabendo da existência do PHP – as páginas web chegam como puros HTML e CSS.

**P:** OK, então como, exatamente, o servidor transforma o código PHP em códigos HTML e CSS?

**R:** Em primeiro lugar, lembre-se de que, por padrão, assume-se que o código em um script PHP é código HTML. Você pode identificar o código PHP dentro de um script colocando-o entre tags `<?php` e `?>`. O servidor vê essas tags e entende que deve rodar o código dentro delas como PHP e todo o código fora dessas tags é informado ao navegador como sendo um HMTL.

**P:** Certo. Mas isso ainda não explica como o código PHP é transformado em HTML/CSS. E então?

**R:** Ah, é aí que o comando `echo` entra em cena. Você pode pensar no `echo` como um comando para enviar as informações que estejam fora do domínio das tags `<?php` e `?>`. Assim, o comando `echo` representa a chave para a capacidade do PHP de gerar código HTML/CSS, dinamicamente. Concatenando strings de texto com variáveis PHP, você pode construir código HTML em tempo real e depois usar `echo` para enviá-lo ao navegador como parte da página web resultante. Um bom exemplo disso, no script `report.php`, é quando a tag `<br />` é anexada ao final de um pedaço de texto para gerar uma quebra de linha no HTML resultante.



### O script PHP ainda precisa enviar os dados do formulário para Owen.

Da forma como está, o script `report.php` está usando os dados do formulário "Relatar uma Abdução" para gerar uma página de confirmação HTML para o usuário. Mas isso ainda não está resolvendo o problema original, que baseia-se no envio de uma mensagem por email para Owen, uma vez submetido o formulário. Ele quer apenas receber uma simples mensagem de email em modo texto, mais ou menos como a seguinte:

De forma semelhante à página web de confirmação, esta mensagem de email consiste em texto estático, combinado com dados do formulário.

Alf Nader foi abduzido em novembro passado e esteve ausente por 11 horas.  
Número de aliens: dúzias  
Descrição dos aliens: homenzinhos verdes  
O que eles fizeram: me perguntaram sobre leis para as OVNIs  
Fang foi visto: não  
Outros comentários: Por favor vote em mim.

Esta mensagem de email pode ser gerada a partir de código PHP, criando-se uma string que combine textos estáticos como "Outros comentários:" com dados do formulário que estejam armazenados em variáveis.

Escreva aqui como você criaria uma mensagem de email a partir de texto estático e variáveis PHP.

\$msg = .....

## Criando o corpo da mensagem de email com PHP

Você já viu como um ponto final pode ser usado em códigos PHP para concatenar várias strings de texto em uma só. Agora precisa usar novamente a concatenação para criar uma string que será a mensagem de email, com variáveis sendo usadas juntamente com texto estático.

A maioria dos editores de texto, automaticamente, quebra o texto para a linha seguinte, mesmo que você não inclua no código a sua própria quebra de linha (return).

Variáveis e texto estático  
são concatenados em uma  
única string (a mensagem  
de email) usando-se pontos  
finals.

```
$msg = $name . ' was abducted ' . $when_it_happened . ' and was gone for ' . $how_long .  
'Number of aliens:' . $num_of_aliens . 'Alien description:' . $alien_description . 'What they did:' .  
$what_they_did . 'Fang spotted:' . $fang_spotted . 'Other comments:' . $other;
```

Um problema ao se criar uma string desse tamanho é que ela demanda uma grande linha de código PHP, difícil de ler e entender. Você pode dividir o código PHP em várias linhas, para deixá-lo mais fácil de ser estudado. Certifique-se, apenas, de separar o código em pontos onde o espaçamento não faça diferença, como por exemplo, entre duas strings concatenadas e não no meio de uma string. Em seguida, coloque um ponto e vírgula no final da última linha de código, para finalizar a instrução PHP.

Lembre-se: cada  
variável armazena  
uma string de texto  
retirada do formulário  
“Relatar uma Abdução”.

Isto na verdade é apenas uma grande linha  
de código dividida em várias linhas.

```
$msg = $name . ' was abducted ' . $when_it_happened . ' and was gone for ' . $how_long .  
'Number of aliens:' . $num_of_aliens .  
'Alien description:' . $alien_description .  
'What they did:' . $what_they_did .  
'Fang spotted:' . $fang_spotted .  
'Other comments:' . $other;
```

A linha de código é cuidadosamente  
dividida, evitando-se separar uma  
string no meio.

Você ainda precisa finalizar a  
instrução, com um ponto e vírgula.

Quando uma linha de código  
PHP é deliberadamente,  
dividida em várias linhas,  
costuma-se indentar as linhas  
após a primeira, para que se  
perceba, com mais facilidade,  
quais linhas compõem uma  
mesma instrução no seu código.

Uma longa linha de código PHP pode  
ser dividida em várias linhas, desde que  
você tenha cuidado na hora de separar  
o código.



Esse código PHP certamente ficou bonito. Mas sem formatação, a mensagem de email não vai ficar toda bagunçada?

**Sim. Só porque o código PHP está bem organizado, não quer dizer que o seu output (saída), automaticamente, também ficará bonito.**

Organizar o código PHP para que você possa entendê-lo melhor é completamente diferente de formatar o output que os usuários irão ver. Você normalmente usará tags HTML para formatar o output do código PHP, uma vez que, na maioria dos casos, o PHP é usado para se gerar, dinamicamente, uma página web. Porém não neste caso.

Aqui, estamos gerando uma mensagem de email, que é puro texto, e não HTML. Precisamos lidar com o fato de que a mensagem, atualmente, se parece com o resultado seguinte:

Alf Nader foi abduzido em novembro passado e esteve ausente por 11 horas. Número de aliens: dúzias Descrição dos aliens: homenzinhos verdes O que eles fizeram: me perguntaram sobre leis para as OVNIIsFang foi visto: nãoOutros comentários: Por favor vote em mim.

Hm... isto NÃO é o que Owen tinha em mente para a sua mensagem de e-mail.

não existem

## Perguntas Idiotas

**P:** Existe alguma maneira de se usar formatação HTML em emails que você envia a partir de um script PHP?

**R:** Existe. Mas isso requer um passo adicional, que envolve definir o cabeçalho do tipo de conteúdo para a mensagem. Cabeçalhos e tipos de conteúdo vão um pouco além do escopo desta discussão, por isso, estamos nos limitando a mensagens de email em formato de texto simples. Você aprenderá mais sobre os cabeçalhos no Capítulo 6, portanto, não se preocupe – você verá como enviar emails em HTML mais adiante.

## PODER DO CÉREBRO

Como você reformataria a mensagem de email para torná-la mais fácil de ler?

## Até mesmo texto simples pode ser formatado... um pouco

Uma vez que Owen está enviando mensagens de email como texto simples, sem nenhuma formatação especial HTML, ele não pode colocar tags `<br />` para adicionar quebras de linha para separar o conteúdo. Mas ele pode usar caracteres newline (nova linha), que são escapados na forma `\n`. Assim, em qualquer lugar do email onde apareça `\n`, será inserida uma quebra de linha, fazendo o conteúdo que vier a seguir se iniciar na linha seguinte. Eis o novo código para a mensagem de email, com os newlines adicionados:

`\n` é usado para  
que sejam inseridos  
caracteres newline na  
mensagem de email.

```
$msg = $name . ' was abducted ' . $when_it_happened . ' and was gone for ' . Show_long . '\n' .  
'Number of aliens: ' . Show_many . '\n' .  
'Alien description: ' . $alien_description . '\n' .  
'What they did: ' . $what_they_did . '\n' .  
'Fang spotted: ' . $fang_spotted . '\n' .  
'Other comments: ' . $other;
```

Os newlines me  
parecem uma boa ideia...  
pena que o código não  
parece funcionar.

Alf Nader foi abduzido em novembro passado e esteve  
ausente por 11 horas.  
Número de aliens: dúzias  
Descrição dos aliens: homenzinhos verdes  
O que eles fizeram: me perguntaram sobre leis para as OVNIs  
Fang foi visto: não  
Outros comentários: Por favor vote em mim.

O `\n` está aparecendo  
como texto normal,  
em vez de um  
caracter newline...  
nada bom.

P: O que é, exatamente, um caracter de escape?

R: Um caracter de escape é um caracter que: ou é difícil de digitar ou que, por algum motivo, poderia causar confusão no código PHP. Você pode estar familiarizado com os caracteres de escape do HTML, programados de uma forma um pouco diferente, como por exemplo, `&#169` ou `&copy` para o símbolo de copyright. O PHP tem um conjunto bastante pequeno de caracteres de escape que são úteis para se escrever coisas que poderiam se confundir com a própria linguagem, como por exemplo, aspas simples (`'`), aspas duplas (`"`) e, é claro, newlines (`\n`).

*IMPORTANTE*

## Newlines precisam de aspas duplas

O problema com o código de Owen é que o PHP trata as strings de forma diferente, conforme elas se apresentem: dentro de aspas simples ou duplas. Mais especificamente, os caracteres newline (\n) só podem ser escapados dentro de aspas duplas. Assim, a mensagem de email com os dados do formulário precisa ser construída usando-se strings dentro de aspas duplas, para as newlines funcionarem.

Mas a história das aspas simples x duplas não termina aí. As strings dentro de aspas simples são consideradas como texto puro, enquanto que o PHP processa as strings dentro de aspas duplas em busca de variáveis. Quando uma variável é encontrada dentro de uma string em aspas duplas, o PHP insere o seu valor dentro da string, como se tivesse ocorrido uma "concatenação". Assim, a string dentro de aspas duplas não só é necessária para o funcionamento dos newlines na mensagem de email, como também nos permite simplificar o código, colocando as variáveis diretamente na string.

```
$msg = "$name was abducted $when_it_happened and was gone for $how_long.\n"
```

```
"Number of aliens: $how_many\n" .
```

```
"Alien description: $alien_description\n"
```

```
"What they did: $what_they_did\n" .
```

```
"Fang spotted: $fang_spotted\n" .
```

```
"Other comments: $other";
```

Não é preciso de uma nova linha quando for a linha final.

A concatenação não é mais necessária, uma vez que as variáveis podem ser referenciadas diretamente dentro da string entre aspas duplas.

Os caracteres newline agora são interpretados corretamente, graças à string apresentada entre aspas duplas.

Mas ainda precisamos dividir a mensagem em várias strings concatenadas para tornar o código mais legível.

não existem

## Perguntas Ídiotas

**P:** Se as aspas duplas são tão legais, por que é que por que fizemos uso, praticamente, aspas simples, até agora?

**R:** Bem, tenha em mente que as aspas simples não são processadas pelo PHP de nenhuma maneira, o que as torna ideais para strings que são puro texto, sem nenhuma variável inserida. Assim, continuaremos a usar aspas simples ao longo do livro, a não ser que haja um bom motivo para usar as duplas. A coisa mais importante nesta questão de usar aspas simples ou duplas para as strings, é tentar manter-se o mais consistente possível.

**P:** O que acontece se eu tiver de usar um apóstrofo dentro de uma string que esteja dentro de aspas simples, como por exemplo, em 'He's lost!' ("ele está perdido!")?

**R:** É aí que os caracteres de escape são úteis. Para

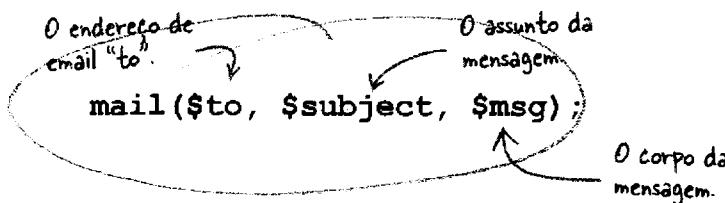
usar um apóstrofo em uma string dentro de aspas simples, basta fazer escape dele como '\', desta forma: 'He\'s lost!'. O mesmo se aplica se você quiser usar uma só aspa dupla dentro de uma string delimitada por aspas duplas – use '\"'. Não é preciso fazer escape das aspas quando não houver conflito, como por exemplo, ao usar um apóstrofo em uma string contida entre aspas duplas: "He's lost!".

**P:** Então, com as aspas simples, pode-se usar '\'' mas não '\n'. Como saber quais caracteres de escape eu posso usar dentro de aspas simples?

**R:** As strings dentro de aspas simples só permitem os caracteres de escape '\'' e '\"'. Todos os outros caracteres de escape só podem ser usados em strings dentro de aspas duplas.

## Enviando uma mensagem de email com PHP

Então, você está pronto para escrever o código PHP que enviará a mensagem a Owen por email. Isto requer a função interna mail() do PHP, a qual envia uma mensagem com base em informações que você fornece.



A função mail(), do PHP, envia uma mensagem de email a partir de um script.

Essas três informações são requeridas pela função mail(), portanto, você sempre tem de fornecê-las. O endereço de email "from" não é exigido, mas ainda é uma boa ideia incluí-lo. Para especificar o campo "from" ao chamar a função mail(), é preciso usar um outro argumento da função, junto a alguma concatenação de strings.

O texto 'From:' precisa ser colocado antes do endereço, ao se especificar o endereço de email do remetente.

```
mail($to, $subject, $msg, 'From: ' . $email);
```

O ponto final é útil mais uma vez, para concatenar 'From:' com o endereço de email de Owen.

não existem

## Perguntas Ídiotas

**P:** Há mais alguma coisa que possa ser especificada como parte de uma mensagem de email, além do endereço "from"?

**R:** Sim. Você pode especificar também recipientes "copy" ("cópia") e "blind copy" ("cópia oculta") da mesma forma que o "from" – basta usar 'Cc:' ou 'Bcc:' em vez de 'From:'. Se quiser especificar tanto um "from" quanto um "copy", você terá de separá-los com uma combinação de caracteres return e newline (\r\n), desta forma:

É isso mesmo, dois caracteres de escape colados um no outro!

"From:" . \$from . "\r\nCc:" . \$cc

Precisamos usar aspas duplas aqui, uma vez que estamos usando os caracteres de escape \r e \n.



Mas então, como é que  
nós usamos, realmente,  
a função mail()?

### Basta adicionar ao seu script o código para chamar mail().

A linha de código que chama a função mail() é tudo o que você precisa para enviar a mensagem de email. Certifique-se de que este código apareça no script após o código que cria as variáveis para o email, e pronto. Eis aqui o código completo para o script report.php de Owen, incluindo a chamada para a função mail().

```

<html>
<head>
    <title>Aliens Abducted Me - Report an Abduction</title>
</head>
<body>
    <h2>Aliens Abducted Me - Report an Abduction</h2>

<?php
    $name = $_POST['firstname'] . ' ' . $_POST['lastname'];
    $when_it_happened = $_POST['whenithappened'];
    $show_long = $_POST['howlong'];
    $show_many = $_POST['howmany'];
    $alien_description = $_POST['aliendescription'];
    $what_they_did = $_POST['whattheydid'];
    $fang_spotted = $_POST['fangspotted'];
    $email = $_POST['email'];
    $other = $_POST['other'];

    $to = 'owen@aliensabductedme.com';
    $subject = 'Aliens Abducted Me - Abduction Report';
    $msg = "$name was abducted $when_it_happened and was gone for $show_long.\n";
    $msg .= "Number of aliens: $show_many\n";
    $msg .= "Alien description: $alien_description\n";
    $msg .= "What they did: $what_they_did\n";
    $msg .= "Fang spotted: $fang_spotted\n";
    $msg .= "Other comments: $other";

    mail($to, $subject, $msg, 'From:' . $email);

    echo 'Thanks for submitting the form.<br />';
    echo 'You were abducted ' . $when_it_happened;
    echo ' and were gone for ' . $show_long . '<br />';
    echo 'Number of aliens: ' . $show_many . '<br />';
    echo 'Describe them: ' . $alien_description . '<br />';
    echo 'The aliens did this: ' . $what_they_did . '<br />';
    echo 'Was Fang there? ' . $fang_spotted . '<br />';
    echo 'Other comments: ' . $other . '<br />';
    echo 'Your email address is ' . $email;
?>

</body>
</html>

```

Envie a  
mensagem.

Gere uma  
página HTML,  
em tempo real,  
para confirmar  
que o formulário  
foi submetido  
com sucesso.

Pegue todos os dados  
do formulário no array  
\$\_POST e coloque-os  
em variáveis individuais.

Certifique-se de  
trocar este endereço  
de email pelo seu  
próprio, para testar o  
script.

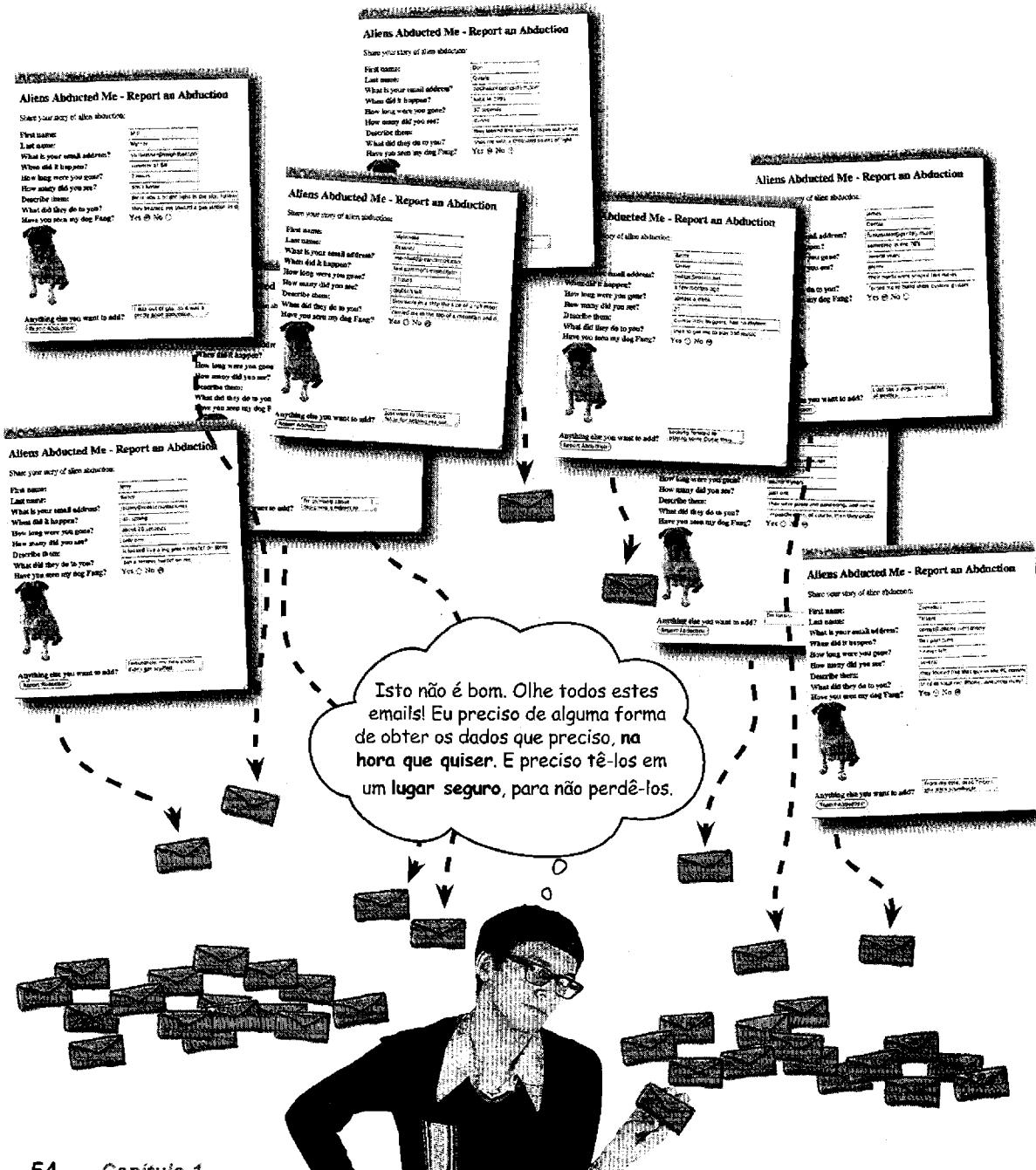
Junte os diferentes  
pedaços da  
mensagem a ser  
enviada para Owen.



report.php

## Owen começa a perder emails

A boa notícia é que Owen está recebendo emails. A má notícia é que ele está recebendo muitos, muitos emails. Tantos que está tendo dificuldade em acompanhá-los. A sua Caixa de Entrada está lotada, e ele já apagou alguns acidentalmente... Owen precisa de uma forma melhor de armazenar os dados sobre as abduções.



## QUAIS FAZ OS ALIENS?

Os aliens mexeram com o seu cérebro? Coloque as coisas no lugar ligando cada componente HTML ou PHP à descrição do que eles fazem.

HTML

Um software que serve para visualizar e interagir com páginas web. Age como o lado do cliente em comunicações na web.

PHP

Um comando PHP usado para fazer o output de algum conteúdo, como, por exemplo, texto simples ou código HTML.

formulário Web

Estas tags são usadas para delimitar o código PHP a fim de que o servidor web saiba que deve processá-lo e executá-lo.

navegador

Um array interno do PHP que armazena os dados submetidos usando o método "post".

<?php ?>

Uma linguagem de programação usada para a criação de scripts, que são executados em um servidor web.

variável

Toda string precisa estar dentro de uma variável.

aspas

Um software para enviar páginas web, o qual age como um servidor em comunicações na web.

echo

Uma linguagem marcação usada para descrever a estrutura do conteúdo das páginas web, que são visualizadas em um navegador web.

\$\_POST

Um nome usado para descrever as variáveis internas do PHP que ficam acessíveis a todos os scripts.

servidor web

Uma série de campos de input, em uma página web, usados para se obter informações dos usuários.

array

Uma função interna do PHP que envia mensagens de email.

superglobal

Um local de armazenamento, em um script PHP, que tem o seu próprio nome e tipo único de dados.

mail()

Um tipo de armazenamento de dados em PHP que permite que várias informações sejam armazenadas em um mesmo local.



## 2 Conectando-se ao MySQL

# Como Tudo se Encaixa

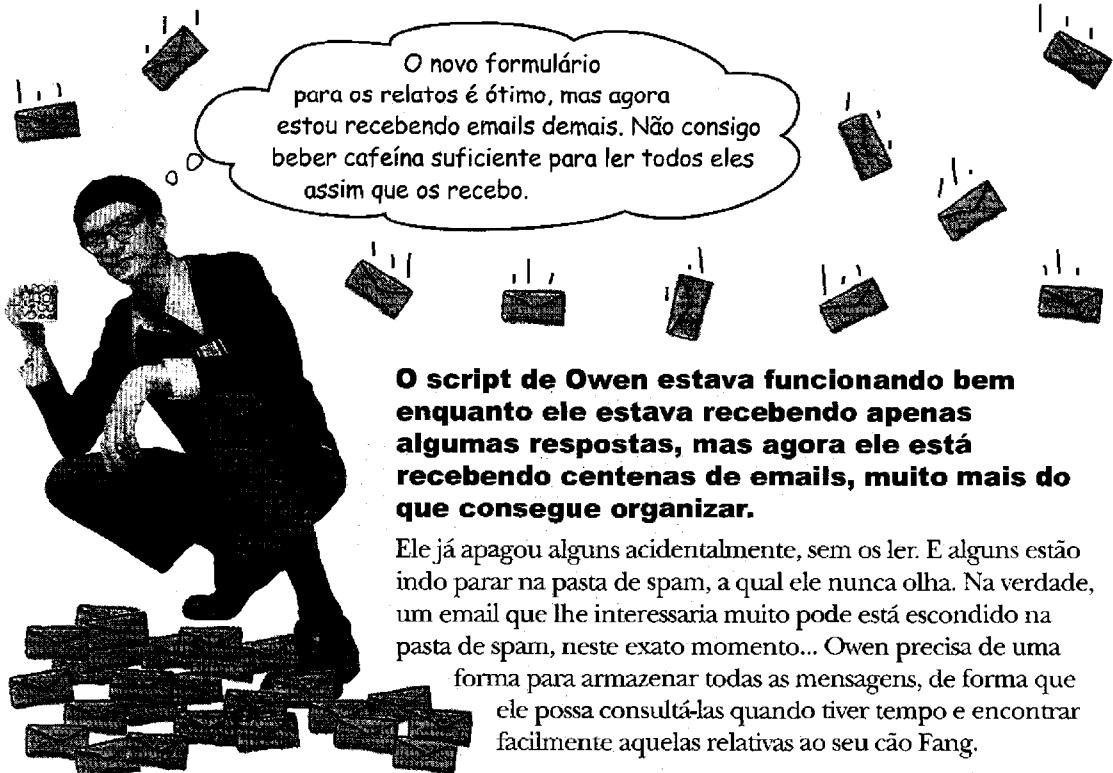
Nós temos que plugar a interweb antes de conectar o configurador do site.

Essa aí não vai chegar nem perto da minha aplicação web.



**Saber como as coisas se encaixam, antes de começar a construir, é uma boa ideia.** Você criou seu primeiro script PHP e ele está funcionando bem. Mas obter os resultados obtidos em um email já não é bom o suficiente. Agora você precisa de uma forma de **guardar os resultados** do seu formulário, para que **possa mantê-los** pelo tempo que precisar e **ter acesso** a eles quando quiser. Um **banco de dados MySQL** pode armazenar estes dados para você. Mas você precisa conectar seu script PHP ao banco de dados MySQL para fazer isso acontecer.

# O formulário de Owen funciona bem. Bem até demais...

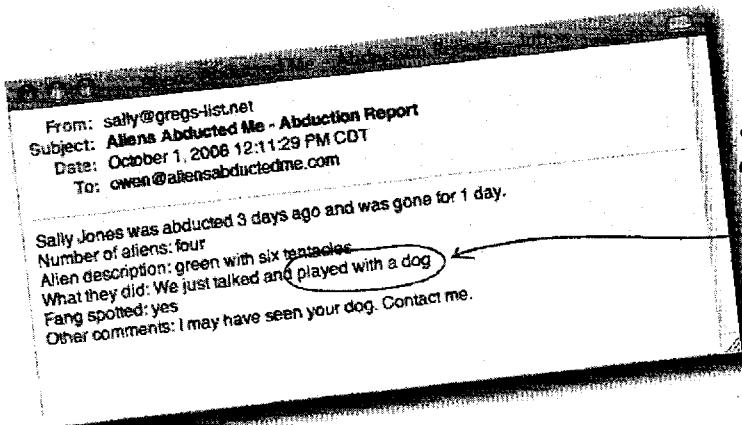


O novo formulário para os relatos é ótimo, mas agora estou recebendo emails demais. Não consigo beber cafeína suficiente para ler todos eles assim que os recebo.

**O script de Owen estava funcionando bem enquanto ele estava recebendo apenas algumas respostas, mas agora ele está recebendo centenas de emails, muito mais do que consegue organizar.**

Ele já apagou alguns accidentalmente, sem os ler. E alguns estão indo parar na pasta de spam, a qual ele nunca olha. Na verdade, um email que lhe interessaria muito pode estar escondido na pasta de spam, neste exato momento... Owen precisa de uma forma para armazenar todas as mensagens, de forma que ele possa consultá-las quando tiver tempo e encontrar facilmente aquelas relativas ao seu cão Fang.

Será preciso mais do que uma xícara de café para Owen conseguir se manter a par de todos os relatos de abduções que estão chegando na sua Caixa de Entrada.



Este relatório perdido menciona um cão (dog)... trata-se de informações de que Owen precisa, desesperadamente.

Owen precisa que mensagens como esta sejam armazenadas de forma segura em algum lugar onde ele possa consultá-las, em busca de possíveis avistamentos de Fang.

É aqui que um banco de dados MySQL viria a calhar...

Caso você não saiba, a maioria das pessoas pronuncia MySQL soletrando as últimas três letras, ou seja, "my é esse quê ele".

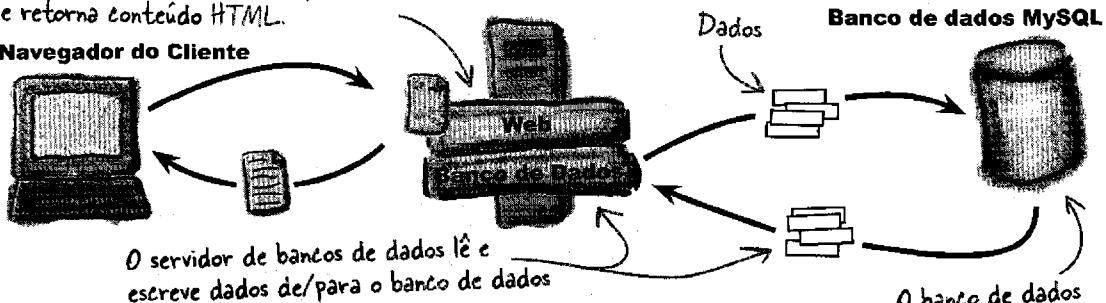
# MySQL é excelente para armazenar dados

Owen realmente precisa de uma forma de armazenar os dados dos relatos de abduções em um lugar seguro, que não a sua Caixa de Entrada de email. O que ele precisa é de um **banco de dados**, que é uma espécie de gaveta de arquivos hi-tech, arrumada e super organizada. Uma vez que as informações localizadas em um banco de dados ficam totalmente organizadas, você pode consultar a informação exata que precisar, quando precisar.

Os bancos de dados são gerenciados por um programa especial, chamado servidor do banco de dados – no nosso caso, um servidor de banco de dados MySQL. Você se comunica com esse servidor usando uma linguagem que ele é capaz de entender, que no nosso caso é o **SQL**. O servidor do banco de dados geralmente roda junto a um servidor web na mesma máquina, com os dois trabalhando em conjunto na leitura e escrita de dados e na entrega de páginas web.

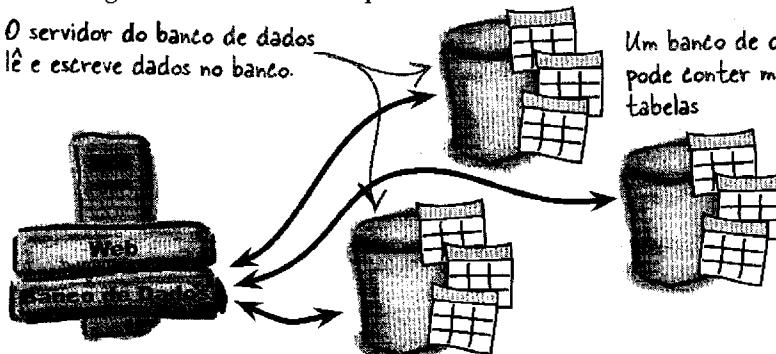
O servidor web processa as requisições por páginas web, executa scripts PHP **Computador Servidor** e retorna conteúdo HTML.

**Navegador do Cliente**



Os bancos de dados MySQL são organizados em **tabelas**, as quais armazenam informações em forma de linhas e colunas de dados relacionados. A maioria das aplicações web usa uma ou mais tabelas dentro de um mesmo banco de dados, de forma mais ou menos parecida como usar diferentes pastas dentro de uma mesma gaveta do armário de arquivos.

O servidor do banco de dados lê e escreve dados no banco.



Com os dados sobre as abduções armazenados de forma segura em um banco de dados MySQL, Owen poderá analisar, da forma que quiser, todos os relatos de pessoas que responderam "sim" à questão de se avistaram ou não Fang. Ele só precisa de um pouco de código SQL para se comunicar com o servidor do banco de dados.

O "SQL" em MySQL significa Structured Query Language ("Linguagem de Consulta Estruturada").

O MySQL armazena dados dentro de tabelas do banco de dados.

**Banco de dados MySQL**

O banco de dados propriamente dito, com frequência, é armazenado na forma de arquivos em um disco rígido, mas não necessariamente precisa ser assim.

SQL é uma linguagem de consulta usada para que ocorra a comunicação com um banco de dados MySQL.

## Owen precisa de um banco de dados MySQL

Então, está decidido: os bancos de dados MySQL são úteis e Owen precisa de um para armazenar dados sobre as abduções alienígenas. Ele poderá, então, modificar o script `report.php` para armazenar dados na tabela, em vez de enviá-los para si mesmo, por email. A tabela manterá os dados recebidos sãos e salvos, dando tempo a Owen para analisá-los e filtrar potenciais avistamentos de Fang. Mas vamos “começar pelo começo”... um banco de dados!

Para se criar um banco de dados MySQL, faz-se necessário um servidor MySQL com um software especial. A razão disso é que, ao contrário de servidores web, um servidor de banco de dados precisa receber instruções usando comandos SQL.

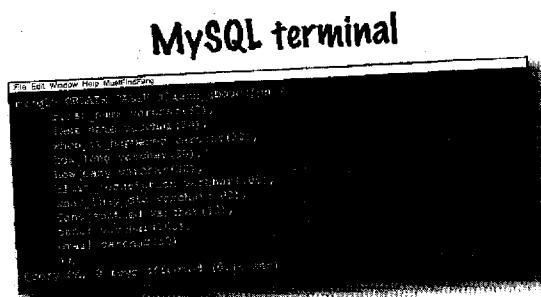
A criação de bancos de dados e tabelas MySQL exige comunicação com um servidor MySQL.

Eu sempre ouvi dizer que a ferramenta utilizada faz toda a diferença para se fazer um trabalho direito. Como eu vou saber qual ferramenta MySQL usar para criar meu banco de dados e minhas tabelas?

O terminal MySQL é uma janela que fornece acesso via linha de comando, onde você pode digitar comandos SQL.

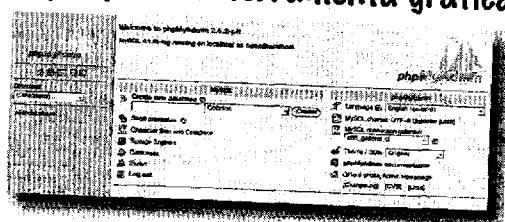


Owen precisa de uma ferramenta MySQL para criar seus novos banco de dados e tabelas.



MySQL terminal

O phpMyAdmin foi escrito, ele mesmo, em PHP.



O phpMyAdmin é uma ferramenta gráfica que lhe permite criar bancos de dados e tabelas através de uma interface web.

Duas ferramentas MySQL populares são o terminal MySQL e o phpMyAdmin. Ambas lhe permitem executar comandos SQL para criar bancos de dados e tabelas, inserir dados, selecionar dados, etc., mas o phpMyAdmin dá um passo além, por fornecer também uma interface gráfica acessível via web. Algumas empresas de web hosting incluem o phpMyAdmin como parte do seu serviço MySQL padrão, já o terminal MySQL pode ser usado para se acessar a maioria das instalações do MySQL.



**Você precisa ter um servidor de banco de dados MySQL instalado antes de virar esta página.**

É impossível ajudar Owen sem um destes! Caso você já tenha um servidor de banco de dados MySQL instalado e funcionando, continue lendo. Se não, consulte o Apêndice ii e siga as instruções de instalação. Caso esteja usando um serviço de web hosting que ofereça MySQL, peça ao administrador que instale o software. Várias informações são necessárias para se acessar um servidor MySQL. Você voltará a precisar delas posteriormente, portanto, agora é uma boa hora para entendermos quais são estas informações. Marque-as após ter escrito a informação correspondente.

- Localização do meu servidor MySQL (endereço IP ou nome do host): *localhost*
- Nome de usuário para o meu banco de dados: *web*
- Senha para o meu banco de dados: *-----*

Você precisa verificar todas estas caixas.

Caso tenha medo que este livro caia em mãos erradas, sinta-se à vontade para não escrever a senha.

Com essas informações em mãos, só o que você precisa fazer é confirmar se o servidor está funcionando. Marque uma das caixas de verificação abaixo para confirmar que você acessou com sucesso o seu servidor MySQL.

- Consigo acessar o meu servidor MySQL com sucesso, usando o terminal MySQL.
- Consigo acessar o meu servidor MySQL com sucesso, usando o phpMyAdmin.
- Consigo acessar o meu servidor MySQL com sucesso, usando .....

Só é preciso verificar uma destas três.

Caso tenha encontrado alguma outra ferramenta MySQL que funcione, escreva o nome dela aqui.

## Crie um banco de dados e uma tabela MySQL

Algumas instalações do MySQL já incluem um banco de dados. Se não for o caso da sua, você terá de criar seu próprio banco, usando o comando SQL CREATE DATABASE no terminal MySQL. Mas, primeiro, abra o terminal em uma janela de linha de comando – em geral, basta digitar mysql. Você saberá se entrou com sucesso no terminal quando o prompt de comando mudar para **mysql>**.

Para criar o novo banco de dados para as abduções alienígenas, digite CREATE DATABASE aliendatabase; desta forma:

```
File Edit Window Help PhoneHome
mysql> CREATE DATABASE aliendatabase;
Query OK, 1 row affected (0.01 sec)
```

O servidor MySQL geralmente apresenta uma resposta, para você saber que o comando teve sucesso.

Quando usar o terminal, você deve colocar um ponto e vírgula após cada comando.

Antes de criar a tabela dentro do banco de dados, certifique-se de que o nosso novo banco está selecionado. Digite o comando

USE aliendatabase;

```
File Edit Window Help PhoneHome
mysql> USE aliendatabase;
Database changed
```

O código SQL usado para se criar uma tabela é um pouco mais complexo, uma vez que ele precisa dizer, exatamente, qual tipo de dados está sendo armazenado. Vamos dar uma olhada no comando SQL antes de digitá-lo no terminal:

```
CREATE TABLE aliens_abduction (
    first_name varchar(30),
    last_name varchar(30),
    when_it_happened varchar(30),
    how_long varchar(30),
    how_many varchar(30),
    alien_description varchar(100),
    what_they_did varchar(100),
    fang_spotted varchar(10),
    other varchar(100),
    email varchar(50)
);
```

Este é um comando SQL usado para que seja criada uma nova tabela.

Todo o resto são informações detalhadas sobre quais tipos de dados podem ser armazenados na tabela.

Todos os comandos SQL digitados no terminal MySQL devem terminar com um ponto e vírgula.

Para que a nova tabela seja criada, digite o comando `CREATE TABLE` no terminal MySQL (ou baixe o código do comando no site [www.altabooks.com.br](http://www.altabooks.com.br)). Após executar o comando com sucesso, aparecerá uma tabela `aliens_abduction` novinha em folha.

A resposta "Query OK" do servidor MySQL lhe permite saber que a tabela foi criada sem problemas.

```

File Edit Window Help PhoneHome
mysql> CREATE TABLE aliens_abduction (
    first_name varchar(30),
    last_name varchar(30),
    when_it_happened varchar(30),
    how_long varchar(30),
    how_many varchar(30),
    alien_description varchar(100),
    what_they_did varchar(100),
    fang_spotted varchar(10),
    other varchar(100),
    email varchar(50)
);
Query OK, 0 rows affected (0.14 sec)
  
```

A sua instalação do MySQL poderá incluir a ferramenta phpMyAdmin, a qual lhe permite acessar seus bancos de dados e tabelas graficamente. É possível usar a interface do phpMyAdmin para criar bancos e tabelas apenas com cliques do mouse ou então digitar comandos SQL diretamente, da mesma forma como no terminal MySQL. Clicando na aba SQL do phpMyAdmin, você acessa uma caixa de texto que age como o terminal MySQL.

The screenshot shows the phpMyAdmin interface with the SQL tab selected. The query input field contains:

```

CREATE TABLE aliens_abduction
(first_name varchar(30),
last_name varchar(30),
when_it_happened varchar(30),
how_long varchar(30),
how_many varchar(30),
alien_description varchar(100),
what_they_did varchar(100),
);
  
```

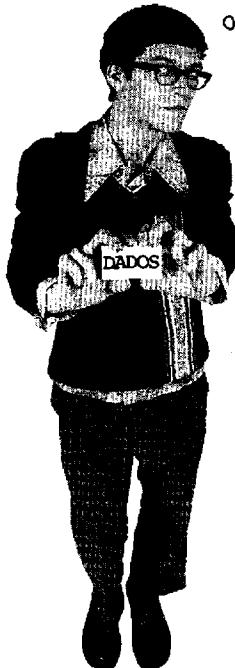
Below the query, there are options to "Run SQL query now" or "Save this query to file". A "Go" button is highlighted with an arrow. To the right, there is a note: "Você pode digitar aqui os mesmos comandos que no terminal MySQL, basta clicar em Go para executá-los."

Você pode digitar aqui os mesmos comandos que no terminal MySQL, basta clicar em Go para executá-los.

Após digitar o código SQL, clique neste botão para criar a tabela.

Assim, a guia SQL do aplicativo phpMyAdmin oferece uma maneira para que comandos SQL sejam emitidos da mesma forma como se o terminal MySQL estivesse sendo usado.

## introduzindo a instrução INSERT

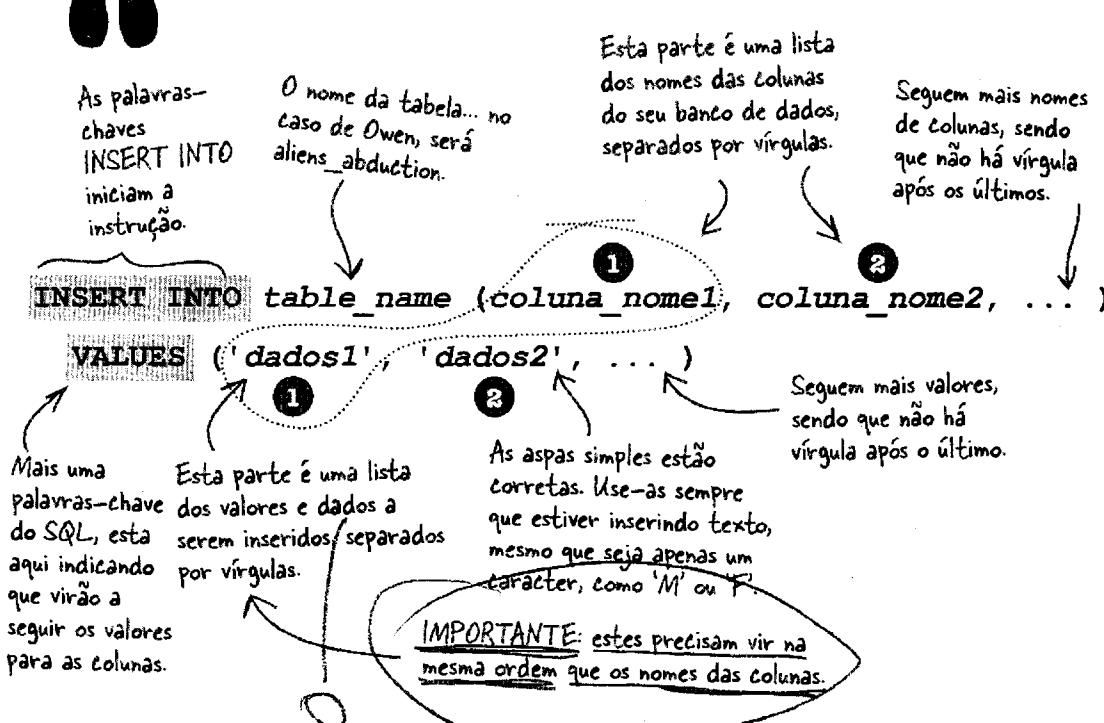


Eu tenho um banco de dados MySQL e uma tabela: agora, como eu insiro dados neles?

### Você usa a instrução SQL INSERT para inserir dados em uma tabela.

A linguagem SQL fornece todo o tipo de instruções interessantes para se interagir com os bancos de dados. Uma das mais usadas é INSERT, que realiza o trabalho de criar e armazenar dados em uma tabela.

Dê uma olhada na instrução abaixo, para ver como INSERT funciona. Tenha em mente que esta não é uma instrução SQL verdadeira, e sim um **modelo da instrução** para lhe mostrar o formato geral de INSERT.



Uma das coisas mais importantes a se notar nesta instrução é que os valores, no segundo conjunto de parênteses, **precisam estar na mesma ordem que os nomes das colunas**. É assim que o comando INSERT sabe qual dado vai para qual coluna, ao inserir os dados.

# A instrução INSERT em ação

Veja a seguir como uma instrução INSERT pode ser usada para que os dados das abduções na recém-criada tabela aliens\_abduction sejam armazenados.

Este é o nome da tabela na qual os dados estão sendo inseridos, e NÃO o nome do banco de dados.

Os nomes das suas colunas estão no primeiro conjunto de parênteses, divididos por vírgulas.

```
INSERT INTO aliens_abduction (first_name, last_name,
    1      2
when_it_happened, how_long, how_many, alien_description,
    3      4      5      6
what_they_did, fang_spotted, other, email)
    7      8      9      10
VALUES ('Sally', 'Jones', '3 dias atrás', '1 dia', 'quatro',
    1      2      3      4      5
'verde com seis tentáculos', 'Nós só conversamos e brincamos com cachorro',
    6      7
'sim', 'acho que posso ter visto seu cão.entre em contato',
    8      9
'sally@gregs-list.net')
```

Os dados para cada coluna estão no segundo conjunto de parênteses, também divididos por vírgulas.

Ao contrário do que ocorre no PHP, as instruções SQL não terminam em ponto e vírgula quando usadas dentro de código PHP.

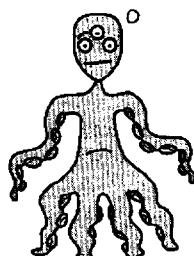
Veja bem!

## A ordem é importante!

Os valores a serem inseridos devem ser listados exatamente na mesma ordem que os nomes das colunas.

Todos estes valores contêm texto, e não números, portanto, nós os colocamos todos entre aspas simples.

Quem é o alienígena esquisito aqui?



## Aponte seu lápis

Estes são os nomes das colunas.

A tabela aliens\_abduction encontra-se abaixo, mas ela ainda não tem quaisquer dados. Escreva os dados da abdução de Sally na tabela. Você pode escrever alguns dos dados fora da tabela, se não houver espaço dentro das células, bastando ligar com uma seta a informação à célula respectiva.

aliens\_abduction

first_name	last_name	when_it_happened	how_long	how_many	alien_description	what_they_did	fang_spotted	other	email
Sally	Jones	3 dias atrás	1 dia	quatro	verde com seis tentáculos	Nós só conversamos e brincamos com cachorro	sim	acho que posso ter visto seu cão.entre em contato	sally@gregs-list.net



## Aponte seu lápis Solução

A tabela `aliens_abduction` encontra-se abaixo, mas ela ainda não tem quaisquer dados. Escreva os dados da abdução de Sally na tabela. Você pode escrever alguns dos dados fora da tabela, se não houver espaço dentro das células, bastando ligar com uma seta a informação à célula respectiva.

verdes com seis tentáculos  
e brincamos com um cachorro.

Nós só conversamos  
e brincamos com um cachorro.

Acho que posso  
ter visto o seu cão.  
Entre em contato.

`sally@gregs-list.net`

first_name	last_name	when_it_happened	how_long	how_many	alien_description	what_they_did	fang_spotted	other
Sally	Jones	3 dias atrás	1 dia	quatro			sim	

*não existem*

## Perguntas Idiotas

**P:** Não sei se entendi bem a diferença entre banco de dados e tabela. Ambos não fazem a mesma coisa: armazenar dados?

**R:** Sim. As tabelas servem como uma forma de dividir os dados de um banco em grupos relacionados, para que você não acabe tendo um único conjunto enorme de dados. É mais ou menos como a diferença entre colocar um monte de sapatos em uma mesma grande caixa e colocar cada par dentro da sua própria caixa – a caixa grande é o banco de dados, as caixas menores são as tabelas. Assim, os dados são armazenados em tabelas, e as tabelas são armazenadas em bancos de dados.

**P:** O que exatamente é o terminal MySQL? Como o encontro no meu computador:

**R:** O terminal MySQL é uma técnica para se acessar um banco de dados MySQL através de uma interface de linha de comando. Em muitos casos, o terminal não é um programa a parte, e sim uma conexão que você estabelece usando a linha de comando de um programa terminal “genérico”, como por exemplo o aplicativo terminal do Mac OS X. O modo como você acessa o terminal MySQL varia bastante, dependendo de qual sistema operacional você está usando e se o servidor MySQL está local ou remoto (localizado em algum outro lugar que não o seu computador). O Apêndice II apresenta mais detalhes sobre como acessar o terminal MySQL.

**P:** E quanto ao phpMyAdmin? Onde posso encontrá-lo?

**R:** Ao contrário do terminal MySQL, o phpMyAdmin é uma aplicação web que permite acesso a um banco de dados MySQL. Trata-se, na verdade, de uma aplicação PHP, motivo pelo qual você sempre a acessa a partir de um servidor web, em vez de instalá-la no seu próprio computador. Muitas empresas de web hosting oferecem o phpMyAdmin como parte do seu serviço MySQL padrão, portanto ele poderá já estar instalado para você. Se não estiver, baixe e instale o programa, você mesmo. Está disponível, gratuitamente, em [www.phpmyadmin.net](http://www.phpmyadmin.net). Lembre-se apenas de que ele precisa ser instalado em um servidor web e configurado para ter acesso aos seus bancos MySQL, da mesma forma como qualquer outra aplicação PHP e MySQL.

**P:** Eu tenho tanto o terminal MySQL quanto o phpMyAdmin disponíveis. Qual deles eu devo usar para acessar meu banco de dados?

**R:** Essa é uma questão pessoal. A vantagem do phpMyAdmin é que você pode explorar seus bancos de dados e tabelas visualmente, sem ter de digitar comandos SQL. Isso pode ser bastante útil, caso você tenha se familiarizado com o SQL e não queira digitar comandos manualmente para cada coisinha que precisar fazer. Entretanto, por agora, é uma boa ideia que você se concentre em realmente entender o modo de interagir com os seus dados usando comandos SQL, e nesse caso, qualquer uma das ferramentas funciona a contento.



## TEST DRIVE

### Armazene um caso de abdução alienígena no seu banco de dados com uma instrução SQL INSERT

Usando uma ferramenta MySQL, como o terminal MySQL ou a guia SQL do phpMyAdmin, digite uma instrução INSERT para armazenar dados de uma abdução. Como exemplo, eis aqui a instrução INSERT para a abdução de Sally Jones:

```
INSERT INTO aliens_abduction (first_name, last_name,
when_it_happened, how_long, how_many, alien_description,
what_they_did, fang_spotted, other, email)
VALUES ('Sally', 'Jones', '3 dias atrás', '1 dia', 'quatro',
'veerde com seis tentáculos', 'Nós só conversamos e brincamos com seu cachorro',
'sim', 'Acho que posso ter visto o seu cão. Entre em contato.',

'sally@gregs-list.net')
```

```
File Edit Window Help PugsInSpace
mysql> INSERT INTO aliens_abduction (first_name, last_name,
when_it_happened, how_long, how_many, alien_description,
what_they_did, fang_spotted, other, email)
VALUES ('Sally', 'Jones', '3 dias atrás', '1 dia', 'quatro',
'veerde com seis tentáculos', 'Nós só conversamos e brincamos com um cachorro',
'sim', 'Acho que posso ter visto o seu cão. entre em contato.',
'sally@gregs-list.net');
Query OK, 1 rows affected (0.0005 sec)
```

  
Executar a instrução INSERT  
no terminal MySQL resulta  
em uma nova linha de dados a  
ser adicionada à tabela alien\_  
description.

A instrução INSERT parece ter sido executada com sucesso. Como você acha que poderíamos confirmar se os dados foram inseridos? Escreva aqui.

...SELECT.....

## Use SELECT para obter dados da tabela

Inserir dados em uma tabela é uma tarefa relativamente fácil, mas é compreensível sentir certa apreensão caso você não tenha confirmado se os dados realmente foram parar na tabela. É mais ou menos como depositar dinheiro na sua conta de poupança, mas nunca poder visualizar o saldo. A instrução SELECT é a forma de “ver o saldo” de uma tabela em um banco de dados. Ou, dizendo propriamente, SELECT permite que você solicite os dados das colunas contidos em uma tabela.

Após SELECT, escreva uma lista das colunas cujos dados quer receber.

`SELECT colunas FROM nome_da_tabela`

A parte FROM da instrução é o que indica a SELECT de qual tabela nós queremos selecionar dados.

Um SELECT sempre é feito em relação a uma tabela específica, e não ao banco de dados, de forma geral.

A instrução SELECT do SQL obtém colunas de dados de uma tabela

As colunas fornecidas a uma instrução SELECT precisam ser separadas por vírgulas. Independente de quantas colunas houver em uma tabela, somente os dados dos nomes das colunas especificadas em SELECT são retornados. Esta instrução SELECT extrai todos os nomes e sobrenomes dos abduzidos contidos na tabela aliens\_abduction:

Somente os dados referentes a estas duas colunas são retornados por esta instrução SELECT.

`SELECT first_name, last_name FROM aliens_abduction`

A instrução SELECT só obtém dados da tabela aliens\_abduction.

Para verificar um INSERT, você precisa de uma forma para visualizar rapidamente, todos os dados da tabela e não apenas algumas colunas. A instrução SELECT oferece um atalho voltado exatamente para isso:

O asterisco, ou “estrela”, diz à instrução SELECT para obter todos os dados de todas as colunas da tabela.

`SELECT * FROM aliens_abduction`

Não é necessário especificar uma lista de colunas, porque \* significa “obtenha todas elas!”



## TEST DRIVE

**Certifique-se de que a instrução INSERT funcionou, usando SELECT para obter dados da tabela.**

Execute uma consulta SELECT, usando uma ferramenta MySQL, para visualizar todo o conteúdo da tabela aliens\_abduction. Certifique-se de que a linha de dados que acabou de inserir está aparecendo nos resultados.

```
SELECT * FROM aliens_abduction
```

Estas são as colunas.

```
File Edit Window Help HaveYouSeenHim
mysql> SELECT * FROM aliens_abduction;
+-----+-----+-----+-----+-----+
| first_name | last_name | when_it_happened | how_long | how_many |
+-----+-----+-----+-----+-----+
| Sally      | Jones     | 3 dias atrás    | 1 dia    | quatro   |
+-----+-----+-----+-----+-----+
1 row in set (0.0005 sec)
```

Abaixo de cada coluna ficam os dados respectivos.

A consulta SELECT revela uma única linha de dados armazenados na tabela.

Quantas linhas de dados há na sua tabela? ..... 1 .....

Então, você está me dizendo que preciso escrever uma instrução INSERT toda vez que quiser adicionar um novo relato de abdução ao meu banco de dados? De repente, esse negócio de MySQL não me parece tão interessante...



```
mysql> INSERT INTO aliens_abduction(first_name, last_name,
   alien_description,
   'yes', 'mysql> INSERT INTO aliens_abduction(first_name, last_name,
   alien_description,
   'Sallying', 'when it happened, how long, alien_name,
   'they did, Fang spotted, other, email'
VALUES ('Tom', 'grayish', 'back in 1991', '30 seconds',
   'tunnel', 'they looked like conkeys made out of metal with some kind
   'of skin on them');
mysql> INSERT INTO aliens_abduction(first_name, last_name,
   alien_description,
   'when it happened, how long, how many, alien_description,
   'they did, Fang spotted, other, email'
VALUES ('Sally', 'Reuter', 'summer of 1993', '2 hours,
   'there was a bright light in the sky, followed by a bark of two',
   'they beamed me toward a gas station in the desert', 'gray',
   'I was out of gas, so it was a pretty good abduction');
Query OK, 1 rows affected (0.0005 sec)

mysql> SELECT * FROM aliens_abduction;
+-----+-----+-----+
| first_name | last_name | alien_description |
+-----+-----+-----+
| Tom        | grayish  | back in 1991      |
| Sally      | Reuter   | summer of 1993    |
+-----+-----+-----+
2 rows in set (0.0005 sec)
```

### **É verdade, cada inserção em um banco de dados MySQL requer uma instrução INSERT.**

E é neste ponto que a comunicação com um banco de dados MySQL realizada simplesmente através de comandos SQL fica tediosa. É claro que há muitos benefícios ao se armazenar os dados de Owen em um banco, em vez de em emails na sua Caixa de Entrada, mas gerenciar os dados manualmente, emitindo-se instruções SQL em uma ferramenta MySQL, não é uma solução razoável.



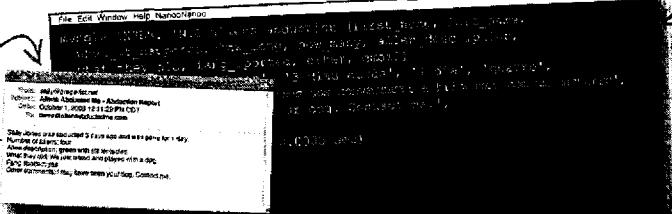
Como você acha que o problema de inserção de dados de Owen pode ser resolvido?

# Deixe o PHP lidar com as coisas tediosas do SQL

A solução para o problema de Owen não é evitar o comando SQL, mas sim **automatizar** o SQL com a ajuda do PHP. O PHP possibilita a emissão de instruções SQL em um código de script que será executado no servidor, de forma que você nem sequer precisa usar uma ferramenta MySQL. Isso significa que o formulário HTML de Owen pode chamar um script PHP para lidar com a inserção dos dados no banco, sempre que novos dados forem submetidos – nada de emails, nem ferramentas SQL, nem confusões!

O formulário HTML gera um email que Owen recebe e agora precisa então adicionar, manualmente, as informações ao banco de dados.

Owen cria uma instrução SQL INSERT, a qual insere os dados do email no banco.



**Sem PHP, é preciso executar uma instrução SQL INSERT manual para armazenar cada relatório de abdução no banco de dados.**

**Com PHP, um script PHP automaticamente executa INSERT quando o formulário é submetido.**

O formulário HTML chama um script PHP e pede a ele que adicione os dados do formulário ao banco.

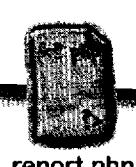
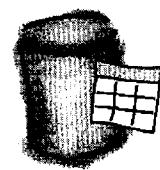
```
<?php
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool', 'aliendatabase');
or die('Error connecting to MySQL server.');

$query = "INSERT INTO aliens_abduction (first_name, last_name, "
        . "what_they_did, how_long, how_many, alien_description, "
        . "VALUES ('Sally', 'Jones', '3 dias atrás', '1 dia', 'quatro', "
        . "'verde com seis tentáculos', 'Nós só conversamos e brincamos com um cachorro', "
        . "'sim', 'Acho que posso ter visto seu cão. entre em contato.', "
        . "'sally@gregs-list.net');";

$result = mysqli_query($dbc, $query)
or die('Error querying database.');

mysqli_close($dbc);
?>
```

O script PHP cria uma instrução INSERT que insere os dados do formulário no banco... não é preciso nenhuma intervenção de Owen!



*como o aplicativo do Owen pode usar php e mysql*

## O PHP permite usar os dados do formulário web de Owen

O PHP melhora o formulário web de Owen ao permitir que um script **envie os dados diretamente para um banco**, em vez de enviá-los para o endereço de email de Owen, que teria então de inseri-los manualmente. Vamos dar uma olhada com mais detalhes no modo exato como a aplicação funciona, agora que o PHP está em ação.

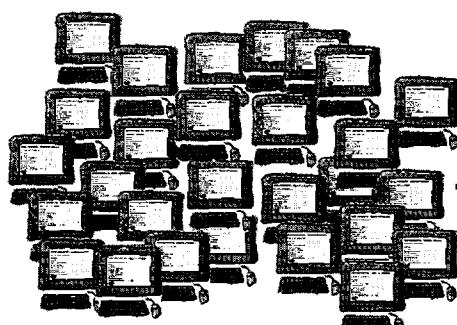
1

Sally preenche o formulário e clica no botão “Relatar Abdução”, para submetê-lo. As informações são enviadas para o script report.php no servidor web.



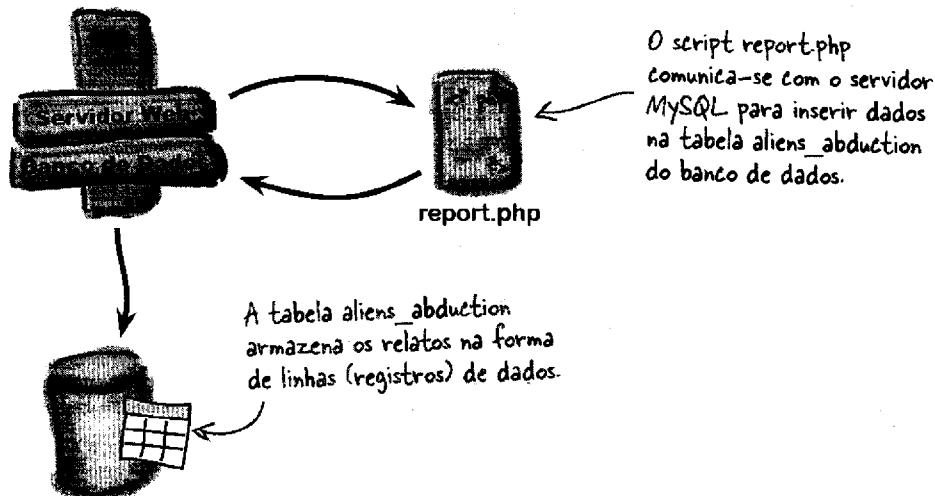
2

Várias pessoas continuam a submeter o formulário.

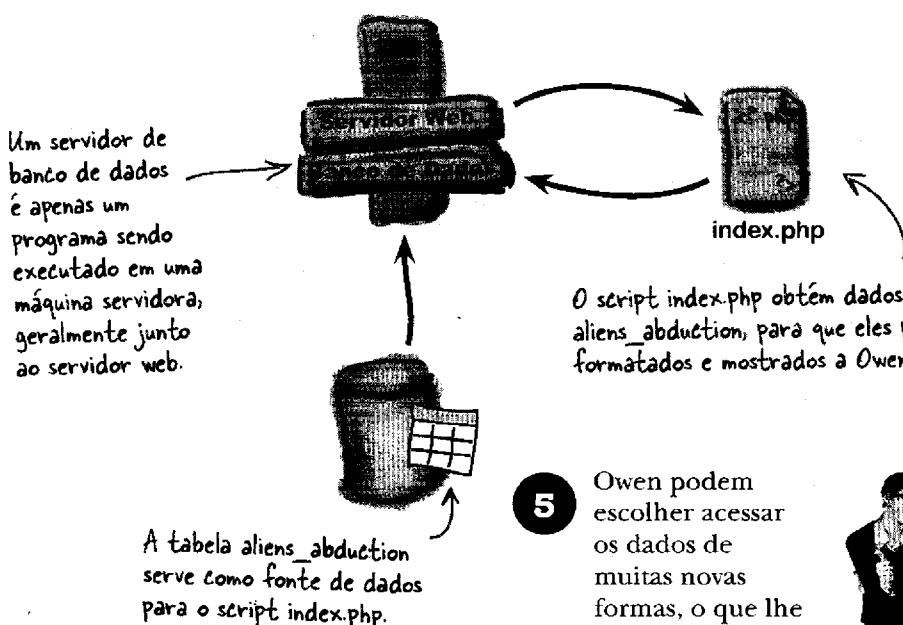


**3**

O script `report.php` de Owen conecta-se a um banco de dados MySQL e insere as informações de cada formulário submetido usando instruções SQL `INSERT`.

**4**

Owen precisa não somente de um script para colocar os dados no banco, como também de outro script para procurar e visualizar os dados. Na verdade, isso poderia servir como a página principal do seu site. O script `index.php` conecta-se ao banco, obtém dados sobre as abduções e os mostra a Owen.

**5**

Owen pode escolher acessar os dados de muitas novas formas, o que lhe permite focar-se em encontrar seu cão Fang.



## Conecte-se ao seu banco de dados a partir do PHP

Antes que um script PHP possa inserir ou obter dados de um banco MySQL, ele precisa se conectar a esse banco. Conectar-se a um banco MySQL a partir do PHP é semelhante, de muitas maneiras, a acessar o banco através de uma ferramenta MySQL e as mesmas informações são requeridas. Lembra-se das três caixas de verificação que você preencheu anteriormente, neste capítulo? Aqui vão elas novamente, junto a mais uma, que indica nome do banco de dados – escreva todas as informações, mais uma vez.

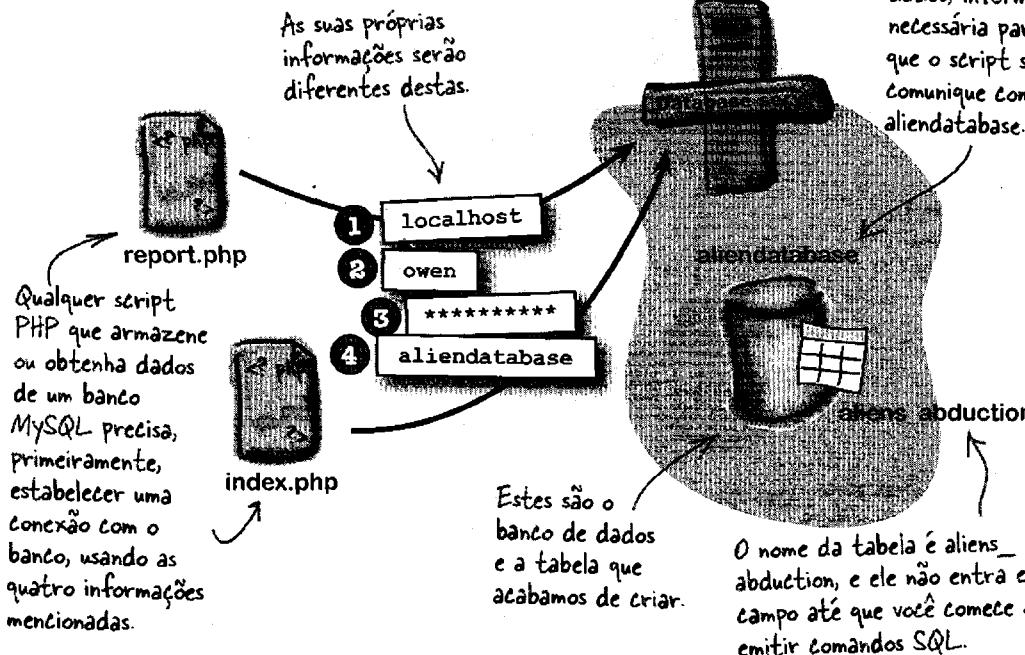
Seu provedor de hospedagem de sites, ou seu webmaster, pode lhe fornecer esta informação. Ou então, se o seu servidor web e o seu servidor de banco de dados MySQL estiverem rodando na mesma máquina, você pode usar a palavra "localhost".

- 1 Localização do meu servidor MySQL (endereço IP ou nome do host): .....
- 2 Nome de usuário para o meu banco de dados: .....
- 3 Senha para o meu banco de dados: .....
- 4 Nome do meu banco de dados: .....

As informações de localização do host, nome do usuário, senha e nome do banco de dados são todas requeridas para se estabelecer uma conexão ao banco MySQL a partir de um script PHP. Uma vez feita a conexão, o script pode executar comandos SQL da mesma forma como se você os estivesse digitando manualmente em uma ferramenta MySQL.

O nome do banco de dados que você criou anteriormente é aliendatabase. Se, por algum motivo, você tiver atribuído algum outro nome ao seu banco, ou tiver optado por usar um banco já existente, use o nome apropriado.

O nome do banco de dados, informação necessária para que o script se comunique com ele, é aliendatabase.



## Insira dados com um script PHP

Executar uma consulta MySQL a partir de código PHP requer que você, primeiramente estabeleça uma conexão com o banco de dados. Em seguida, você constrói a instrução (query) na forma de uma string PHP. A instrução (query) não é executada até que repasse a string para o servidor do banco de dados. Finalmente, uma vez terminado o acesso ao banco de dados, você encerra a conexão. Todas essas tarefas são executadas através de código em um script PHP. Eis um exemplo que insere uma nova linha de dados de abduções:

```
<?php
    $dbc = mysqli_connect("data.alienabduction.com.br", "owen", "senhasally", "aliens_abduction");
    or die('Error connecting to MySQL server.');

    $query = "INSERT INTO aliens_abduction (first_name, last_name, when_it_happened, how_long,
    " . "how_many, alien_description, what_they_did, fang_spotted, other, email) "
    . "VALUES ('Sally', 'Jones', '3 dias atrás', '1 dia', 'quatro', 'verde com seis tentáculos', "
    . "'Nós só conversamos e brincamos com um cachorro', 'sim', 'Acho que posso ter visto seu cão. entre em contato',
    . "'sally@gregs-list.net')";

    $result = mysqli_query($dbc, $query);
    or die('Error querying database.');

    mysqli_close($dbc);
?>
```

*Conecta-se ao banco de dados MySQL.*

*Estes devem ser os SEUS valores, e não os de Owen.*

*Você talvez possa usar 'localhost' para a localização do seu banco, em vez de um nome de domínio.*

*Construa a instrução (query)  
INSERT como uma string em código PHP.*

*Emita a consulta INSERT no banco de dados MySQL.*

*Tenha muito cuidado com as aspas simples e duplas aqui, e também com os espaços antes e depois das aspas!*

*Estas funções requerem que o seu servidor web tenha o PHP versão 4.1 ou acima.*

### PODER DO CÉREBRO

O que você acha que cada uma destas funções PHP está fazendo no script?

mysqli\_connect() → conecta ao bd  
 mysqli\_query() → insere em banco de dados  
 mysqli\_close() → fecha a conexão

*três funções PHP para conexão mysql*

## Use funções PHP para falar com o banco de dados

Existem três funções PHP principais, usadas para que se realize a comunicação com bancos de dados MySQL: `mysqli_connect()`, `mysqli_query()` e `mysqli_close()`. Caso tenha percebido um padrão aí, não é coincidência – todas as funções modernas do PHP que interagem com o MySQL começam com `mysqli_`.

Um conjunto antigo de funções PHP que interajam com o MySQL começará com `mysql`, sem o “`i`”. O “`i`” significa “improved” (“aprimorado”), e hoje em dia prefere-se usar as funções `mysqli`.

### `mysqli_connect()`

Conecta-se a um banco de dados MySQL usando as quatro informações que você já viu anteriormente.

### `mysqli_query()`

Emite uma instrução (query) a um banco de dados MySQL, o que, frequentemente, envolve armazenar ou obter dados de uma tabela.

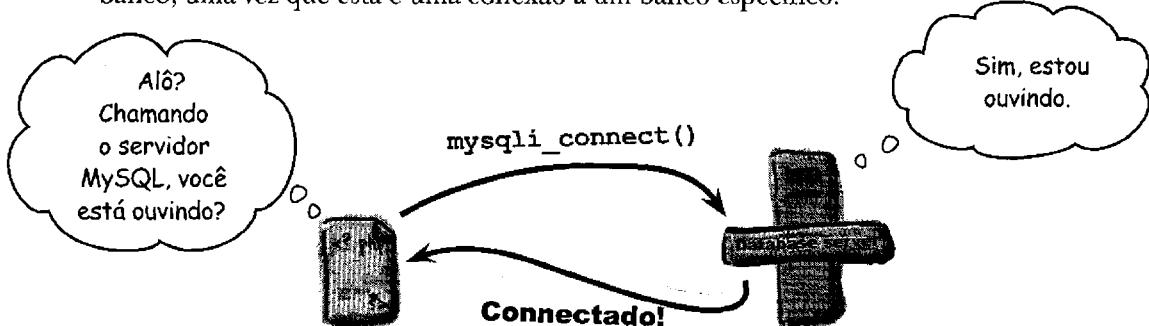
### `mysqli_close()`

Encerra uma conexão a um banco de dados MySQL.

O uso destas três funções geralmente envolve uma sequência previsível de etapas.

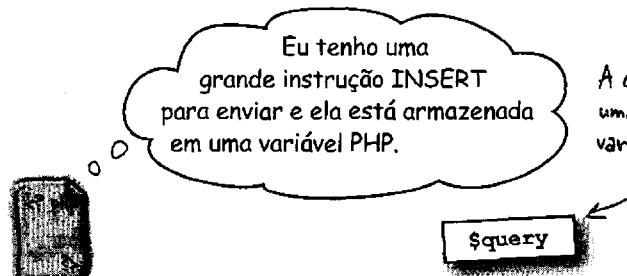
#### 1 Conecte-se a um banco de dados com a função `mysqli_connect()`.

Forneça a localização, o nome do usuário e a senha, para obter permissão para interagir com o servidor do banco de dados MySQL. Especifique também o nome do banco, uma vez que esta é uma conexão a um banco específico.



#### 2 Crie uma query SQL e a armazene como uma string em uma variável PHP.

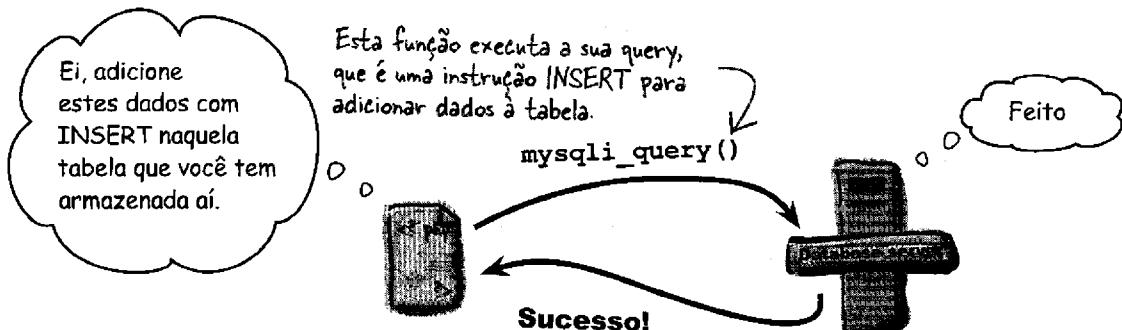
Para se comunicar com o servidor do banco de dados, você tem de usar comandos SQL. Por exemplo, é necessário uma instrução `INSERT` para que dados sejam adicionados à tabela `aliens_abduction`. Não há nada de especial a ser citado sobre o nome da variável que escolhermos, e um nome simples como `$query` funciona perfeitamente bem.



A consulta é criada como uma string e armazenada na variável `$query`.

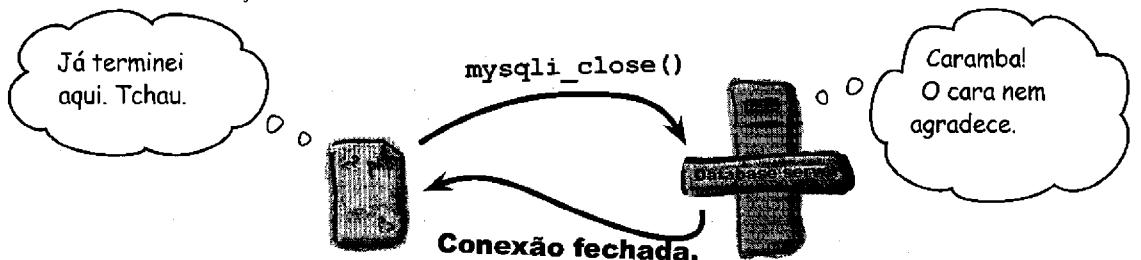
### 3 Emita a instrução (query) com a função mysqli\_query().

Use a variável \$query com a função mysqli\_query() para falar com o servidor do banco MySQL e adicionar dados à tabela aliens\_abduction. Você tem que informar à mysqli\_query() tanto o nome da conexão que criou no Passo 1 quanto o nome da variável que armazena a sua consulta, do Passo 2.



### 4 Encerre a conexão ao banco com a função mysqli\_close().

Finalmente, mysqli\_close() diz ao servidor do banco MySQL que você já terminou a sua comunicação com ele.



```
<?php
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool', 'aliendatabase')
or die('Error connecting to MySQL server.');

$query = "INSERT INTO aliens_abduction (first_name, last_name, when_it_happened, how_long,
    "how_many, alien_description, what_they_did, fang_spotted, other, email)
VALUES ('Sally', 'Jones', '3 days ago', '1 day', 'four', 'green with six tentacles',
    "'We just talked and played with a dog', 'yes', 'I may have seen your dog. Contact me.', '',
    "'sally@gregs-list.net');";

mysqli_query($dbc, $query)
or die('Error querying database.');

mysqli_close($dbc);
?>
```

Annotations for the code:

- Annotation 1: "Este é o nome da sua variável da conexão."
- Annotation 2: "Se algo der errado, isto envia uma mensagem para você e para tudo."
- Annotation 3: "Isto é uma query SQL INSERT que adiciona dados ao nosso banco."
- Annotation 4: "mysqli\_query() é a função usada pelo PHP para se comunicar com o servidor MySQL. O código armazenado na variável \$query é SQL, e não PHP."
- Annotation 5: "E aqui, nós fechamos a conexão."

Vamos dar uma olhada com mais atenção em cada uma dessas funções PHP para bancos de dados, começando por mysqli\_connect() ...

## *usando mysqli\_connect()*

# Conecte-se com mysqli\_connect()

Para o nosso script PHP poder criar uma conexão com o banco de dados usando a função `mysqli_connect()`, você precisará de algumas informações, com as quais já está começando a se familiarizar. Sim, são as mesmas informações que usou anteriormente, quando trabalhou com o terminal MySQL, mais o nome do banco de dados.

- 1 Conecte-se com `mysqli_connect()`
- 2 Monte a string da query.
- 3 Execute a query com `mysqli_query()`.
- 4 Feche a conexão com `mysqli_close()`.

### **Seu nome de usuário e senha**

#### **Quem?**

Você precisará ter seu próprio nome de usuário e senha para o seu próprio servidor do banco de dados. Essas informações ou serão definidas por você ou lhe serão entregues pela sua empresa de hospedagem de sites, quando o MySQL for instalado. Se você configurar seu próprio MySQL, siga as instruções para criar um nome e uma senha seguros.

### **O nome do banco de dados**

#### **O quê?**

No nosso exemplo, demos ao banco o nome `aliendatabase`. O seu terá qualquer nome que você decida lhe dar, ou então terá o nome dado pela sua empresa de hospedagem de sites, caso ela tenha criado o banco para você.

### **A localização do banco (um nome de domínio, um endereço IP ou localhost)**

#### **Onde?**

No nosso exemplo, estamos usando a localização do banco (fictício) de Owen. Você precisa usar a localização do seu próprio servidor MySQL. Frequentemente, você usará `localhost`, caso o servidor do banco esteja na mesma máquina que o seu servidor web. A sua empresa de web hosting poderá informar-lhe a respeito disso. A localização poderá ser também um endereço IP ou um nome de domínio, como por exemplo, `seuservidor.seuisp.com`.

A localização, o nome do usuário, a senha e o nome do banco MySQL na função `mysqli_connect()` precisam estar, todos, entre aspas.

```

$dbc = mysqli_connect(
    'data.aliensabductedme.com',
    'owen',
    'aliensrool',
    'aliendatabase');

Use esta variável para realizar
outras ações no banco de dados.

Nome do usuário.      Nome do banco de dados.
Senha.                Localização
                        do banco de
                        dados.
  
```

A função `mysqli_connect()` lida com a localização, o nome do usuário, a senha e o nome do banco de dados como strings, portanto, você tem de colocar essas informações entre aspas.

O resultado ao se chamar a função, é uma conexão ao banco de dados e uma variável PHP que você pode usar para interagir com o banco. A variável recebeu o nome `$dbc` no exemplo, mas você pode atribuir qualquer nome que quiser.



## Aponte seu lápis

Eis alguns exemplos de strings PHP para conexão a bancos de dados. Observe cada uma delas e escreva se a string vai funcionar ou não, e como consertá-la, neste caso. Adicionalmente, circule qualquer parte de código que você considere problemática.

```
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool',
'aliendatabase');
```

*OK*

```
$dbe = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool',
'aliendatabase');
```

*Alien database*

```
$fangisgone = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool',
'aliendatabase');
```

*dbc*

```
$dbc = mysqli_connect('localhost', 'owen', 'aliensrool',
'aliendatabase');
```

*OK*

```
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', '',
'aliendatabase');
```

*falta a senha*

```
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool');
mysqli_select_db($dbc, 'aliendatabase');
```

*?*

## Solução do aponte

# Aponte seu lápis

## Solução

Eis alguns exemplos de strings PHP para conexão a bancos de dados. Observe cada uma delas e escreva se a string vai funcionar ou não, e como consertá-la neste caso. Adicionalmente, circule qualquer parte de código que você considere problemática.

```
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool',  
'aliendatabase');
```

Esta string de conexão funciona sem problemas.

É preciso um ponto e vírgula aqui  
para terminar a instrução PHP.

Neste livro, estamos usando aspas simples para strings  
PHP e reservando as aspas duplas para consultas SQL.

```
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool',  
"aliendatabase")
```

Isto não funciona, porque está faltando um ponto e vírgula. As aspas duplas funcionam do  
mesmo jeito que as simples.

→ Não é um nome muito descriptivo para  
uma conexão a banco de dados.

```
$fangisgone = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool',  
'aliendatabase');
```

Isto funciona, embora "fangisgone" ("fangestádesaparecido") não seja um bom nome para uma  
conexão a banco de dados.

→ Aqui, está nítido assumindo que o servidor do banco de  
dados se localiza no mesmo computador que o servidor web.

```
$dbc = mysqli_connect('localhost', 'owen', 'aliensrool', 'aliendatabase');
```

Isto funciona desde que o servidor web e o servidor do banco estejam na mesma máquina.

Uma senha vazia para o banco  
de dados não é uma boa ideia.

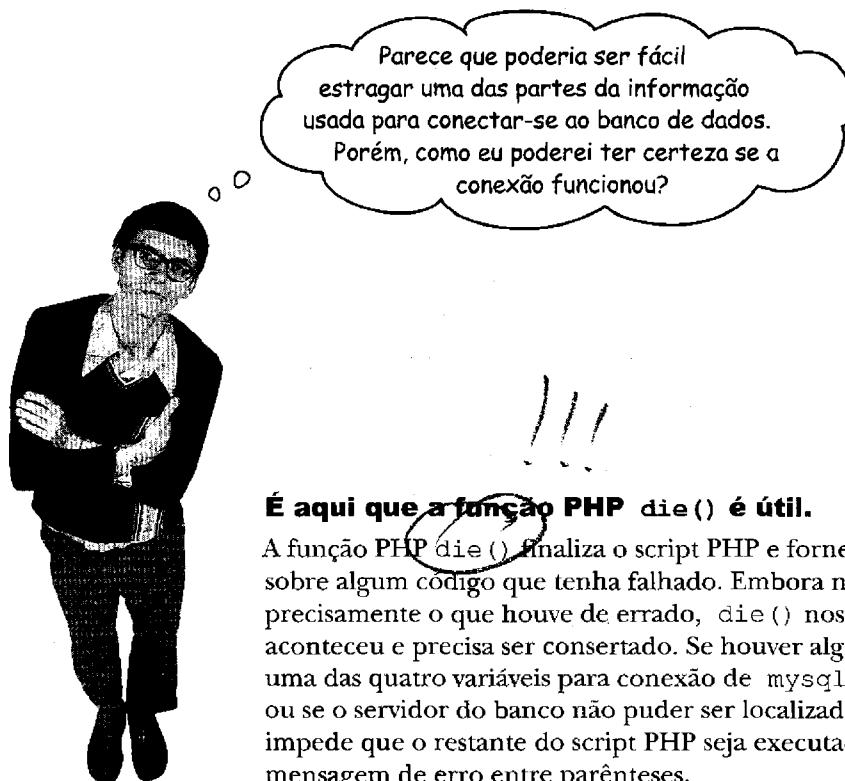
```
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', '', 'aliendatabase');
```

Isto só funcionará se você definir uma senha em branco para o banco de dados. Não é uma  
boa ideia, mas funciona! Você sempre deve ter uma senha para cada banco de dados.

Deixar de fora o quarto argumento  
requer que chame mysqli\_select\_db()  
para selecionar o banco de dados.

```
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool');  
mysqli_select_db($dbc, 'aliendatabase');
```

Me desculpe, mas isto é uma "pegadinha". Em mysqli\_connect(), o quarto item, o nome do banco  
de dados, é opcional. Você pode deixá-lo de fora da função e usar mysqli\_select\_db() para  
especificar o nome do banco. Assim, este código é o mesmo que se você tivesse passado todos os  
quatro argumentos para mysqli\_connect().



### É aqui que a função PHP die() é útil.

A função PHP die() finaliza o script PHP e fornece conhecimento sobre algum código que tenha falhado. Embora não revele precisamente o que houve de errado, die() nos informa que algo aconteceu e precisa ser consertado. Se houver algo de errado com uma das quatro variáveis para conexão de mysqli\_connect(), ou se o servidor do banco não puder ser localizado, a função die() impede que o restante do script PHP seja executado e mostra a mensagem de erro entre parênteses.

A função die() é solicitada,  
caso a conexão não tenha sido  
estabelecida.

Se uma das nossas quatro strings na função  
mysqli\_connect não estiver correta, receberemos  
uma resposta sobre isso.

```
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool', 'aliendatabase')
or die('Erro ao conectar o banco de dados.');
```

Esta mensagem é enviada para a  
página web, caso a conexão falhe.

Não é preciso colocar ponto  
e vírgula aqui, uma vez que  
"or die(...)" corresponde a,  
teoricamente, continuação de  
uma mesma instrução.

## construindo consultas no php



OK, então nós temos uma conexão PHP com o banco de dados. E agora? Podemos simplesmente começar a executar querys (instrução) como se estivéssemos no terminal MySQL?

**Sim! Uma vez feita a conexão ao banco com `mysqli_connect()`, você pode emitir consultas SQL diretamente do PHP.**

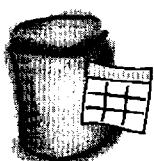
Praticamente, tudo que pode fazer em um terminal MySQL pode também ser feito em código PHP, com a conexão ao banco que acabou de ser estabelecido. É essa conexão que disponibiliza um canal de comunicação entre um script PHP e um banco de dados MySQL. Por exemplo, agora que Owen tem uma conexão ao seu banco, ele pode começar a inserir dados na tabela `aliens_abduction` com a função `mysqli_query()` e algum código de consulta SQL.

```
File Edit Window Help View  
Query OK, rows affected: 1  
Time: 0.000 sec  
+-----+  
| id | alien | victim | date |  
+-----+  
| 1 | Sally Jones | 1 day ago | 1 dog |  
+-----+
```

`mysqli_query($dbc, $query)`

Lembre-se, nosso objetivo é automatizar esta query `INSERT` usando código PHP.

Sally Jones  
3 days ago 1 d  
yes 1 d and played with a dog  
I may have seen your four  
in six tentacles  
sally@gregs-list.net  
me.



A query SQL é passada para `mysqli_query()` como uma string PHP.

A função `mysqli_query()` precisa ter uma query SQL armazenada em uma string PHP (`$query`) para executar a inserção dos dados das abduções alienígenas.

# Construa a query INSERT no PHP

As query SQL, em PHP, são representadas como strings e costuma-se armazenar a query em uma string antes de passá-la para a função `mysqli_query()`. Uma vez que as consultas SQL podem ser razoavelmente grandes, frequentemente é necessário construí-las a partir de strings menores, que se estendem por várias linhas de código. A query `INSERT` de Owen é um bom exemplo disso:

```
$query = "INSERT INTO aliens_abduction (first_name, last_name, "
    . "when_it_happened, how_long, how_many, alien_description, "
    . "what_they_did, fang_spotted, other, email) "
    . "VALUES ('Sally', 'Jones', '3 dias atrás', '1 dia', 'quatro', "
    . "'verde com seis tentáculos', 'Nós só conversamos e brincamos com um cachorro', "
    . "'sim', 'Acho que posso ter visto seu cão. Entre em contato.', "
    . "'sally@gregs-list.net')";
```

← Esta é uma variável string do PHP, que agora armazena a query `INSERT`.

↑ Uma vez que tudo isto é código PHP, é preciso finalizá-lo com ponto e vírgula.

A string de consulta se divide em várias linhas, para torná-la mais legível – os pontos finais instruem o PHP a transformar tudo isto em uma única string.

↑ Com a query insert armazenada em uma string, a mesma estará pronta para ser transferida via função `mysqli_query()`.

↑ não existem

- 1 Conecte-se com `mysqli_connect()`.
- 2 Monte a string da query.
- 3 Execute a consulta com `mysqli_query()`.
- 4 Feche a conexão com `mysqli_close()`.

O ponto final diz ao PHP para juntar esta string com a que vier na linha seguinte.

**P:** Por que um `INSERT` feito em um banco é chamado de query? O termo “query” não significa que nós estamos pedindo alguma coisa ao banco de dados?

**R:** Sim, “query” significa que algo está sendo requisitado... você está pedindo ao banco que faça alguma coisa. Nas aplicações de bancos de dados MySQL, o termo “query” é genérico, referindo-se a qualquer comando SQL que seja executado, incluindo tanto o armazenamento quanto a obtenção de dados.

**P:** Por que a instrução `INSERT` não é criada como uma única string gigante?

**R:** Tenha em mente que a instrução `INSERT` é armazenada como uma única string gigante, mesmo que tenha sido criada a partir de várias strings menores. O ideal seria que a instrução `INSERT` fosse programada como uma única string. Mas, assim como várias instruções SQL, `INSERT` é bastante longa e não cabe em uma linha “normal” de código. Assim, ela é mais fácil de ler se programada como strings menores, conectadas por pontos finais.

**P:** É realmente necessário listar os nomes das colunas ao se fazer um `INSERT`?

**R:** Não. Você pode omitir os nomes das colunas na instrução `INSERT`. Nesse caso, terá de fornecer valores para todas as colunas da tabela, na mesma ordem em que aparecem na estrutura da tabela. Tendo isso em mente, em geral é mais seguro e mais conveniente especificar os nomes das colunas.

a função `mysqli_query()`

## Consulte o banco de dados MySQL com PHP

A função `mysqli_query()` precisa de duas informações para executar uma query: uma conexão ao banco e uma string de query SQL.

```
mysqli_query(conexão_ao_banco, consulta);
```

Esta é uma conexão que já foi estabelecida, através da função `mysqli_connect()`.

- 1 Conecte-se com `mysqli_connect()`.
- 2 Monte a string da query.
- 3 Execute a consulta com `mysqli_query()`.
- 4 Feche a conexão com `mysqli_close()`.

Esta é a query SQL que será realizada... aquela que armazenamos em uma string.  
Lembre-se, estas variáveis serão diferentes no seu próprio banco de dados.

A conexão requerida pela função `mysqli_query()` foi retornada para você por `mysqli_connect()`. Caso isso tudo esteja um pouco difícil de visualizar, eis o código que estabeleceu a conexão:

```
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensroot', 'aliendatabase')
```

↑ or die('Erro ao conectar ao servidor mysql.');

A conexão ao banco de dados já foi armazenada na variável \$dbc.

Assim, você tem uma conexão ao banco (\$dbc) e uma query SQL (\$query). Só o que falta é repassá-los para a função `mysqli_query()`.

```
$result = mysqli_query($dbc, $query);
```

↑ or die('Error querying database.')

O resultado da query

A query  
↑ A conexão ao banco de dados.

Uma query SQL corresponde a uma requisição escrita em código SQL, que é enviada ao servidor do banco de dados.

Este código mostra que ou solicitar a função `mysqli_query()` não corresponde a, apenas, uma comunicação de uma só via. A função responde a você, retornando uma informação que é armazenada na variável \$result. Mas nenhum dado é realmente retornado a partir da consulta **INSERT** – a variável \$result apenas registra a informação referente ao sucesso – ou não – da query emitida por `mysqli`.

A função `mysqli_query()` requer uma conexão ao banco de dados e uma string de query para executar uma consulta SQL.

## Encerre sua conexão com mysqli\_close()

Uma vez que só estamos interessados em executar a instrução INSERT simples, a interação com o banco está terminada, pelo menos no que concerne ao nosso script. E, ao terminar de usar uma conexão ao banco, feche-a. As conexões se fecham sozinhas quando o usuário fecha a página, porém, assim como fechar uma porta no mundo real, é um bom hábito tomar a iniciativa de fechar as conexões quando tiver terminado de usá-las. A função PHP mysqli\_close() realiza essa tarefa.

- 1 Conecte-se com mysqli\_connect().
- 2 Monte a string da query.
- 3 Execute a consulta com mysqli\_query().
- 4 Feche a conexão com mysqli\_close().

É um bom hábito fechar a conexão a um banco de dados MySQL, após terminar de usá-lo.

### mysqli\_close(conexão\_ao\_banco);

Aqui, você informa a variável da conexão, utilizada para interagir com o banco de dados.

No caso do script de Owen, precisamos executar a mysqli\_close() e informar qual é a conexão a ser fechada, informação essa que se encontra armazenada na variável \$dbc.

`mysqli_close($dbc);`

Esta variável armazena uma referência à conexão que foi criada por mysqli\_connect() lá atrás, quando a conexão foi aberta pela primeira vez.

Mas se as conexões aos bancos se fecham automaticamente, por que se dar ao trabalho de fechá-las de forma manual?



**Os servidores de bancos de dados só dispõem de um certo número de conexões disponíveis por vez, portanto, elas precisam ser preservadas tanto quanto for possível.**

E quando você fecha uma conexão, isso libera o recurso para que uma nova conexão possa ser criada. Se estiver em um banco de dados compartilhado, apenas cinco conexões poderão ser alocadas para você, por exemplo. E, à medida que você for criando novas aplicações que usem bancos de dados, tenha certeza que quererá manter o maior estoque possível de conexões disponíveis.

*não existem perguntas idiotas e pontos de bala*

*não existem*

## Perguntas Idiotas

*deveria estar fazendo algo com o resultado?*

**P:** Você não poderia, simplesmente, colocar todo o código SQL diretamente na função `mysqli_query()`, em vez de na variável `$query`?

**R:** Seria possível, mas ficaria muito bagunçado. É mais fácil gerenciar o seu código quando você armazena suas queries em variáveis, e depois usa essas variáveis na função `mysqli_query()`.

**P:** O código que referencia a consulta `INSERT`

**R:** Talvez sim. Até aqui, estivemos usando `die()` para finalizar um script e enviar uma mensagem ao navegador caso algo de errado acontecesse. Chegará um momento em que você poderá querer fornecer mais informações ao usuário quando a query não tiver sucesso, nesse caso, você pode usar o resultado da query para determinar o sucesso desta.

## PONTOS DE BALA

- As conexões a bancos de dados precisam de uma localização, um nome de usuário, uma senha e um nome do banco.
- A função `mysqli_connect()` cria uma conexão entre o seu script PHP e o servidor do banco de dados MySQL.
- A função `die()` finaliza o script e retorna uma mensagem caso a sua conexão falhe.
- Para se executar uma query SQL a partir de código PHP, você constrói a query em uma string e depois a executa com uma chamada a `mysqli_query()`.
- Chame a função `mysqli_close()` para fechar uma conexão ao banco MySQL, a partir do PHP, quando tiver terminado de usá-la.



## TEST DRIVE

**Modifique o código do script report.php de forma que, em vez de enviar os dados por email para Owen, eles sejam inseridos no banco de dados MySQL. Depois, teste as modificações que você fez.**

Remova a parte do código do script report.php que envia os dados para Owen. No lugar, coloque o código que inicia a conexão ao seu banco de dados MySQL, construa uma query SQL como uma string PHP, execute a query no banco de dados e depois encerre a conexão.

Eis o novo código PHP em que vimos trabalhando. Não coloque as tags <?php ?> em report.php, uma vez que está adicionando este código a um ponto no script que já se encontra dentro das tags.

```
<?php
$dbc = mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool', 'aliendatabase')
or die('Error connecting to MySQL server.');

$query = "INSERT INTO aliens_abduction (first_name, last_name, "
        . "when_it_happened, how_long, how_many, alien_description, "
        . "what_they_did, tang_spotted, other, email) "
        . "VALUES ('Sally', 'Jones', '3 dias atrás', '1 dia', 'quadro', '',
        . "'verde com seis tentáculos', 'Nós só conversamos e brincamos com um cachorro', '',
        . "'sim', 'Acho que posso ter visto o seu cão. Entre em contato!', '',
        . "'sally@grags-list.net')";

$result = mysqli_query($dbc, $query)
or die('Error querying database.');

mysqli_close($dbc);
?>
```

Envie o novo report.php para o seu servidor web e depois abra a página report.html em um navegador para acessar o formulário “Relatar uma Abdução”. Preencha o formulário e clique no botão “Relatar Abdução” para armazenar os dados no banco. Agora, inicie a sua ferramenta MySQL e realize uma query SELECT para visualizar as modificações feitas no banco.

Relatório de Abdução - FROM alien_abduction						
Nome	Sobrenome	Quando	Por quanto tempo	Quantos	Alien	Ação
Sally	Jones	3 dias atrás	1 dia	1	quadro	Nós só conversamos e brincamos com um cachorro
Danny	Griggs	3 dias atrás	1 dia	1	verde com seis tentáculos	Acho que posso ter visto o seu cão. Entre em contato!

Isto está correto? Escreva abaixo se você acha que é isto que o script deveria estar fazendo, e por quê.

*deveria utilizar a classe PDO*

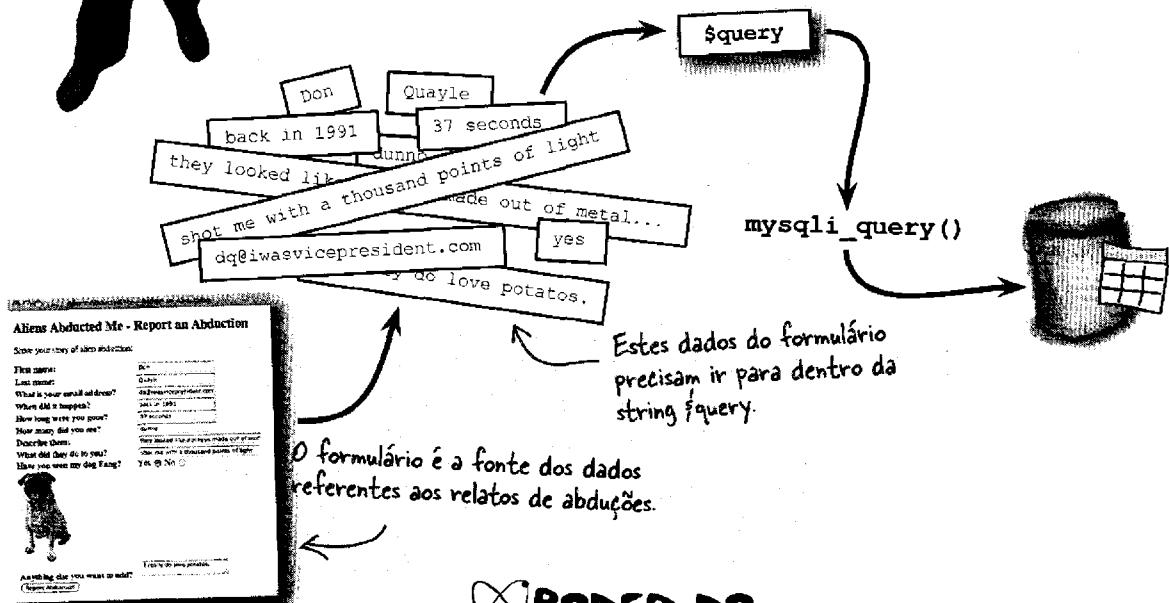
`use $_POST na query INSERT`



Espere um minuto. Esse trabalho todo que estamos fazendo não é com o objetivo de pegar dados diretamente do formulário e armazená-los no banco? Parece que a consulta está inserindo os mesmos dados, não importa o que tenha sido digitado no formulário. Não estou vendo nenhuma automação com este script PHP.

**Isto é um grande problema. A consulta `INSERT` precisa inserir os dados do formulário, e não strings estáticas.**

A query que construímos consiste de strings definidas diretamente no código, e não de strings resultantes do texto que foi digitado no formulário. Para que o script funcione com o formulário, precisamos fazer com que os dados digitados sejam colocados dentro da string de query.



## PODER DO CÉREBRO

Qual código PHP poderia nos ajudar a enviar os valores do formulário de Owen para dentro da query `INSERT`?

## \$\_POST fornece os dados do formulário

A boa notícia é que o script report.php já tem os dados do formulário armazenados em variáveis, graças à superglobal `$_POST`.  
Você lembra deste código PHP?

```
$name = $_POST['firstname'] . ' ' . $_POST['lastname'];
$when_it_happened = $_POST['whenithappened'];
$how_long = $_POST['howlong'];
$how_many = $_POST['howmany'];
$alien_description = $_POST['aliendescription'];
$what_they_did = $_POST['whattheydid'];
$fang_spotted = $_POST['fangspotted'];
$email = $_POST['email'];
$other = $_POST['other'];
```

A superglobal `$_POST` já está sendo usada para extrair os dados de cada campo do formulário de Owen e para inseri-los em variáveis.

Lembre-se, o nome que você usar para `$_POST` precisa ser o mesmo do campo correspondente, no formulário HTML.

Assim, você já tem os dados do formulário e só precisa incorporá-los na instrução `INSERT`. Mas é preciso fazer uma pequena modificação, primeiro. Agora que você não está mais enviando os dados por email, não precisa da variável `$name`. Você ainda precisa do nome e do sobrenome do usuário, para que o relato possa ser adicionado ao banco de dados – mas precisa tê-los em variáveis separadas.

```
$first_name = $_POST['firstname'];
$last_name = $_POST['lastname'];
```

O nome e o sobrenome do usuário agora são armazenados em variáveis separadas, para que possam ser inseridos em colunas diferentes da tabela `aliens_abduction`.



### Exercício

Escreva o código PHP para criar a string de query `INSERT` armazenada na variável `$query`, certificando-se de que ela esteja armazenando os dados do formulário na tabela `aliens_abduction`, ao ser executada.

```
$query = "INSERT INTO aliens_abduction (".
..... "firstname, lastname, ... )",
..... "VALUES ('$first_name', '$last_name', ... )";
```



## Exercício Solução

Escreva o código PHP para criar a string de query INSERT armazenada na variável \$query, certificando-se de que ela esteja armazenando os dados do formulário na tabela aliens\_abduction, ao ser executada.

Os nomes das colunas  
aparecem na instrução SQL  
exatamente como antes.

```
$query = "INSERT INTO aliens_abduction (first_name, last_name, when_it_happened, how_long, " .  
        "how_many, alien_description, what_they_did, fang_spotted, other, email)" .  
        "VALUES ('$first_name', '$last_name', '$when_it_happened', '$how_long', '$how_many', "  
        "'$alien_description', '$what_they_did', '$fang_spotted', '$other', '$email');
```

Em vez de dados estáticos sobre a  
abdução de Sally Jones, agora nós  
inserimos os dados que o usuário  
digitou no formulário.

A ordem das variáveis precisa ser a mesma dos  
nomes das colunas, para que os dados sejam  
armazenados nas colunas corretas da tabela.

não existem

## Perguntas Ídiotas

**P:** Eu tenho mesmo que criar todas essas variáveis para armazenar os dados de \$\_POST? Não posso, simplesmente, referenciar os dados de \$\_POST diretamente para dentro da string \$query?

**R:** Pode. Não há nada que lhe impeça de colocar \$\_POST diretamente em uma query. Porém, é um bom的习惯 de programação isolar os dados do formulário antes de fazer qualquer coisa com eles. Isso acontece devido ao fato de ser relativamente comum a necessidade de processar os dados de alguma forma, antes de inseri-los no banco. Por exemplo, existem técnicas inteligentes que hackers podem usar para "sequestrar" suas query, digitando dados perigosos no formulário. Você verá como evitar tais ataques no Capítulo 6. Para manter a simplicidade, neste capítulo não fazemos nenhum processamento nos dados, mas isso não significa que não deva adquirir o hábito de armazenar os dados do formulário nas suas próprias variáveis antes de colocá-los em uma query.

**P:** Faz alguma diferença usar aspas simples ou duplas? Posso usar aspas simples em torno de toda a consulta, e aspas duplas em torno de cada variável?

**R:** Sim, faz diferença. E não, você não pode usar aspas simples em torno da query inteira, com aspas duplas em

torno das variáveis. O motivo é que o PHP trata as strings de forma diferente, dependendo da forma como elas aparecem: dentro de aspas simples ou duplas. A diferença é que as aspas simples representam **exatamente** o texto contido nelas, enquanto, ao se fazer uso das aspas duplas, significa que há algum processamento adicional no texto quando se usam aspas duplas. Esse processamento resulta no fato de que os valores das variáveis colocadas dentro de aspas duplas são mostradas em strings, no lugar do nome da variável. Isso é bastante útil, e é por isso que as aspas duplas geralmente são preferidas para se criar strings de query SQL.

**P:** Não seria possível criar strings de query simplesmente concatenando-se as variáveis com o código SQL?

**R:** Sim, e se você decidir usar esse método, certamente poderá usar aspas simples em vez das duplas. Mas as strings de query já tendem a ser bagunçadas por natureza, de modo que tudo o que puder fazer para torná-las mais legíveis é bom – inserir variáveis diretamente em uma string dentro de aspas duplas, em vez de as concatená-las com aspas simples, definitivamente torna as strings mais fáceis de serem entendidas.



## Exercício

Vamos usar tudo o que já aprendemos para terminar o script PHP de Owen, para que ele possa armazenar com sucesso os dados sobre as abduções em um banco de dados. Escreva abaixo o código PHP necessário para completar o script `report.php`.

```
<?php
.....
$when_it_happened = $_POST['whenithappened'];
$how_long = $_POST['howlong'];
$how_many = $_POST['howmany'];
$alien_description = $_POST['aliendescription'];
$what_they_did = $_POST['whattheydid'];
$fang_spotted = $_POST['fangspotted'];
$email = $_POST['email'];
$other = $_POST['other'];

$dbc =
.....
$query = "INSERT INTO aliens_abduction (first_name, last_name, when_it_happened, how_long, "
        . "how_many, alien_description, what_they_did, fang_spotted, other, email) "
        . "VALUES ('$first_name', '$last_name', '$when_it_happened', '$how_long', '$how_many', "
        . "'$alien_description', '$what_they_did', '$fang_spotted', '$other', '$email')";

$result =
.....
echo 'Thanks for submitting the form.<br />';
echo 'You were abducted ' . $when_it_happened;
echo ' and were gone for ' . $how_long . '<br />';
echo 'Number of aliens: ' . $how_many . '<br />';
echo 'Describe them: ' . $alien_description . '<br />';
echo 'The aliens did this: ' . $what_they_did . '<br />';
echo 'Was Fang there? ' . $fang_spotted . '<br />';
echo 'Other comments: ' . $other . '<br />';
echo 'Your email address is ' . $email;
?>
```

## Exercício Solução

Vamos usar tudo o que já aprendemos para terminar o script PHP de Owen, para que ele possa armazenar com sucesso os dados sobre as abduções em um banco de dados. Escreva abaixo o código PHP necessário para completar o script report.php.

```
<?php
$first_name = $_POST['firstname'];
$last_name = $_POST['lastname'];
$when_it_happened = $_POST['whenithappened'];
$how_long = $_POST['howlong'];
$show_many = $_POST['howmany'];
$alien_description = $_POST['aliendescription'];
$what_they_did = $_POST['whattheydid'];
$fang_spotted = $_POST['fangspotted'];
$email = $_POST['email'];
$other = $_POST['other'];

$dbc =
    mysqli_connect('data.aliensabductedme.com', 'owen', 'aliensrool', 'aliendatabase')
or die('Error connecting to MySQL server.');

$query = "INSERT INTO aliens_abduction (first_name, last_name, when_it_happened, how_long, "
    . "how_many, alien_description, what_they_did, fang_spotted, other, email) "
    . "VALUES ('$first_name', '$last_name', '$when_it_happened', '$show_long', '$show_many', "
    . "'$alien_description', '$what_they_did', '$fang_spotted', '$other', '$email')";

$result =
    mysqli_query($dbc, $query)
or die('Error querying database.');

mysqli_close($dbc);
?>
```

As novas variáveis armazenam o nome e o sobrenome do usuário, do jeito que ele digitou no formulário.

Você precisa se conectar ao banco de dados e fornecer as informações necessárias antes de realizar quaisquer consultas SQL a partir do PHP.

A query é construída como uma string PHP. Certifique-se de usar os dados extraídos dos campos do formulário.

Execute a query ao banco de dados - isto é o que insere os dados!

Feche a conexão ao banco.

Confirme se o formulário foi submetido com sucesso, da mesma forma como fazia no script antigo.



# TEST DRIVE

## Modifique o script de Owen de forma a usar os dados do formulário para fazer um INSERT.

Remova a variável \$name do script report.php, adicione as variáveis \$first\_name e \$last\_name e modifique \$query para usar as variáveis do formulário, em vez de texto estático na instrução INSERT. Envie a nova versão do script para o servidor e teste-o, submetendo o formulário da página report.html algumas vezes, digitando dados diferentes a cada vez.

**Aliens Abducted Me - Report an Abduction**

Share your story of alien abduction:

First name:  Last name:  What is your email address?  When did it happen?  How long were you gone?  How many did you see?  Describe them:  What did they do to you?  Yes  No  Have you seen my dog Fung?

**Aliens Abducted Me - Report an Abduction**

Share your story of alien abduction:

First name:  Last name:  What is your email address?  When did it happen?  How long were you gone?  How many did you see?  Describe them:  What did they do to you?  Have you seen my dog Fung?  Yes  No  Please vote for me.

Agora, use sua ferramenta MySQL para executar um SELECT e visualizar o conteúdo da tabela aliens\_abduction.

Os novos relatos de abduções aparecem na tabela, exatamente como você espera!

```

File Edit Window Help Standard
SELECT * FROM aliens_abduction;
+-----+-----+-----+-----+
| first_name | last_name | email | when |
+-----+-----+-----+-----+
| Dan        | Quive    | dquive@verizon.net | back in 1991 |
| Af         | Nader    | aliensaremyfathers.com | last November |
| Sally      | Jones    | sallyjones@earthlink.net | summer of 1999 |
| Sally      | Jones    | sallyjones@earthlink.net | was out of gas, so it was a pretty gooditudin |
+-----+-----+-----+-----+

```

Há uma linha extra de dados para Sally Jones, que foi colocada antes de você corrigir a consulta INSERT. Não se preocupe, você aprenderá como remover dados indesejados no próximo capítulo.

## Owen precisa de ajuda para peneirar seus dados

O novo e aprimorado script report.php está fazendo o seu trabalho, automatizando o processo de adicionar relatos de abduções ao banco de dados. Owen pode relaxar e esperar que os relatórios cheguem... exceto pelo fato de que há um novo problema. Receber mais dados não está, exatamente, tornando mais fácil selecionar os relatos que representem um potencial avistamento de Fang.

Eu realmente estou feliz, agora que tenho um banco de dados sendo automaticamente preenchido com os relatos de abduções submetidos pelos usuários. Mas isso não está ajudando a isolar os relatos que poderiam me ajudar a encontrar Fang.

### Owen precisa de uma forma para localizar dados específicos, como por exemplo, as abduções nas quais Fang foi avistado.

Você sabe qual coluna do banco contém as informações em questão: fang\_spotted. Esta coluna contém yes (sim) ou no (não), dependendo do fato de o abduzido ter relatado se viu Fang – ou não. Assim, o que precisa é de uma forma de selecionar apenas os relatos da tabela aliens\_abduction que tenham um valor yes na coluna fang\_spotted.

Você sabe que a seguinte query SQL retorna todos os dados da tabela:

```
SELECT * FROM aliens_abduction
```

A instrução SQL SELECT lhe permite incluir uma cláusula para controlar os dados retornados pela query. Trata-se de WHERE, e com ela você diz exatamente como quer filtrar os resultados da consulta. No caso de Owen, isso significa selecionar apenas os relatos de abduções onde a coluna fang\_spotted seja igual a yes.

```
SELECT * FROM aliens_abduction WHERE fang_spotted = 'yes'
```

Esta parte da consulta SELECT permanece a mesma – a cláusula WHERE cuida de filtrar os resultados.

O valor da coluna precisa ser definido, para que os dados possam ser selecionados.

Esta cláusula reduz o volume de dados retornados pela query, incluindo apenas os dados em que a coluna fang\_spotted esteja definida como yes.



## TEST DRIVE

### Teste a consulta SELECT com uma cláusula WHERE para encontrar dados específicos.

Use uma consulta SELECT com uma cláusula WHERE na sua ferramenta MySQL para procurar por dados de abduções que envolvam, especificamente, avistamentos de Fang.

```

File Edit Window Help HaveYouSeenHim
mysql> SELECT * FROM alien_abduction WHERE fang_spotted = 'yes';

+-----+-----+-----+-----+-----+
| first_name | last_name | when_it_happened | how_long | how_many |
+-----+-----+-----+-----+-----+
| Sally      | Jones     | 3 days ago       | 1 day    | food      |
| Sally      | Jones     | 3 days ago       | 1 day    | fear      |
| Don        | Osgood    | back in 1991     | 37 seconds | dreams    |
| Bill       | Manner   | summer of '69     | 2 hours   | I don't know |
| Mickey     | Mikens   | Just now         | 143 minutes...and counting | hundreds |
+-----+-----+-----+-----+-----+
5 rows in set (0.0005 sec)

```

```

+-----+-----+
| fang_spotted | other |
+-----+-----+
| yes          | I may have seen your dog. Contact me. |
| yes          | I may have seen your dog. Contact me. |
| yes          | I really do love potatos. |
| yes          | I was out of gas, so it was a pretty good abduction. |
| yes          | I'm thinking about designing a helmet to thwart future abductions. |
+-----+-----+

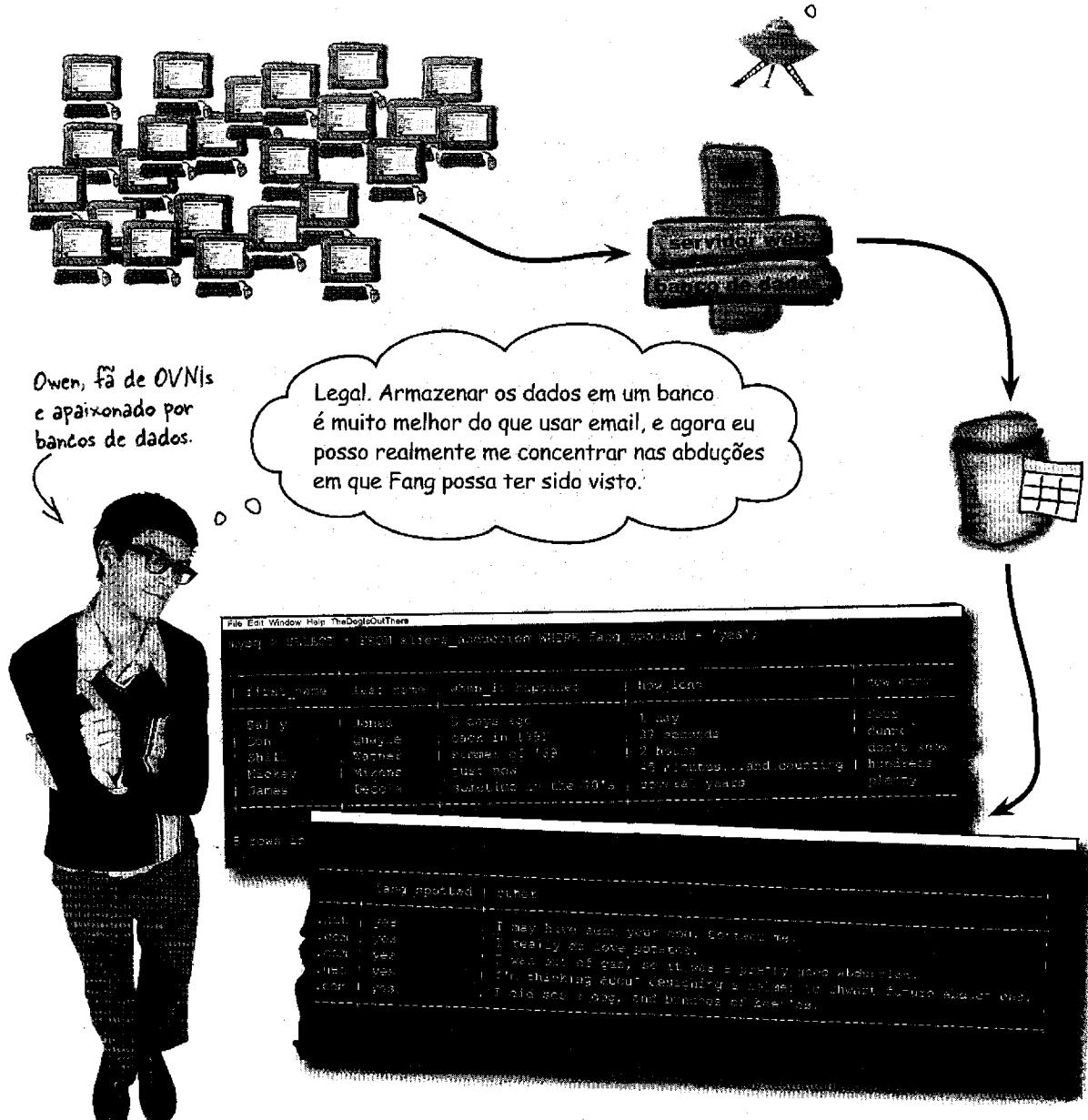
```

Todos estes registros têm  
a coluna fang\_spotted  
definida como yes.

*banco de dados mysql do owen está funcionando*

## Owen sai em busca de Fang

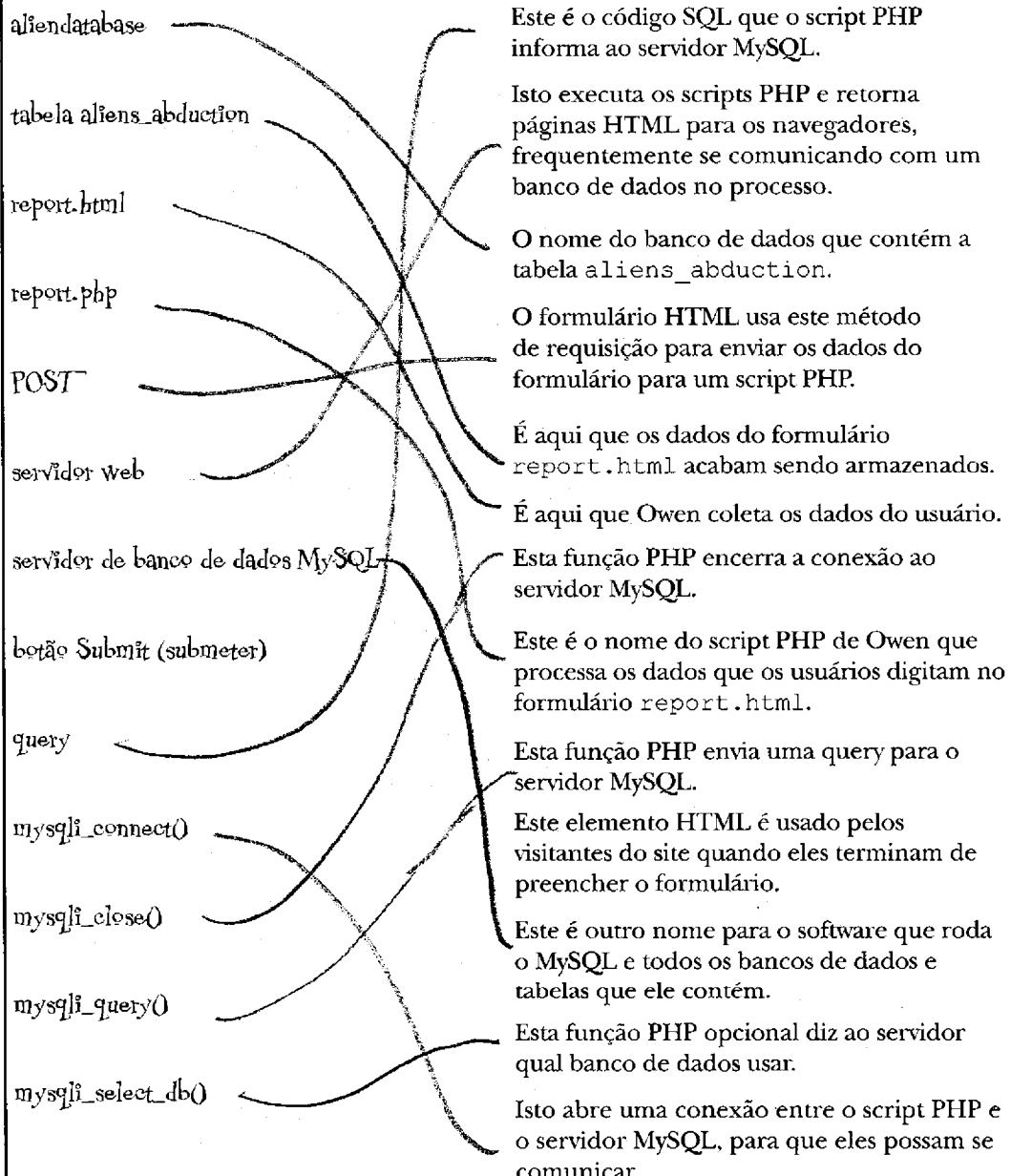
Graças ao PHP e às suas funções que fazem interface com o MySQL, o servidor de banco de dados MySQL de Owen está recebendo os dados das abduções a partir de um formulário HTML e os está armazenando em uma tabela do banco. Os dados ficam guardados de forma segura na tabela, até que Owen tenha a oportunidade de peneirá-los. E, quando está pronto para fazê-lo, basta uma simples query SELECT para isolá-los dos relatos de abduções que, potencialmente, envolvam Fang.



# 2

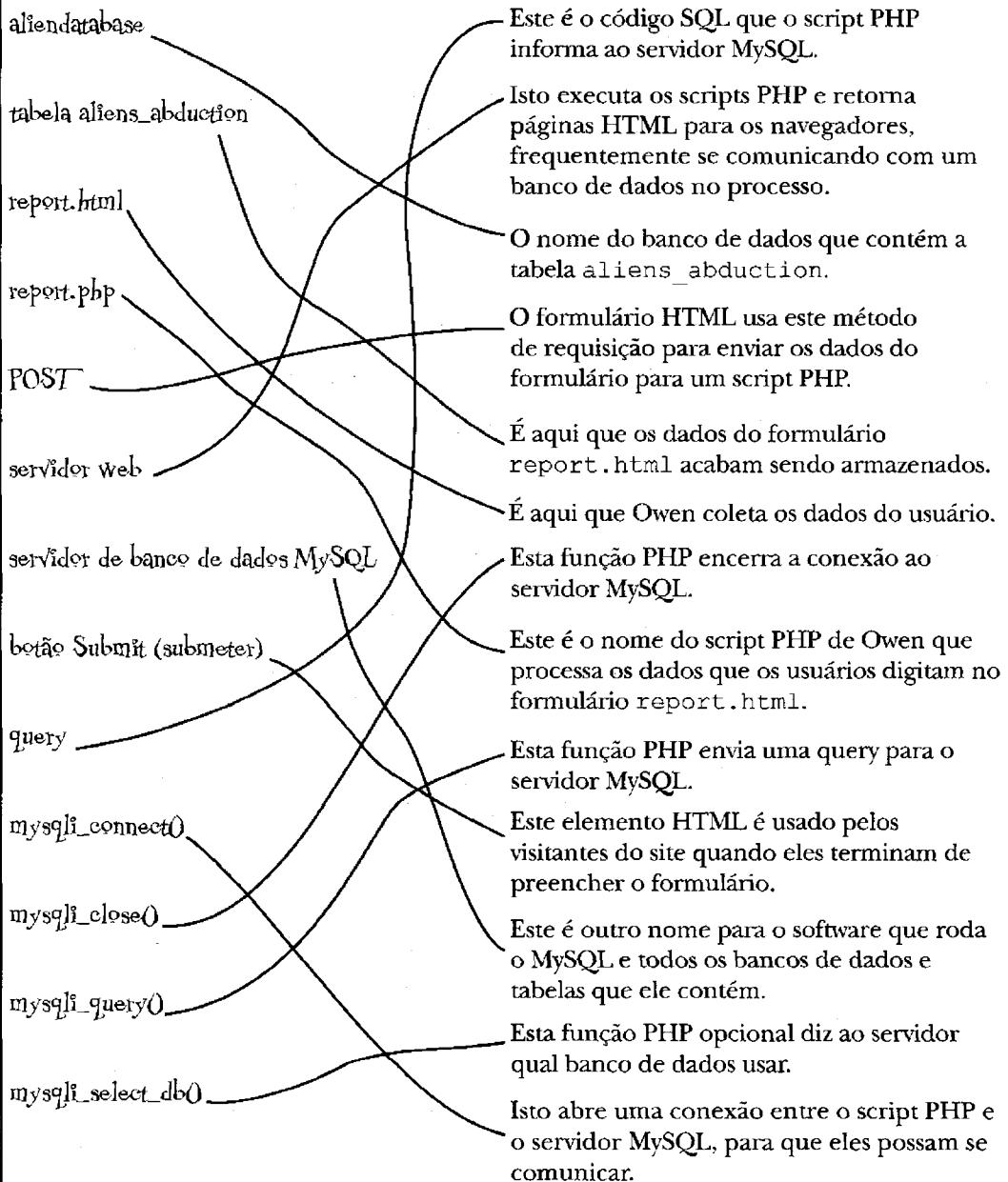
## Owen faz o quê?

Mesmo que você ainda não tenha organizado tudo, ligue cada componente HTML, PHP e MySQL ao que você acha que ele faz.



# QUEM FAZ O QUE? SOLUÇÃO

Mesmo que você ainda não tenha organizado tudo, ligue cada componente HTML, PHP e MySQL ao que você acha que ele faz.





Confira as sugestões abaixo para este desafio envolvendo novas ideias e pensar sobre a melhor forma de representá-las. Esse é o foco principal do próximo capítulo... você está pronto?



### 3 Crie e Preencha um Banco de Dados

# **Criando os Seus Próprios Dados**



**Nem sempre você tem os dados de que precisa.** Às vezes, você mesmo tem que *criar os dados* que pretende usar. E, outras vezes, você precisa *criar as tabelas* que irão armazenar esses dados. E, ainda, você precisa *criar o banco* que armazenará os dados que você precisa criar. Confuso? Você não ficará. Prepare-se para aprender como criar seus próprios bancos de dados e tabelas. E se isso não for o suficiente, você ainda criará, no processo, a sua primeira aplicação combinando PHP e MySQL.

precisa-se do aplicativo lista de email

## A loja Elvis está aberta

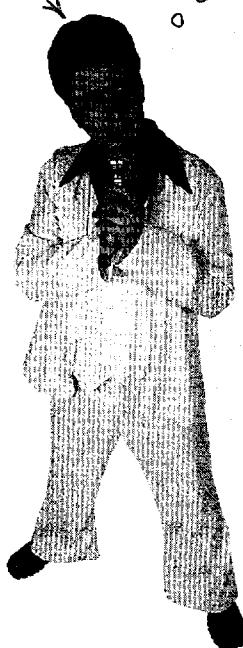
Elmer Priestley abriu uma loja especializada em produtos Elvis, QueroSerElvis.com (makemeelvis.com). A demanda tem sido enorme. Ele já vendeu um monte de ternos de poliéster, muitas costeletas falsas e centenas de pares de óculos.

Cada vez que alguém compra algo, Elmer coleta um novo endereço de email. Ele os utiliza para enviar avisos sobre promoções na sua loja. Da forma como está agora, Elmer tem que copiar, manualmente, cada endereço da sua lista e depois colar um por um nos seus emails promocionais. Funciona, mas demanda muito tempo e esforço.

Elmer, o  
inquestionável  
Rei da lojas  
online de  
produtos Elvis  
Presley.

Isto está demorando  
muito. Eu preferiria passar  
o meu tempo imitando  
Elvis, e não enviando emails  
manualmente.

Elmer escreve  
este email e  
copia e cola  
cada endereço  
no campo "To".



To:

Subject: Grande promoção

Caros Elvismaníacos,

Grande promoção esta semana em Makemeelvis.com! Costeletas de pêlo de cavalo genuíno com 20% de desconto!

E não se esqueçam da promoção "compre um e leve outro de graça" para os ternos de poliéster - só dura mais três dias!

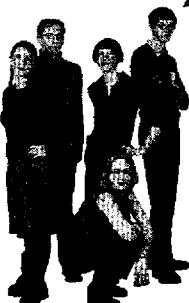
Elmer está passando tempo demais copiando e colando emails no campo "To" do seu programa de email. Ele deseja simplificar a tarefa de adicionar novos endereços e de enviar emails em massa.

Elmer já coletou 328 endereços de email até o momento, e a cada dia chegam mais.

### Elmer's customer mailing list:

Anderson	Jillian	jill_anderson@breakneckpizza.com
Joffe	Kevin	joffe@simuduck.com
Newsome	Amanda	aman2uv@breakneckpizza.com
Garcia	Ed	ed99@bot0msup.com
Roundtree	Jo-Ann	jojoround@breakneckpizza.com
Briggs	Chris	cbriggs@boards-r-us.com
Harte	Lloyd	hovercraft@breakneckpizza.com
Toth	Anne	AnneToth@leapinlimos.com
Wiley	Andrew	andrewwiley@objectville.net
Palumbo	Tom	paloftmine@mightygumball.net
Ryan	Alanna	angrypirate@breakneckpizza.com
McKinney	Clay	clay@starbuzzcoffee.com
Meeker	Ann	annmeeker@chocoholic-inc.com
Powers	Brian	bp@honey-dalt.com
Manson	Anne	am86@objectville.net
Mandel	Debra	debmamonster@breakneckpizza.com
Tedesco	Janis	janistedesco@starbuzzcoffee.com
Taiwar	Vikram	viki@starbuzzcoffee.com
Szwed	Joe	szwedjoe@objectville.net
Sherdan	Diana	dianerid@mightygumball.net
Snow	Edward	snowman@tikibeanlounge.com
Otto	Glenn	glenn0098@objectville.net
Hardy	Anne	anneh@bot0msup.com
Deal	Mary	nobideal@starbuzzcoffee.com
Jagel	Ann	dreamgirl@breakneckpizza.com
Melfi	James	dmelfi@bot0msup.com
Oliver	Lee	leeliver@weatherorama.com
Parker	Annie	annepe@starbuzzcoffee.com
Ricci	Peter	ricciman@tikibeanlounge.com
Reno	Grace	grace23@objectville.net
Moss	Zelda	zelda@weatherorama.com
Day	Clifford	cliffmighi@breakneckpizza.com
Boiger	Joyce	joyce@chocoholic-inc.com
Blunt	Anne	anneblunt@breakneckpizza.com
Bolling	Lindy	lindy@tikibeanlounge.com
Gares	Fred	fgares@objectville.net
Jacobs	Anne	anne99@objectville.net

Estas pessoas estão  
na lista de emails de  
Elmer e querem a sua  
ajuda para ficar um  
pouco mais parecidas  
com Elvis.



## **Elmer precisa de uma aplicação**

Uma aplicação é um software elaborado para atender a um determinado propósito dos seus usuários. Elmer precisa de uma aplicação que mantenha o registro da sua lista de endereços de email e que lhe permita enviar emails para as pessoas da lista clicando apenas em um botão. Eis a forma como ele quer que funcione:

- Ir até uma página web e digitar uma mensagem de email.
- Clicar em um botão Submit na página, e a mensagem ser automaticamente enviada para toda a lista de "QueroSerElvis.com".
- Fazer com que a lista de emails cresça sozinha, permitindo que novos clientes se cadastrem através de um formulário web.

Com essa lista de necessidades que a aplicação precisa atender, Elmer pode visualizá-la em detalhes...

Uma aplicação web é um site dinâmico elaborado para atender a um determinado propósito dos seus usuários.

Esta parte de emails se parece bastante com a aplicação Abdução Alienígena de Owen, mas a diferença aqui é que a lista de emails de Elmer irá crescer sozinha, e as suas mensagens serão enviadas para a lista inteira. O objetivo da aplicação de Elmer é automação!

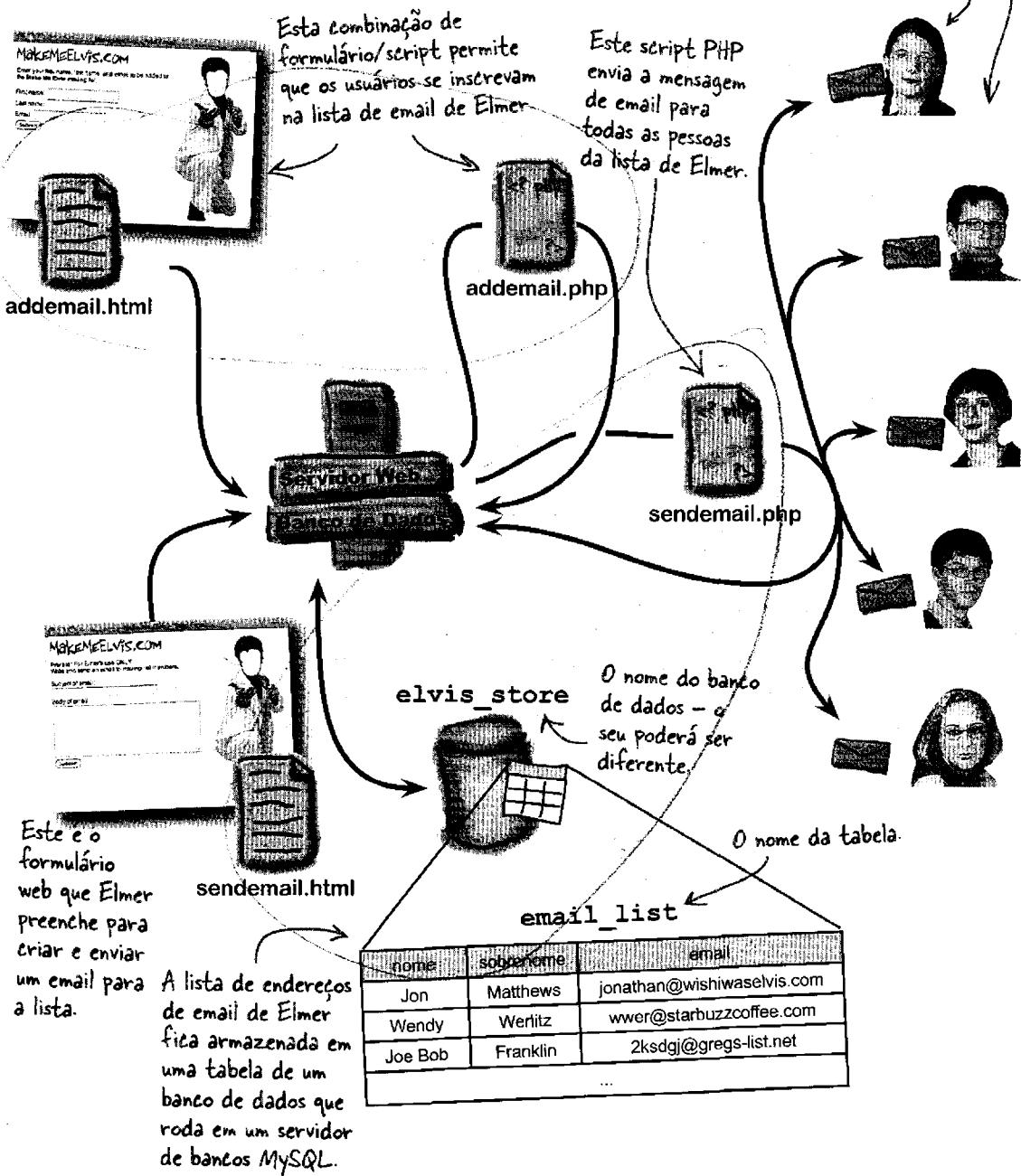


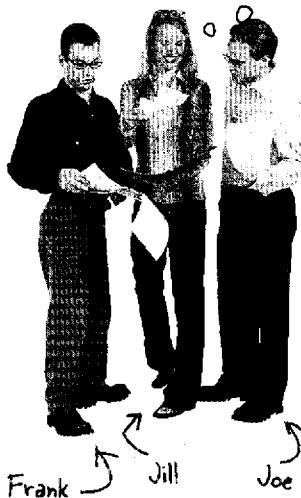
A aplicação web QueroSerElvis.com (MakeMeElvis.com) consiste de dois componentes principais: um formulário para enviar mensagem de email para as pessoas na lista de Elmer e um formulário para permitir que novos clientes se inscrevam na lista. Com esses dois formulários em mente, faça um esboço de projeto para a aplicação de Elmer.

## Visualize o projeto da aplicação de Elmer

Sempre é útil visualizar o projeto de uma aplicação antes de mergulhar nos detalhes do desenvolvimento. Isso implica descobrir quais páginas e scripts estarão envolvidos, como eles se conectam e, talvez o mais importante, como você irá armazenar os dados no banco MySQL.

Estas pessoas estão na lista de emails de Elmer e recebem os e-mails que ele envia para a lista.





Então, por onde começamos a criar uma aplicação com PHP e MySQL? Será que nós escrevemos o script PHP primeiro e depois criamos a tabela para armazenar os dados? Ou será que devemos criar a tabela primeiro e depois o script?

**Joe:** Não entendo que diferença faz. Nós vamos precisar da tabela e do script para a aplicação funcionar.

**Frank:** Isso é verdade, mas acho que nós devemos escrever o script primeiro, para podermos testar o código PHP antes de conectá-lo ao banco de dados.

**Jill:** Mas o script PHP é completamente dependente do banco de dados. Será difícil testar o script se não tivermos um banco ao qual ele possa se conectar.

**Frank:** Será que não poderíamos criar o script e omitir apenas a parte específica do código que faz a conexão com o banco? Poderíamos fazer tudo exceto interagir realmente com o banco de dados. Isso ainda seria útil, certo?

**Joe:** Não necessariamente. Lembre-se, o único trabalho do script é selecionar os dados digitados em um formulário HTML e colocá-los em um banco. Ou, se a função for enviar emails para uma lista, o script lê os dados do banco e gera uma mensagem de email para cada usuário. De uma forma ou de outra, o banco é essencial para o script.

**Jill:** É verdade, mas ainda nem começamos a pensar no formulário HTML. Onde ele se encaixa nisto tudo? Estou achando que precisamos criar o banco de dados antes de sequer pensarmos em escrever o script.

**Frank:** É isso! Primeiro criamos o formulário HTML, depois analisamos que tipos de dados vão ser colocados no banco e, feito isso, nós finalizamos tudo com o script.

**Joe:** Não sei se isso realmente faz sentido. Como podemos criar um formulário HTML se não temos 100% de certeza de que tipo de dados precisamos obter do usuário?

**Jill:** Joe está certo. O formulário HTML ainda nos leva de volta ao problema de precisarmos saber quais serão os tipos de dados usados pela aplicação. Os dados comandam tudo, então nós provavelmente deveríamos criar o banco de dados e a tabela primeiro, depois o formulário HTML e, finalmente, o script que reage à submissão do formulário.

**Frank:** Gostei. Vamos fazer isso!

**Joe:** Eu ainda acho que nós, provavelmente, precisaríamos elaborar os passos específicos de como esta aplicação será montada...

**Escreva abaixo os passos específicos que você acha que serão necessários para ir do projeto até a implementação de QueroSerElvis.com.**

*Criar um banco, criar a tabela, inserir dados, conectar ao banco, enviar emails, etc.*

# PLANEJE ANTES

Nós realmente precisamos de um plano para montar a aplicação de Elmer. Passo-a-passo, nós podemos nos concentrar em uma coisa de cada vez, sem que fiquemos sobrecarregados.

## 1 Criar um banco de dados e uma tabela para a lista de emails.

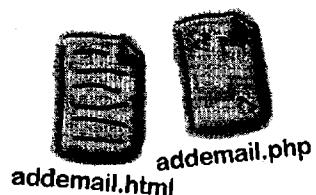
Esta tabela irá armazenar os nomes, sobrenomes e endereços de email de todos os inscritos na lista de Elmer.

elvis\_store



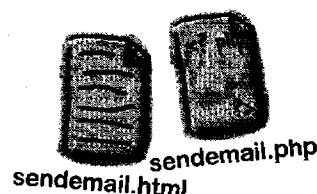
## 2 Criar um formulário web e um script PHP Adicionar Email para inserir novos clientes à lista.

Aqui, nós criaremos um formulário e um script que permitirão ao cliente digitar seu nome, sobrenome e endereço de email, e então adicioná-los à lista.



## 3 Criar um formulário web e um script PHP Enviar Email para enviar emails para a lista.

Finalmente, iremos criar um formulário web que permitirá a Elmer compor uma mensagem de email e, o que é mais importante, um script que enviará esta mensagem para todos os nomes inscritos na tabela da lista de emails.



## Tudo começa com uma tabela

Na verdade, tudo começa com um banco de dados, que é basicamente um container para o armazenamento de dados. Já vimos, no capítulo anterior, o modo como os bancos de dados são divididos internamente em mais containers, chamados **tabelas**.

Assim como os dias e semanas em um calendário, a tabela se compõe de **colunas** e linhas de dados. As colunas consistem de um tipo específico de dados, como por exemplo "nome", "sobrenome" e "email". As **linhas** são conjuntos de colunas em que uma linha consiste de um elemento de cada uma das colunas. Um exemplo de linha seria "Wendy, Werlitz, wwer@starbuzzcoffee.com."

calendario						
Domingo	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
...						

The diagram shows two tables side-by-side. The left table is labeled 'email\_list' and has columns 'name', 'sobrenome', and 'email'. It contains four rows with data: Jon Mathews, jonathan@wishiwaselvis.com; Wendy Werlitz, www@starbuzzcoffee.com; and Joe Bob Franklin, 2ksdg@gregs-list.net. The right table is also labeled 'email\_list' and has the same three columns. It contains the same four rows of data. Arrows point from the text 'Uma coluna' to the column headers 'name', 'sobrenome', and 'email', and from the text 'Uma linha' to the first row of each table.

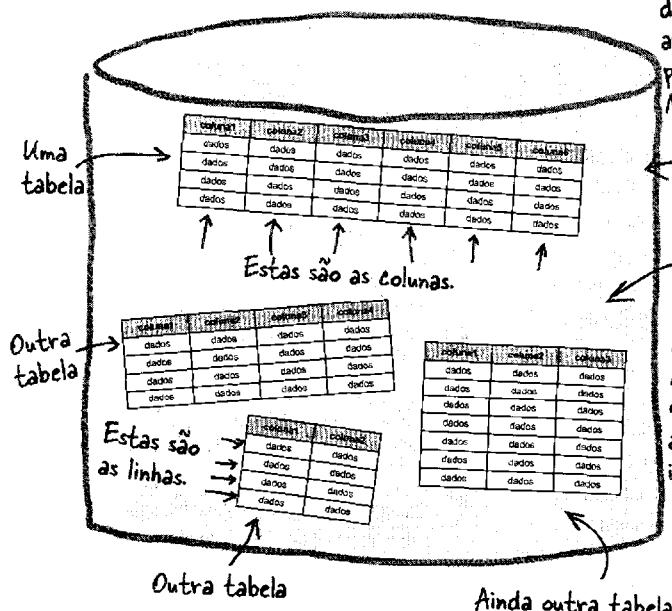
email_list		
name	sobrenome	email
Jon	Mathews	jonathan@wishiwaselvis.com
Wendy	Werlitz	www@starbuzzcoffee.com
Joe Bob	Franklin	2ksdg@gregs-list.net

Ambas estas estruturas de dados são tabelas.

Em geral, todas as tabelas de um banco de dados tem alguma relação umas com as outras, mesmo que essa associação às vezes seja superficial. É comum uma aplicação web consistir de várias tabelas conectadas umas às outras por meio dos seus dados. Mas todas as tabelas ainda são feitas de colunas e linhas.

Um banco de dados, que é armazenado por um servidor MySQL.

Pense no banco de dados como um container que armazena informações.



As tabelas armazenam dados seguindo um padrão de colunas e linhas.

não existem

## Perguntas Ídiotas

P: Onde os dados do banco são realmente armazenados? É possível ver os arquivos?

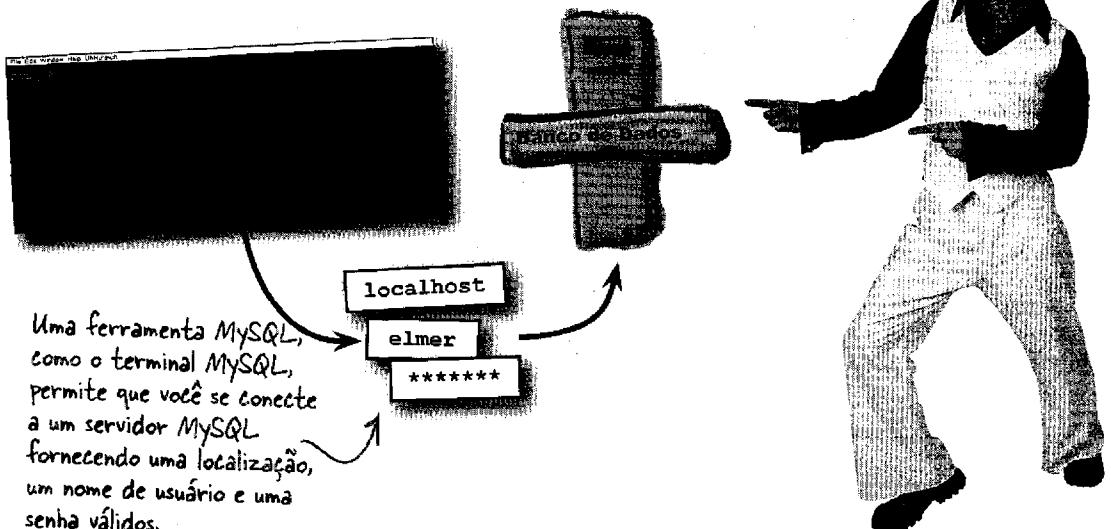
R: Os dados de um banco geralmente são armazenados em um disco rígido. E, embora você certamente possa vê-los, eles não lhe diriam muita coisa. Os arquivos de bancos de dados são binários, e não podem simplesmente ser abertos e ter suas informações visualizadas. É por isso que temos SQL – para permitir que nós olhemos o conteúdo do banco e interajamos com os dados armazenados dentro dele.

**CREIE** seu banco de dados

## Faça contato com o servidor MySQL

O projeto da aplicação de Elmer exige um banco de dados e uma tabela. A maior parte do trabalho cotidiano com um banco de dados envolve a interação com tabelas, mas você não pode simplesmente sair criando tabelas sem criar, primeiro, um banco onde elas fiquem armazenadas.

`CREATE DATABASE` é o comando SQL usado para se criar um banco de dados. Uma vez feito isso, você pode então criar uma tabela com o comando `CREATE TABLE`. Mas antes de usar qualquer um desses comandos, você tem de se conectar ao seu servidor de bancos de dados MySQL. Você fez isso no capítulo anterior e precisou de algumas informações importantes.



Além de permitirem que um script PHP se conecte ao banco de dados e realize ações nele, a localização, o nome de usuário e a senha do banco são a chave para que o terminal MySQL ou o phpMyAdmin sejam utilizados. E essas ferramentas são bastante úteis para que realizemos os primeiros passos da aplicação: a criação do banco e da tabela.

Já que a criação de um banco de dados e de uma tabela para a aplicação de Elmer é uma tarefa que só precisa ser realizada uma vez, faz sentido usar uma consulta SQL para criá-los manualmente. Portanto, inicie sua ferramenta MySQL preferida e prepare-se para realizar o primeiro passo na elaboração da aplicação de Elmer, criando um banco de dados e uma tabela para a lista de emails.

- Você está aqui.*
- 1 Crie um banco de dados e uma tabela para a lista de emails.
  - 2 Crie um formulário web e um script PHP "Adicionar Email" para adicionar novos clientes à lista.
  - 3 Crie um formulário web e um script PHP "Enviar Email" para enviar um email para a lista.

## Crie um banco de dados para os emails de Elmer

Para criar uma nova tabela e um novo banco para a lista de emails de Elmer, primeiro nós temos de criar o banco de dados `elvis_store`, o qual irá armazenar a tabela `email_list`. Nós usaremos comandos SQL para criar ambos. O comando SQL usado para criar um banco de dados é `CREATE DATABASE`, que você usou, rapidamente, no capítulo anterior. Vamos dar uma olhada mais detalhada no seu funcionamento.

`CREATE DATABASE nome_do_banco`

O nome do novo banco de dados à ser criado

É preciso especificar o nome do novo banco de dados após o comando `CREATE DATABASE`. Eis a instrução SQL para criar o banco de dados de Elmer:

`CREATE DATABASE elvis_store`

`CREATE DATABASE` é o comando SQL usado para se criar um novo banco de dados.

Quando você executar esta instrução em um servidor MySQL, o banco será criado.

Ao executar comandos SQL no terminal, você sempre adiciona um ponto-e-vírgula ao final... mas não quando fizer consultas SQL através da função PHP `mysqli_query()`.

```
File Edit Window Help Don't Be Cruel
mysql> CREATE DATABASE elvis_store;
Query OK, 1 row affected (0.01 sec)
```

`elvis_store`

Criar o banco de dados `elvis_store` com o comando `CREATE DATABASE` resulta em um banco de dados novinho em folha, mas ainda não há uma tabela para armazenar os dados...



O banco de dados foi criado, mas ele não pode armazenar dados se não houver uma tabela.



Veja bem!

**As instruções SQL só terminam em ponto-e-vírgula quando você usa o terminal.**

No seu código PHP, as suas instruções SQL não precisam terminar em ponto-e-vírgula. O terminal MySQL é diferente, porém, e requer o ponto-e-vírgula ao final de cada instrução SQL. Isso ocorre porque o terminal é capaz de rodar múltiplas instruções SQL, enquanto que, no PHP, você só pode submeter uma instrução por vez.

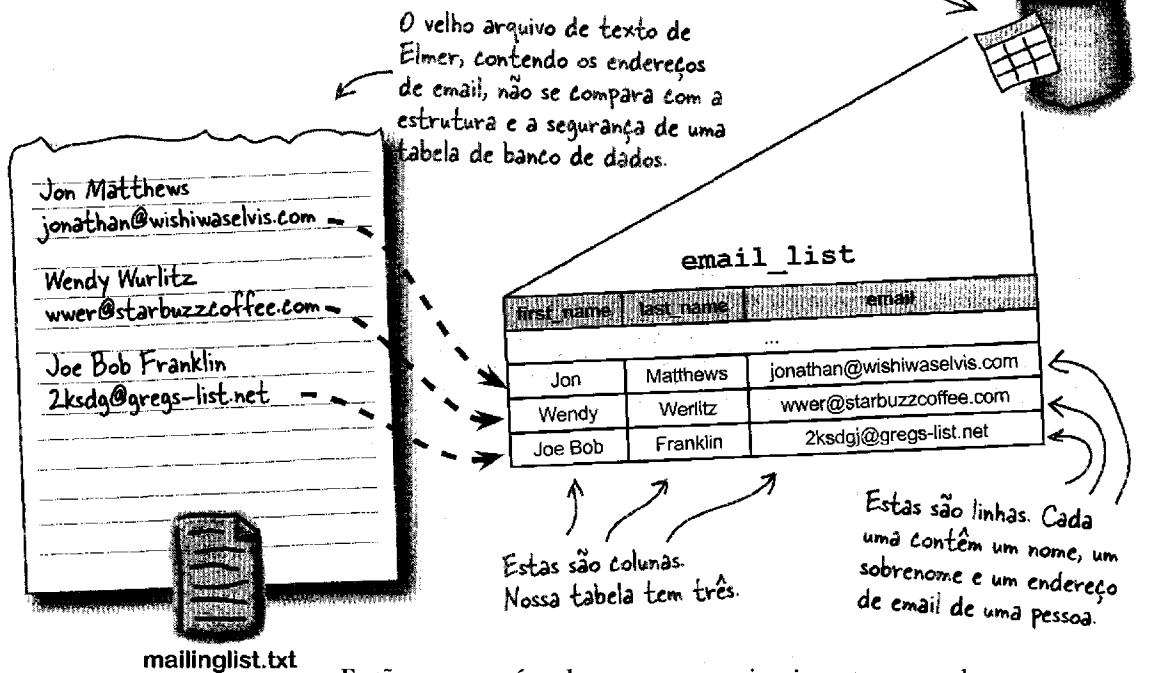
agora CRIE sua tabela

## Crie uma tabela dentro do banco de dados

Você precisa saber qual tipo de dados deseja armazenar na tabela, antes de criá-la. Elmer deseja usar o nome e o sobrenome das pessoas da sua lista de email, para tornar um pouco mais pessoais as mensagens que irá enviar. Adicionando a essas informações o endereço de email, a tabela `email_list` de Elmer terá de armazenar três informações para cada entrada.

Cada informação de uma tabela fica em uma coluna, que precisa ter um nome que descreva os dados. Vamos usar `first_name` ("primeiro nome"), `last_name` ("sobrenome") e `email` como o nome das nossas colunas. Cada linha da tabela consiste de uma única informação para cada uma dessas colunas, e constitui uma única entrada na lista de emails de Elmer.

`email_list` é uma das muitas tabelas que podem ser armazenadas no banco de dados `elvis_store`.



As linhas da tabela  
são horizontais  
e as colunas são  
verticais.

Então, agora nós sabemos que o primeiro nome, o sobrenome e o endereço de email do cliente precisam ser criados como colunas na tabela `email_list`. O problema é que as tabelas do MySQL são altamente estruturadas e esperam que você forneça mais do que apenas o nome da coluna. Você precisa informar ao banco de dados um pouco mais sobre que tipo de dados você pretende armazenar na coluna.



## Precisamos definir os nossos dados

Quando você cria uma tabela, precisa informar ao servidor MySQL qual o tipo de dados que cada coluna terá. A definição do tipo de dados é exigida para todas as colunas MySQL, e cada coluna de uma tabela armazena um tipo particular de dados. Isso significa que algumas colunas poderão conter texto, algumas poderão conter valores numéricos, algumas poderão conter horas ou datas, e assim por diante. O MySQL possui uma série de tipos de dados, e você precisa saber qual tipo corresponde aos seus dados específicos. Vamos supor que Elmer tenha uma tabela chamada `products` ("produtos"), que registra os itens à venda na sua loja:

Esta coluna contém descrições textuais de cada produto da loja de Elmer.

products			
id	product	inventory	price
1	Blue Suede Shoes	24	59.00
2	Polyester Pants with Sequins	16	23.50
3	Stick-On Sideburns	93	1.99
4	Elvis wig	7	48.00
...			

A coluna `ID` contém valores de identificação únicos para cada produto da loja.

A coluna `inventory` ("inventário") contém um valor que é um número inteiro informando quantas unidades de cada produto há no estoque.

Número Inteiro

product
Blue Suede Shoes
Polyester Pants with Sequins
Stick-On Sideburns
Elvis wig

Texto

inventory
24
16
93
7

Número Inteiro

Repare que `product` é a única coluna de texto na tabela `products`. Há também números decimais para `price`, e números inteiros para `inventory` e `id`. O MySQL tem os seus próprios nomes para cada um desses tipos de dados, bem como alguns outros, como por exemplo para datas e horas.

É importante usar os tipos de dados apropriados ao se criar colunas, para que as suas tabelas sejam precisas e eficientes. Por exemplo, **dados do tipo texto ocupam mais espaço do que números inteiros**; assim, se uma coluna só precisa armazenar números inteiros, é uma prática inteligente usar o tipo de dados inteiro para ela. Além disso, se o servidor web souber qual tipo de dados uma determinada coluna armazena, ele não permitirá que você, acidentalmente, insira o tipo de dado errado. Assim, em uma coluna definida para armazenar datas, você receberá um erro se tentar inserir qualquer coisa que não seja uma data, nessa coluna.

Para criar uma tabela, você precisa saber qual tipo de dados deve ser armazenado em cada coluna da tabela.

price
59.00
23.50
1.99
48.00

Número Decimal

### PODER DO CÉREBRO

Por que você acha que usar diferentes tipos de dados é melhor do que usar apenas texto, para armazenar tudo?

## alguns tipos de dados MySQL

Estes são alguns dos tipos de dados mais úteis do MySQL. Lembre-se, você pode usar qualquer um deles para descrever os dados armazenados dentro de uma determinada coluna da tabela. A função desses tipos é armazenar os seus dados sem criar confusões.

**CHAR ou CHARACTER.** Ela é rígida e prefere que os seus dados tenham um tamanho definido. Ela pode ser altamente eficiente se você tiver algum texto que tenha sempre a mesma extensão.

**INT ou INTEGER** acha que os números devem ser inteiros, mas não tem medo de números negativos. Ele é capaz, também, de armazenar números inteiros pequenos, cujo caso é chamado de **TINYINT**.

**DEC**, abreviação de **DECIMAL**. Ele lhe fornecerá todas as casas decimais que você precisar, pelo menos até se encher.

Ele se chama **BLOB**. Ele gosta de grandes quantidades de dados binários.

Ela atende tanto por **DATETIME** quanto por **TIMESTAMP**. Ela mantém o registro da data e da hora.

Uma boa amiga de **BLOB**, esta é **TEXT**, ela é excelente para armazenar grandes quantidades de texto – muito mais do que **CHAR** ou **VARCHAR**.

Este é **VARCHAR**, abreviação de **VARIABLE CHARacter** ("caráter variável"). Ele armazena dados em modo texto. É flexível e capaz de se adaptar à extensão dos seus dados, armazenando apenas o que você precisa, sem preencher lacunas com espaços em branco.

**DATE** mantém registro das suas datas. Ela não se interessa pelas horas, porém. Ela tem uma irmã gêmea, **TIME**, que, ao contrário dela, não se importa com as datas.

Dependendo da sua versão do MySQL, a extensão pode ser de 255 caracteres antes da versão 5.0.3, e até 65.535 caracteres na versão 5.0.3 e acima.

não existem

## Perguntas Ídiotas

**P:** Por que eu usaria CHAR, se VARCHAR faz a mesma coisa, com mais flexibilidade?

**R:** A resposta está na precisão e na eficiência. Do ponto de vista do projeto, você deve sempre projetar suas tabelas de forma a modelar os seus dados o mais rigidamente possível. Se você souber, sem sombra de dúvida, que uma coluna para armazenar a sigla dos estados irá sempre armazenar exatamente uma abreviatura composta de duas letras, então faz sentido permitir apenas dois caracteres de espaço para ela, com CHAR (2). Entretanto, se por exemplo uma coluna para senhas puder conter até 10 caracteres, então VARCHAR (10) faz mais sentido. Esse é o ponto de vista do projeto. Assim, CHAR é um pouco mais eficiente do que VARCHAR, porque ele não precisa se preocupar com a variação de tamanho dos dados. Assim, é mais desejável usar quando você sabe, com certeza, que uma coluna de texto terá uma extensão fixa.

**P:** Por que eu preciso desses tipos numéricos, como INT e DEC?

**R:** Trata-se do armazenamento e eficiência do banco de dados. Escolher o melhor tipo de dados para cada coluna da sua tabela reduz o tamanho desta, e torna mais rápidas as operações feitas nos seus dados. Armazenar um número como número propriamente (INT, DEC, etc.), em vez de caracteres de texto, em geral é mais eficiente.

**P:** É só isso? Esses são todos os tipos?

**R:** Não, mas estes são os mais comumente usados. Nós praticaremos com estes por agora, em vez de complicar as coisas incluindo tipos de dados que, talvez, você nunca vá precisar.

## QUAL É MEU PROPÓSITO

Ligue cada tipo de dados MySQL à descrição dos dados que você poderia armazenar em uma tabela.

### Tipos de Dados

INT

CHAR (1)

DATE

TIME

VARCHAR (2)

DEC (4, 2)

VARCHAR (60)

CHAR (2)

DATETIME

DEC (10, 2)

### Descrição

Seu nome completo.

Uma abreviatura de estado, com duas letras.

Preço de uma peruca do Elvis: 48.99

Quanto dinheiro rendeu o disco mais vendido de Elvis.

Data da abdução alienígena: 2/12/2004

Número de costeletas do Elvis em estoque: 93

Você viu o cachorro de Owen? S ou N

Seu endereço de email

A que horas você janta

Quantos aliens você viu quando foi abduzido

Quando foi que Elvis nasceu

## QUAL É MEU PROPÓSITO

Ligue cada tipo de dados MySQL à descrição dos dados que você poderia armazenar em uma tabela.

Não é necessário. Embora funcione para a abreviatura de estado, CHAR(2) é uma opção melhor, porque geralmente é um pouco mais eficiente.

Quando o tamanho de um valor de texto puder variar, VARCHAR é uma boa opção. Faça-o longo o suficiente para armazenar qualquer valor que alguém precise armazenar.

### Tipos de Dados

TIPO DE DADO	DESCRIÇÃO
INT	Seu nome completo.
CHAR(1)	Uma abreviatura de estado, com duas letras.
DATE	Preço de uma peruca do Elvis: 48.99
TIME	Quanto dinheiro rendeu o disco mais vendido de Elvis.
VARCHAR(2)	Data da abdução alienígena: 2/12/2004
DEC(4, 2)	Número de costeletas do Elvis em estoque: 93
VARCHAR(60)	Você viu o cachorro de Owen? S ou N
CHAR(2)	Seu endereço de email
DATETIME	A que horas você janta
DEC(10, 2)	Quantos aliens você viu quando foi abduzido
	Quando foi que Elvis nasceu

DEC  
geralmente é  
usado para  
armazenar  
preços, além de  
outros valores  
decimais.

Estes dois  
números mostram  
quants dígitos o  
banco de dados  
deve esperar  
na frente  
do decimal, e  
quants depois.

Você pode ter  
respondido DATE aqui,  
mas os Elvismaniacos  
verdadeiros sabem a  
data e a hora exatas.

Podem haver outras (e  
potencialmente melhores)  
formas de se representar  
um valor sim/não no MySQL  
do que usando CHAR(1),  
mas esta forma é simples e,  
razoavelmente, eficiente.

# Crie sua tabela com uma consulta

Já temos todos os pedaços de que precisamos para criar nossa tabela, inclusive um bom nome (`email_list`). Também temos nomes para as colunas de dados: `first_name`, `last_name` e `email`. Só o que falta são os tipos de dados para cada coluna, bem como uma instrução SQL para juntar tudo isso e criar a tabela. O comando SQL para criar sua tabela é `CREATE TABLE`.

Ele começa com `CREATE TABLE` seguido do nome da sua tabela. Os nomes de todas as colunas vem entre parênteses, separados por vírgula e seguidos, cada um, de um tipo de dados. Eis como seria o comando:

```
CREATE TABLE nome_da_tabela
(
    nome_da_coluna1 tipo_da_coluna1,
    nome_da_coluna2 tipo_da_coluna2,
    ...
)
```

O nome da tabela  
O nome da coluna  
O tipo de dados da coluna  
Mais colunas, se necessário

Você não precisa nomear suas tabelas e colunas com um underscore separando as palavras, mas é uma boa ideia se manter consistente em relação a qualquer convenção de nomeação que decidir usar.

Sim, ainda estamos aqui... mas estamos quase prontos para seguir adiante.

- 1 Crie um banco de dados e uma tabela para a lista de emails.
- 2 Crie um formulário web e um script PHP "Adicionar Email" para adicionar novos clientes à lista.
- 3 Crie um formulário web e um script PHP "Enviar Email" para enviar um email para a lista.

O comando SQL `CREATE TABLE` é usado para que seja criada uma nova tabela em um banco de dados.



Escreva uma query SQL para criar a tabela `email_list` de Elmer com as três colunas de dados requeridas: `first_name`, `last_name` e `email`.

```
CREATE TABLE email_list(
    first_name VARCHAR(20),
    last_name VARCHAR(20),
    email VARCHAR(60));
```

test-drive suas queries CREATE

## Aponte seu lápis Solução

Escreva uma query SQL para criar a tabela email\_list de Elmer com as três colunas de dados requeridas: first\_name, last\_name e email.

Este é o comando SQL para criar a tabela - repare nas letras maiúsculas.

O parêntese de abertura abre a lista de colunas a serem criadas.

CREATE TABLE email\_list

O nome da sua tabela deve ficar em caixa baixa e deve ter um underscore no lugar de quaisquer espaços.

first\_name VARCHAR(20),

A vírgula separa as colunas que serão criadas.

last\_name VARCHAR(20),

email VARCHAR(60)

Isto diz ao MySQL que a coluna email tem o tipo de dados VARCHAR. O (60) significa que o texto armazenado pode ter até 60 caracteres.

O parêntese final fecha a lista das colunas.



## TEST DRIVE

### Crie o banco de dados e a tabela de Elmer.

Execute as queries CREATE DATABASE e CREATE TABLE usando uma ferramenta MySQL para criar o banco de dados elvis\_store e a tabela email\_list dentro dele.

```
CREATE DATABASE elvis_store
```

```
CREATE TABLE email_list(first_name VARCHAR(20), last_name VARCHAR(20), email VARCHAR(60))
```

Ambas as queries foram executadas sem problemas? Caso contrário, escreva aqui o que você acha que pode ter dado errado.

*Resposta: erro de nome de tabela.*

Espere aí, tem algo de errado aqui. Eu digitei o código para criar a tabela exatamente da forma como estava escrito... e agora estou recebendo uma mensagem de erro esquisita.

A instrução CREATE TABLE está OK, mas o terminal MySQL não sabe em que banco de dados a tabela está sendo criada... isso não é bom



```

File Edit Window Help Oops
mysql> CREATE TABLE email_list
(
    first_name VARCHAR(20),
    last_name VARCHAR(20),
    email VARCHAR(60)
);
ERROR 1046 (3D000): No database selected

```

Por algum motivo, a instrução CREATE TABLE falhou no terminal MySQL.

## a tabela do banco de dados Colocando o carro na frente dos bois

Elmer tem um problema que tem a ver com o fato de que o terminal MySQL não sabe, automaticamente, a qual banco você está se referindo ao emitir comandos. É claro que ele sabe que você acabou de criar o banco de dados elvis\_store, mas poderia haver diversos outros bancos de dados armazenados no mesmo servidor – ele não pode simplesmente assumir que você está se referindo àquele que acabou de criar.

Felizmente, existe uma solução simples, que envolve informar ao terminal MySQL qual banco de dados você quer que seja o alvo de todas as instruções subsequentes...

não existem

### Perguntas Ídiotas

**P:** Qual é a desse prompt esquisito -> que eu às vezes vejo no terminal MySQL?

**R:** O prompt -> indica que você está digitando uma mesma instrução ao longo de várias linhas – o MySQL está basicamente lhe dizendo que sabe que você ainda está digitando a mesma instrução, mesmo que você tenha pressionado Enter para separá-la em mais de uma linha. Uma vez que você termine a instrução e coloque o ponto-e-vírgula no final, o MySQL irá executá-la.

*não esqueça do comando USE!*

## SELECIONE o banco de dados antes de usá-lo

Para que a instrução CREATE TABLE funcione, Elmer precisa selecionar o banco de dados no terminal MySQL, para que este saiba a qual banco de dados pertence a nova tabela. O comando USE escolhe um banco de dados como sendo o padrão no terminal, o que significa que todos os comandos subsequentes irão se aplicar a esse banco de dados.

Funciona desta maneira:

O comando USE  
seleciona um banco  
de dados como  
o padrão para  
instruções SQL  
subsequentes.

**USE nome\_do\_banco**

O comando USE informa ao MySQL qual banco de dados você pretende usar.

Elmer deve especificar o nome do seu banco de dados (`elvis_store`) em uma instrução USE para selecioná-lo e acessar a sua nova tabela.

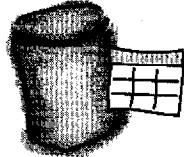
O comando USE escolhe o banco de dados com o qual você deseja trabalhar.

**USE elvis\_store**

Nome do banco de dados que você quer selecionar com USE.



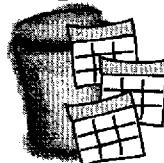
`elvis_sightings`



`elvis_lyrics`



`elvis_fans`



Uma vez que você tenha escolhido um banco de dados para usar, os demais bancos de dados do servidor são ignorados... até que você escolha selecionar um outro banco de dados.



## TEST DRIVE

### Execute USE primeiro para depois criar a tabela.

Execute a consulta USE em uma ferramenta MySQL, para selecionar o banco de dados elvis\_store de Elmer e depois execute a query CREATE TABLE para criar a tabela email\_list dentro do banco.

```
USE elvis_store
```

```
CREATE TABLE email_list(first_name VARCHAR(20), last_name VARCHAR(20), email VARCHAR(60))
```

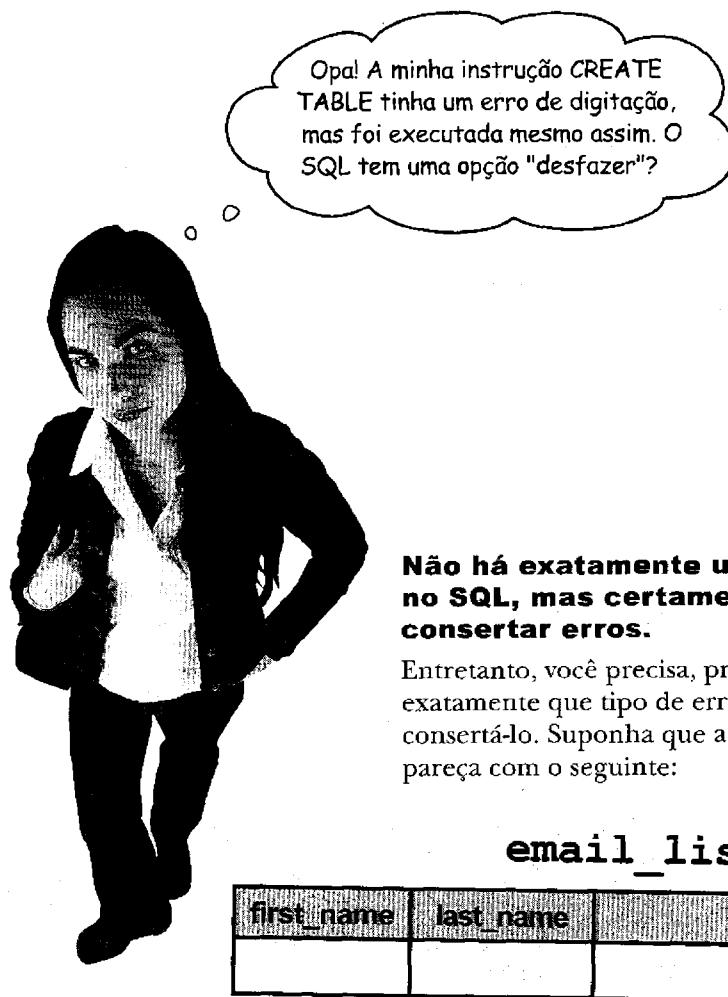
A instrução USE não é necessária caso você esteja usando uma ferramenta SQL gráfica, como o phpMyAdmin – nesse caso, você irá selecionar o banco gráficamente, antes de emitir os comandos SQL.

```
File Edit Window Help LisaMarie
mysql> USE elvis_store;
Database changed
mysql> CREATE TABLE email_list
(
    first_name VARCHAR(20),
    last_name VARCHAR(20),
    email VARCHAR(60)
);
Your SQL query has been executed successfully (Query took 0.4481 sec)
```

O código para a criação da tabela é o mesmo de antes – só era preciso selecionar o banco para que funcionasse.

Com o banco de dados selecionado através do comando USE, a criação da tabela, agora, ocorre sem problemas.

**o comando DESCRIBE**



Opa! A minha instrução CREATE TABLE tinha um erro de digitação, mas foi executada mesmo assim. O SQL tem uma opção "desfazer"?

**Não há exatamente uma opção desfazer no SQL, mas certamente é possível consertar erros.**

Entretanto, você precisa, primeiramente, descobrir exatamente que tipo de erro foi cometido, para consertá-lo. Suponha que a tabela email\_liste se pareça com o seguinte:

**email\_list**

first_name	last_name	email

Circule o que você acha que há de errado com esta tabela.  
Alguma ideia sobre como consertá-la?



# DESCRIBE revela a estrutura das tabelas

Para consertar um erro em uma tabela, é preciso, primeiro, achar o erro. Mesmo que você não suspeite que haja um, nunca é uma má ideia verificar o seu trabalho. O comando SQL DESCRIBE analisa a estrutura de uma tabela, exibindo uma lista de nomes de colunas, tipos de dados e outras informações.

**DESCRIBE nome\_da\_tabela**

Usando esse comando na tabela de Elmer, teremos a seguinte instrução SQL:

**DESCRIBE email\_list** *Este é o nome da tabela cuja descrição queremos ver.*

Em "Field" ("Campo") você encontra os nomes de cada coluna.

Em "Type" ("Tipo") você vê os tipos de dados que definimos para cada coluna.

O MySQL não faz distinção entre maiúsculas e minúsculas no que se refere às palavras reservadas, como os tipos de dados; por isso, você poderá, às vezes, vê-las escritos em minúsculas.

não existem

## Perguntas Idiotas

**P:** O que são aquelas outras colunas: Null (Nulo), Key (Chave), Default (Padrão) e Extra?

**R:** O MySQL lhe permite definir uma série de opções para cada coluna da sua tabela. Essas opções controlam opções como, por exemplo, se uma coluna pode ser deixada em branco ou se ela terá um valor padrão. Nós vamos explorar isso um pouco mais adiante, quando essas opções se tornarem mais vitais para a aplicação.

**P:** Se eu já tivesse dados armazenados na minha tabela, eles teriam aparecido aqui?

**R:** Não. DESCRIBE só lhe mostra a estrutura da tabela, e não os dados armazenados nela. Mas não se preocupe, você logo verá os dados da sua tabela... mas primeiro temos de aprender como inserir esses dados nela.

**P:** Eu posso visualizar a mesma estrutura usando o phpMyAdmin?

**R:** Sim. Ferramentas gráficas como o phpMyAdmin lhe permitem visualizar a estrutura das tabelas através de uma instrução DESCRIBE ou clicando em uma representação visual da tabela. Você é livre para escolher que tipo de ferramenta deseja usar para analisar suas tabelas.

## APAGUE sua tabela

Eu consertei o erro de digitação e tentei rodar a consulta CREATE TABLE novamente. Não funcionou. É claro que eu não preciso apagar primeiro a tabela que contém o erro... ou preciso?

O nome da coluna first\_name foi acidentalmente digitado como first\_naem... ops!

```
File Edit Window Help Type?
mysql> DESCRIBE email_list;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| first_naem | varchar(30) | YES |   | NULL    |       |
| last_name  | varchar(30) | YES |   | NULL    |       |
| email      | varchar(60)  | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```



**Na verdade, precisa sim. Não é possível recriar uma tabela com CREATE TABLE uma vez que ela já tenha sido criada.**

Uma vez que você tenha criado uma tabela, ela não pode ser sobreescrita através de uma nova consulta CREATE TABLE. Se quiser recriar a tabela do zero, você terá de apagar primeiro a existente, e começar tudo de novo.

No SQL, o comando DROP TABLE é usado para que se possa apagar uma tabela de um banco de dados. Ele apaga a tabela e tudo que você tiver armazenado nela. Uma vez que não existem dados em uma tabela recém-criada, não iremos perder nada ao apagá-la e criar uma nova, com first\_name escrito corretamente.

DROP TABLE email\_list

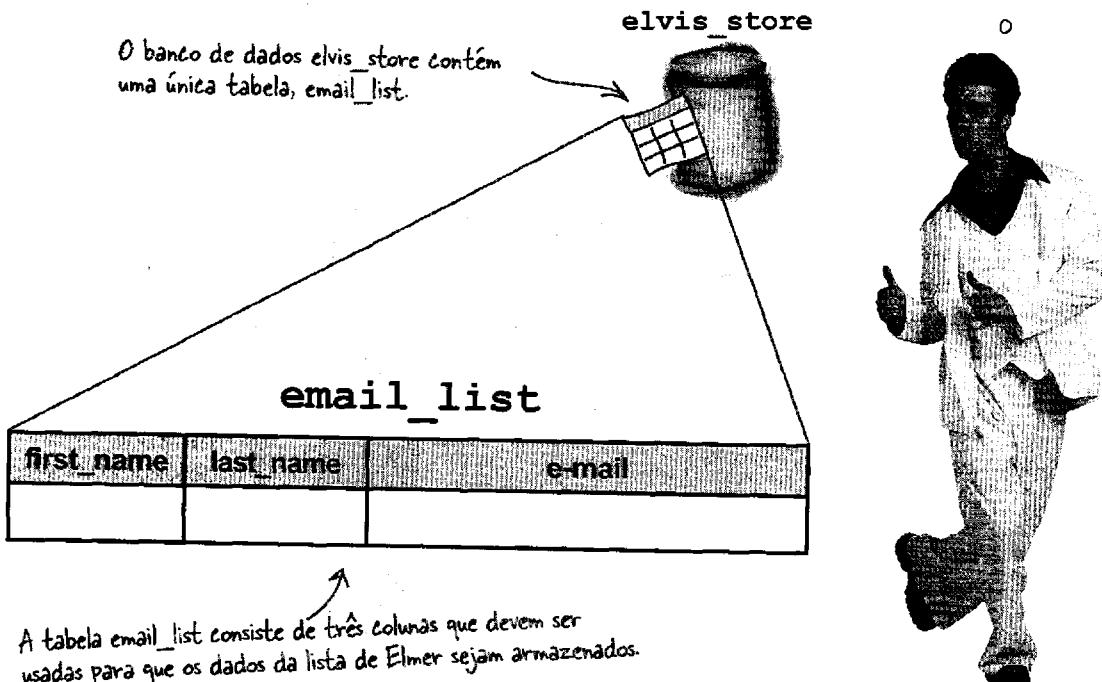
Nome da tabela que você gostaria de apagar do banco de dados.

O comando DROP TABLE apaga a tabela e todos os dados contidos nela.

## Elmer está pronto para armazenar dados

Os comandos SQL CREATE DATABASE, USE e CREATE TABLE foram usados com sucesso para criar o banco de dados e a tabela de Elmer. Ele não poderia estar mais satisfeito, a não ser que a tabela já viesse preenchida com os dados dos clientes. Esse é um trabalho para o PHP...

Legal. Com o banco de dados e a tabela criados, estou pronto para começar a armazenar alguns dados da minha lista de emails.



### não existem Perguntas Ídiotas

**P:** El, eu tenho uma cópia de Use a Cabeça! SQL (ótimo livro, a propósito). Naquele livro, toda vez que você mostrava o código de uma instrução SQL, colocava um ponto-e-vírgula ao final dela. Por que aqui é diferente?

**R:** Ficamos felizes de você ter gostado de Use a Cabeça! SQL. A diferença é que, quando você fala diretamente ao MySQL, é preciso colocar o ponto-e-vírgula para que ele saiba onde a instrução termina. Isso porque é possível emitir múltiplas instruções ao MySQL, diretamente. No PHP, quando é usada função mysqli\_query(), você só executa um comando SQL por vez, de modo que não é preciso usar o ponto-e-vírgula. Mas não se esqueça de que você ainda precisa colocar o ponto-e-vírgula ao final de cada instrução PHP!

**P:** Se a minha tabela tiver dados e eu a apagar, todos os dados serão apagados, também?

**R:** Sim, é verdade. Portanto, cuidado ao apagar tabelas!

**P:** Então, se eu precisar modificar uma tabela que já tenha dados, estou ferrado?

**R:** Ei, ninguém é perfeito. Todo mundo comete erros, e o SQL oferece a instrução ALTER para nos ajudar a modificar tabelas existentes. Falaremos desse comando mais adiante, neste livro.

## *o script addemail.php*

# Crie o script "Adicionar Email"

Elmer precisa de um formulário HTML para coletar nomes e endereços de email dos clientes. Uma vez que ele tenha essas informações, precisa filtrá-las com um script PHP e armazená-las na tabela email\_list. O formulário web (addemail.html) requer três campos de input (entrada) e um botão. A ação é a parte mais importante do código do formulário, uma vez que o seu trabalho é repassar os dados para o script addemail.php, que estamos prestes a criar.

Você agora está aqui.

- 1 Crie um banco de dados e uma tabela para a lista de emails.
- 2 Crie um formulário web e um script PHP "Adicionar Email" para adicionar novos clientes à lista.
- 3 Crie um formulário web e um script PHP "Enviar Email" para enviar um email para a lista.

A ação é o elemento que conecta o formulário HTML ao script PHP (addemail.php) que processará os dados.

MAKEELVIS.COM

Enter your first name, last name, and email to be added to the Make Me Elvis mailing list.

First name:	<input id="firstname" name="firstname" type="text"/>
Last name:	<input id="lastname" name="lastname" type="text"/>
Email:	<input id="email" name="email" type="text"/>
<input type="submit" value="Submit"/>	

```
<form method="post" action="addemail.php">
  <label for="firstname">First name:</label>
  <input type="text" id="firstname" name="firstname" /><br />
  <label for="lastname">Last name:</label>
  <input type="text" id="lastname" name="lastname" /><br />
  <label for="email">Email:</label>
  <input type="text" id="email" name="email" /><br />
  <input type="submit" name="submit" value="Submit" />
</form>
</body>
</html>
```

O script addemail.php é executado quando o formulário é submetido, e o seu trabalho é processar os dados e adicionar o cliente à lista de emails (na tabela do banco de dados).

addeemail.html

addeemail.php

Os novos clientes podem se cadastrar na lista de email de Elmer (ou seja, serem adicionados ao banco de dados) simplesmente usando o formulário web.

elvis\_store

email\_list

first name	last name	email



O script addemail.php processa os dados do formulário Add Email ("Adicionar Email"). O script precisa filtrar os dados do formulário, conectar-se ao banco de dados elvis\_store e inserir (INSERT) os dados na tabela email\_list. Ajude Elmer a escrever um exemplo de query SQL para inserir um novo cliente, e depois use essa query para finalizar o código do script PHP.

INSERT

Escreva aqui um exemplo  
de consulta para inserir  
dados na tabela de Elmer.

```
<?php
$dbc = mysqli_connect('localhost', 'user', 'pass',
    'db') or die('Error');
$first_name = $_POST['firstname'];
$last_name = $_POST['lastname'];
$email = $_POST['email'];
$query = "INSERT INTO db (first_name, last_name, email) VALUES ('$first_name', '$last_name',
'$email')";
$result = mysqli_query($dbc, $query);

echo 'cliente adicionado.';

?>
```



addemail.php

**solução do exercício**



**Exercício  
Solução**

O script addremail.php processa os dados do formulário "Adicionar Email". O script precisa filtrar os dados do formulário, conectar-se ao banco de dados elvis\_store e inserir (INSERT) os dados na tabela email\_list. Ajude Elmer a escrever um exemplo de query SQL para inserir um novo cliente, e depois use essa query para finalizar o código do script PHP.

`INSERT INTO email_list (first_name, last_name, email)  
VALUES ('Julian', 'Oates', 'julian@breakneakpizza.com')`

Aqui estão os valores do array \$\_POST que contém as informações submetidas.

A query INSERT de exemplo é reescrita como uma string PHP que depende dos dados do formulário para a inserção.

```
<?php  
$dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking', 'elvis_store')  
or die('Erro ao se conectar com o servidor MySQL server.');//  
  
$firstname = $_POST['firstname'];  
$lastname = $_POST['lastname'];  
$email = $_POST['email'];  
  
$query = "INSERT INTO email_list (first_name, last_name, email)" .  
"VALUES ('$firstname', '$lastname', '$email');"  
  
mysqli_query($dbc, $query)  
or die('Erro ao acessar o banco de dados');//  
  
echo 'cliente adicionado.';  
  
mysqli_close($dbc);
```

?>



Se quiséssemos nos exibir aqui, poderíamos colocar um link de volta para o nosso formulário, com uma tag HTML <a>.



## TEST DRIVE

### Teste o formulário de “Adicionar Email”.

Baixe o código da página web “Adicionar Email” no site da Alta Books, em [www.altabooks.com.br](http://www.altabooks.com.br). O código está na pasta **capítulo03**. Ele consiste do formulário web **addemail.html**, uma folha de estilo (**style.css**) e duas imagens (**elvislogo.gif** e **blankface.jpg**).

Em seguida, crie um arquivo de texto chamado **addemail.php** e digite nele todo o código da página anterior. Este é o script que irá processar os dados do formulário web de Elmer e adicionar novos clientes à tabela **email\_list**.

Envie todos esses arquivos ao seu servidor web e abra a página **addemail.html** em um navegador. Digite os dados de um novo cliente no formulário e clique em Submit (Submeter).

Não se esqueça de trocar as variáveis de conexão ao banco de dados pelas suas próprias.

A inserção do novo cliente à lista de email é confirmada pelo script **addemail.php**.

Verifique se o cliente foi adicionado ao banco de dados emitindo uma consulta SELECT em alguma ferramenta MySQL.

```

File Edit Window Help BlueSuedeShoes
mysql> SELECT * FROM email_list;
+-----+-----+-----+
| customer_id | first_name | last_name | email           |
+-----+-----+-----+
|          1 | Julian     | Oates      | julian@breakneckpizza.co |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> EXIT;

```



## Aponte seu lápis Solução

Com a lista de Elmer começando a ser preenchida, ajude-o a escrever algumas consultas SQL que ele possa usar para encontrar dados específicos dos clientes.

Selecionar todos os dados de clientes que tenham o primeiro nome Martin:

```
SELECT * FROM email_list WHERE first_name = 'Martin'
```

A estrela seleciona todas as colunas da tabela.

Esta cláusula WHERE limita os resultados da consulta aos clientes cujo primeiro nome seja Martin.

Selecionar apenas o sobrenome de clientes que tenham o primeiro nome Bubba:

```
SELECT last_name FROM email_list WHERE first_name = 'Bubba'
```

Somente a coluna last\_name é retornada nos resultados da consulta.

Selecionar o nome e o sobrenome do cliente que tenha o endereço de email ls@objectville.net:

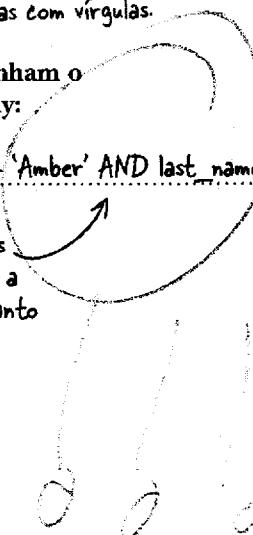
```
SELECT first_name, last_name FROM email_list WHERE email = 'ls@objectville.net'
```

Para especificar múltiplas colunas de resultados, você separa os nomes das colunas com vírgulas.

Selecionar todas as colunas dos clientes que tenham o primeiro nome Amber e o sobrenome McCarthy:

```
SELECT * FROM email_list WHERE first_name = 'Amber' AND last_name = 'McCarthy'
```

A cláusula WHERE pode ser realizada, mas depende de várias informações, neste caso a ocorrência tanto de um primeiro nome quanto de um sobrenome.



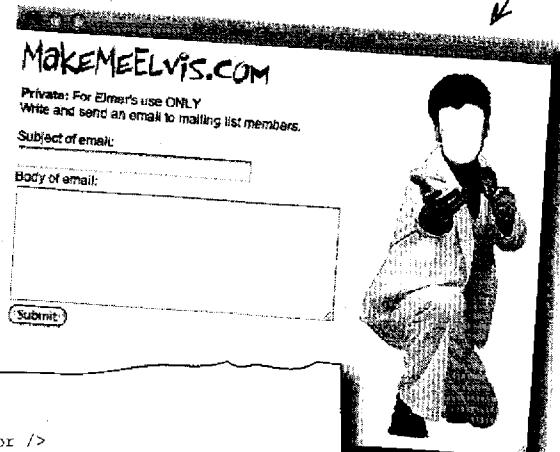
# O outro lado da aplicação de Elmer

Enviar mensagens de email para as pessoas na lista de Elmer é uma tarefa semelhante à de adicionar as pessoas à lista, porque ambas envolvem um formulário HTML e um script PHP. A grande diferença é que, para enviar uma mensagem de email para a lista, é preciso lidar com o conteúdo inteiro da tabela `email_list`, enquanto que o script `addemail.php` lida apenas com uma linha de dados.

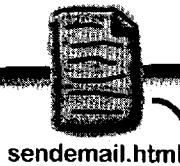
O formulário web Enviar Email permite a Elmer digitar o assunto e o corpo de uma mensagem de email, e depois enviá-los para toda a lista.

- 1 Crie um banco de dados e uma tabela para a lista de emails.
- 2 Crie um formulário web e um script PHP "Adicionar Email" para adicionar novos clientes à lista.
- 3 Crie um formulário web e um script PHP "Enviar Email" para enviar um email à lista.

Ufa, finalmente chegamos ao último Passo.

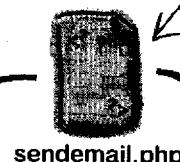
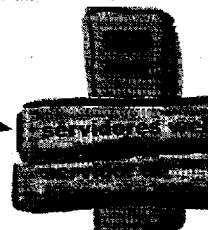


```
...<form method="post" action="sendemail.php">
<label for="subject">Subject of email:</label><br />
<input type="text" id="subject" name="subject" size="60" /><br />
<label for="elvismail">Body of email:</label><br />
<textarea id="elvismail" name="elvismail" rows="8" cols="60"></textarea><br />
<input type="submit" name="submit" value="Submit" />
</form>
</body>
</html>
```



A ação form dispara o script `sendemail.php`.

O script `sendemail.php` lê as informações dos clientes na tabela, e envia o email de Elmer para cada um deles.



`elvis_store`

Nome	last name	e-mail
Julian	Oates	julian@breakneckpizza.com
Kevin	Jones	jones@simuduck.com
Amanda	Sanchez	sunshine@breakneckpizza.com

## O funcionamento interno do script "Enviar Email"

O script sendemail.php precisa combinar dados de duas fontes diferentes para gerar e enviar mensagens de email. Por um lado, o script precisa retirar os nomes e endereços de email da tabela email\_list do banco de dados elvis\_store. Porém, ele precisa, também, obter o assunto e o corpo da mensagem digitados por Elmer no formulário web "Enviar Email" (sendemail.html). Vamos detalhar os passos da operação.

- 1 Use o array `$_POST` para obter o assunto e o corpo da mensagem de email do formulário.

Nada de novo aqui. Clicar no botão Submit do formulário sendemail.html para enviar os dados para sendemail.php, onde nós os capturamos em variáveis com uma pequena ajuda do array `$_POST`.

- 2 Execute uma consulta SELECT na tabela email\_list.

A função PHP `mysqli_query()` roda uma consulta SELECT para obter os dados da lista de emails. Uma vez que queremos todos os dados da tabela, podemos usar `SELECT *`.

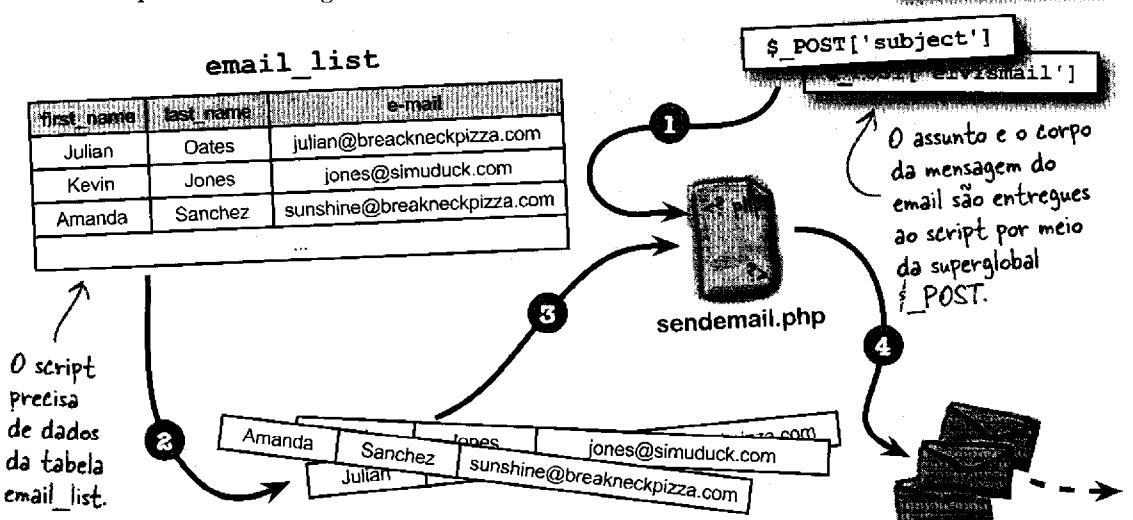
- 3 Acesse os dados do email no resultado da consulta.

Apenas executar uma consulta não fornece acesso aos dados. Nós precisamos acessar cada linha de dados do resultado da consulta, para termos acesso ao nome, sobrenome e endereço de email de cada cliente.

- 4 Recorra à função `mail()` para enviar um email a cada cliente.

Para enviar os emails, é preciso fazer um loop através de cada cliente na lista de emails, o que corresponde a cada linha de dados nos resultados da consulta. O loop que criamos aqui começa na primeira linha de dados, depois passa para a segunda linha, e assim por diante através de todas as linhas de dados obtidos pela consulta SELECT.

Nós paramos ao atingirmos o final dos dados.



## Em primeiro lugar, obtenha os dados

Nós já estamos bem experientes em extrair dados de formulários no PHP, portanto, o primeiro passo não traz nada de novo; basta usar a superglobal `$_POST` para armazenar o assunto e o corpo da mensagem do email em variáveis. E vamos aproveitar também para armazenar o endereço de email de Elmer em uma variável, uma vez que iremos precisar dele posteriormente, quando enviarmos os emails.

```
$from = 'elmer@makemeelvis.com';
```

O endereço de email de Elmer é armazenado em uma variável, para que saibamos exatamente onde ele está, caso precisemos modificá-lo.

```
$subject = $_POST['subject'];
```

Os dados do formulário para a mensagem de email também são armazenados em variáveis.

```
$text = $_POST['elvismail'];
```

Os demais dados requeridos pelo script `sendemail.php` são extraídos do banco de dados MySQL de Elmer. Para obter dados dos clientes a partir da tabela `email_list` e colocá-los no script, precisamos de uma consulta `SELECT`. Ao contrário do que fizemos anteriormente, quando usamos o terminal MySQL para emitir um `SELECT` e olhar os dados da tabela, desta vez iremos fazê-lo no script `sendemail.php`, e iremos emitir a consulta com `mysqli_query()`.

A variável `$query` armazena a consulta SQL na forma de uma string de texto.

Eis a nossa consulta, que seleciona todas as colunas da tabela `email_list`.

```
$query = "SELECT * FROM email_list";
```

```
$result = mysqli_query($dbc, $query);
```

`mysqli_query` executa a consulta usando uma variável de conexão (`$dbc`) e uma string de consulta (`$query`).

É preciso haver uma conexão com o banco de dados para que a consulta possa ser submetida – os detalhes da conexão são armazenados na variável `$dbc`.

Então, tudo o que temos de fazer é usar os resultados da consulta, presentes na variável `$result`, certo?



**Não, a variável `$result`, na verdade, não contém quaisquer dados.**

Se você tentar ecoar a variável `$result` diretamente, verá algo como:

```
Resource id #3
```

A variável `$result` armazena um número de identificação de um **recurso** MySQL, e não os dados, propriamente ditos, que são retornados pela consulta. O que acontece é que o servidor MySQL armazena, temporariamente, os resultados da sua consulta e fornece a eles um número de recurso para identificação. Você então usa essa ID do recurso com a função PHP `mysqli_fetch_array()` para obter os dados, uma linha de cada vez.

`fetch_array()` para obter os resultados da consulta

## `fetch_array()` obtém os resultados da consulta

Uma vez executada a nossa consulta, podemos obter os resultados com a variável `$result`. Essa variável é usada com a função `mysqli_fetch_array()` para se obter os dados da tabela, uma linha por vez. Cada linha de dados é retornada como um array, que podemos armazenar em uma nova variável chamada `$row`.

```
$row = mysqli_fetch_array($result);
```

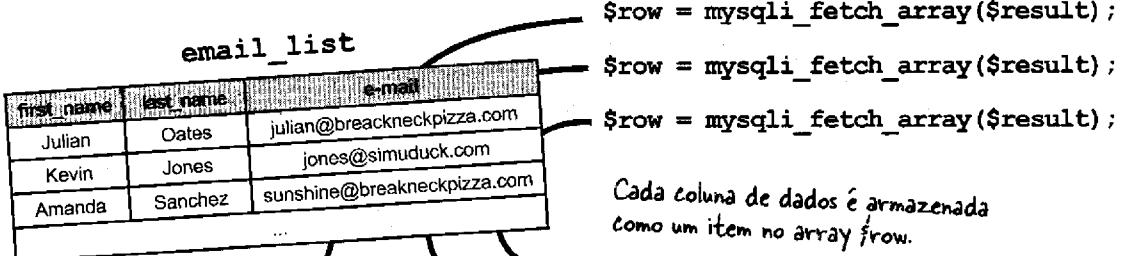
A variável `$row` é um array que, inicialmente, armazena a primeira linha de dados dos nossos resultados.

Esta função obtém uma linha de dados a partir dos resultados da consulta, e a armazena em um array.

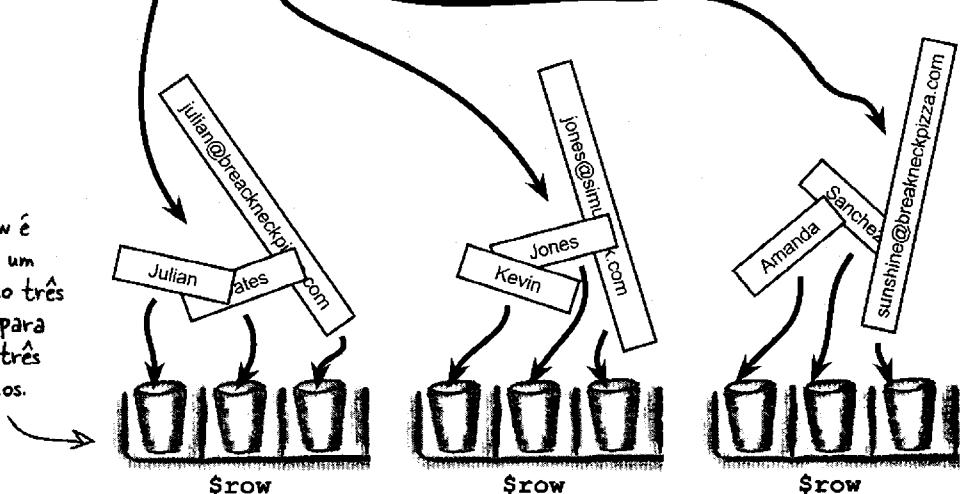
Cada consulta SQL tem o seu próprio número de ID, que é usado para se tenha acesso aos dados, associados aos seus resultados.

Cada vez que este código é executado pelo servidor web, uma linha de dados dos resultados da consulta é armazenada no array `$row`. Repetidamente, você solicita que a função `mysqli_fetch_array()` seja transmitida através de cada linha dos resultados. Assim, as três primeiras chamadas à função `mysqli_fetch_array()` obtém as três primeiras linhas de dados da tabela, armazenando cada coluna da linha como um item no array `$row`.

A função `mysqli_fetch_array()` armazena uma linha de dados em um array.



A variável `$row` é definida como um array contendo três elementos, um para cada uma das três colunas de dados.



## Aponte seu lápis

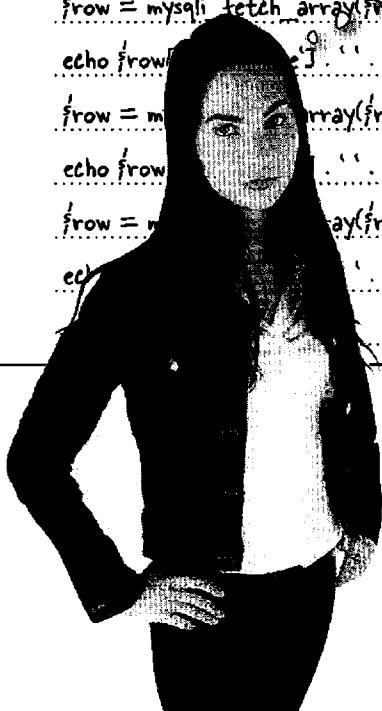
Como um teste para nos certificarmos de que realmente conseguimos obter os dados dos clientes, termine este código PHP para exibir o primeiro nome, sobrenome e endereço de email de cada cliente presente na tabela email\_list, uma linha de cada vez.

```
$query = "SELECT * FROM email_list";  
$result = mysqli_query($dbc, $query);  
$row = mysqli_fetch_array($result);
```

```
echo draw(firstname) . " " . draw(lastname) .  
    "  
    $row[email] . '  
';
```

nte seu lápis  
**Solução**

Como um teste para nos certificarmos de que realmente conseguimos obter os dados dos clientes, termine o código PHP para exibir o primeiro nome, sobrenome e endereço de email de cada cliente presente na tabela `email_list`, uma linha de cada vez.



**Existe uma maneira melhor – nós precisamos de um loop.**

Um **loop** é um mecanismo da linguagem PHP que repete um pedaço de código até que uma certa condição seja atendida, por exemplo, até se esgotarem os dados. Dessa forma, o loop é **capaz de analisar, ciclicamente, cada linha de dados de um resultado de consulta**, executando qualquer ação que quisermos, em qualquer linha.

## Loop com um WHILE

Um loop usando o `while` é um loop especificamente voltado à tarefa de ~~repetir o código enquanto uma~~ determinada condição esteja sendo atendida. Por exemplo, você pode ter uma variável em uma aplicação de serviço ao cliente chamada `$got_customers` (`$tem_cliente Esperando`), que serve para registrar a informação sobre o tempo de espera para a realização dos serviços, por parte do cliente. Se `$tem_cliente Esperando` estiver definida como `true` ("verdadeiro"), você sabe que ainda há clientes para atender, então você poderia, consequentemente, chamar a função `next_customer()` (`próximo_cliente`) para ajudar o próximo cliente da fila. Eis como poderíamos programar isso usando um loop `while`:

Um loop while repete o código enquanto uma condição estiver sendo atendida.

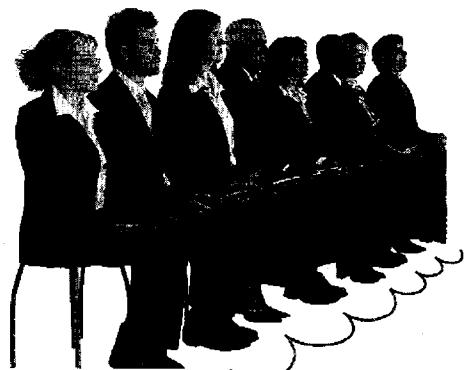
```
Enquanto tivermos clientes,  
continue o loop.  
while ($got_customers) {
```

```
while ($got_customers) {  
    next_customer(); } } Este é o código que  
    ... é executado a cada  
        iteração do loop.
```

} Colocar o código do loop dentro de chaves lhe permite executar quantas linhas de código quiser.

Quando verificamos “se há mais clientes a atender”, estamos testando uma **condição**. A condição é o código entre parênteses, e ela sempre apresenta uma questão que resulta em uma resposta sim/não. Se sim, ou **true** (“verdadeiro”), então a ação é realizada. Se não, ou **false** (“falso”), então terminamos o loop.

Quando chamamos `next_customer()` e passamos a atendê-lo, estamos **realizando uma ação**. A ação é o código dentro das chaves, e é repetida enquanto a condição permanecer `true`. Se a condição se tornar `false`, o loop termina e a ação não é mais repetida. Eis o formato geral de um loop usando o `while`:



Um loop while nos permite fazer loop através dos clientes até que não sobre nenhum para atender.

A condição de teste sempre resulta em verdadeiro ou falso... continuar olhando (verdadeiro) ou parar o loop (falso).

```
while (condição de teste) {  
    ação  
}
```

# PODER DO CÉREBRO

Como você acha que um loop while poderia ser usado na tabela email\_list de Elmer?

## como while() funciona

# Loop através dos dados com while

Aplicando um loop com while para os dados do email de Elmer, acessarmos os dados, uma linha de cada vez, sem duplicar nenhum código. Nós sabemos que `mysqli_fetch_array()` é capaz de tomar uma linha da tabela e colocar os valores das respectivas colunas no array `$row`, mas a função, sozinha, não passa através de todos os nossos dados – ela armazena a primeira linha e então para. Um loop while pode chamar `mysqli_fetch_array()` para atravessar todas as linhas dos dados do resultado, uma de cada vez, até chegar ao final.

A condição do loop com while é o valor retornado pela função `mysqli_fetch_array()`, que é interpretado como true, se houver dados disponíveis, ou false, se os dados tiverem acabado.

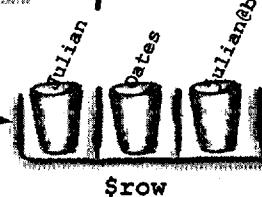
```
while($row = mysqli_fetch_array($result)) {
```

```
    echo $row['first_name'] . ' ' . $row['last_name'] . '  
        ' . $row['email'] . '<br>';
```

A ação do loop é executada uma vez a cada iteração.

A ação do loop consiste de uma instrução echo que junta os dados da linha, colocando uma quebra de linha ao final.

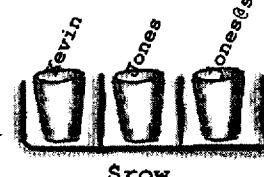
Na primeira passada do loop, o array `$row` armazenará a primeira linha da tabela `email_list`.



1º loop!

first_name	last_name	e-mail
Julian	Oates	julian@breackneckpizza.com
Kevin	Jones	jones@simuduck.com
Amanda	Sanchez	sunshine@breakneckpizza.com
...		

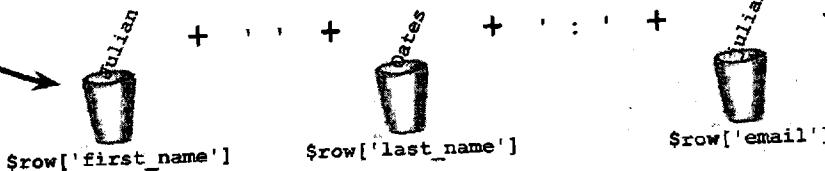
2º loop!



Mais loops...

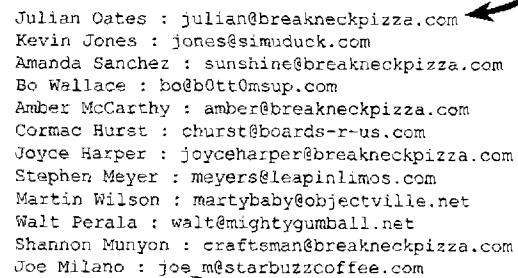
Na segunda passada do loop, o array `$row` armazenará a segunda linha da tabela `email_list`... está vendo o padrão?

A instrução echo dentro do loop while extrai os dados do array \$row e exibe o conteúdo formatado como HTML.

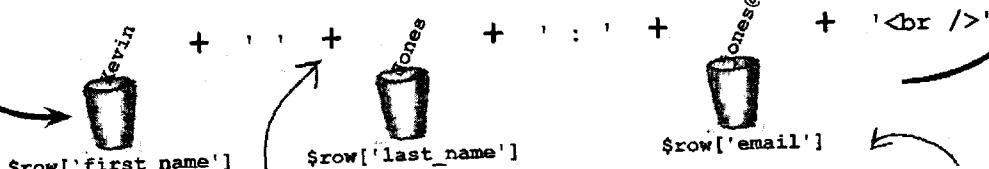


A chave usada para acessar o elemento do array deve ter o mesmo nome que uma coluna.

O loop while perpassa os dados da tabela, linha por linha. Quando não houver mais linhas de dados, ele finaliza a consulta.



Na segunda passada do loop, as instruções echo exibem uma outra sequência de texto, mas desta vez são usados os dados da segunda linha da tabela.



Na verdade, não se usa o sinal de mais para juntar duas strings – usa-se o operador ponto.

A cada passada do loop, os valores armazenados no array \$row se modificam para refletir a linha atual. Os nomes das colunas são usados para se acessar os valores do array.

Uma quebra  
HTML coloca  
cada linha de  
dados em uma  
linha própria na  
página resultante.

*não existem perguntas idiotas sobre while()*

*não existem*

## Perguntas Idiotas

**P:** Como exatamente o loop `while` sabe que deve continuar olhando? Quer dizer, o loop `while` é controlado por uma condição **verdadeiro/falso**, e `mysqli_fetch_array()` retoma algum tipo de ID do recurso, que é armazenado em `$row`... Isso certamente não se parece com uma condição de teste **verdadeiro/falso**.

**R:** Bem observado. Ocorre que o PHP é bem liberal no que diz respeito ao modo como ele interpreta a condição "verdadeira". Resumindo, qualquer valor que não seja zero (0) ou `false` é considerado como verdadeiro, em uma condição de teste. Assim, quando a função `mysqli_fetch_array()` retoma uma linha de dados, o array `$row` é interpretado como verdadeiro, uma vez que não está definido como 0 nem `false`. E, uma vez que a condição é verdadeira, o loop segue em frente. O interessante é o que acontece quando não há mais dados disponíveis – a função `mysqli_fetch_array()` retoma `false`, o que encerra o loop.

**P:** Então eu posso controlar um loop `while` com qualquer tipo de dados, e não apenas valores **verdadeiro ou falso**?

**R:** Correto. Mas tenha em mente que, no fim das contas, o loop `while` está interpretando os dados como **verdadeiro ou falso**. Assim, a coisa mais importante a se entender é o que constitui **verdadeiro ou falso** no que diz respeito à interpretação de outros tipos de dados. E a resposta simples é que qualquer coisa diferente de 0 ou `false` é sempre interpretada como verdadeiro.

**P:** O que acontece com o loop `while` se nenhum dado for retomado pela função `mysqli_fetch_array()`?

**R:** Se a consulta não resultar em quaisquer dados, então a função `mysqli_fetch_array()` retoma `false`. E isso faz o loop `while` nunca chegar ao código da ação, nem mesmo uma vez.

**P:** Então é possível ter um loop que nunca faz repetições?

**R:** É, sim. Também é possível ter um loop que nunca termina. Considere este loop `while`:

```
while (true) {
```

Isto é conhecido como **loop infinito**, porque a condição de teste nunca faz o loop terminar. Loops infinitos representam algo muito ruim.

## PONTOS DE BALA

- Um banco de dados é um container para armazenar dados de uma forma altamente estruturada.
- As tabelas armazenam dados em um padrão de colunas e linhas dentro de um banco de dados.
- O comando SQL `CREATE DATABASE` é usado para que seja criado um novo banco de dados.
- O comando SQL `CREATE TABLE` cria uma tabela dentro de um banco de dados e requisita informações detalhadas sobre as colunas de dados dentro da tabela.
- Você pode apagar uma tabela de um banco de dados com o comando SQL `DROP TABLE`.
- A função `mysqli_fetch_array()` obtém uma linha de dados a partir dos resultados de uma consulta ao banco de dados.
- Um loop `while` repete um pedaço de código PHP enquanto uma condição de teste continua sendo atendida.

1 Crie um banco de dados e uma tabela para a lista de emails.

2 Crie um formulário web e um script PHP "Adicionar Email" para adicionar novos clientes à lista.

3 Crie um formulário web e um script PHP "Enviar Email" para enviar um email para a lista.

7

Não se esqueça, ainda temos este último Passo para terminar.



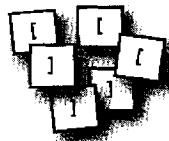
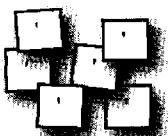
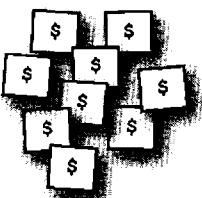
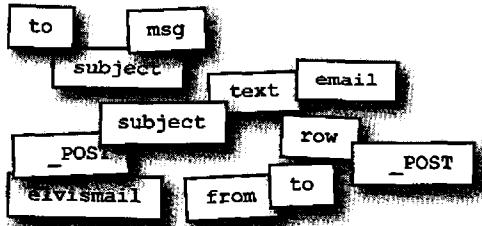
## Ímãs de Geladeira PHP & MySQL

Use os ímãs abaixo para completar o código do script "Enviar Email" de modo que Elmer possa começar a enviar emails para a sua lista de clientes. Para refrescar sua memória, eis aqui o modo como mail() funciona:

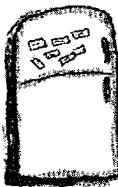
**mail(to, subject, msg, 'From:' . from);**

```
<?php
$from = 'elmer@makemeelvis.com';
$subject = 'Assunto';
$text = 'Corpo da mensagem';
$dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking', 'elvis_store')
or die('Erro ao se conectar com o servidor MySQL.');
$query = "SELECT * FROM email_list";
$result = mysqli_query($dbc, $query)
or die('Erro ao consultar o banco de dados.');
while($row = mysqli_fetch_array($result)) {
    $first_name = $row['first_name'];
    $last_name = $row['last_name'];
    $msg = "Dear $first_name $last_name,\n$text";
    $to = $row['email'];
    mail($to, $subject, $msg, 'From:' . $from);
    echo 'Email sent to: ' . $to . '<br />';
}
mysqli_close($dbc);
?>
```

sendemail.php



*script sendemail.php finalizado*



## Ímãs de Geladeira PHP & MySQL - Solução

Use os ímãs abaixo para completar o código do script "Enviar Email" de modo que Elmer possa começar a enviar emails para a sua lista de clientes. Para refrescar sua memória, eis aqui o modo como mail() funciona:

```
mail(to, subject, msg, 'From:' . from);
```

Certifique-se de trocar  
isto aqui pelo seu próprio  
endereço de email.

```
<?php
    $from = 'elmer@makemeelvis.com';
    $subject = $_POST['subject'];
    $text = $_POST['elvismail'];
    $dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking', 'elvis_store');
    or die('Erro ao se conectar ao servidor MySQL.');
    $query = "SELECT * FROM email_list";
    $result = mysqli_query($dbc, $query);
    or die('Erro ao consultar o banco de dados.');
    while($row = mysqli_fetch_array($result)) {
        $first_name = $row['first_name'];
        $last_name = $row['last_name'];
        $msg = "Dear $first_name $last_name,\n" . $text;
        $to = $row['email'];
        mail($to, $subject, $msg, 'From:' . $from);
        echo 'Email sent to: ' . $to . '<br />';
    }
    mysqli_close($dbc);
?>
```

O campo referente ao assunto chama-se "subject", que é o mesmo nome usado para acessá-lo no array \$\_POST.

O texto da mensagem de email é digitado no campo do formulário denominado "elvismail".

O destinatário do email, o assunto e o corpo da mensagem são passados para a função mail(), junto com o endereço "from" de Elmer.

O email destinatário, assunto da mensagem e corpo da mensagem, são passado na função mail(), com o endereço "de" Elmer."

A coluna "email" do banco de dados armazena os endereços dos clientes, aos quais a mensagem deve ser enviada.

Uma mensagem de confirmação é enviada para a página, com o endereço de cada cliente a quem foi enviada a mensagem.

Em geral, não é uma boa ideia, em termos de segurança, informar o que o usuário digitou diretamente para a função mail() sem fazer, antes uma verificação primeiro. O Capítulo 6 mostrará algumas técnicas para resolver este problema.

sendemail.php



## TEST DRIVE

### Envie um email para a lista usando o formulário “Enviar Email”

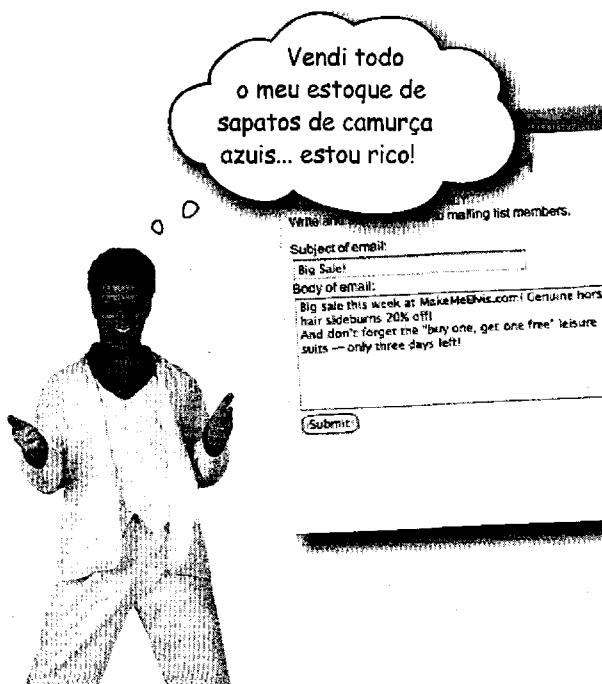
Baixe o código para a página “Enviar Email” no site da Alta Books, [www.altabooks.com.br](http://www.altabooks.com.br). Está na pasta **capítulo03**. Da mesma forma que a página “Adicionar Email” que você viu anteriormente, este código consiste de um formulário web em `sendmail.html`, uma folha de estilo (`style.css`) e algumas imagens (`elvislogo.gif` e `blankface.jpg`).

Crie um arquivo de texto chamado `sendmail.php` e digite nele todo o código da página anterior. Envie todos os arquivos para o seu servidor web e abra a página `sendmail.html` em um navegador. Digite uma mensagem de email no formulário e clique em Submit.

Tenha em mente que o seu endereço de email precisará estar na lista, para que você receba a mensagem.

### Você tem email... de Elmer!

Finalmente, Elmer pode enviar os seus emails anunciando promoções de QueroSerElvis.com (MakeMeElvis.com) para todos os inscritos na sua lista de email, usando o seu novo formulário e script PHP de “Enviar Email”. Ele pode, também, usar o output do script para confirmar que cada mensagem foi enviada com sucesso. Cada vez que o código no loop while do script é executada, ele vê “Email enviado para alguém@algumlugar.com”, com os endereços das pessoas no seu banco de dados. O resultado final é mais exposição para os seus produtos e, para o bem ou para o mal, mais sósias de Elvis Presley por aí!



O script “Send Email”  
realmente envia mensagens  
para os endereços  
presentes no banco de  
dados, portanto cuidado  
ao fazer alterações nele!

Writing a message to all members of mailing list members.

**Subject of email:** Big Sale!  
**Body of email:**

Big sale this week at MakeMeElvis.com! Genuine horse hair sideburns 20% off! And don't forget the "buy one, get one free" leisure suits — only three days left!

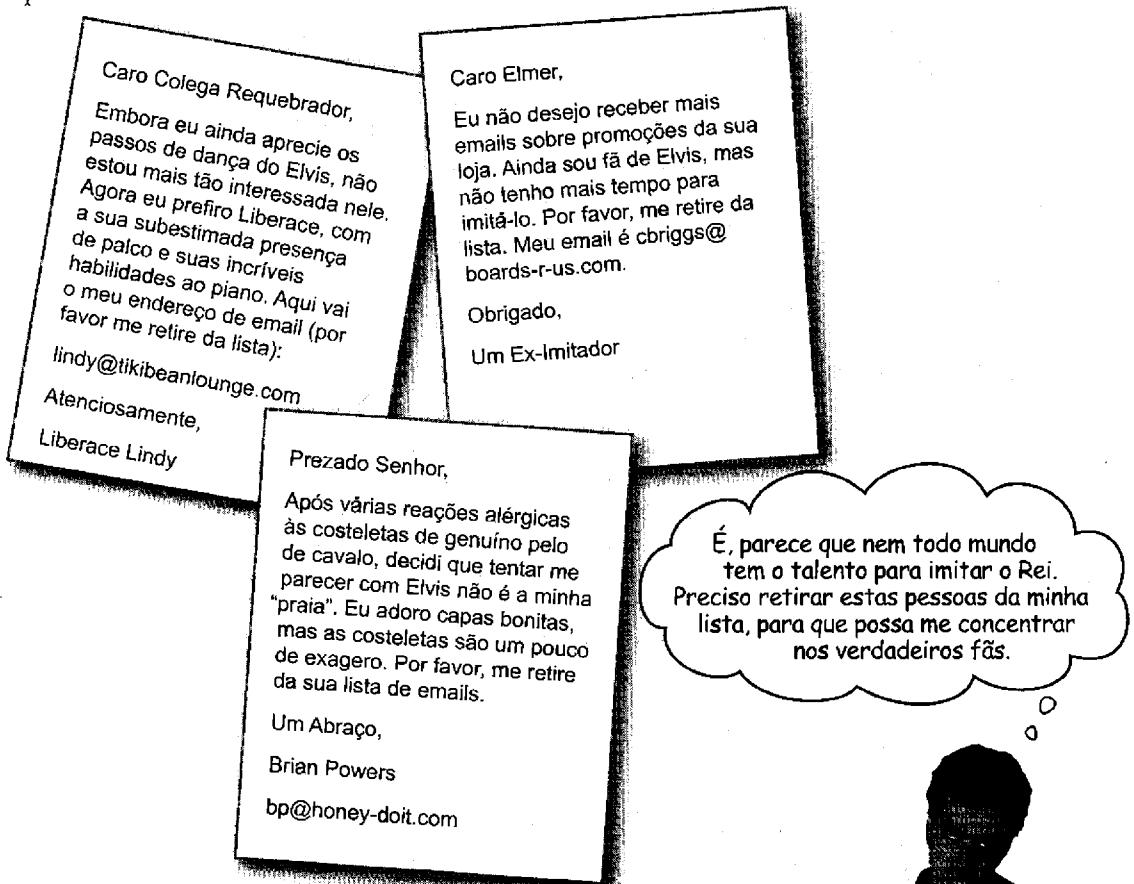
**Submit**

```
Email sent to: julian@breakneckpizza.com
Email sent to: jones@simduck.com
Email sent to: sunshine@breakneckpizza.com
Email sent to: bo@b0tflmsup.com
Email sent to: amber@breakneckpizza.com
Email sent to: churst@boards-r-us.com
Email sent to: joyceharper@breakneckpizza.com
Email sent to: meyern@leapinlimos.com
Email sent to: martybaby@objectville.net
Email sent to: walt@mighthygumball.net
Email sent to: craftsman@breakneckpizza.com
Email sent to: joe_m@starbuzzcoffee.com
Email sent to: bruce@chocoholic-inc.com
Email sent to: pr@honey-doi.com
Email sent to: bertieh@objectville.net
Email sent to: gregeck@breakneckpizza.com
Email sent to: wilnawni@starbuzzcoffee.com
Email sent to: sanjaffe@starbuzzcoffee.com
Email sent to: ls@objectville.net
Email sent to: bshakes@mighthygumball.net
```

*nosso aplicativo precisa da funcionalidade delete*

## Às vezes, as pessoas querem sair

Como acontece com qualquer nova empresa, existem obstáculos no caminho. Parece que alguns fãs de Elvis pularam do navio do Rei e querem sair da lista de Elmer. Ele quer atender a esses pedidos, mas para isso é necessário remover os clientes do seu banco de dados.



Elmer não está muito feliz em perder clientes, mas quer atender aos pedidos de remoção da sua lista de emails.

É um fato na vida do MySQL – às vezes você precisa remover dados do seu banco. Elmer precisa atualizar a sua aplicação para poder apagar usuários da tabela `email_list`.

Escreva aqui os novos componentes da aplicação que você acha que Elmer irá precisar para implementar o recurso de “Remover Email”:



## Removendo dados com DELETE

Para apagar dados de uma tabela, precisamos de um novo comando SQL, DELETE. Usaremos DELETE em um novo script "Remover Email", o qual apagará dados dos usuários da lista de emails de Elmer. Na verdade, precisamos de um novo script e de um novo formulário web para ativá-lo... mas antes de tudo precisamos da função DELETE.

O comando SQL DELETE remove linhas de dados de uma tabela. Isto o torna um comando que você deve usar com muito cuidado, uma vez que ele é capaz de eliminar uma tabela inteira, e todos os seus dados, num piscar de olhos. Sabendo disso, eis aqui a forma mais perigosa de DELETE, que apaga todas as linhas de uma tabela.

**DELETE FROM nome\_da\_tabela**

Este é o nome da tabela da qual você deseja apagar linhas.

Sem nenhum outro qualificador, o comando DELETE esvazia completamente a tabela, removendo todos os seus dados!

Então não podemos nunca apagar algo de uma tabela sem apagar tudo?

Parece que precisamos de um novo Passo... às vezes os projetos precisam se modificar!

**Não, de jeito nenhum. DELETE pode ser usado especificando-se uma determinada linha, ou linhas, para serem apagadas.**

Para especificar precisamente a linha – ou as linhas – que se deseja apagar com DELETE, você precisa adicionar uma cláusula WHERE. Se você se lembra de como a usamos com o comando SELECT, WHERE tem a mesma função aqui: permitir que você isole linhas específicas em uma consulta.



### Aponte seu lápis

Suponha que Elmer tenha 23 clientes cujo primeiro nome seja Anne, 11 clientes cujo sobrenome seja Parker, e uma cliente com o nome Anne Parker. Escreva abaixo quantas linhas de dados seriam apagadas por cada uma destas consultas.

DELETE FROM email\_list WHERE first\_name = 'Anne';

23

DELETE FROM email\_list WHERE first\_name = 'Anne' OR last\_name = 'Parker';

33

DELETE FROM email\_list WHERE last\_name = 'Parker';

6 (11)

## **DELETE com WHERE**



### **Aponte seu lápis Solução**

Suponha que Elmer tenha 23 clientes cujo primeiro nome seja Anne, 11 clientes cujo sobrenome seja Parker, e uma cliente com o nome Anne Parker. Escreva abaixo quantas linhas de dados seriam apagadas por cada uma destas consultas.

`DELETE FROM email_list WHERE first_name = 'Anne';`

23

`DELETE FROM email_list WHERE first_name = 'Anne' OR last_name = 'Parker';`

33

`DELETE FROM email_list WHERE last_name = Parker;`

0

Pegadinha! O sobrenome não está entre aspas, portanto, nenhuma linha é apagada - todos os valores de texto precisam estar entre aspas.

## **Use WHERE e DELETE para apagar dados específicos**

Usando uma cláusula WHERE com o comando DELETE, nós especificamos algumas linhas para serem apagadas, em vez de tudo que houver na tabela inteira. A cláusula WHERE permite que nos concentremos apenas nas linhas que desejamos remover, neste caso um dos clientes de Elmer que pediu para ser retirado da lista.

`DELETE FROM email_list`

O valor a ser procurado.

`WHERE email = 'pr@honey-doit.com'`

O nome de  
uma coluna da  
tabela

Esta parte da cláusula WHERE realiza um teste em todas as linhas, para ver quais atendem ao valor que está sendo procurado.

O teste propriamente dito, dentro da cláusula WHERE, realiza uma comparação que é executada em cada linha da tabela. Neste exemplo, o sinal de igual (=) testa cada valor da coluna email para ver quais linhas são iguais a "pr@honey-doit.com". Se o valor da coluna email bater com o que está sendo procurado, então a linha em questão é apagada.

A cláusula WHERE limita a consulta, de forma que o foco volte-se às linhas específicas da tabela.

Escreva aqui por que você acha que a coluna email é usada na cláusula WHERE, e não first\_name ou last\_name:

...  
...  
...  
...  
...

# Minimize o risco de apagamentos acidentais

É importante entender que, embora qualquer coluna possa ser usada em uma cláusula WHERE para selecionar linhas, há um motivo muito especial para termos escolhido a coluna email para a consulta DELETE de Elmer. Considere que, se mais de uma linha atender à cláusula WHERE, todas as linhas que também atenderem, serão apagadas. Assim, é importante que a cláusula WHERE de Elmer indique exatamente, a linha que deve ser apagada.

Estamos nos referindo, aqui à unicidade. É razoavelmente seguro assumir que, entre os endereços de email da tabela email\_list, não haverá dois emails idênticos, enquanto nomes e sobrenomes não forem detectados como únicos. Você não vai querer criar uma cláusula WHERE que procure por "Pat" na coluna first\_name para apagar um único cliente – você acabará apagando todos os clientes chamados Pat! É por isso que a cláusula WHERE de Elmer tem de ser cuidadosamente elaborada, de forma a procurar algo bastante específico, na coluna email.

**DELETE FROM email\_list**

WHERE email = 'pr@honey-doit.com'

Usar a coluna email na cláusula WHERE ajuda a estabelecer a unicidade e a reduzir o risco de se apagar uma linha acidentalmente.

Se tivéssemos usado first\_name na cláusula WHERE, em vez de email, este usuário teria sido, acidentalmente, deletado.

## Uma cláusula

WHERE em uma instrução DELETE lhe permite indicar a linha que você quer remover.

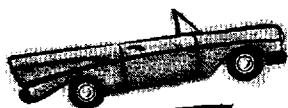
A consulta DELETE remove esta linha do banco de dados... para nunca mais ser vista!

first_name	last_name	e-mail
Joe	Milano	joe_m@starbuzzcoffee.com
Bruce	Spence	bruce@chocoholic-inc.com
Pat	Rice	pr@honey-doit.com
Bertie	Henderson	bertieh@objectville.net
Greg	Eckstein	gregeck@breakneckpizza.com
Wilma	Wu	wilmawu@starbuzzcoffee.com
Sam	Jaffe	samjaffe@starbuzzcoffee.com
Louis	Shaffer	ls@objectville.net
Bubba	Shakespeare	bshakes@mightygumball.net
John	Doe	johndoe@tikibeanlounge.com
Pat	Grommet	grommetp@simuduck.com

File Edit Window Help ByeBye

```
mysql> DELETE FROM email_list WHERE email = 'pr@honey-doit.com';
1 row deleted (0.005 sec)
```

**test-drive o comando DELETE**



## **TEST DRIVE**

### **Teste o comando DELETE no banco de dados de Elmer.**

Inicie uma ferramenta MySQL e emita alguns comandos DELETE para apagar linhas individuais da tabela email\_list, com base nos endereços de email dos clientes. Certifique-se de incluir uma cláusula WHERE em todas as instruções DELETE, para não acabar, acidentalmente, deletando a tabela inteira!

O comando DELETE é bem útil, mas o ideal seria apagarmos linhas de dados usando um formulário web e um script PHP, certo?



**Correto. Apagar usuários manualmente, com consultas individuais, não é a forma ideal de se gerenciar uma lista de emails.**

Já que Elmer, inevitavelmente, irá se deparar com usuários que desejarão ser removidos da sua lista, faz todo o sentido futuramente, desenvolver uma interface web para a remoção de usuários da lista. Um formulário web HTML e um script PHP resolvem o problema, juntamente com uma consulta DELETE FROM auxiliada por uma cláusula WHERE...



Elmer criou um formulário web (`removeemail.html`) para apagar clientes da sua lista. Só o que o formulário está aceitando apenas um endereço de email, o qual é digitado em um campo chamado `email`. Complete o código do script `removeemail.php` que é chamado pelo formulário para executar cada remoção de usuário.

Este campo do formulário chama-se "email".

Clicar no botão "Remove" envia o formulário, na forma de uma requisição POST, ao script PHP.



<?php

```
$dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking', 'elvis_store')
or die('Erro ao se conectar com o servidor MySQL.')
$email = $_POST['email'];
$query = "DELETE FROM email_list WHERE
email = '$email'";
$mysql_query($query);
$result = mysqli_query($dbc, $query)
or die('Erro');
echo 'Email: ' . $email . ' apagado!';
die;
```

removeemail.php

## *o script removeemail.php finalizado*



### Exercício Solução

Elmer criou um formulário web (`removeemail.html`) para apagar clientes da sua lista. Só o que o formulário está aceitando apenas um endereço de email, o qual é digitado em um campo chamado `email`. Complete o código do script `removeemail.php` que é chamado pelo formulário para executar cada remoção de usuário.

Este campo do formulário chama-se "email".

Clicar no botão "Remove" envia o formulário, na forma de uma requisição POST, ao script PHP.

Os dados do formulário presentes em `$_POST['email']` são armazenados em uma variável, e depois usados na consulta DELETE.

```
$_email = $_POST['email'];
```

```
$query = "DELETE FROM email_list WHERE email = '$email';  
mysql_query($dbc, $query)
```

```
or die('Erro ao consultar o banco de dados');
```

```
echo 'Cliente removido: ' . $email;
```

Não se esqueça de fechar a conexão com o banco de dados.

Este campo do formulário chama-se "email".

Clicar no botão "Remove" envia o formulário, na forma de uma requisição POST, ao script PHP.

removeemail.html

Cuidado com estas aspas simples e duplas aqui! As aspas duplas ficam em torno de toda a consulta SQL, e as simples ficam em torno do endereço de email armazenado em `$email`.

Nunca é demais confirmar o que aconteceu, principalmente quando se trata da remoção de dados do banco.

removeemail.php



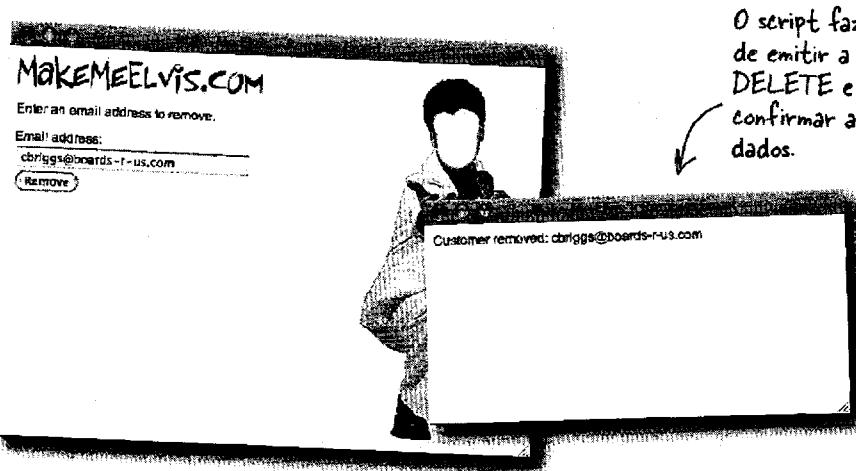
- 1 Crie um banco de dados e uma tabela para a lista de emails.
- 2 Crie um formulário web e um script PHP "Adicionar Email" para adicionar novos clientes à lista.
- 3 Crie um formulário web e um script PHP "Enviar Email" para enviar um email à lista.
- 4 Crie um formulário web e um script PHP "Remover Email" para remover clientes da lista.

## TEST DRIVE

### Remova um cliente da lista de emails usando o formulário "Remover Email".

Isto está começando a lhe parecer familiar, não? Baixe o código para a página "Remover Email" no site da Alta Books, [www.altabooks.com.br](http://www.altabooks.com.br). Está na pasta **capítulo03**. O código consiste de um formulário web em **removeemail.html**, uma folha de estilo (**style.css**) e algumas imagens (**elvislogo.gif** e **blankface.jpg**).

Crie um arquivo de texto chamado **removeemail.php** e digite nele todo o código da página anterior. Envie todos estes arquivos para o seu servidor web e abra a página **removeemail.html** em um navegador. Digite o endereço de email de um cliente no formulário e clique em "Remove" para apagá-lo do banco de dados.



*o aplicativo lista de email está completo!*

## "QueroSerElvis.com" é uma aplicação web

É oficial. Com a ajuda do PHP e do MySQL, o site QueroSerElvis.com de Elmer agora é digno de ser chamado de aplicação web. Elmer agora é capaz de armazenar dados, permanentemente, em um banco de dados MySQL e também de interagir com esses dados através de formulários web. Uma combinação de páginas HTML scripts PHP e consultas SQL inseridas permite a Elmer adicionar e remover clientes da sua lista de email (os clientes também podem se inscrever sozinhos), bem como enviar mensagens de email para a lista inteira.



Viva o PHP e o MySQL! Isso é o que eu chamo de aplicação web. Eu posso gerenciar minha lista de email, enviar mensagens para todos os meus clientes e até mesmo excluir usuários da lista... tudo sem sair do meu navegador web.

addemail  
addemail.php

A página "Adicionar Email" adiciona novos clientes à lista de email de Elmer.

sendemail  
sendemail.php

A página "Enviar Email" envia uma mensagem para todos os inscritos na lista, ao clique de um botão.

removeemail  
removeemail.php

Devolver ao remetente!  
Por favor, me retire da lista de emails.

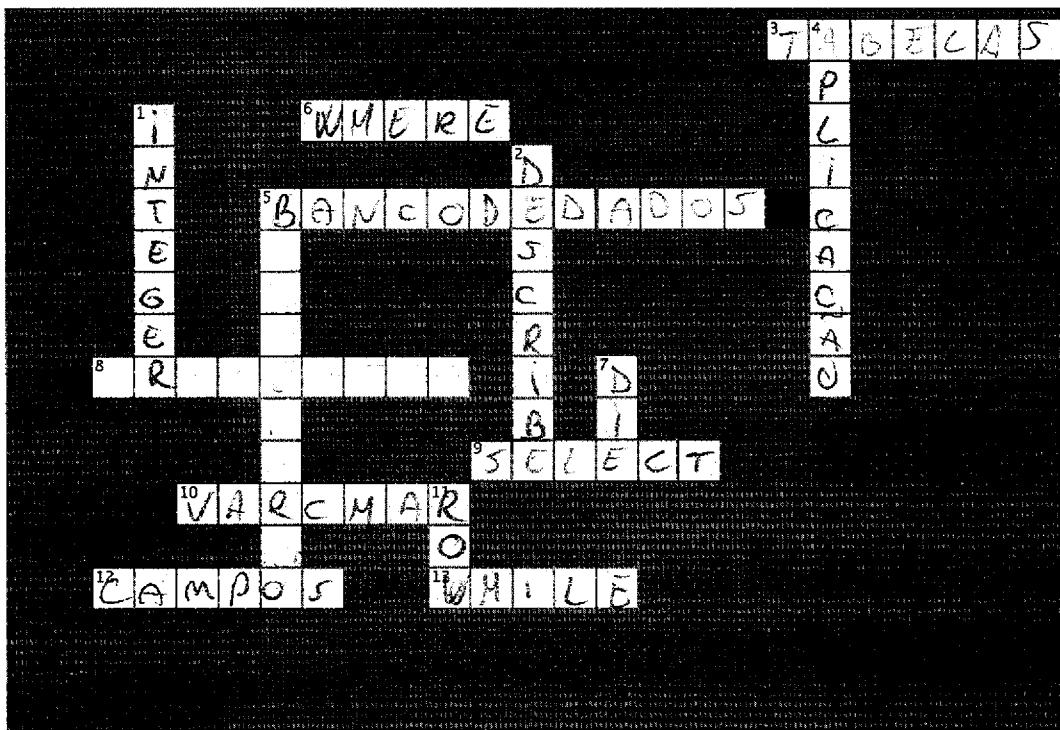
A página "Remover Email" exclui um usuário da lista de emails.





# Cruzadas PHP & MySQL

Depois que você tiver treinado, os passos de dança de Elmer, veja se consegue cantar junto e completar estas palavras cruzadas.



### Horizontais

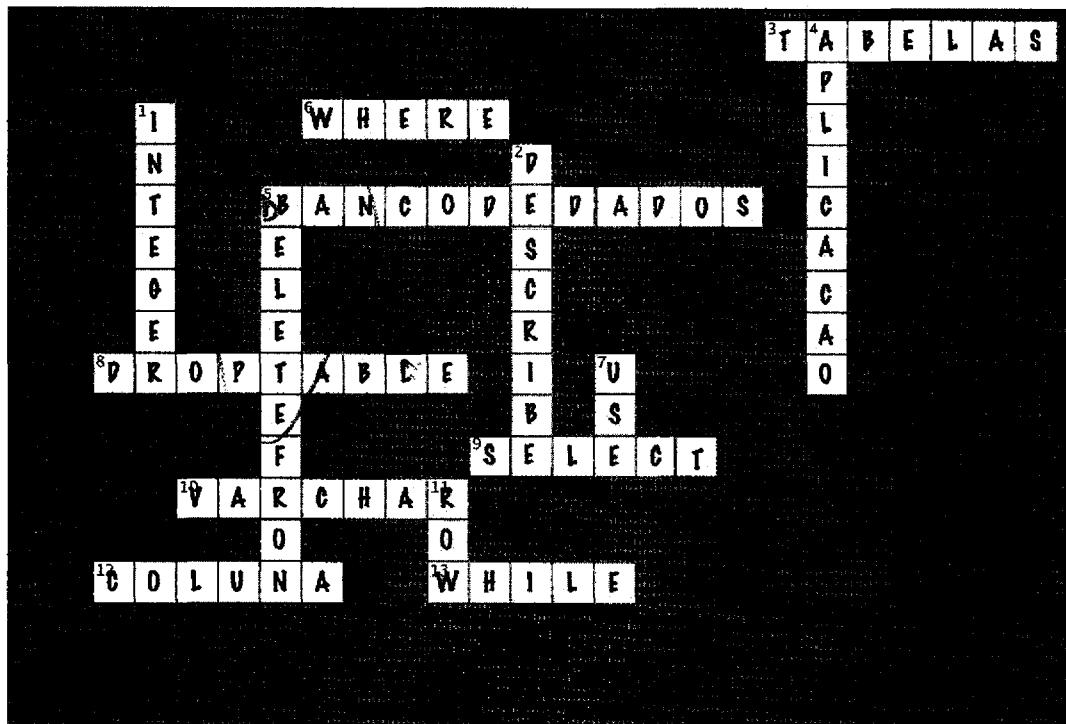
3. Um banco de dados MySQL divide-se nas
  4. Uma estrutura de dados persistente, altamente organizada, que geralmente é armazenada em um arquivo, no disco rígido.
  5. Esta cláusula condicional pode ser adicionada a instruções SQL para controlar quais linhas são especificadas.
  6. Este comando SQL remove uma tabela inteira do banco de dados.
  7. Use este comando SQL para selecionar linhas de uma tabela.
  8. Use este tipo de dados do MySQL para armazenar uma quantidade variável de texto.
  9. Dentro de uma tabela MySQL, isto armazena um tipo específico de dados.
  10. Continua fazendo algo, enquanto uma determinada condição de teste permanecer verdadeira.

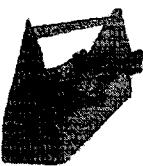
## Verticais

1. Um tipo de dados do MySQL que armazena números sem casas decimais.
  2. Use este comando SQL para olhar a estrutura de uma tabela.
  4. Quando funcionalidades dinâmicas são adicionadas a um web site usando-se PHP e MySQL, o site torna-se uma .....
  5. Use este comando SQL para destruir linhas de uma tabela.
  7. Após ter criado um novo banco de dados em um terminal MySQL, você deve exexutar este comando antes de fazer qualquer coisa com o banco.
  11. Termo em inglês que representa um conjunto singular de dados de uma tabela, consistindo de um elemento de cada coluna.



## Cruzadas PHP & MySQL - Solução





## Sua caixa de ferramentas PHP & MySQL

Você não só ajudou Elmer a montar sua aplicação web, como também desenvolveu valiosas habilidades com PHP e MySQL neste capítulo. Por exemplo...

### **while**

Um construto de loops do PHP que lhe permite repetir uma seção do código enquanto uma determinada condição permanecer verdadeira. Um uso particularmente interessante do loop while é voltado à realização de um ciclo através de linhas de dados, em um resultado de consulta SQL.

### **DELETE FROM**

*nomeDaTabela*

Use esta instrução SQL para apagar linhas de uma tabela. Dependendo de como você usar a instrução, é possível apagar linhas individuais ou então mais de uma linha.

### **WHERE**

Esta cláusula SQL é usada em conjunto com outros comandos SQL para se elaborar instruções SQL que especificuem determinadas linhas de uma tabela. Por exemplo, você pode isolar linhas que tenham uma coluna contendo um determinado valor.

### **mysqli\_fetch\_array()**

Esta função interna do PHP obtém uma linha de dados a partir dos resultados de uma consulta ao banco de dados. Você pode chamar esta função repetidamente, para ler várias linhas de dados.

### **DROP TABLE**

*nomeDaTabela*

Esta instrução SQL apaga uma tabela inteira do banco de dados, significando que a tabela é removida junto com quaisquer dados armazenados dentro dela.

### **DESCRIBE *nomeDaTabela***

Se precisar saber como é a estrutura de uma tabela, esta instrução SQL é o que você precisa. Ela não revela quaisquer dados, mas mostra os nomes das colunas e os seus respectivos tipos de dados.

### **SELECT \* FROM**

*nomeDaTabela*

Esta instrução SQL seleciona linhas de uma tabela. Quando a estrutura (\*) é usada, todas as colunas de todas as linhas da tabela são retornadas. Você pode ser mais específico, listando nomes de colunas individuais em vez de \*, caso não queira receber, da consulta, todos os dados registrados na tabela.



## 4 Aplicações Realistas e Práticas

# Sua Aplicação na Web

Se eu colocar uma banana no cano do escapamento, o carro da professora não vai pegar, e com isso não teremos prova. Mas aí o professor substituto pode aplicar a prova, então é preciso enguiçar o carro dele, também. Mas aí tem o substituto do substituto...

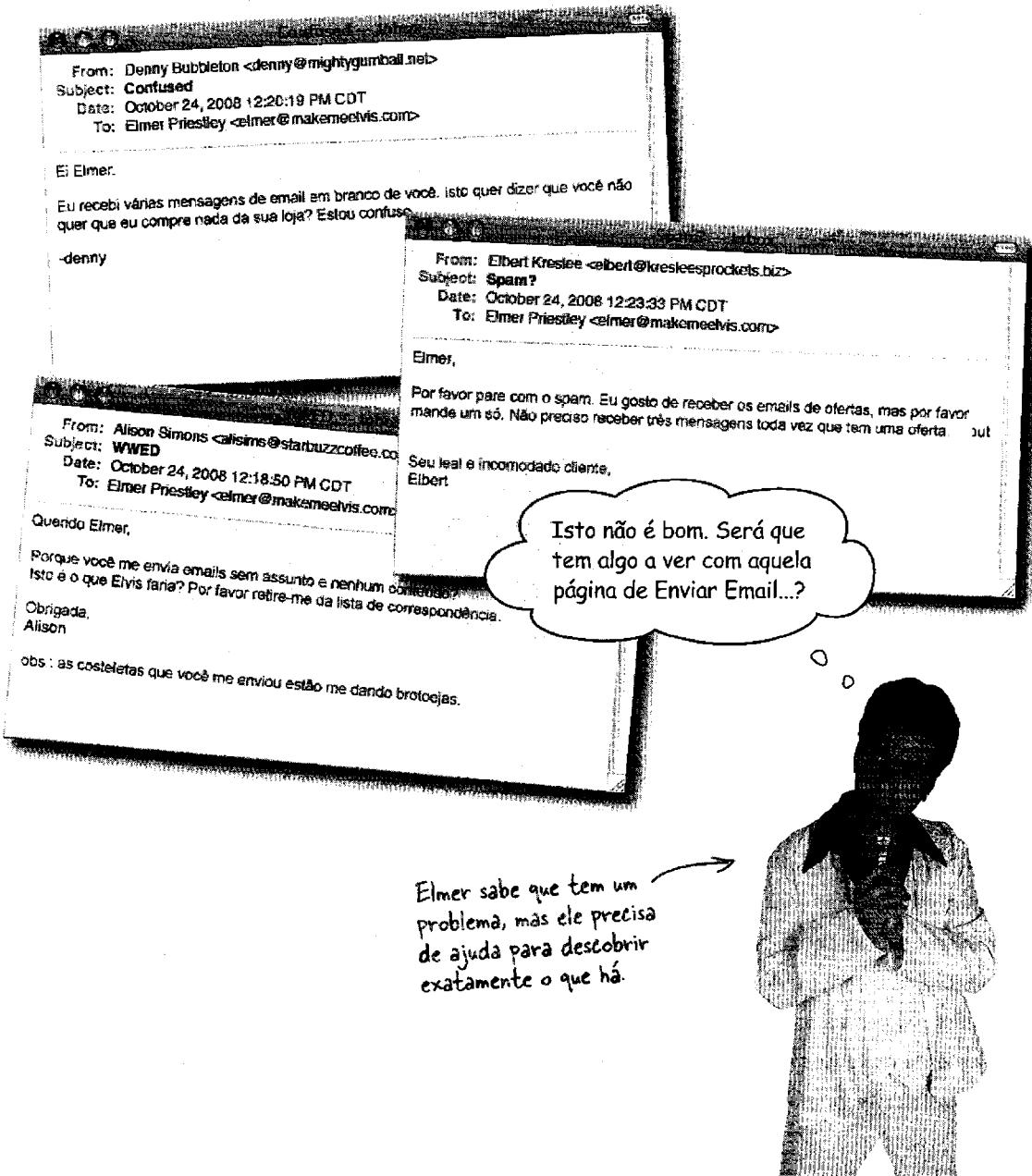


**Às vezes você precisa ser realista e repensar seus planos.** Ou então planejar com mais cuidado logo no começo. Uma vez lançada a sua aplicação na Web, você poderá descobrir que não planejou suficientemente bem. Coisas que você pensou que funcionariam podem não ser boas o suficiente no mundo real. Este capítulo dá uma olhada em alguns *problemas do mundo real* que podem ocorrer quando você **transfere sua aplicação do ambiente de testes para um site real**. E enquanto isso, nós mostraremos exemplos de códigos PHP e SQL importantes.

*elmer precisa de um aplicativo de lista de email melhor*

## Elmer tem alguns clientes irritados

A lista com os emails dos clientes de Elmer cresceu exponencialmente, mas os seus emails têm gerado algumas reclamações. Elas são variadas, mas todas parecem ter algo a ver com os usuários recebendo mensagens em branco, ou múltiplas mensagens, o que não é bom em nenhum dos casos. Elmer precisa descobrir o que deu errado e consertá-lo. O seu negócio depende disso.



## SINTA-SE como Elmer, o administrador da lista de emails



Sua tarefa é interpretar o papel de Elmer e descobrir como esses emails em branco estão sendo enviados. Ele suspeita que tem algo a ver com o formulário sendemail.html.

Escreva aqui o que Elmer acha que pode ser o problema.

**MAKEMEELVIS.COM**

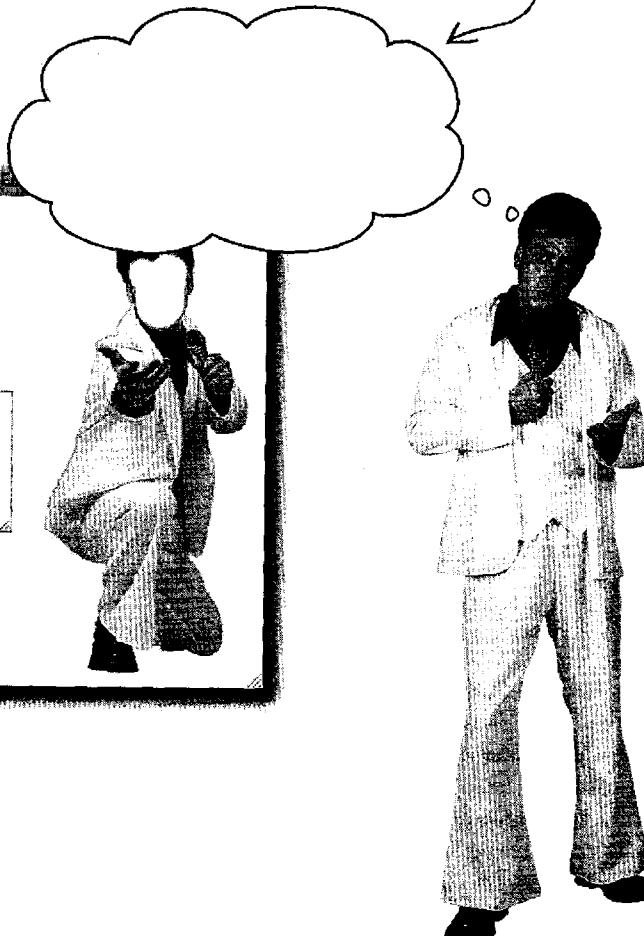
Private: For Elmer's use ONLY  
Write and send an email to mailing list members.

Subject of email:

Body of email:

Submit

sendemail.html



## sinta-se como elmer solução

# SINTA-SE como Elmer, o administrador da lista de emails - Solução



Sua tarefa é interpretar o papel de Elmer e descobrir como esses emails em branco estão sendo enviados. Ele suspeita que tem algo a ver com o formulário sendemail.html.

Escreva aqui o que Elmer acha que pode ser o problema.

http://makemeelvis.com

Private: For Elmer's use ONLY  
Write and send an email to mailing list members.

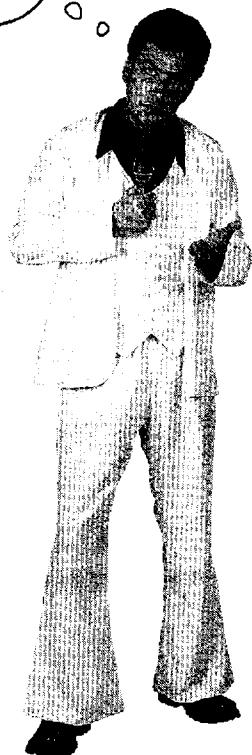
Subject of email:

Body of email:

Submit

sendemail.html

Se eu clicar no botão Submit sem digitar uma mensagem, é enviado um email em branco.



Se o botão Submit for pressionado no formulário sem haver nada no campo Body (corpo da mensagem), é enviado um email em branco. E agora que estamos pensando no assunto, um campo Subject (assunto) vazio também é problemático.

# Protegendo Elmer de... Elmer

Portanto, o problema aqui está naquela peça que fica entre o teclado e a cadeira – Elmer está clicando accidentalmente em Submit (Submeter) sem digitar uma mensagem, e com isso emails em branco são enviados para a lista inteira. Nunca é seguro presumir que um formulário web será usado exatamente da forma que foi planejada. É por isso que você, um programador PHP atento, tem a responsabilidade de tentar evitar esses tipos de problemas, antevendo que alguns usuários irão usar os seus formulários de forma errada.

Vamos dar uma olhada no código do nosso script sendemail.php atual, para ver como as mensagens vazias de Elmer estão sendo criadas.

Nosso script Enviar Email usa o texto do formulário para criar um email, mesmo que o usuário não tenha digitado nada.

```
<?php
$from = 'elmer@makemeelvis.com';
$subject = $_POST['subject'];
$text = $_POST['elvismail'];
```

O texto do formulário é obtido de `$_POST['subject']` e `$_POST['elvismail']`, e é salvo em `$subject` e `$text`, respectivamente...

```
$dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking', 'elvis_store')
or die('Erro ao se conectar ao banco de dados.');

$query = "SELECT * FROM email_list";
$result = mysqli_query($dbc, $query)
or die('Erro ao consultar o banco de dados.');

while ($row = mysqli_fetch_array($result)){
    $to = $row['email'];
    $first_name = $row['first_name'];
    $last_name = $row['last_name'];
    $msg = "Dear $first_name $last_name, \n$text";
    mail($to, $subject, $msg, 'From:' . $from);
    echo 'Email enviado: ' . $to . '<br />';
}

mysqli_close($dbc);
```

O problema é que nós usamos `$text` na nossa mensagem, independentemente de a variável conter algum texto ou não...

...e nós também usamos `$subject`, independentemente de haver texto nela ou não.

Escreva aqui o que você acha que deveria ser modificado no código do script sendemail.php para consertar o problema dos emails em branco:

*Adicionar uma condição para verificar se \$text não é vazio*

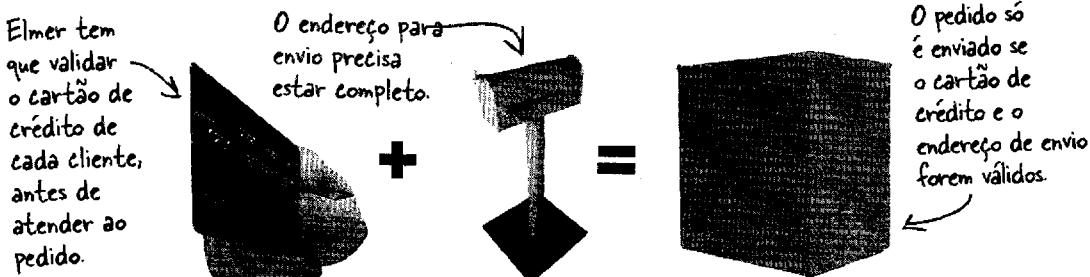
`sendemail.php` precisa de validação

## Exija bons dados do formulário

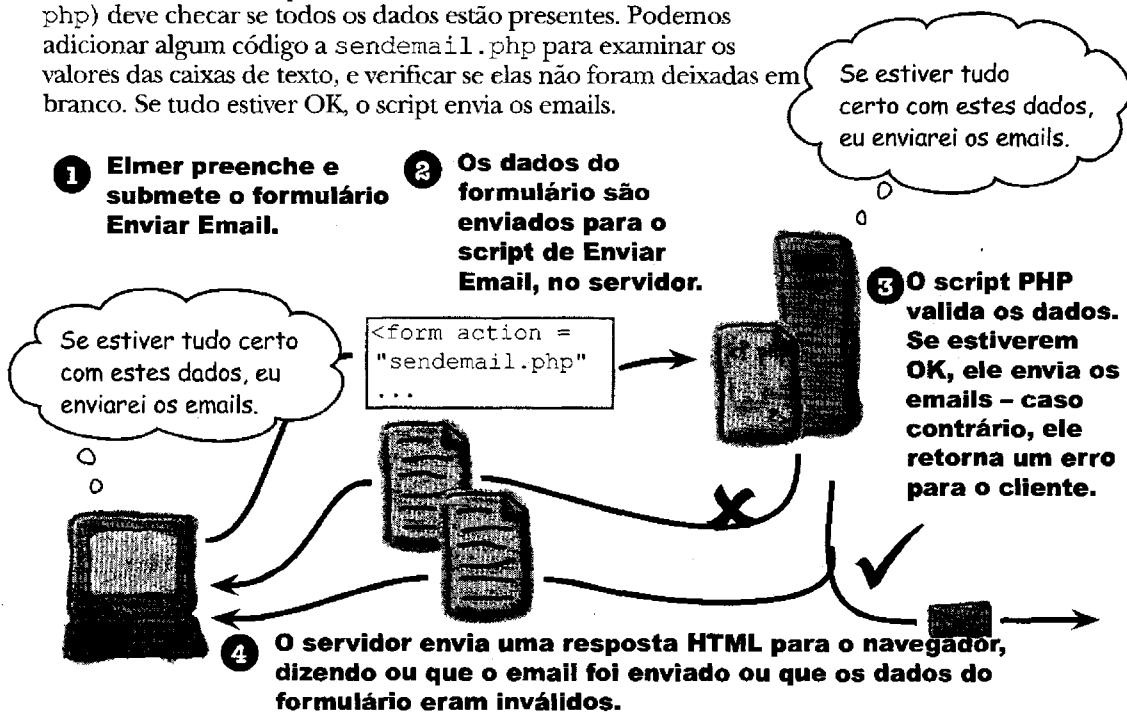
O formulário de Enviar Email de Elmer precisa de alguma **validação**, que é o processo de se verificar se os dados do formulário estão OK antes de fazer qualquer coisa com eles. Elmer já está usando validação, embora não a esteja chamando por esse nome. Sempre que ele recebe um pedido através do site, ele não envia o produto imediatamente... ele valida o pedido antes.

No caso de um pedido, Elmer primeiramente verifica se o cartão de crédito do cliente é **válido**. Caso afirmativo, ele prepara o produto para envio. Mas para isso, ele precisa verificar se o endereço do cliente está completo. Se estiver tudo certo, então Elmer faz o envio. Para que um pedido seja processado com sucesso na loja de Elmer, é necessário validar vários dados referentes a esse pedido.

Validar significa certificar-se de que os dados que você está recebendo são os esperados.



Para resolver o problema dos emails em branco, precisamos validar os dados do formulário entregues ao script `sendemail.php`. Isso significa que os dados devem ser submetidos da página do cliente (`sendemail.html`) para o servidor e o servidor (`sendemail.php`) deve checar se todos os dados estão presentes. Podemos adicionar algum código a `sendemail.php` para examinar os valores das caixas de texto, e verificar se elas não foram deixadas em branco. Se tudo estiver OK, o script envia os emails.



# A lógica por trás da validação de Enviar Email

Elmer precisa **validar** os dados que obtém do formulário `sendemail.html`, para poder enviar os emails. Na verdade, a situação ideal é que o envio dos emails dependa totalmente da validação dos dados. O que realmente precisamos que o PHP faça é **tomar uma decisão** com base na validade dos dados recebidos pelo script `sendemail.php`. Precisamos de algum código que diga “*se os dados forem válidos, vá em frente e envie os emails*”.

Estas duas condições precisam ser verdadeiras para o dado ser considerado válido.

**SE** Subject contiver texto **E** Body contiver texto

✓ **ENTÃO** enviar email

Se ambas as condições forem atendidas, é sinal de que está tudo certo e que podemos enviar os emails.

Estávamos mandando email sem nos preocuparmos se algo fosse digitado nestes campos do formulário.

Com a ajuda da validação, podemos nos certificar de que os emails só sejam enviados se ambos os campos contiverem dados.

não existem  
Perguntas Idiotas

**P:** Eu já ouvi falar em validar os dados no cliente, em vez de no servidor. Como isso funciona?

**R:** O navegador web é considerado como o cliente, de modo que a validação no lado do cliente seria qualquer verificação que ocorra antes de os dados serem enviados para o script PHP. Linguagens como JavaScript são capazes de fazer validação no lado do cliente. Se estiver interessado em aprender mais, dê uma olhada em *Use a Cabeça! JavaScript*, o qual aborda em detalhes a validação no lado do cliente.

**P:** Então por que usar a validação no lado do servidor, em vez de no lado do cliente?

**R:** Se validarmos no cliente, apenas parte do problema é resolvida. Elmer poderia navegar diretamente até `sendemail.php` e enviar um email em branco. Porém, se validarmos no servidor, isso resolve ambos os problemas. Dados em branco no formulário serão detectados, bem como dados em branco sendo carregados diretamente no script PHP. Isso não equivale a dizer que é errado validar no cliente. Na verdade, é uma ideia muito boa. Mas o servidor é a última linha de defesa para se capturar dados ruins, portanto a validação no lado do servidor não pode ser ignorada.

## Seu código pode tomar decisões com IF

A declaração PHP if permite ao seu código tomar decisões com base em se algo é verdadeiro ou não. Considere novamente os pedidos da loja de Elmer. Antes de atender um pedido, Elmer precisa receber o pagamento, o que significa cobrar no cartão de crédito do cliente. Se o cliente fornecer a Elmer um número errado de cartão de crédito, ele não pode atender ao pedido. Assim, Elmer realiza uma espécie de validação-no-mundo-real em cada pedido, que acontece mais ou menos assim:

**Se o cartão de crédito do cliente for válido, vá em frente e atenda ao pedido**

Nós podemos traduzir este cenário para código PHP usando a declaração if, que serve para lidar exatamente com este tipo de tomada de decisões.

### A instrução if básica tem três partes:

#### 1 A palavra-chave if

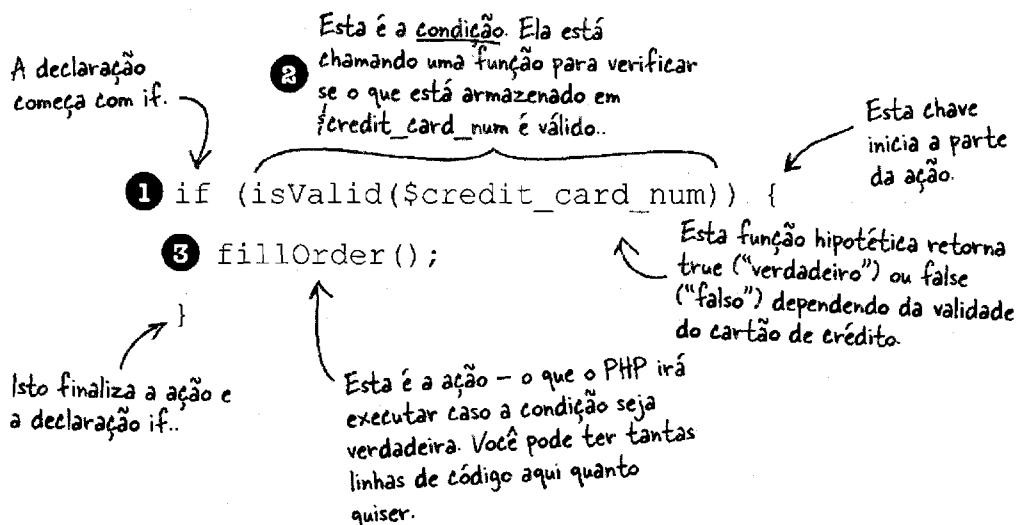
Ela inicia a declaração

#### 2 A condição de teste

A condição de teste ou **expressão condicional**, localiza-se entre **parênteses** logo após a palavra-chave if. É aqui que você coloca a declaração cuja validade ou veracidade, você deseja determinar.

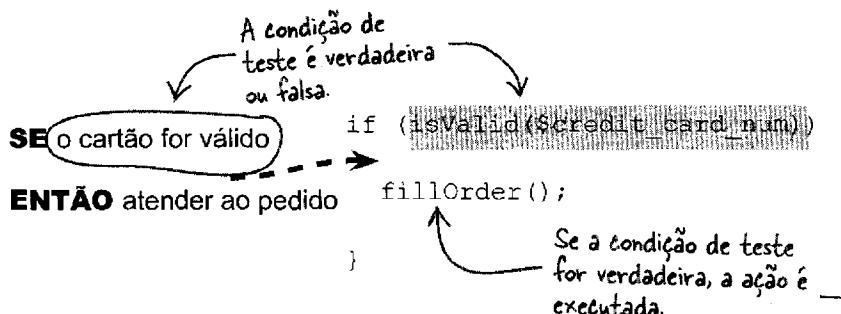
#### 3 A ação

A ação de uma declaração if vem logo após a condição de teste, e fica dentro de **chaves**. É aqui que você coloca o código PHP que deseja executar quando a condição for verdadeira.

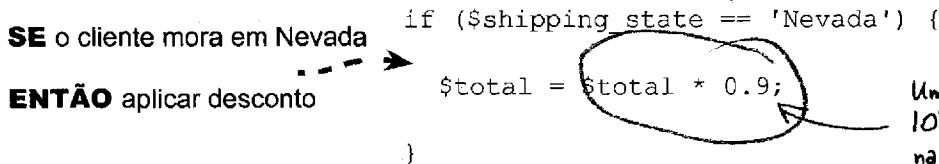


## Testando em busca da verdade

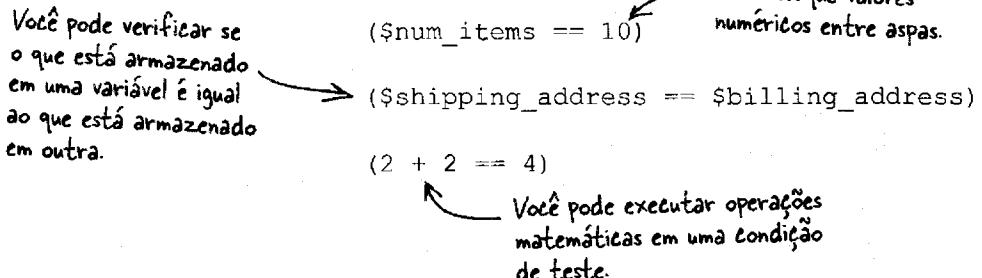
O cerne da declaração `if` é a sua condição de teste, que é sempre interpretada como verdadeira ou falsa. A condição de teste pode ser uma variável, uma chamada a função ou uma comparação de uma coisa com outra, por exemplo. A validação de cartão de crédito feita por Elmer usa uma chamada a função como a condição de teste, o que significa que o valor retornado pela função será `true` ("verdadeiro") ou `false` ("falso")..



É bastante comum usar uma comparação como condição de teste, o que normalmente significa comparar uma variável com algum outro valor. Por exemplo, talvez Elmer queira oferecer um desconto para clientes que morem em Nevada. Ele poderia criar uma declaração `if` para executar uma comparação em parte do endereço de envio, desta forma:



Esta condição de teste realiza uma comparação de igualdade, a qual envolve dois sinais de igual (`==`). As comparações de igualdade não são apenas para variáveis e strings. Você pode comparar variáveis com números, variáveis com variáveis e até mesmo realizar cálculos.



## a mais do que apenas igualdade

Uma declaração if pode verificar mais do que apenas igualdade.

A condição de teste da sua declaração if pode também verificar se um valor é maior do que outro, por exemplo. Se for, o resultado da condição é true e o código da ação é executado. Eis aqui alguns outros testes que você pode usar para controlar a decisão de uma declaração if.

Comete com estas duas variáveis.

Existem duas formas de se verificar se as coisas não são iguais: `<>` e `!=`. Isto lhe dá o resultado oposto de um teste de igualdade com `=`.

O sinal maior que (`>`) verifica se o valor da esquerda é maior do que o valor da direita. Se for, a condição é true; caso contrário, é false.

O sinal menor que (`<`) compara o valor da esquerda com o valor da direita. Se o da esquerda for menor do que o da direita, a condição é true.

Maior ou igual a (`>=`) funciona como maior que (`>`), exceto pelo fato de que ele também resulta em true se os dois valores forem iguais.

Menor ou igual a (`<=`) é semelhante a menor que, exceto pelo fato de que também resulta em true se os valores forem iguais.

Não tem problema escrever uma declaração if em uma só linha, desde que a ação seja relativamente simples.

```
$small_number = 2;
```

```
$big_number = 98065;
```

Ambas estas condições são verdadeiras.

```
if ($small_number <> $big_number) { echo 'True'; }
```

```
if ($small_number != $big_number) { echo 'True'; }
```

Esta condição é falsa.

```
if ($small_number > $big_number) { echo 'True'; }
```

Esta condição é verdadeira.

```
if ($small_number < $big_number) { echo 'True'; }
```

Esta condição é falsa.

```
if ($small_number >= $big_number) { echo 'True'; }
```

Esta condição é verdadeira.

```
if ($small_number <= $big_number) { echo 'True'; }
```



E quanto as strings? Será que ("cachorro" > "gato") funcionaria?

**Sim, você pode comparar strings em condições de teste**

A comparação funciona com base no alfabeto, com a letra a sendo considerada como menor do que a letra z. Usar os sinais de maior que ou menor que pode lhe ajudar quando você tem que apresentar informações em ordem alfabética

# SINTA-SE como uma condição de teste da declaração if



Sua tarefa é interpretar o papel da condição de teste if e decidir se você é verdadeiro ou falso, dadas as seguintes variáveis.

```
$my_name = 'Buster';
$a_number = 3;
$a_decimal = 4.6;
$favorite_song = 'Trouble';
$another_number = 0;
$your_name = $my_name;
```

|   |                            |
|---|----------------------------|
| <code>(\$a_number == 3)</code>                | <b>verdadeiro ou falso</b> |
| <code>(\$another_number == "")</code>         | <b>verdadeiro ou falso</b> |
| <code>(\$favorite_song == "Trouble")</code>   | <b>verdadeiro ou falso</b> |
| <code>(\$my_name == '\$your_name')</code>     | <b>verdadeiro ou falso</b> |
| <code>(\$my_name == "\$your_name")</code>     | <b>verdadeiro ou falso</b> |
| <code>(\$your_name == \$my_name)</code>       | <b>verdadeiro ou falso</b> |
| <code>(\$favorite_song == 'Trouble')</code>   | <b>verdadeiro ou falso</b> |
| <code>(\$a_number &gt; 9)</code>              | <b>verdadeiro ou falso</b> |
| <code>(\$favorite_food == 'hamburger')</code> | <b>verdadeiro ou falso</b> |

sinta-se como uma condição de teste solução

## SINTA-SE como uma condição de teste da declaração if - Solução

Sua tarefa é interpretar o papel da condição de teste if e decidir se você é verdadeiro ou falso, dadas as seguintes variáveis.



```
$my_name = 'Buster';
$a_number = 3;
$a_decimal = 4.6;
$favorite_song = 'Trouble';
$another_number = 0;
$your_name = $my_name;
```

`($a_number == 3)`

verdadeiro ou falso

`($another_number == "")`

verdadeiro ou falso

`($favorite_song == "Trouble")`

verdadeiro ou falso

`($my_name == '$your_name')`

verdadeiro ou falso

`($my_name == "$your_name")`

verdadeiro ou falso

`($your_name == $my_name)`

verdadeiro ou falso

`($favorite_song == 'Trouble')`

verdadeiro ou falso

`($a_number > 9)`

verdadeiro ou falso

`($favorite_food == 'hamburger')` verdadeiro ou falso

Deveria ter sido ==, se a intenção aqui era fazer uma comparação..

não existem

### Perguntas Ídiotas

P: Então a condição de teste é a mesma coisa que usamos para controlar os loops while no Capítulo 3??

Q: É exatamente a mesma coisa. No capítulo 3, a usamos para saber se ainda tínhamos linhas de dados resultantes da consulta, mas podemos pensar em condições de teste mais interessantes para loops while, usando diferentes tipos de comparações. Você verá isso mais adiante no livro..

O e uma string vazia são considerados como iguais.

Devido às aspas simples, a condição na verdade está perguntando se a string Buster é igual à string "\$your\_name", e não ao valor contido na variável \$your\_name.

`$a_number` é 3, o que não é maior que 9.

OK, esta aqui é difícil. Uma vez que está sendo usado apenas um sinal de igual aqui, isto é na verdade uma atribuição (=), e não uma comparação (==). E ela acaba sendo verdadeira, porque qualquer coisa que não seja 0, NULL ou false é interpretada como true pelo PHP.

# A lógica por trás da validação de Enviar Email

Elmer precisa **validar** os dados que obtém do formulário sendemail.html, para poder enviar os emails. Na verdade, a situação ideal é que o envio dos emails dependa totalmente da validação dos dados. O que realmente precisamos que o PHP faça é **tomar uma decisão** com base na validade dos dados recebidos pelo script sendemail.php. Precisamos de algum código que diga “*se os dados forem válidos, vá em frente e envie os emails.*”

Mas primeiro nós precisamos obter os dados do formulário e os armazenar em algumas variáveis:

```
$subject = $_POST['subject'];
$text = $_POST['elvismail'];
```

Isto é tudo de que precisamos para verificar se existem dados em cada um dos campos do formulário. A lógica é algo como o seguinte:

**SE** Subject contiver texto **E** Body contiver texto

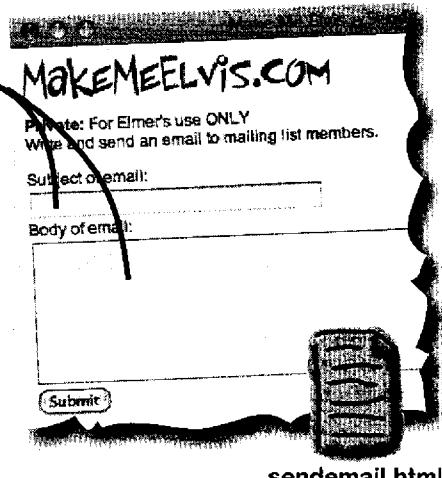
**ENTÃO** enviar email

Ou então poderíamos adotar a abordagem oposta e verificar se os campos do formulário estão ambos vazios, em cujo caso poderíamos exibir um aviso para o usuário:

**SE** Subject estiver vazio **E** Body estiver vazio

**ENTÃO** exibir mensagem de erro

Ambos os exemplos têm um problema, porque a lógica deles requer que nós façamos duas comparações em uma mesma declaração if. Uma solução possível é usar duas declarações if...



sendemail.html



## Aponte seu lápis

Escreva duas declarações if para verificar se tanto o assunto quanto o corpo da mensagem do formulário Enviar Email estão vazios. Emite um aviso caso estejam.

IF (\$subject == NULL AND  
\$body == NULL) then.....

.....  
.....  
.....

## nte seu lápis

### Solução

Escreva duas declaração if para verificar se tanto o assunto quanto o corpo da mensagem do formulário Enviar Email estão vazios. Emite um aviso caso estejam..

Colocando a segunda instrução if dentro da primeira, o código está dizendo que ambas precisam ser verdadeiras para que a instrução echo seja executada.

```
if ($subject == '') {  
    if ($text == '') {  
        echo 'Você se esqueceu de digitar o assunto e o texto do email<br />';  
    }  
}  
A indentação ajuda a mostrar onde a declaração if interna termina e onde começa a declaração if externa.
```

## Funções PHP para verificar variáveis

Usar == para checar se uma string está vazia funciona, mas existe uma forma melhor de fazer isso, através de funções internas do PHP. A função **isset()** verifica se uma determinada variável existe, o que significa dizer que ela tem um valor atribuído a si. A função **empty()** dá um passo além e determina se a variável contém um **valor vazio**, que o PHP define como um 0, uma string vazia (' ' ou "") ou os valores false ou NULL. Assim, **isset()** só retorna true se um valor tiver sido atribuído à variável, enquanto que **empty()** só retorna true se a variável tiver sido definida como 0, um string vazia, false ou NULL.

Vejamos como essas funções trabalham:

```
$v1 contém  
um valor.  
$v2 é uma  
string vazia..  
$v2 está  
definida,  
mesmo que  
contenha  
uma string  
vazia..  
$v3 não  
existe.
```

Tanto \$v1 quanto \$v2 são consideradas como tendo sido definidas, mesmo que somente \$v1 tenha um valor.

Somente o código sombreado é executado!!

\$v1 não está vazia, ela contém texto. Assim, esta condição if é falsa..

\$v2 está vazia porque a string que ela contém é vazia.

\$v3 é considerada vazia, mesmo que nem exista..

```
$v1 = 'aloha';  
$v2 = '';  
  
if (isset($v1)) { echo '$v1 está definida<br />'; }  
if (empty($v1)) { echo '$v1 está vazia<br />'; }  
  
if (isset($v2)) { echo '$v2 está definida<br />'; }  
if (empty($v2)) { echo '$v2 é uma string vazia<br />'; }  
  
if (isset($v3)) { echo '$v3 está definida<br />'; }  
if (empty($v3)) { echo '$v3 está vazia<br />'; }
```



Entendi. Nós podemos usar `isset()` e `empty()` para validar os dados de `$subject` e `$text`.

### **Quase isso. Na verdade, nós só estamos verificando se os dados não estão vazios, por isso `empty()` é o que precisamos.**

As variáveis de texto `$subject` e `$text` recebem valores das superglobais `$_POST['subject']` e `$_POST['elvismail']`. Se você testar essas variáveis com `isset()`, o teste sempre retornará true, independentemente de elas realmente conterem ou não algum texto. Em outras palavras, `isset()` não lhe mostra a diferença entre um campo em branco e um preenchido. A função `empty()` verifica se a variável está realmente vazia, que é o que precisamos para a validação do formulário.

isset() verifica se a variável existe e se está definida.

empty() verifica se a variável tem algum conteúdo.

não existem

### **Perguntas Ídiotas**

**P:** Então qual o propósito de usar `isset()`?

**R:** A função `isset()` é extremamente útil quando você precisa saber se um determinado dado existe. Por exemplo, você pode verificar se um formulário foi submetido através de uma requisição POST repassando `$_POST` à função `isset()`. Isso acaba sendo uma técnica bastante útil, como você descobrirá um pouco mais adiante neste capítulo.



### **Aponte seu lápis**

Reescreva as duas instruções `if` que criamos para verificar se tanto o assunto quanto o corpo da mensagem do formulário Enviar Email estão vazios, porém, desta vez, use a função `empty()` em vez de `==` nas condições de teste.

*if(empty(\$subject) OR empty(\$text)) {  
 echo "Email";  
}*

## Raponte seu lápis Solução

Reescreva as duas declarações if que criamos para verificar se tanto o assunto quanto o corpo da mensagem do formulário Enviar Email estão vazios, porém, desta vez, use a função empty() em vez de == nas condições de teste..

Uma chamada à função empty() substitui o operador de igualdade (==) em cada uma das condições de teste.

```
if (empty($subject)) { esse escopo só é válido dentro de um if
    echo 'Você esqueceu de digitar o assunto e o texto do email' para o tipo de variável que é
    } de classificação, por exemplo.
} O restante do código é o mesmo que antes.
} mas o teste não é mais o mesmo, ele não é mais
assumido como se estivesse correta.
```

E se nós precisarmos executar uma determinada ação apenas se o campo do formulário não estiver vazio? Existe uma função notempty()?

**Não, mas há uma maneira fácil de inverter a lógica de qualquer condição de teste... o operador de negação.**

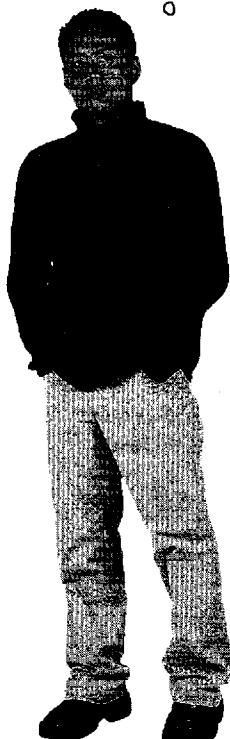
Nós sabemos que a condição de teste que controla uma declaração if sempre resulta em um valor true ou false. Mas e se a nossa lógica determinar que precisamos verificar o inverso do que uma condição nos apresenta? Por exemplo, seria útil saber se os campos do formulário de Elmer não estão vazios, antes de enviar um monte de emails. O problema é que não existe uma função notempty(). A solução é o operador de negação (!), que transforma true em false ou false em true. Assim,!empty () literalmente chama a função empty() e inverte o seu resultado, desta forma:

O operador NOT (!)  
transforma true em  
false ou false em true.

```
if (!empty($subject)) {
```

...  
}

↑ Esta condição está perguntando "O campo do Assunto está não-vazio", ou seja, ele contém dados?





Preencha as lacunas no código sendemail.php de Elmer, de modo que o email só seja enviado quando tanto \$subject e \$text não estiverem vazias. Use declaração if e a função empty().

Todos os  
meus campos  
precisam ter  
valores.

Makemeelvis.com

Private: For Elmer's use ONLY  
Write and send an email to mailing list members.

Subject of email:

Body of email:

Submit

sendemail.html

```
<?php
$from = 'elmer@makemeelvis.com';
$subject = $_POST['subject'];
$text = $_POST['elvismail'];

if(empty($subject)) { $erro = 'y' ;}
if(empty($text)) { $erro = 'y' ;}

if($erro == 'y') {
    $dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking',
    'elvis_store')
    or die('Erro ao se conectar com o servidor MySQL.');

    $query = "SELECT * FROM email_list";
    $result = mysqli_query($dbc, $query)
    or die('Erro ao consultar o banco de dados.');

    while ($row = mysqli_fetch_array($result)) {
        $to = $row['email'];
        $first_name = $row['first_name'];
        $last_name = $row['last_name'];
        $msg = "Caro(a) $first_name $last_name, \n$text";
        mail($to, $subject, $msg, 'From:' . $from);
        echo 'Email enviado para ' . $to . '<br />';
    }
    mysqli_close($dbc);
}

?>
```

**sendemail.php**—agora com validação!



## **Exercício**

ଶ୍ରୀବଦ୍ଧେନ

Todos  
os meus campos  
precisam ter valores..

O ponto de exclamação inverte a lógica da função empty().

```

<?php
    $from = 'elmer@makemeelvis.com';
    $subject = $_POST['subject'];
    $text = $_POST['elvismail'];

    if... (!empty($subject)) {
        if... (!empty($text)) {
            $dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking',
            'elvis_store')
                or die('Erro ao se conectar com o servidor MySQL.');

            $query = "SELECT * FROM email_list";
            $result = mysqli_query($dbc, $query)
                or die('Erro ao consultar o banco de dados');

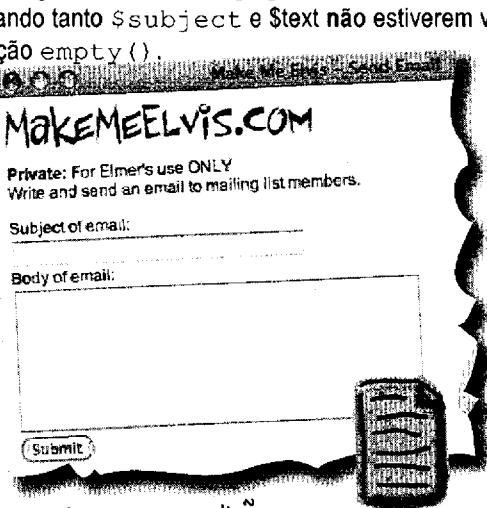
            while ($row = mysqli_fetch_array($result)) {
                $to = $row['email'];
                $first_name = $row['first_name'];
                $last_name = $row['last_name'];
                $msg = "Caro(a) $first_name $last_name, \n$text";
                mail($to, $subject, $msg, 'From:' . $from);
                echo 'Email enviado para: ' . $to . '<br />';
            }
            mysqli_close($dbc);
        }
    }
?>

```

A primeira condição verifica se \$subject não está vazia...  
 ...se não estiver, ótimo! Agora verificamos se \$text não está vazia.  
 Nós tivemos de colocar uma declaração if dentro da outra, para que o código funcionasse. Isso se chama aninhado.

Se qualquer uma das variáveis estiver vazia, uma das declarações if será falsa, e nada deste código será executado, o que significa que nenhum email em branco será enviado - exatamente o que queríamos!

Nós temos de finalizar a parte relativa à ação de ambas as declarações if. A primeira chave finaliza a declaração if interna, enquanto que a segunda finaliza a declaração if externa.



A primeira condição verifica se subject não está vazia...

...se não estiver, ótimo! Agora  
verificamos se `text` não está vazia.

Nós tivemos de colocar uma declaração if dentro da outra, para que o código funcionasse. Isso se chama aninhado.

Se qualquer uma das variáveis estiver vazia, uma das declarações if será falsa, e nada deste código será executado, o que significa que nenhum email em branco será enviado – exatamente o que queríamos!

Nós temos de finalizar a parte relativa à ação de ambas as declarações if. A primeira chave finaliza a declaração if interna, enquanto que a segunda finaliza a declaração if externa.



## TEST DRIVE

### Verifique se a validação dos campos do formulário funciona.

Modifique o código de sendemail.php para usar declarações if que verifique os dados do formulário antes de enviar as mensagens de email. Envie a nova versão do script para o seu servidor web e abra a página sendemail.html em um navegador. Certifique-se de deixar pelo menos um dos campos do formulário em branco, e clique em Submit.

O corpo da mensagem  
está vazio, o que  
faz os dados do  
formulário falharem  
na validação.

**MAKEMEELVIS.COM**

Private: For Elmer's use ONLY  
Write and send an email to mailing list members.

Subject of email:

Blue Suede Clearance

Body of email:

Submit



A ausência de confirmações  
de emails revela que nada  
foi enviado, que é o que nós  
queríamos. Mas algum tipo de  
aviso aqui seria mais útil do que  
uma página em branco.

## conversa a três



E se nós tivéssemos um monte de campos no formulário? Teríamos de aninhar um monte de instruções if para validar tudo??

O formulário de Owen, visto anteriormente neste livro, é um exemplo de como o fato de termos mais campos no formulário resulta em termos também um monte de declarações if aninhadas.

```
if (!empty($first_name)) {
    if (!empty($last_name)) {
        if (!empty($when_it_
happened)) {
            if (!empty($show_long)) {
                if (!empty($show_many)) {
```

**Joe:** Acho que você tem razão. Se quisermos nos certificar de que todos os campos estejam preenchidos, precisaremos colocar uma declaração if para cada campo.

**Frank:** Mas desde que façamos a indentação da linha de cada declaração, estará tudo bem, certo?

**Jill:** Tecnicamente, sim. Quer dizer, o código certamente vai funcionar, não importa quantos if's nós colocarmos. Mas eu me preocupo se ele não vai ficar difícil de entender, com tanto aninhamento. Colocar as chaves corretamente poderia se tornar um problema.

**Frank:** Isso é verdade. E eu acho que também seria trabalhoso ter de indentar o código de ação que temos até aqui... vejamos, são dez campos no formulário, o que nos dá dez if's aninhados, com dez níveis de indentação. Mesmo se cada if for indentado com apenas dois espaços, isso dá 20 espaços antes de cada linha do código. Nada bom.

**Joe:** Mas se indentarmos com tabs, isso diminui pela metade – 10 tabs em vez de 20 espaços não é tão ruim assim.

**Jill:** Pessoal, a questão não é o modo como nós indentamos os if's aninhados. É que simplesmente não é uma boa prática de programação colocar tantos if's uns dentro dos outros. Pensem desta forma – o que estamos realmente falando é de uma condição de teste lógica, “todos os campos do nosso formulário estão não-vazios?” O problema é que essa condição de teste envolve dez dados diferentes, fazendo com que nós tenhamos de dividi-la em dez declarações if separadas.

**Frank:** Ah, entendi. Então o que precisamos é de uma forma de verificar todos os dez dados em uma só condição de teste, certo?

**Jill:** Isso.

**Joe:** Então nós poderíamos escrever uma só condição de teste gigante, para verificar todos os campos do formulário de uma só vez. Seria ótimo.

**Jill:** Sim, mas ainda não temos o pedaço do quebra-cabeça que nos permitirá combinar diversas comparações em uma mesma condição de teste...

**Aliens Abducted Me - Report an Abduction**

Share your story of alien abduction:

|   |  |
|---|--|
| First name:                                     | <input type="text"/>   |
| Last name:                                      | <input type="text"/>   |
| What is your email address?                     | <input type="text"/>   |
| When did it happen?                             | <input type="text"/>   |
| How long were you gone?                         | <input type="text"/>   |
| How many did you see?                           | <input type="text"/>   |
| Describe them:                                  | <input type="text"/>   |
| What did they do to you?                        | <input type="text"/>   |
| Have you seen my dog Fang?                      | <input checked="" type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> |
|   |  |
| Anything else you want to add?                  |  |
| <input type="button" value="Report Abduction"/> |  |

Tanto aninhamento torna difícil manter o controle de quantas chaves precisam ser usadas em que lugares.

# Teste múltiplas condições com AND e OR

É possível construir uma condição de teste para uma declaração if com diversas verificações conectando-as através de um operador lógico. Vamos ver como isso funciona com duas condições familiares, `!empty($subject)` e `!empty($text)`. Este primeiro exemplo envolve duas expressões conectadas através do operador lógico AND ("e"), que é codificado usando-se `&&`.

Os parênteses extras ajudam a deixar claro que o operador de negação só se aplica à função `empty()`.

O operador lógico AND...  
`if ((!empty($subject)) && (!empty($text))) {`  
 Esta condição de teste só é verdadeira se tanto \$subject quanto \$text não estiverem vazias.

O operador AND toma dois valores true/false e lhe retorna true somente se ambos forem verdadeiros; caso contrário, o resultado é false. Assim, neste caso, ambos os campos precisam estar não-vazios para que a condição de teste seja verdadeira e o código de ação da declaração if possa rodar.

O operador lógico OR ("ou"), codificado como `||`, é semelhante a AND, exceto pelo fato de que ele resulta em true se pelo menos um dos dois valores true/false for verdadeiro. Eis um exemplo:

`if ((!empty($subject)) || (!empty($text))) {`  
 Esta condição de teste é verdadeira se \$subject OU \$text estiver não-vazia.

Assim, o código de ação para esta declaração if é executado se um dos campos do formulário estiver não-vazio. As coisas ficam ainda mais interessantes se você quiser isolar um campo como estando vazio, mas com o outro tendo dados, desta forma:

`if (empty($subject) && (!empty($text))) {`

Uma vez que esta condição de teste usa AND, ambas as expressões dentro dela precisam ser true para que o código da ação possa se executar. Isso significa que o campo Subject ("assunto") do formulário precisa estar vazio, mas o campo Body ("corpo da mensagem") precisa ter dados nele. Você pode inverter essa verificação passando o operador de negação (`!`) para a outra função `empty()`:

`if ((!empty($subject)) && empty($text)) {`

Isto aqui não é o número onze, e sim duas barras verticais - fica na parte de cima da tecla barra invertida (`\`), no seu teclado.

\$subject precisa estar vazia e \$text precisa estar não-vazia para esta condição de teste ser verdadeira.

Isto só é verdadeiro se \$subject não estiver vazio e \$text estiver.

Os operadores lógicos AND (`&&`) e OR (`||`) tornam possível elaborar condições de teste muito mais poderosas que, de outra forma, necessitariam de muitas, e frequentemente bagunçadas declarações if.

A lógica do PHP torna possível elaborar declarações if mais elegantes.

O AND lógico é codificado como `&&`, enquanto que o OR lógico é codificado como `||`.

## **eliminando o aninhamento da declaração if**



Reescreva as seções salientadas do script sendemail.php para que ele use operadores lógicos em uma única condição de teste, em vez de declarações if aninhadas.

```
<?php
    $from = 'elmer@makemeelvis.com';
    $subject = $_POST['subject'];
    $text = $_POST['elvismail'];

    if (!empty($subject)) {
        if (!empty($text)) {
            if ((empty($subject)) || (empty($text))) {
                .....
                $dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking',
                    'elvis_store')
                    or die('Erro ao se conectar com o servidor MySQL.');

                $query = "SELECT * FROM email_list";
                $result = mysqli_query($dbc, $query)
                    or die('Erro ao consultar o banco de dados.');

                while ($row = mysqli_fetch_array($result)) {
                    $to = $row['email'];
                    $first_name = $row['first_name'];
                    $last_name = $row['last_name'];
                    $msg = "Caro(a) $first_name $last_name, \n$text";
                    mail($to, $subject, $msg, 'From:' . $from);
                    echo 'Email enviado para: ' . $to . '<br />';
                }

                mysqli_close($dbc);
            }
        }
    }
?>
```

Aqui estão nossas declarações if aninhadas. Re-escreva usando uma única declaração if e com operadores lógico

Estas chaves fecham as duas declarações if.



## TEST DRIVE

**Certifique-se de que os operadores lógicos no script "Enviar Email" estejam fazendo o mesmo trabalho que as declarações if aninhadas.**

Modifique o código de `sendemail.php` para usar uma única declaração if que tire proveito dos operadores lógicos para verificar os dados do formulário, antes de enviar as mensagens de email. Consulte a solução do exercício na próxima página, caso não tenha certeza sobre como fazer as modificações.

Envie a nova versão do script para o seu servidor web e abra a página `sendemail.html` em um navegador. Certifique-se de deixar pelo menos um dos campos em branco e clique em Submit. O script ainda impede as mensagens de serem enviadas quando um campo é deixado em branco?

não existem

### Perguntas Ídiotas

**P:** Faz diferença a ordem em que você coloca as duas conexões ligadas por `&&` ou `||` em uma declaração if?

**R:** Sim. O motivo é que o funcionamento desses dois operadores pega atalhos sempre que possível. O que isso significa é que, se o primeiro operando for suficiente para determinar o resultado da expressão, o segundo é ignorado. Por exemplo, se o primeiro operando de uma expressão AND for false, isto é suficiente para fazer a expressão inteira ser false, independentemente do segundo operando, de modo que este é ignorado. A mesma regra se aplica quando o primeiro operando de uma expressão OR é true.

**P:** vi códigos PHP que usam `and` e `or` em vez de `&&` e `||`. Como isso funciona?

**R:** Eles são praticamente a mesma coisa que `&&` e `||`. Há uma leve diferença no modo como eles são avaliados em relação a outros operadores, mas, se você tiver o cuidado de usar parênteses para tornar as suas condições de teste claras, então não haverá nenhuma diferença essencial.



## Exercício Solução

Reescreva as seções salientadas do script *sendemail.php* para que ele use operadores lógicos em uma única condição de teste, em vez de declarações if aninhadas.

```
<?php
    $from = 'elmer@makemeelvis.com';
    $subject = $_POST['subject'];
    $text = $_POST['elvismail'];

    if (!empty($subject)) {
        if (!empty($text)) {
            if ((!empty($subject)) && (!empty($text))) { ←
                $dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking',
                    'elvis_store')
                or die('Erro ao se conectar com o servidor MySQL.');

                $query = "SELECT * FROM email_list";
                $result = mysqli_query($dbc, $query)
                or die('Erro ao consultar o banco de dados.');

                while ($row = mysqli_fetch_array($result)) {
                    $to = $row['email'];
                    $first_name = $row['first_name'];
                    $last_name = $row['last_name'];
                    $msg = "Caro (a) $first_name $last_name,\n$text";
                    mail($to, $subject, $msg, 'From:' . $from);
                    echo 'Email enviado para: ' . $to . '<br />';
                }
            }
            mysqli_close($dbc);
        }
    }
?>
```

A negação ou o operador NOT (!), é usada para checar se há campos não-vazios no formulário.

Nós podemos usar AND para verificar ambas as condições, em uma só declaração if.

Lembre-se, && é o modo como você especifica o operador lógico AND.

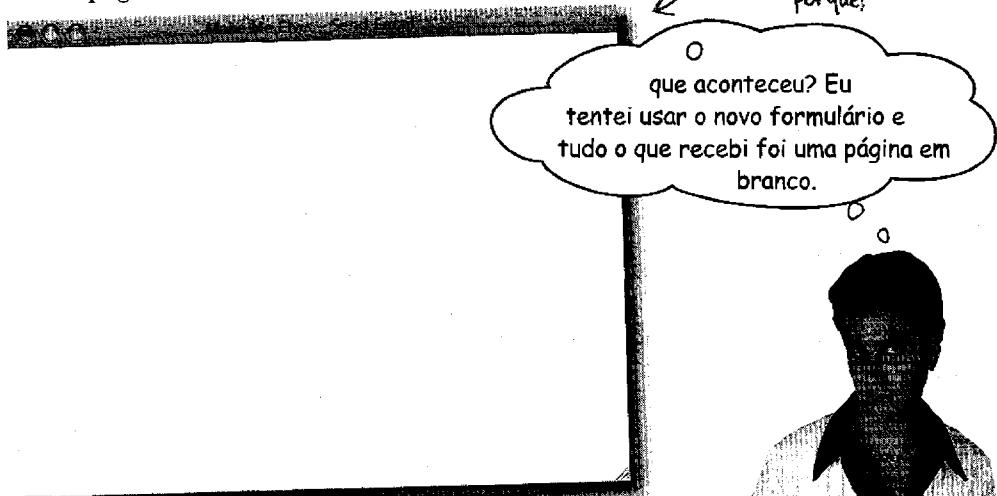
Todo o código dentro da declaração if deve ser recuado em um nível de indentação, uma vez que agora ele reside dentro de uma única declaração if.

Tendo apenas uma declaração if, nós só precisamos de uma chave de fechamento.

# Os usuários do formulário precisam de uma resposta

O código do nosso script `sendemail.php` faz um bom trabalho ao validar os dados do formulário, com o objetivo de impedir que emails sejam enviados caso o campo Subject (Assunto) ou o Body (Texto) sejam deixados em branco. Mas quando a validação falha, e não são enviados emails, o script não informa a Elmer o que aconteceu. Ele vê apenas uma página web em branco.

Elmer vê esta página quando submete o formulário... e não tem a menor ideia do porquê!



O problema é que o nosso código só reage a uma validação com sucesso, em cujo caso ele envia as mensagens de email. Mas se a declaração if acabar sendo false (dados do formulário inválidos), o código não faz nada, deixando Elmer sem saber se foram enviados emails ou não, ou então o que deu errado. Eis aqui o código abreviado, que revela o problema da página em branco:

```
<?php
$from = 'elmer@makemeelvis.com';
$subject = $_POST['subject'];
$text = $_POST['elvismail'];

if ((!empty($subject)) && (!empty($text))) {
    $dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking',
'elvis_store')
    ...
    mysqli_close($dbc);
?>
```

Não acontece nada se a declaração if falhar ao executar o código de ação, e é por isso que é gerada uma página em branco quando algum campo do formulário tenha ficado em branco.

Precisamos avisar a Elmer que houve um problema, de preferência dizendo a ele quais campos do formulário ficaram em branco, para que ele possa tentar digitar novamente a mensagem.

## a cláusula else



Sem problemas. Basta colocar uma declaração echo depois da chave que fecha a declaração if.

### **Isso não vai funcionar, porque o código depois da declaração if sempre será executado.**

Colocar a declaração echo depois da declaração if só significa que a primeira vai rodar depois da segunda, mas ela sempre roda – independentemente do resultado de if. Não é disso que precisamos. Precisamos que a declaração echo mostre uma mensagem de erro apenas se a condição de teste da declaração if for false. Você pode expressar esta lógica da seguinte forma:

**SE** Subject contiver texto **E** Body contiver texto

✓ **ENTÃO** enviar email

✗ **CASO CONTRÁRIO** exibir mensagem de erro

A declaração if oferece uma cláusula opcional, else, que executa algum código no caso de a condição de teste ser false. Assim, a nossa mensagem de erro é colocado dentro de uma cláusula else, e portanto, só é executado quando um dos campos do formulário for deixado em branco. Basta colocar a palavra else depois da declaração if, e depois colocar o respectivo código de ação dentro de chaves

A cláusula else  
começa logo  
depois da chave  
que fecha a  
declaração if.

```
if ((!empty($subject)) && (!empty($text))) {  
    ...  
}  
else {  
    echo 'Você esqueceu do assunto e/ou do corpo da  
    mensagem.<br />';  
}
```

Aqui você insere o código  
que envia as mensagens  
de email.

Assim como o  
código de ação em  
if, o código de else  
também é colocado  
dentro de chaves.

Este código só é executado se a  
declaração if resultar em false.

A cláusula else executa algum código quando uma condição de teste if resulta em false.



Segue abaixo o novo código para o script sendemail.php de Elmer, usando declarações if e cláusulas else para fornecer feedback, mas partes do código se perderam. Coloque os ímãs corretamente no lugar do código que está faltando.

```

1 // Nós sabemos que tanto $subject e $text estão
   // faltando
2 // $subject está vazia
3 // Tudo está certo, enviar emails
4 // $text está vazia
5 if (!empty($subject)) && (!empty($text))
6 {
7     echo 'Email enviado com sucesso!';
8 } else {
9     if (!empty($text))
10    {
11        echo 'Você esqueceu do assunto.<br />';
12    } else {
13        if (!empty($subject))
14        {
15            echo 'Você esqueceu do corpo da mensagem.<br />';
16        }
17    }
18 } else {
19
20     ...
21     while ($row = mysqli_fetch_array($result)) {
22         $to = $row['email'];
23         $first_name = $row['first_name'];
24         $last_name = $row['last_name'];
25         $msg = "Caro(a) $first_name $last_name, \n$text";
26         mail($to, $subject, $msg, 'From:' . $from);
27         echo 'Email enviado para: ' . $to . '<br />';
28     }
29     mysqli_close($dbc);
30 }
31 ?>

```

Diagrama de fluxo de dados:

- Ímã 1: { (abre bloco)
- Ímã 2: } (fecha bloco)
- Ímã 3: && (operador lógico AND)
- Ímã 4: ( ( abre parêntese)
- Ímã 5: ) (fecha parêntese)
- Ímã 6: if (condição)
- Ímã 7: if (condição)
- Ímã 8: empty(\$text) (verificação de vazio)
- Ímã 9: empty(\$subject) (verificação de vazio)
- Ímã 10: empty(\$subject) (verificação de vazio)
- Ímã 11: empty(\$subject) (verificação de vazio)
- Ímã 12: empty(\$text) (verificação de vazio)



## Exercício

### Solução

```
<?php
$from = 'elmer@makemeelvis.com';
$subject = $_POST['subject'];
$text = $_POST['elvismail'];
```

O if externo checa se o assunto e o texto estão vazios. Se não há apenas 3 opções ambas estão preenchidos o assunto está faltando, ou o texto do corpo do email está faltando

```
if [empty($subject) && empty($text)] {
    // Nós sabemos que tanto $subject e $text estão faltando
```

}

else {

```
    if ([empty($subject) || empty($text)]) {
```

// Nós sabemos que um dos dois, \$subject ou \$text, está faltando - vamos descobrir qual deles

```
        if ([empty($subject)]) {
```

// \$subject está vazia

```
            echo 'Você esqueceu do assunto..<br />';
```

}

else {

// \$text está vazia

```
            echo 'Você esqueceu do corpo da mensagem..<br />';
```

}

}

else {

// Tudo está certo, enviar emails

```
    ...
    while ($row = mysqli_fetch_array($result)) {
```

```
        $to = $row['email'];
```

```
        $first_name = $row['first_name'];
```

```
        $last_name = $row['last_name'];
```

```
        $msg = "Caro(a) $first_name $last_name, \n$text";
```

```
        mail($to, $subject, $msg, 'From:' . $from);
```

```
        echo 'Email enviado para: ' . $to . '<br />';
```

}

```
    mysqli_close($dbc);
```

}

?

Neste ponto passamos por todos as possibilidades, então, sabemos que ambos campos do formulário contém valores.



**É sempre uma boa ideia simplificar código sempre que possível, especialmente, código com muitos níveis de aninhamento.**

Uma quantidade muito grande de cláusulas `else` com declarações `if` aninhadas pode tornar o seu código difícil de ler. Talvez isso não fizesse diferença caso nós nunca mais tivéssemos que olhar esse código, mas isso é bem improvável. Se algum dia precisarmos modificar o formulário para adicionar um novo campo, validá-lo seria mais difícil do que precisa ser, porque seria muito difícil ler o código e entender onde as modificações precisam ser colocadas.

## o código if limpo

faxineiro do

## SINTA-SE como código IF Solução



Sua tarefa é interpretar o papel do código IF e limpar o código dos IF's e ELSE's aninhados. Reescreva o código de modo a se livrar do aninhamento, mas certifique-se de que ela ainda funcionará corretamente.

```
if (empty($subject) && empty($text)) {  
    echo 'Você esqueceu do assunto e do corpo da mensagem.<br />';  
} else {  
    if (empty($subject) || empty($text)) {  
        if (empty($subject)) {  
            echo 'Você esqueceu do assunto.<br />';  
        } else {  
            echo 'Você esqueceu do corpo da mensagem.<br />';  
        }  
    } else {  
        // Tudo está certo, enviar emails  
    }  
}
```

Aqui, estamos  
testando se  
tanto a variável  
\$subject quanto  
a \$text estão  
vazias.

→ if (empty(\$subject) && empty(\$text)) {

echo 'Você esqueceu do assunto e do corpo da mensagem.<br />';

Aqui, estamos  
testando se  
\$text está vazia  
e \$subject está  
não-vazia.

if (empty(\$subject) || empty(\$text)) {

Este código verifica se \$subject  
está vazia e se \$text está não-  
vazia.

→ if ((empty(\$subject)) && empty(\$text)) {

echo 'Você esqueceu do assunto.<br />';

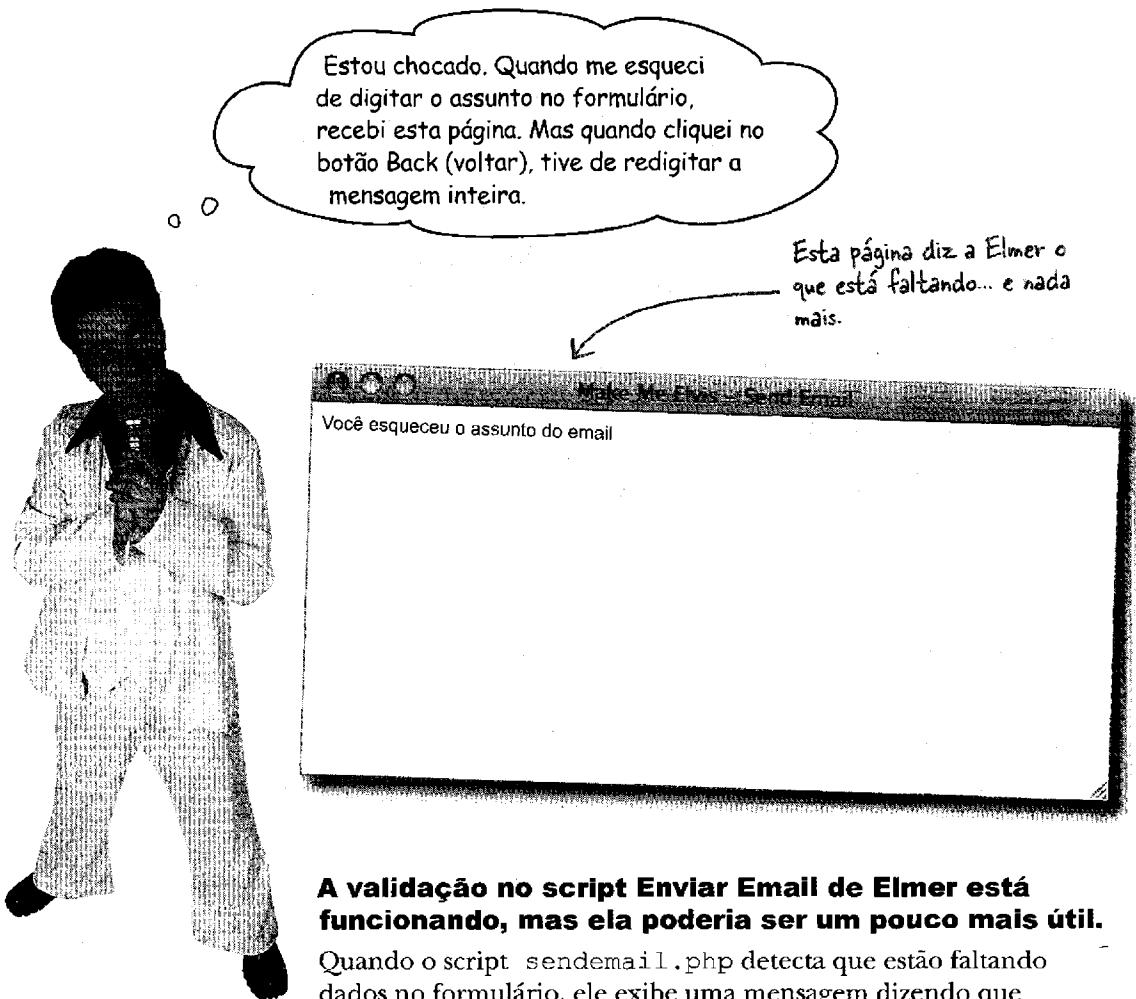
Se não estivéssemos usando o AND  
(&&) para isolar o texto não-vazio de  
\$subject e \$text, poderíamos acabar  
recendo uma mensagem de erro  
desnecessária. O mesmo acontece para  
\$subject e \$text não-vazias.

→ if ((!empty(\$subject)) && (!empty(\$text))) {

// Tudo está certo, enviar emails.

E aqui, estamos  
testando se  
nem \$subject  
nem \$text  
estão vazias.

O operador NOT (!) verifica se  
\$subject e \$text estão não-vazias.



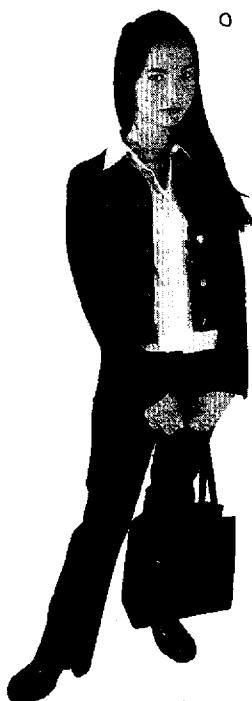
**A validação no script Enviar Email de Elmer está funcionando, mas ela poderia ser um pouco mais útil.**

Quando o script `sendemail.php` detecta que estão faltando dados no formulário, ele exibe uma mensagem dizendo que alguma informação está faltando, mas só isso. Não há um link para voltar ao formulário original, por exemplo. E, pior ainda, quando Elmer navega de volta para o formulário original, as informações que digitou não estão mais lá. Ele tem de redigitar o assunto e o corpo da sua mensagem.

**PODER DO CÉREBRO**

O que você faria para melhorar o tratamento de erros do script Enviar Email, de modo a torná-lo mais útil?

## **regenerando o código HTML do formulário**



O O

Seria legal exibir o formulário junto com a mensagem de erro. Será que não poderíamos simplesmente exibir o formulário com echo, no caso de o assunto e o corpo do texto estarem vazios?

### **Exibir o formulário definitivamente seria útil, uma vez que assim Elmer não precisaria navegar de volta à página original.**

Assim, além de apresentar uma mensagem de erro quando um dos campos do formulário é deixado em branco, nós precisamos também “ressuscitar” o código HTML a partir do PHP, usando echo para enviá-lo ao navegador. Este código mostra que o PHP é capaz de gerar código HTML razoavelmente complexo:

Este código PHP gera o formulário HTML inteiro, começando pela tag <form>.

```
echo '<form method="post" action="sendemail.php">';  
echo ' <label for="subject">Subject of email:</label><br />';  
echo ' <input id="subject" name="subject" type="text" ' .  
     'size="30" /><br />';  
echo ' <label for="elvismail">Body of email:</label><br />';  
echo ' <textarea id="elvismail" name="elvismail" rows="8" ' .  
     'cols="40"></textarea><br />';  
echo ' <input type="submit" name="submit" value="Submit" />';  
echo '</form>';
```

Esta indentação  
não é estritamente  
necessária, mas ela  
ajuda a visualizar  
a estrutura do  
código HTML  
original.

Uma vez que o código HTML  
tem aspas duplas por toda a  
parte, é mais fácil usar aspas  
simples para escrever strings de  
código HTML em PHP.

Se você estiver achando este código um pouco caótico, é porque ele é mesmo. Só porque você **pode** fazer algo em PHP, não quer dizer que você deva fazê-lo. Neste caso, a complexidade adicional de se enviar todo esse código HTML através de echo é um problema. O código é tão grande que gerá-lo através do PHP, com a declaração echo, realmente não é uma boa opção...

## Facilite a entrada e a saída do PHP

Às vezes é fácil esquecer que um script PHP é na verdade apenas uma página HTML capaz de armazenar código PHP. Qualquer código, em um script PHP, que não esteja dentro das tags <?php e ?> é presumido como sendo HTML. Isso significa que você pode fechar um bloco de código PHP e reverter ao HTML conforme necessário, e depois começar de novo com um novo bloco de código PHP. Esta é uma técnica extremamente útil para se transmitir um pedaço de código HTML que seja grande demais para ser gerado através de declarações echo do PHP... como por exemplo, o código do nosso formulário Enviar Email..

```
<?php
Esta tag      $from = 'elmer@makemeelvis.com';
?> fecha o    $subject = $_POST['subject'];
bloco PHP,    $text = $_POST['elvismail'];
nos levando   if (empty($subject) && empty($text)) {
de volta ao/  // Nós sabemos que tanto $subject e $text estão faltando
HTML.         echo 'Você esqueceu do assunto e do corpo da mensagem..<br />';
?>
```

Você pode fechar e abrir blocos de código PHP para enviar pedaços de código HTML de dentro do script PHP.

O formulário é codificado como HTML normal, uma vez que este código fica fora das tags PHP.

```
<form method="post" action="sendemail.php">
  <label for="subject">Assunto do email:</label><br />
  <input id="subject" name="subject" type="text" size="30" /><br />
  <label for="elvismail">Corpo da mensagem:</label><br />
  <textarea id="elvismail" name="elvismail" rows="8" cols="40">
    </textarea><br />
  <input type="submit" name="submit" value="Submit" />
</form>
```

A tag <?php inicia um novo bloco de código PHP. Uma vez que ainda estamos dentro da ação de if, temos de fechar a declaração if antes de continuarmos.

```
<?php
}
if (empty($subject) && (!empty($text))) {
  echo 'Você esqueceu do assunto.<br />';
}

if ((!empty($subject)) && empty($text)) {
  echo 'Você esqueceu do corpo da mensagem.<br />';
}

if ((!empty($subject)) && (!empty($text))) {
  // Código para enviar o email
  ...
}
```

Uma vez que ainda estamos dentro da ação de if, o código HTML só é exibido se ambos os campos do formulário estiverem vazios.

Escreva abaixo qualquer coisa que você considere que esteja limitando este código. Como você o consertaria?

*• O erro é quando eu a tento enviar isso*

evitando code duplicado com uma flag

## Use um flag para evitar código duplicado

O problema com o código anterior é que ele precisa sair do PHP e repetir o código do formulário em três lugares diferentes (um para cada validação). Nós podemos usar uma variável true/false, conhecida como **flag**, para nos mantermos a par sobre a necessidade ou não de enviar o formulário. Vamos chamá-la de \$output\_form. Então, poderemos verificar a variável mais adiante no código, e exibir o formulário caso ela seja true.

Assim, nós começamos o script com \$output\_form definida como false, e só a modificamos para true se algum campo estiver vazio e precisarmos exibir o formulário:

initialize \$output\_form **como** false

Definir \$output\_form inicialmente como false significa que o formulário não será exibido a não ser que haja um problema de validação que faça o valor se modificar.

**SE** Subject está vazio **E** Body está vazio

Estas mensagens de erro são ligeiramente diferentes, para indicar os(s) campo(s) específico(s) que esteja(m) vazio(s).

✓ **ENTÃO** exibir mensagem de erro, definir \$output\_form **como** true

**SE** Subject estiver vazio **E** Body **NÃO** estiver vazio

Se um campo for deixado em branco, a variável \$output\_form é definida como true, mas o formulário não é exibido... ainda!

✓ **ENTÃO** exibir mensagem de erro, definir \$output\_form **como** true

**SE** Subject **NÃO** estiver vazio **E** Body estiver vazio

✓ **ENTÃO** exibir mensagem de erro, definir \$output\_form **como** true

Se ambos os campos contiverem dados, vá em frente e envie os emails.

**SE** Subject **NÃO** estiver vazio **E** Body **NÃO** estiver vazio

✓ **ENTÃO** enviar emails

**SE** \$output\_form **for** true

✓ **ENTÃO** exibir formulário

Finalmente, verificamos a variável \$output\_form para ver se o formulário precisa ser exibido. De uma forma ou de outra, só precisamos que o código HTML apareça uma vez.

# Codifique o formulário HTML apenas uma vez

Transformar o novo código de validação em código PHP envolve a criação e a inicialização da variável \$output\_form, e depois garantir que ela vá ser definida ao longo de todo o código de validação. Mais importante é a nova declaração if ao final do código, a qual só exibe o formulário se \$output\_form for definida como true.

```
<?php
    $from = 'elmer@makemeelvis.com';
    $subject = $_POST['subject'];
    $text = $_POST['elvismail'];
    $output_form = false; // Nós criamos nossa
                          // nova variável aqui, e a
                          // definimos inicialmente
                          // como false.

    if (empty($subject) && empty($text)) {
        // Nós sabemos que tanto $subject e $text estão faltando
        echo 'Você esqueceu do assunto e do corpo da mensagem.<br />';
        $output_form = true; // Define a variável como true se tanto
                           // $subject quanto $text estiverem vazias,
                           // para que o formulário seja exibido.

    if (empty($subject) && (!empty($text))) {
        echo 'Você esqueceu do assunto.<br />';
        $output_form = true; // Também define a variável como
                           // true se $subject estiver vazia.

    if ((!empty($subject)) && empty($text)) {
        echo 'Você esqueceu do corpo da mensagem.<br />';
        $output_form = true; // E define a variável como true
                           // se $text estiver vazia.

    if ((!empty($subject)) && (!empty($text))) {
        // Código para enviar o email
        ...
    if ($output_form) // Esta declaração if verifica a
                     // variável $output_form e exibe o
                     // formulário caso ela seja true.
    ?>

<form method="post" action="sendemail.php">
    <label for="subject">Assunto do email:</label><br />
    <input id="subject" name="subject" type="text" size="30" /><br />
    <label for="elvismail">Corpo da mensagem:</label><br />
    <textarea id="elvismail" name="elvismail" rows="5" cols="40"></textarea><br />
    <input type="submit" name="submit" value="Submit" />
</form>
```

Não se esqueça de voltar ao código PHP e fechar a declaração if.

O código HTML só aparece uma vez desde que resolvemos colocar toda a lógica da sua exibição em uma só variável, \$output\_form.

Ao tornar o código HTML dependente de uma declaração if, nós evitamos duplicação de código no nosso script.

Aqui, já saímos do código PHP, mas tudo que estiver antes da } de fechamento ainda é considerado como parte da ação de if - neste caso, é o código HTML do formulário.

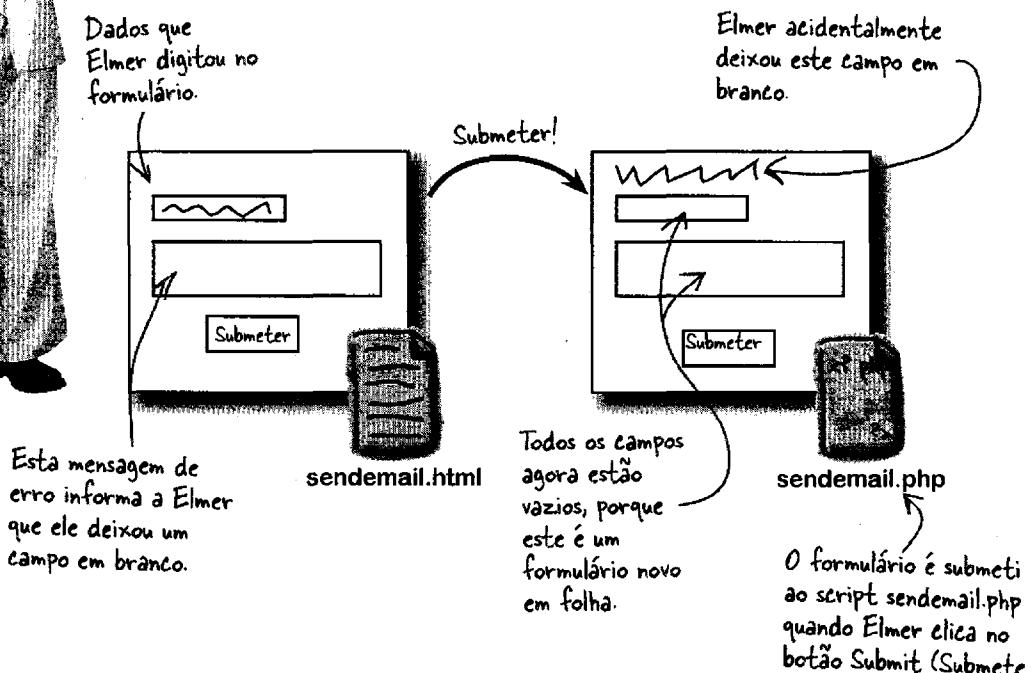
**os dados do formulário continuam desaparecidos**



O novo formulário é melhor, mas eu ainda tenho de refazer o que eu já havia digitado corretamente, o que é muito irritante.

### **O HTML sozinho não é capaz de preservar dados de formulários.**

Quando Elmer submete o formulário Enviar Email com um campo vazio, o script sendemail.php captura o erro e gera um novo formulário. Mas o novo formulário é puro código HTML, o qual não tem como saber nada sobre quaisquer dados que Elmer possa ter digitado anteriormente. Assim, estamos gerando um formulário totalmente novo, como parte da validação, o que apaga quaisquer dados que Elmer possa ter digitado



Péssimo. Não conseguimos contornar o fato de que um novo formulário terá de ser gerado no script PHP. Nós precisamos de uma forma de preservar quaisquer dados que Elmer já tiver digitando, colocando-os no novo formulário para que ele possa se concentrar em preencher apenas o campo que accidentalmente deixou vazio...



## Aponte seu lápis

Esboce aqui como você gostaria que fosse o comportamento do formulário de Elmer depois que ele o submete com apenas o primeiro campo preenchido. Em seguida, escreva como você acha que cada um dos arquivos (HTML e PHP) deve ser alterado para executar esta nova funcionalidade.

A diagram of a form window with a dark border. Inside, there are two rectangular input fields stacked vertically, followed by a horizontal "Submitter" button at the bottom.



sendemail.php

erro... entao aparece...  
sem o corretor é o  
formulário entro...  
não... em... a... de... domínio



sendemail.html

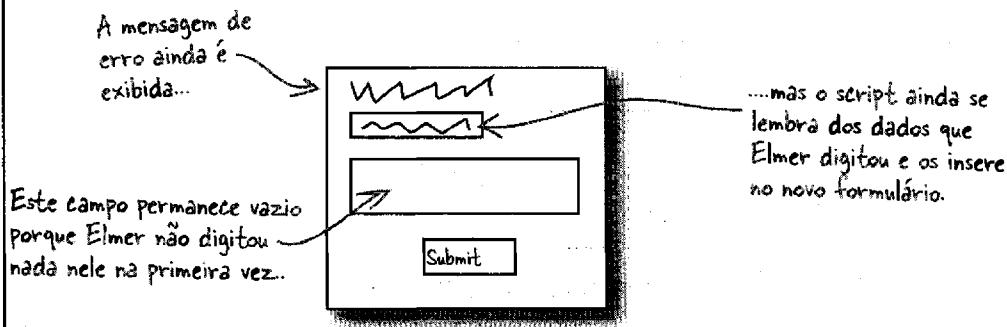
formulário

.....  
.....  
.....

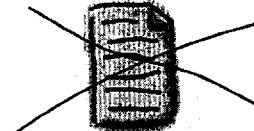
faça o formulário "permanente"

## Aponte seu lápis

Esboce aqui como você gostaria que fosse o comportamento do formulário de Elmer depois que ele o submete com apenas o primeiro campo preenchido. Em seguida, escreva como você acha que cada um dos arquivos (HTML e PHP) deve ser alterado para executar esta nova funcionalidade.



sendemail.php



sendemail.html

O script PHP assume a tarefa de exibir o formulário, antes e depois da submissão. É uma vez que o script tem acesso a quaisquer dados que tenham sido digitados, ele pode inserir esses dados no novo formulário, quando este for gerado. Isto resolve o problema de Elmer ter que redigitar dados que já tenha preenchido.

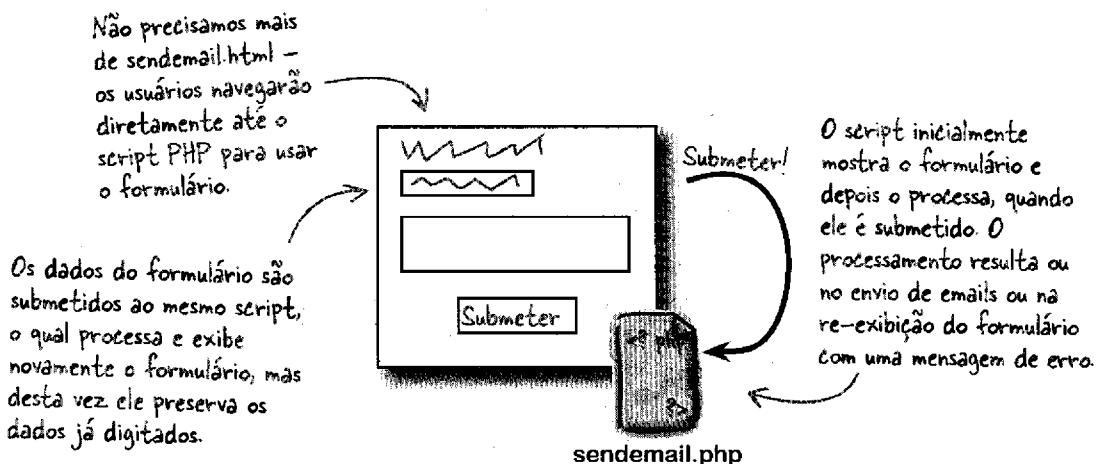
Se exibirmos o formulário inteiramente através do script PHP, podemos eliminar a página HTML, uma vez que o script PHP irá exibir e processar o formulário. Ao fazer isso, o script PHP pode acessar e usar quaisquer dados digitados no formulário, o que é impossível com código HTML puro.

## Um formulário que referencia a si mesmo

Como é possível ter o formulário Enviar Email sem o arquivo sendemail.html? A resposta é que nós não estamos na verdade eliminando o código HTML, estamos apenas transferindo-o para o script PHP. Isso é possível devido ao fato de que um script PHP pode conter código HTML, como uma página web normal. Assim, nós podemos estruturar nosso script de modo que ele não somente processe o formulário após a submissão, mas também o apresente desde o início, que é o que sendemail.html estava fazendo.

A chave para o script sendemail.php ser capaz de desempenhar o papel de sendemail.html é a ação do formulário. Uma vez que o próprio script agora contém o formulário HTML, a ação nos leva de volta ao script... um formulário que referencia a si mesmo.

**Um formulário HTML que faça parte de um script PHP que o processe é conhecido como auto-referente.**



Para entender o que está acontecendo aqui, pense na primeira vez que Elmer visita a página (script). Um formulário vazio é gerado como código HTML, e é exibido. Elmer preenche um campo do formulário e clica em Submit. O script processa o seu próprio formulário e exibe uma mensagem de erro se quaisquer dados estiverem faltando. Mais importante, o script exibe o formulário novamente, mas desta vez inclui quaisquer dados que Elmer já tenha digitado. Quando um formulário é inteligente o suficiente para se lembrar de dados inseridos em submissões anteriores, ele é conhecido como permanente... os dados permanecem nele!

Os formulários permanentes se lembram dos dados que o usuário já tiver digitado corretamente.

### PODER DO CÉREBRO

Como você acha que podemos modificar a aplicação de Elmer para tornar os campos do formulário permanentes?

faça a ação do formulário auto-referente

## Aponte a ação do formulário para o script

Como já vimos algumas vezes, o atributo action da tag <form> é o que conecta um formulário a um script PHP que o processa. Para permitir que o formulário de Elmer processe a si mesmo, o que é o primeiro passo para torná-lo permanente, você direciona a ação do formulário para sendemail.php. Na verdade, o formulário já tem o seu atributo action direcionado para o script:

O atributo action da tag <form> conecta o formulário ao script que o processa – neste caso o mesmo script sendemail.php que armazena o formulário.

```
<form action = "sendemail.php" method="post">
```

Esta é uma tag <form> padrão, que usa POST para submeter os dados do formulário ao script.

Este código funciona, presumindo-se que você nunca renomeie o script e se esqueça de atualizar o código. Mas há uma forma melhor, que funciona em qualquer caso, porque ela não depende de um script específico. Trata-se da variável superglobal interna do PHP `$_SERVER['PHP_SELF']`, que armazena o nome do script atual. Você pode substituir a URL do script, na ação do formulário, por `$_SERVER['PHP_SELF']`, e então jamais terá de se preocupar em atualizar nada, se algum dia tiver de mudar o nome do script.

O único senão é que `$_SERVER['PHP_SELF']` é código PHP, o que significa que você tem de enviar o seu valor através do comando echo, para que ele seja exibido como parte do código HTML, desta forma::

Em vez de escrever o nome do script permanentemente no código, podemos instruí-lo a referenciar a si mesmo, usando a superglobal `$_SERVER['PHP_SELF']`.

```
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
```

É claro que usar `$_SERVER['PHP_SELF']` em vez do nome do script não é um aprimoramento espetacular, mas é uma das pequenas coisas que você pode fazer para facilitar a manutenção posterior dos seus script.

`$_SERVER['PHP_SELF']` armazena o nome do script atual.



## TEST DRIVE

### Teste o novo script autorreferente, com a lógica de validação aprimorada.

Modifique o código do script sendemail.php de modo que ele use a variável \$output\_form para exibir o formulário seletivamente, como mostrado algumas páginas atrás. Modifique também o atributo action da tag <form> de modo a tornar o formulário autorreferente.

Você não precisa mais da página sendemail.html no seu servidor web, portanto, esteja à vontade para apagá-la. Em seguida, envie a nova versão do script sendmail.php para o seu servidor e abra-o em um navegador. Está tudo funcionando?

Por algum motivo, o script está mostrando uma mensagem de erro, embora o formulário não tenha nem sido submetido... nada bom.

E não é só isso: o formulário ainda não está preservando os dados digitados. Ainda temos trabalho a fazer!

Mas vamos começar pelo começo.  
Logo passaremos para a parte  
da preservação dos dados.

Escreva aqui o motivo pelo qual você acha que o script está mostrando uma mensagem de erro na primeira vez que o formulário é mostrado.

*Algumas das coisas que eu posso dizer é que o formulário não está preservando os dados digitados.*

*verifique o \$\_POST['submit']*

## Verifique se o formulário foi submetido

O problema é que o script não é capaz de distinguir entre o formulário estar sendo exibido pela primeira vez e estar sendo submetido com dados incompletos. Assim, o script relata o problema de dados faltando na primeira vez que o formulário é exibido, o que é confuso. A questão é: como podemos verificar se o formulário está sendo submetido? Se tivermos essa informação, podemos nos certificar de só validar os dados quando houver uma submissão.

Você se lembra de como os dados são armazenados no array `$_POST`, quando um formulário é submetido usando-se o método POST? Se o formulário ainda não tiver sido submetido, então o array `$_POST` ainda não terá sido preenchido com quaisquer dados. Ou, colocando de outra forma, o array `$_POST` ainda não terá sido definido. Alguma sugestão de qual função poderíamos chamar para ver se o array `$_POST` está definido?

A função `isset()` verifica se uma variável foi definida.

```
if (isset($_POST['submit'])) {  
    ...  
}
```

Qualquer código aqui só será executado se o formulário tiver sido submetido.

Este deve ser o mesmo nome da tag `<input>` usada com o seu botão Submit.

A superglobal `$_POST` nos permite verificar se um formulário foi submetido.

Uma vez que todo formulário tem um botão Submit, uma forma fácil de se verificar se o formulário foi submetido é ver se existem dados em `$_POST` para o botão Submit. Os dados são apenas o rótulo do botão, o que não é importante. O importante é simplesmente a existência de `$_POST['submit']`, o que nos diz que o formulário já foi submetido. Apenas certifique-se de que o texto 'submit' seja o mesmo do atributo id do botão Submit, no código do formulário.

## não existem Perguntas Ídiotas

**P:** Como o conhecimento de se o formulário foi submetido ou não nos impede de exibir mensagens de erro de validação acidentais?

**R:** O motivo pelo qual as mensagens de erro estão sendo mostradas incorretamente é que o script não consegue distinguir entre o formulário ser submetido e ser exibido pela primeira vez. Assim, precisamos de uma maneira de saber se esta é a primeira vez que o formulário está sendo mostrado, em cujo caso é perfeitamente OK ter campos vazios – não se trata de um erro. Nós só devemos validar os campos se o formulário tiver sido submetido; portanto, a capacidade de detectar a submissão do formulário é muito importante.

**P:** Então por que não verificamos se existem dados diretamente nos campos do formulário, em vez de no botão Submit

**R:** Seria perfeitamente aceitável verificar `$_POST['subject']` ou `$_POST['elvismail']`, mas apenas para este formulário em particular. Uma vez que todo formulário tem um botão Submit que pode ser consistentemente chamado de submit, verificar `$_POST['submit']` lhe fornece um modo confiável de checar a submissão do formulário em todos os seus scripts.



## O script Enviar Email em detalhes

```

<?php
if (!isset($_POST['submit'])) { ←
    $from = 'elmer@makemeelvis.com';
    $subject = $_POST['subject'];
    $text = $_POST['elvismail'];
    $output_form = false;

    if (empty($subject) && empty($text)) {
        // Nós sabemos que tanto $subject e $text estão faltando
        echo 'Você esqueceu do assunto e do corpo da mensagem.<br />';
        $output_form = true;
    }

    if (empty($subject) && (!empty($text))) {
        echo 'Você esqueceu do assunto.<br />';
        $output_form = true;
    }

    if ((!empty($subject)) && empty($text)) {
        echo 'Você esqueceu do corpo da mensagem.<br />';
        $output_form = true;
    }

    if ((!empty($subject)) && (!empty($text))) {
        // Código para enviar o email
        ...
    } ← Esta chave fecha a primeira
    else { ← declaração if, que nos diz se o
        $output_form = true; ← formulário foi submetido.

        if ($output_form) { ← Caso o formulário jamais tenha sido
            submetido, nós definitivamente precisamos
            mostrá-lo!
        }
    }
}
?>

<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
    <label for="subject">Assunto do email:</label><br />
    <input id="subject" name="subject" type="text" size="30" /><br />
    <label for="el_vismail">Corpo da mensagem:</label><br />
    <textarea id="elvismail" name="elvismail" rows="8" cols="40"></
        textarea><br />
    <input type="submit" name="submit" value="Submit" />
</form>

<?php
}
?>

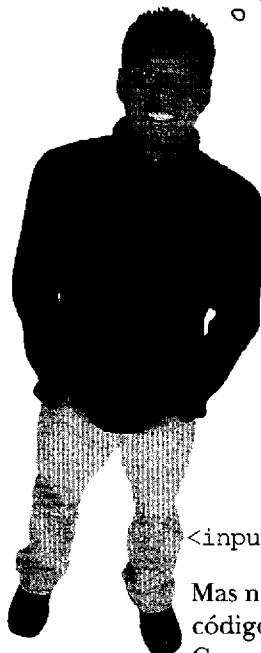
```

Nós verificamos o valor de `$_POST['submit']`. Se o formulário jamais tiver sido submetido, ela estará indefinida.

Esta chave fecha a primeira declaração if, que nos diz se o formulário foi submetido.

Caso o formulário jamais tenha sido submetido, nós definitivamente precisamos mostrá-lo!

**faça os campos dos formulários “permanentes”**



Legal. Então agora nós podemos detectar a submissão do formulário e mostrar as mensagens de erro corretamente. Mas ainda não tornamos os campos do formulário permanentes, certo?

**É verdade. Detectar a submissão do formulário é importante, mas ainda precisamos de uma forma para manter os dados permanentes.**

Saber se o formulário foi submetido é uma parte importante de torná-lo permanente, mas não é a única. A parte que está faltando é a que toma quaisquer dados que tenham sido submetidos e os coloca de volta no formulário, quando este é re-exibido. Você pode definir um campo de input do formulário usando o atributo value da tag HTML <input>. Por exemplo, o seguinte código pré-define o valor de um campo de input usando o atributo value:

Este valor está escrito permanentemente no código – ele é sempre o mesmo, a cada vez que o formulário é mostrado..

```
<input name="subject" type="text" value="Introdução de Outono!"/>
```

Mas não queremos escrever um valor específico de forma permanente no código. O que queremos é inserir a informação retirada de uma variável PHP. Como fazer isso? Lembre-se de que usamos echo para gerar código HTML dinamicamente a partir de PHP, em outras situações. Neste caso, podemos usar echo para gerar um valor para o atributo value a partir de uma variável PHP, desta forma::

Uma vez que estamos entrando no “modo PHP” para enviar a variável, temos de usar uma tag <?php.

A variável é enviada usando-se a familiar declaração echo.

```
<input name="subject" type="text" value="php echo $subject; ?"/>
```

Para um campo de input de texto, nós enviamos os dados permanentes que se encontram dentro das tags <textarea> e /<textarea>, em vez de usar o atributo value.

E para voltar ao HTML, nós fechamos o código PHP com a tag ?>.

Então, o formulário de Elmer pode ser modificado de forma semelhante, para tirar proveito dos dados permanentes:

```
<form method="post" action="php echo $_SERVER['PHP_SELF']; ?&gt;&gt;
  &lt;label for="subject"&gt;Assunto do email:&lt;/label&gt;&lt;br /&gt;
  &lt;input id="subject" name="subject" type="text" size="30"
        value="<?php echo $subject; ?&gt;"/&gt;&lt;br /&gt;
  &lt;label for="elvismail"&gt;Assunto do email:&lt;/label&gt;&lt;br /&gt;
  &lt;textarea id="elvismail" name="elvismail" rows="8" cols="40"&gt;
    &lt;?php echo $text; ?&gt;&lt;/textarea&gt;&lt;br /&gt;
  &lt;input type="submit" name="submit" value="Submit" /&gt;
&lt;/form&gt;</pre
```



## TEST DRIVE

### Verifique se os dados de Elmer realmente estão permanecendo.

Modifique o código de sendemail.php de modo que ele verifique em `$_POST` se o formulário foi submetido, e adicione também o código echo para tornar os campos permanentes. Envie a nova versão script para o seu servidor web e abra-o em um navegador. Experimente diferentes valores para os campos do formulário, deixando um ou ambos em branco, e submeta-o algumas vezes.

Cara, foi burrice minha deixar o corpo da mensagem em branco. Felizmente, nunca mais vou fazer isso, já que o formulário está atento para esse erro. E também nunca mais terei de ficar redigitando os mesmos dados.

**MAKEMEELVIS.COM**

Private: For Elmer's Use ONLY  
Write and send an email to mailing list members.

You forgot the email body text.  
Subject of email:  
Fall Clearance!  
Body of email:

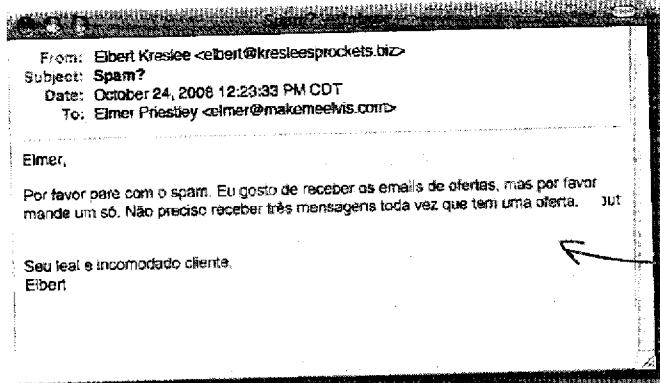
Submit

O script de Enviar Email agora mostra uma mensagem de erro quando Elmer deixa um campo do formulário em branco, mas ele mantém os dados digitados anteriormente.

*quando bons DELETEs viram maus*

## Alguns usuários ainda estão insatisfeitos

A validação do formulário nos ajudou bastante a lidar com os clientes insatisfeitos de Elmer, particularmente aqueles que estavam recebendo emails em branco. Mas nem todos estão felizes. Parece que algumas pessoas estão recebendo emails duplicados... lembra-se deste cara, do início do capítulo?

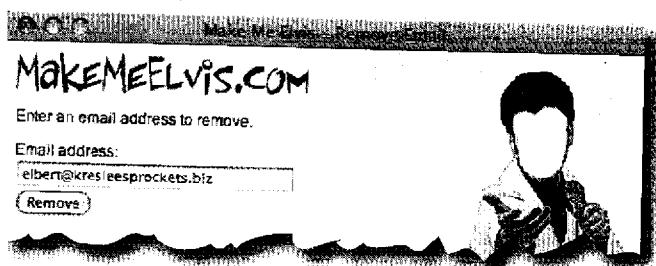


Este cliente está frustrado porque está recebendo múltiplas cópias dos emails de Elmer.

Elmer sabe que não digitou a mesma mensagem mais de uma vez, o que o leva a suspeitar que talvez alguns usuários accidentalmente tenham se inscrito na sua lista mais de uma vez. Sem problemas, basta usar a página/script de Remover Email do capítulo anterior para remover o usuário, certo?

Infelizmente, não é tão simples assim. Remover Elbert usando o endereço de email dele fará com que ele seja completamente apagado da tabela email\_list, e com isso ele nunca mais receberá nenhum email de Elmer. Precisamos de uma maneira de apagar apenas as linhas duplicadas de Elbert na tabela, deixando uma.

Usar a página de Remover Email do capítulo anterior removeria inteiramente o cliente do banco de dados de Elmer, e não é isso o que queremos.



 PODER DO CÉREBRO

Como Elmer poderia apagar todas as linhas da sua tabela que tiverem endereços de email idênticos, exceto uma?



Hmm. O problema é que há linhas repetidas na tabela, mas não temos nenhuma forma de distingui-las umas das outras. Sem uma forma de isolá-las, qualquer DELETE que tentarmos usar irá apagá-las todas.

**Joe:** Talvez o nosso formulário Adicionar Email devesse verificar os endereços de email antes de adicionar novos usuários. Isso consertaria o problema, certo?

**Frank:** Excelente ideia.

**Jill:** Sim, isso resolveria o problema para o futuro, mas não nos ajuda a lidar com endereços de email duplicados que já estejam no banco de dados.

**Frank:** Certo. E se tentássemos usar uma coluna diferente da tabela para apagar as linhas extras, como por exemplo, last\_name?

**Jill:** Também pensei nisso, mas usar um sobrenome pode acabar sendo pior do que um endereço de email. E se quiséssemos apagar alguém chamado John Smith da nossa lista, usando o seguinte código SQL:

```
DELETE FROM email_list WHERE last_name = 'Smith'
```

**Joe:** Isso não apagaria apenas John Smith da tabela; estariamos apagando também Will Smith, Maggie Smith, Emmitt Smith...

**Frank:** É, isso não seria bom. A probabilidade de sobrenomes se repetirem em várias linhas é maior do que a dos endereços de email, e primeiros nomes seriam ainda piores. Poderíamos perder dezenas de linhas com uma simples consulta.

**Jill:** Exatamente. Não podemos nos arriscar a usar uma cláusula WHERE que apague linhas que queremos manter. Precisamos ter certeza de que podemos especificar apenas aquelas que queremos remover.

**Joe:** Então o que fazer? Não podemos usar email, last\_name nem first\_name em nossa cláusula WHERE.

**Frank:** E não temos outras colunas para usar, na nossa tabela. Parece que estamos sem sorte.

**Jill:** Não necessariamente. O que realmente precisamos é de algo para tornar cada linha da tabela identificável individualmente – assim poderíamos especificar linhas sem nenhum problema. E só porque não temos atualmente uma coluna com um valor único para cada linha, não significa que não podemos adicionar uma.

**Joe:** Uma nova coluna? Mas já temos uma estrutura para a nossa tabela.

**Frank:** Sim, mas o que temos não está atendendo às nossas necessidades. Você tem razão, teria sido melhor se tivéssemos previsto esta situação antes, para que pudéssemos ter elaborado nossa tabela de modo a evitar este tipo de problemas, mas não é tarde demais para consertar o que temos.

**Joe:** OK, mas que nome daremos à nossa nova coluna? E que dados colocaremos nela?

**Jill:** Bem, uma vez que o propósito seria identificar individualmente cada linha da tabela, poderíamos chamá-la de identificador, ou apenas id, para abreviar.

**Frank:** Ótimo, e podemos preencher a coluna id com um número de identificação para cada linha, de modo que quando executarmos nosso DELETE, poderemos remover linhas com base em um número individual, em vez de em um endereço de email ou sobrenome.

**Joe:** Exatamente. É uma ótima ideia, não? Fico feliz de ter pensado nela.

*adicionando uma coluna com chave primária para uma tabela*

## As linhas da tabela precisam ser identificáveis individualmente

Parte de todo o processo de se armazenar algo em um banco de dados é que, posteriormente, você irá querer procurar esses dados e fazer algo com eles. Sabendo disso, é incrivelmente importante que cada linha de uma tabela seja identificável individualmente, o que significa que você pode acessar cada linha especificamente (e somente a linha em questão!). A tabela email\_list de Elmer faz uma perigosa assunção de que os endereços de email serão únicos. Isso funciona, desde que ninguém accidentalmente se inscreva na lista duas vezes, mas quando alguém o fizer (e alguém o fará, pode ter certeza!), o seu endereço de email será armazenado na tabela duas vezes... e acabou-se a unicidade!

### O que a tabela de Elmer contém agora:

id	first_name	last_name	email
	Denny	Bubbleton	denny@mightygumball.net
	Irma	Werlitz	iwer@aliensabductedme.com
	Elbert	Kreslee	elbert@kresleespockets.biz
	Irma	Kreslee	elbert@kresleespockets.biz

Mais de uma pessoa pode ter o mesmo primeiro nome, portanto, esta não é uma boa opção para coluna identificadora.

Nada na estrutura desta tabela garante identidade para as linhas.

E embora os endereços de email sejam quase sempre únicos, não podemos confiar que esse será sempre o caso.

O mesmo aqui, não podemos contar com sobrenomes para identificação.

Se você não tem uma coluna de valores realmente únicos em uma tabela, então deve criar uma. O MySQL lhe oferece uma forma de adicionar uma coluna de números inteiros individuais, o que também se chama de chave primária, para cada linha da sua tabela.

### O que a tabela de Elmer deveria conter:

id	first_name	last_name	email
1	Denny	Bubbleton	denny@mightygumball.net
2	Irma	Werlitz	iwer@aliensabductedme.com
3	Elbert	Kreslee	elbert@kresleespockets.biz
4	Irma	Kreslee	elbert@kresleespockets.biz

Precisamos de uma nova coluna que contenha um valor único para cada linha da tabela.

Agora que esta coluna contém um valor único, podemos ter certeza de que cada linha da nossa tabela será realmente única.

Os dados duplicados nas outras colunas não afetam mais a unicidade das linhas, porque a nova coluna id cuida disso.



Ei, gênio, você sabe que, se quisermos fazer uma modificação na estrutura de uma tabela, temos de usar um **DROP TABLE** e recriá-la do zero. Os dados de Elmer serão perdidos!

**É verdade que **DROP TABLE** destruiria os dados de Elmer. Mas o SQL tem outro comando que lhe permite fazer modificações em uma tabela existente, sem perda de dados.**

Trata-se de **ALTER TABLE**, e podemos usá-lo para criar uma nova coluna sem ter de excluir a tabela e destruir os seus dados. Eis o formato geral de uma instrução **ALTER TABLE** para se adicionar uma nova coluna à tabela:

```
ALTER TABLE nome_da_tabela ADD nome_da_coluna tipo_da_coluna
```

O nome da tabela a ser alterada.

O nome da coluna a ser adicionada.

O tipo de dados da nova coluna.

Nós podemos usar o comando **ALTER TABLE** para adicionar uma nova coluna à lista `email_list`, a qual chamaremos de `id`. Nós daremos à coluna `id` o tipo `INT`, uma vez que números inteiros são ideais para estabelecer a unicidade. Algumas outras informações são também necessárias, como revela este código:

```
ALTER TABLE email_list ADD id INT NOT NULL AUTO_INCREMENT FIRST,
ADD PRIMARY KEY (id)
```

O nome da tabela que queremos alterar.

Queremos adicionar (**ADD**) uma nova coluna, que chamaremos de `id`.

Isto diz ao servidor MySQL para adicionar 1 ao valor armazenado nesta coluna para cada nova linha inserida.

Este pequeno pedaço de código diz ao MySQL que a nova coluna `id` é a chave primária da tabela. Ficaremos mais sobre isso mais adiante!

O tipo de dados da coluna, `INT`.

FIRST instrui o MySQL a fazer com que a nova coluna seja a primeira da tabela. Isto é opcional, mas é uma boa prática colocar a sua coluna de identificação em primeiro lugar.

Esta instrução **ALTER TABLE** tem um monte de coisas acontecendo, e isso é porque as chaves primárias precisam ser criadas com características bastante específicas. Por exemplo, `NOT NULL` diz ao MySQL que é preciso haver um valor na coluna `id` – não se pode nunca deixá-la em branco. `AUTO_INCREMENT` descreve ainda mais as características da coluna `id`, fazendo com que ela receba automaticamente um valor numérico único a cada vez que uma nova linha é inserida. Como o nome sugere, `AUTO_INCREMENT` automaticamente adiciona 1 ao último valor `id` usado em uma linha, e coloca o resultado na coluna `id` da sua nova linha. Finalmente, `PRIMARY KEY` diz ao MySQL que cada valor da coluna `id` é único, mas existem mais coisas aí do que apenas a unicidade....

## As chaves primárias possibilitam a identificação individual

Uma chave primária é uma coluna que distingue, em uma tabela, cada linha como sendo única. Ao contrário das colunas normais, que também podem ser elaboradas de forma a serem únicas, apenas uma coluna pode ser designada como a chave primária. Isto cria a possibilidade de se fazerem consultas que precisem apontar uma ou mais linhas específicas.

Para garantir a unicidade das chaves primárias, o MySQL impõe uma série de restrições sobre a coluna declarada como PRIMARY KEY.

Você pode pensar nessas restrições como regras a serem seguidas, no seu trabalho com as chaves primárias:



Uma chave primária é uma coluna da sua tabela que torna cada linha única.

### As cinco regras das chaves primárias:

**Os dados de uma chave primária não podem ser repetidos.**



Nunca duas linhas podem ter os mesmos dados nas suas chaves primárias. Sem exceções – uma chave primária deve sempre ter valores únicos dentro de uma dada tabela.

**Uma chave primária deve sempre ter uma valor.**



Se uma chave primária fosse deixada vazia (NULL), ela poderia não ser única, porque outras linhas poderiam também ser NULL. Sempre defina as suas chaves primárias com valores únicos!

**A chave primária deve ser definida sempre que uma nova linha for inserida.**



Se houver a possibilidade de você inserir uma linha sem uma chave primária, você correria o risco de ter chaves NULL e linhas duplicadas na sua tabela, o que acabaria com todo o propósito.

**A chave primária deve ser o mais eficiente possível.** Uma chave primária deve conter apenas as informações de que precisa para ser única, e nada mais. É por isso que números inteiros rendem boas chaves primárias – eles permitem a unicidade sem exigir muito espaço de armazenamento.



**O valor de uma chave primária não pode ser modificado.**



Se houver a possibilidade de você modificar o valor da sua chave, você correria o risco de acidentalmente defini-la como um valor já usado.

Lembre-se, a chave precisa se manter única, a qualquer custo.

A coluna id na tabela de Elmer não tem dados repetidos, tem um valor diferente para cada linha, é automaticamente definida quando uma nova linha é inserida, é compacta e não se modifica. Perfeita!

id	first_name	last_name	email
1	Denny	Bubbleton	denny@mightygumball.net
2	Irma	Werlitz	iwer@aliensabductedme.com
...			



## TEST DRIVE

### **Altere a tabela de Elmer e tente inserir uma nova linha de dados com uma chave primária.**

Usando uma ferramenta MySQL, como o terminal MySQL ou a guia SQL de phpMyAdmin, digite a instrução ALTER TABLE para adicionar uma coluna de chave primária chamada id:

```
ALTER TABLE email_list ADD id INT NOT NULL AUTO_INCREMENT FIRST,
ADD PRIMARY KEY (id)
```

Agora insira um novo cliente no banco de dados, para ver se a coluna id é automaticamente definida na nova linha. Eis um exemplo de instrução INSERT para ser usada (repare que a chave primária não é mencionada):

```
INSERT INTO email_list (first_name, last_name, email)
VALUES ('Don', 'Draper', 'draper@sterling-cooper.com')
```

Finalmente, emita uma instrução SELECT para visualizar o conteúdo da tabela e ver a nova chave primária em toda a sua glória! Caso tenha se esquecido, eis aqui a instrução SELECT:

```
SELECT * FROM email_list
```

A nova coluna id é automaticamente incrementada, de modo a permanecer única com a adição da nova linha de dados.

```
File Edit Window Help Email
mysql> SELECT * FROM email_list;
+----+----+----+
| id | first_name | last_name | email           |
+----+----+----+
| 1  | Denny      | Bubbleton | denny@mightygumball.net |
| 2  | Irma       | Werlitz   | iwer@aliensabductedme.com |
| 3  | Elbert     | Kreslee   | elbert@kresleesprockets.biz |
| 4  | Irma       | Kreslee   | elbert@kresleesprockets.biz |
| 5  | Don        | Draper    | draper@sterling-cooper.com |
+----+----+----+
5 rows in set (0.0005 sec)
```

## conversa à três



OK, agora cada linha da tabela tem uma chave primária única. Como isso ajuda? Elmer ainda está deletando os usuários com base nos endereço de email.

**Joe:** O problema é que o usuário precisa especificar as linhas usando a chave primária, em vez do endereço de email.

**Frank:** É isso! Então só precisamos modificar o formulário para que o usuário digite a ID do cliente, em vez do endereço de email. Sem problemas!

**Jill:** Na verdade, há um problema sim. O usuário não tem como saber qual é a ID do cliente sem ter alguma forma de procurar essa informação no banco. Na verdade, o usuário não sabe nada sobre a estrutura do banco. Talvez o que precisamos aqui é de modificar o formulário, de forma que ele mostre todos os nomes e endereços de email em uma lista, com caixas de verificação ao lado de cada nome. Seria mais ou menos assim:

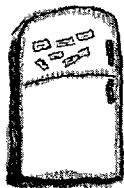
**Frank:** Bonito desenho, mas como isso ajuda Elmer a isolar um cliente para remoção, usando a ID?

**Joe:** Hmm. E se armazenássemos a ID do cliente no valor da caixa de verificação? A ID não ficaria visível ao usuário, mas o script teria acesso a ela.

**Jill:** Ótima ideia. Poderíamos então gerar o formulário automaticamente, em um loop, fazendo um SELECT para obter os dados e depois criando cada caixa de verificação a partir de uma linha dos dados da consulta.

**Joe:** Ótimo. Mas o que acontece quando o botão Submit é pressionado? Qual será o conteúdo de \$\_POST?

**Frank:** Um minuto Joe, chegaremos lá em um instante. Mas vamos começar criando esta parte do script, a parte que exibe todos os dados da tabela e gera as caixas de verificação...



## Ímãs de Geladeira PHP & MySQL

Use os ímãs abaixo para completar o código do script Remover Email, o qual deverá apresentar uma caixa de verificação para cada cliente presente no banco de dados de Elmer. Repare que este código apenas cria o formulário; por enquanto, não se preocupe com o código que realiza o **DELETE**.

```


<p> Por favor selecione os endereços de email a serem apagados e clique em
Remove.</p>

<form method="post" action=" ..... echo $_SERVER['PHP_SELF']; ..... ">

<?php
$dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking', 'elvis_store')
or die('Erro ao conectar servidor MySQLr.');
// Exibe as linhas com os clientes e as caixas de verificação para apagá-los
$query = "SELECT * FROM email_list";
$result = mysqli_query($dbc, $query);

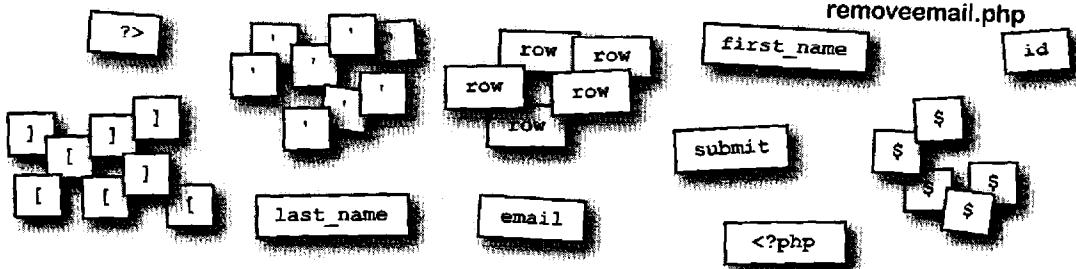
while ( ..... mysqli_fetch_array($result)) {
    echo '<input type="checkbox" value="'
name="todelete[]" />';
    echo ' ..... ;
    echo ' ..... ;
    echo ' ..... ;
    echo '<br />';
}

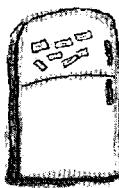
mysqli_close($dbc);
?>

<input type="submit" name=" ..... " value="Remove" />
</form>
```



## removeemail.php





## Ímãs de Geladeira PHP & MySQL - Solução

Use os ímãs abaixo para completar o código do script Remover Email, o qual deverá apresentar uma caixa de verificação para cada cliente presente no banco de dados de Elmer. Repare que este código apenas cria o formulário; por enquanto, não se preocupe com o código que realiza o DELETE.

```



<p>Por favor, selecione os endereços de email a serem apagados e clique em
Remove.</p>
<form method="post" action="removeemail.php">
    <?php echo $_SERVER['PHP_SELF']; ?>
    Este código é autorreferente!
    <?php
        $dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking', 'elvis_store')
        or die('Erro ao conectar ao serviço MySQL.');
        // Exibe as linhas com os clientes e as caixas de verificação para apagá-los
        $query = "SELECT * FROM email_list";
        $result = mysqli_query($dbc, $query);
        while ($row = mysqli_fetch_array($result)) {
            echo '<input type="checkbox" value="';
            echo $row['id'];
            echo '" name="todelete[]" />';
            echo ' ' . $row['first_name'] . ' ';
            echo ' ' . $row['last_name'] . ' ';
            echo ' ' . $row['email'] . ' ';
            echo '<br />';
        }
        mysqli_close($dbc);
    ?>
    .....<br />
    <input type="submit" name="submit" value="Remove" />
</form>

```

O código PHP de uma só linha ainda precisa ser colocado dentro das tags <?php e ?>

É aqui que a chave primária é usada nas caixas de verificação – podemos usar isto aqui mais adiante, para remover quaisquer usuários selecionados.

Cada caixa de verificação é construída a partir de uma linha de dados dos clientes.

O script ainda não está apagando nada. Por enquanto, ele apenas apresenta uma lista de caixas de verificação.

Você pode dar qualquer nome que quiser ao seu botão Submit – mas precisará lembrar-se do nome posteriormente, caso decida usar \$\_POST para verificar se o formulário foi submetido.

## Das caixas de verificação até as IDs dos clientes

O código para as caixas de verificação geradas pelo script Remover Email é simples HTML, com a nossa chave primária (`id`) colocada no atributo valor da tag `<input>`. No entanto, há uma pequena, porém, muito importante modificação em relação ao código HTML comum. Você pode ter reparado que há colchetes (`[]`) no final do nome da caixa de verificação – eles servem a um propósito vital.

```
echo '<input type="checkbox" value="' . $row['id'] . '" name="todelete[]">';
```

Os colchetes resultam na criação de um array dentro de `$_POST` que armazena o conteúdo do atributo (valor) de cada caixa de verificação marcada no formulário. Uma vez que o atributo value de cada caixa contém uma chave primária, **cada valor no array `todelete` é a ID de uma linha da nossa tabela que precisa ser apagada**. Isso nos possibilita fazer um loop através do array `todelete` e emitir uma consulta SQL para apagar cada cliente marcado no formulário.

Os colchetes ao final do nome da caixa de verificação automaticamente colocam os valores da caixa em um array a que demos o nome “`todelete[]`”.

Cada caixa de verificação tem a ID do cliente armazenada, o que fica acessível através da superglobal `$_POST`.

**MAKEMEELVIS.COM**

Please select the email addresses to delete from the email list and click Remove.

- Denny Bubbleton denny@mightygumball.net
- Irne Wurtz iwen@aliensabducredme.com
- Elbert Kreese elbert@kresleesrockerz.blz
- Irne Kreese elbert@kresleesrockerz.blz
- Don Draper draper@sterling-cooper.com



Entendi. Podemos simplesmente usar um loop com while para fazer um ciclo através do array `todelete` e apagar cada um dos clientes usando as suas IDs.

**Nós poderíamos usar um loop while, mas há uma solução mais elegante que usa um tipo diferente de loop.**

O loop `foreach` é um tipo especial de loop elaborado especificamente para se fazer um ciclo através de valores armazenados em um array. Tudo o que você precisa fazer é especificar o array através do qual quer fazer o loop, além de uma variável onde armazenar os valores, e o PHP cuidará de fazer as iterações neles, um por um... não é necessária nenhuma condição de teste!

Escreva aqui como você acha que um loop `foreach` poderia fazer iterações através de um array contendo as IDs dos clientes de Elmer:

## Faça loop através de um array com foreach

O loop foreach toma um array e faz loop através de cada elemento do array sem a necessidade de uma condição de teste nem um contador de loops. À medida que passa através de cada elemento do array, ele armazena o valor desse elemento em uma variável temporariamente. Assumindo-se que o array está armazenado em uma variável chamada \$customers, este código faz o loop:

O array através do qual você deseja fazer o loop aparece primeiro.

À medida que o loop passa através de cada elemento do array, ele os armazena temporariamente em uma variável com este nome.

```
foreach ($customers as $customer) {
```

```
    echo $customer;
```

```
};
```

Dentro do loop, você pode acessar cada elemento, usando o nome da variável.

Assim, se quiséssemos fazer um loop através das IDs dos clientes armazenadas no array \$\_POST do script Remover Email, poderíamos usar o seguinte código com foreach:

Aqui, o array é armazenado dentro da superglobal \$\_POST, e é identificado por "todelete".

Cada elemento do array fica acessível através da variável \$delete\_id.

```
foreach ($_POST['todelete'] as $delete_id) {
```

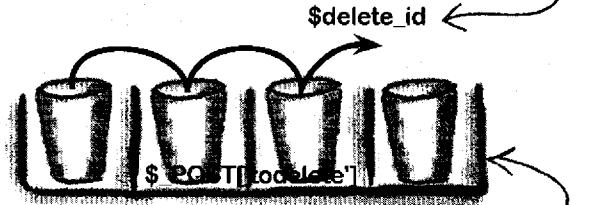
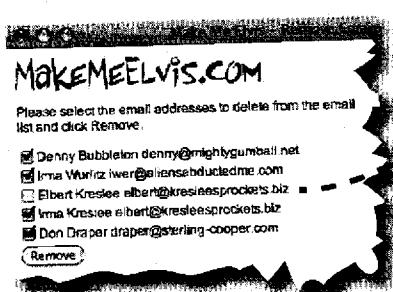
```
    // Apaga uma linha da tabela
```

```
};
```

Podemos usar \$delete\_id para remover cada um dos clientes do banco de dados.

A variável \$delete\_id armazena o valor de cada elemento do array à medida que o loop passa por eles, um de cada vez.

Podemos usar esta variável para aceitar o ID de cada cliente e então apagá-lo da tabela.



Com o loop foreach agora passando através de cada caixa marcada no formulário Remover Email, só precisamos adicionar algum código dentro do loop para emitir uma consulta DELETE, a qual fará a remoção propriamente dita de cada linha marcada.

Nós construímos o array de forma tal que ele só armazene os clientes que tiverem sido marcados no formulário Remover Email.



## Exercício

Complete o código do novo e melhorado script `removeemail.php` de Elmer, para fazê-lo excluir os clientes que tiverem sido marcados no formulário, quando este for submetido.

```

...
$dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking', 'elvis_store')
or die('Erro ao conectar no servidor MySQL.');

// Exclui as linhas dos clientes (somente se o formulário tiver sido submetido)
if (
    ) {
    foreach($_POST['todelete'] as $delete_id) {
        .....
        .....
        .....
    }

    echo 'Cliente(s) removido(s).<br />';
}

// Exibe as linhas dos clientes com caixas de verificação
$query = "SELECT * FROM email_list";
$result = mysqli_query($dbc, $query);
while ($row = mysqli_fetch_array($result)) {
    echo '<input type="checkbox" value="' . $row['id'] . '" name="todelete[]" />';
    echo $row['first_name'];
    echo ' ' . $row['last_name'];
    echo ' ' . $row['email'];
    echo '<br />';
}

mysqli_close($dbc);
?>

<input type="submit" name="submit" value="Remove" />
</form>

```



removeemail.php



## Exercício Solução

Complete o código do novo e melhorado script removeemail.php de Elmer, para fazê-lo excluir os clientes que tiverem sido marcados no formulário, quando este for submetido.

```
...  
$dbc = mysqli_connect('data.makemeelvis.com', 'elmer', 'theking', 'elvis_  
store')  
  
or die('Erro ao conectar no servidor MySQL.');//  
Só excluir os clientes se o  
formulário tiver sido submetido!  
// Exclui as linhas dos clientes (somente se o formulário tiver sido submetido)  
if ( isset($_POST['submit']) ) {  
    foreach ($_POST['todelete'] as $delete_id) {  
        $query = "DELETE FROM email_list WHERE id = '$delete_id';"  
        mysqli_query($dbc, $query)  
        or die('Erro consultando o banco de dados.');//  
    }  
  
    echo 'Cliente(s) removido(s).<br />';  
}  
  
// Exibe as linhas dos clientes com caixas de verificação  
$query = "SELECT * FROM email_list";  
$result = mysqli_query($dbc, $query);  
while ($row = mysqli_fetch_array($result)) {  
    echo '<input type="checkbox" value="' . $row['id'] . '" name="todelete[]" />;'  
    echo $row['first_name'];  
    echo ' ' . $row['last_name'];  
    echo ' ' . $row['email'];  
    echo '<br />';  
}  
mysql_close($dbc);  
?  
</form>
```

Usa \$delete\_id para  
escolher o cliente  
exato a ser excluído.

O código para gerar as caixas  
de cada cliente é o mesmo de  
antes.



removeemail.php



## TEST DRIVE

### Teste o novo e aprimorado script de Remover Email de Elmer.

Modifique o código do script `removeemail.php` de modo que ele gere caixas de verificação para os clientes, em vez de usar o velho campo de texto com os emails. Em seguida, adicione o código para excluir os clientes sempre que o formulário for submetido. Além disso, modifique o atributo `action` da tag `<form>` para tornar o formulário autorreferente.

Agora que `removeemail.php` usa um formulário autorreferente, você não precisa mais da página `removeemail.html` no seu servidor web, portanto, esteja à vontade para apagá-la. Em seguida, envie a nova versão de `removeemail.php` para o seu servidor e abra-o em um navegador. Marque as caixas de alguns clientes e clique em Submit. O formulário automaticamente se modifica para informar a remoção dos clientes..

**MAKEMEELVIS.COM**

Please select the email addresses to delete from the email list and click Remove.

- Denny Bubbleton denny@mightygumball.net
- Irma Wurlitz iwer@aliensabductedme.com
- Elbert Kreslee elbert@kresleespockets.biz
- Irma Kreslee elbert@kresleespockets.biz
- Don Draper draper@sterling-cooper.com

**Remove**

Quando você marca um cliente e clica em Submit, ele é removido do banco de dados.

**MAKEMEELVIS.COM**

Please select the email addresses to delete from the email list and click Remove.

Customer(s) removed.

- Denny Bubbleton denny@mightygumball.net
- Irma Wurlitz iwer@aliensabductedme.com
- Elbert Kreslee elbert@kresleespockets.biz
- Don Draper draper@sterling-cooper.com

**Remove**

O script confirma a remoção do cliente e também atualiza a lista – o cliente excluído não aparece mais.

viva [makemeelvis.com](http://makemeelvis.com)

# WELCOME TO Fabulous LAS VEGAS NEVADA

Estou curtindo o meu novo  
formulário de Remover Email.  
Já posso sair de férias. Viva Las  
Vegas, baby!

**MAKEMEELVIS.COM**

Enter your first name, last name, and email to be added to  
the Make Me Elvis mailing list.

First name: Julian  
Last name: Costa  
Email: julian@redactedpizza.com

**MAKEMEELVIS.COM**

Private: For Editors use ONLY  
Write and send an email to mailing list members.

Subject of email:

Big Sale  
Body of email:  
Big sale this week at MakeMeElvis.com  
not to be missed 20% off  
And don't forget the "buy one, get a  
free" — only three days left.

**MAKEMEELVIS.COM**

Please select the email address to delete from the email  
list and click Remove.

Customer(s) removed:  
 Danny Bubbleten deamy@mglygumail.net  
 Anna Wuritz twr@lensabductions.com  
 Albert Krebsle abert@salesandorder.biz  
 Glyn Draper drapergathering-cooper.com



# Sua caixa de ferramentas PHP & MySQL

**Você embolsou uma boa quantidade de novas habilidades em PHP e MySQL ao trabalhar para fazer a aplicação de Elmer subir de nível...**

## ALTER TABLE

Esta instrução SQL modifica a estrutura de uma tabela. Por exemplo, adicionando uma nova coluna de dados. Isto lhe permite alterar uma tabela estruturalmente sem ter de apagá-la e começar tudo de novo.

!

O operador de negação, ou operador NOT, inverte um valor true/false. Assim, true se torna false e vice-versa.

## foreach

Um construtor para loops do PHP que lhe permite fazer um loop através de um array, em um elemento de cada vez, sem usar uma condição de teste. Dentro do loop, você pode acessar cada elemento do array.

## isset(), empty()

A função interna `isset()` do PHP verifica se uma determinada variável existe, o que significa dizer que ela tenha recebido um valor. A função `empty()` dá um passo adiante e define se uma determinada variável contém um valor vazio (0, uma string vazia, false ou NULL).

## if, else

A declaração `if` do PHP toma decisões com base em se algo é verdadeiro ou não. Forneca a ela uma condição de teste do tipo true/false e algum código de ação, e a declaração `if` lhe permitirá tomar todo o tipo de decisões bacanas. Você pode adicionar uma cláusula `else` à declaração `if` para lhe fornecer uma ação alternativa.

## ==, !=, <, >, . . .

Operadores de comparação, que podem ser usados para se construir condições de teste para comparar valores uns com os outros. Estes frequentemente são usados para se controlar instruções `if` e loops.

## &&, OR

Estes são operadores lógicos, usados para se elaborar expressões que envolvam valores true/false. A combinação de dois valores com `&&` (AND) resulta em true somente se ambos os valores forem true. A combinação de valores com `||` (OR) resulta em true se qualquer um dos valores for true.



## 5 Trabalhando com Dados Armazenados em Arquivos

# Quando um Banco de Dados Não é o Suficiente



**Não acredite no hype... pelo menos não naquele sobre os bancos de dados.** Certamente os bancos de dados são maravilhosos para se armazenar todo tipo de dados que envolvam texto, **mas e quanto aos dados binários?** Coisas como **imagens JPEG** e **documentos PDF**? Faz sentido armazenar todas essas figuras da sua coleção de palhetas raras de guitarra em uma tabela de banco de dados? Geralmente não. Esses tipos de dados normalmente são armazenados em arquivos, e nós os deixaremos nos arquivos. Mas é inteiramente possível ter o melhor de dois mundos – este capítulo revela que você pode **usar arquivos e bancos de dados juntos para criar aplicações PHP** cheias de dados binários.

## Guitarristas virtuais gostam de competir

Aparentemente, o conceito de arte pela arte não é mais o suficiente, porque os jogadores do novo Guitar Wars estão fascinados com a possibilidade de tocar guitarra virtual competitivamente. Tanto que eles regularmente publicam as suas maiores pontuações no site do Guitar Wars, cuja manutenção você agora está encarregado de fazer. O problema é que, atualmente, não há um bom sistema para se verificar as pontuações.

A aplicação Guitar Wars permite que os usuários adicionem as suas pontuações à lista.

**Guitar Wars - High Scores**

Welcome, Guitar Warrior, do you have what it takes to crack the high score list? If so, just add your own score.

Score	Name	Date
127650	Paco Jastorius	2008-04-22 14:37:34
98430	Nevil Johansson	2008-04-22 21:27:54
345900	Eddie Vanilli	2008-04-23 09:06:35
282470	Belita Chevy	2008-04-23 09:12:53
368420	Ashion Simpson	2008-04-23 09:13:34

Sem alguma forma de verificação, não temos como saber quais pontuações são legítimas e quais não são.

Isso é totalmente falsificado. Não é possível que essas pontuações sejam verdadeiras. Gostaria de ver provas delas!

Esta é Belita, incrédula roqueira e jogadora do Guitar Wars.



## **Não dá para confiar no texto**

Do jeito como funciona agora, os jogadores estão publicando as suas pontuações simplesmente como texto, e tem havido muitas discussões sobre quais pontuações são reais e quais não são. Só existe uma forma de pôr um fim a todo esse bate-boca e coroar o verdadeiro campeão do Guitar Wars...

## O rock A imagem é a prova

O que precisamos para determinar quem está falando a verdade ou não é de algum tipo de verificação visual. Assim, a aplicação Guitar Wars terá de permitir que os usuários submetam uma captura de tela contendo a sua pontuação, quando forem incluir esta última. Isso significa que a lista dos recordes não conterá apenas pontuações, nomes e datas, mas será também uma lista de imagens (telas capturadas).

Com a verificação  
fotográfica, descobrimos  
que Eddie é uma fraude!

Score	Date
345900	Name: Eddie Vanilli Date: 2008-04-23 09:06:35
282470	Name: Belita Chevy Date: 2008-04-23 09:12:53
368420	Name: Ashton Simpson Date: 2008-04-23 09:13:34

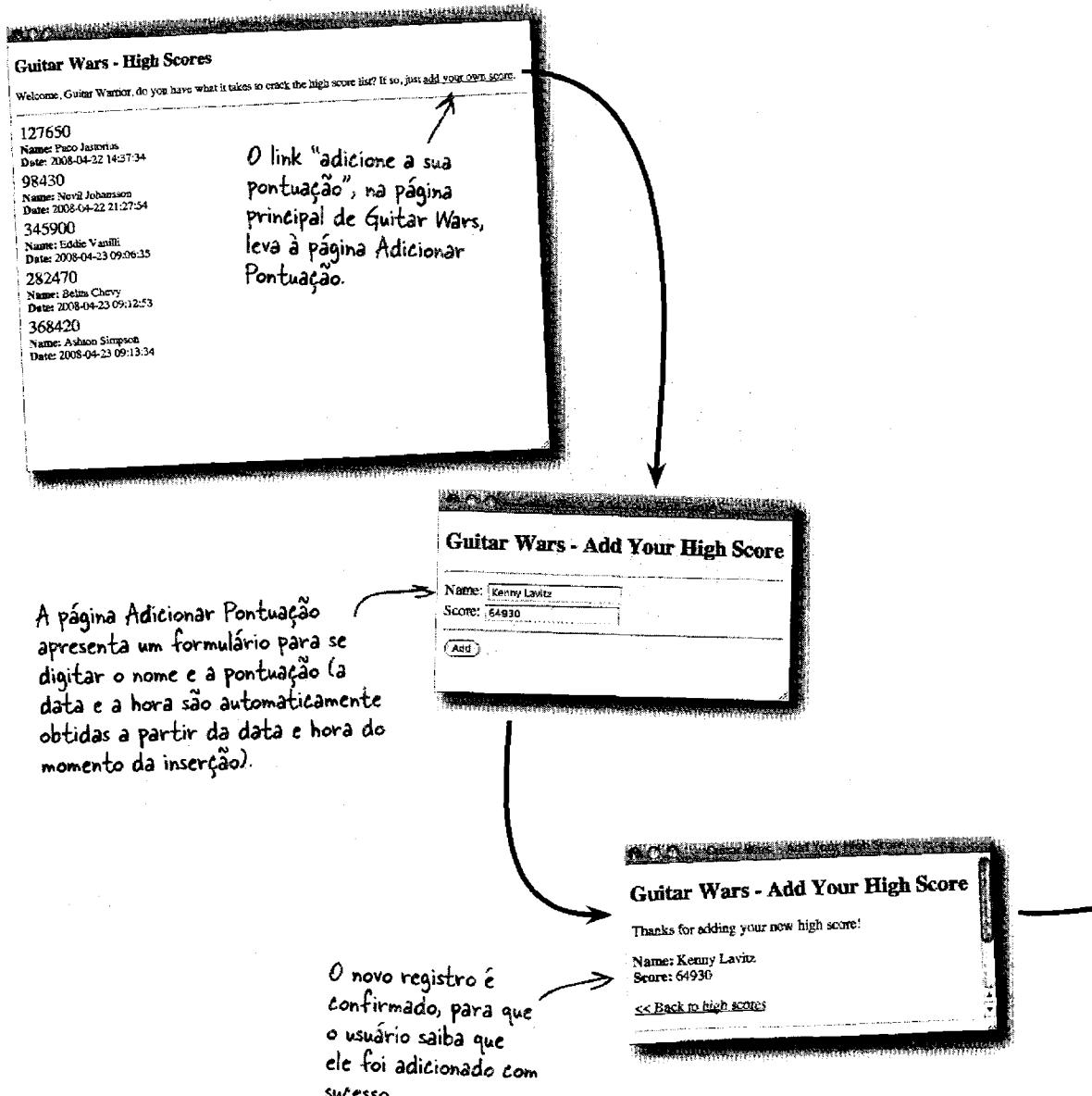
A pontuação de Belita  
é verdadeira, como  
prova a imagem que  
ela enviou.

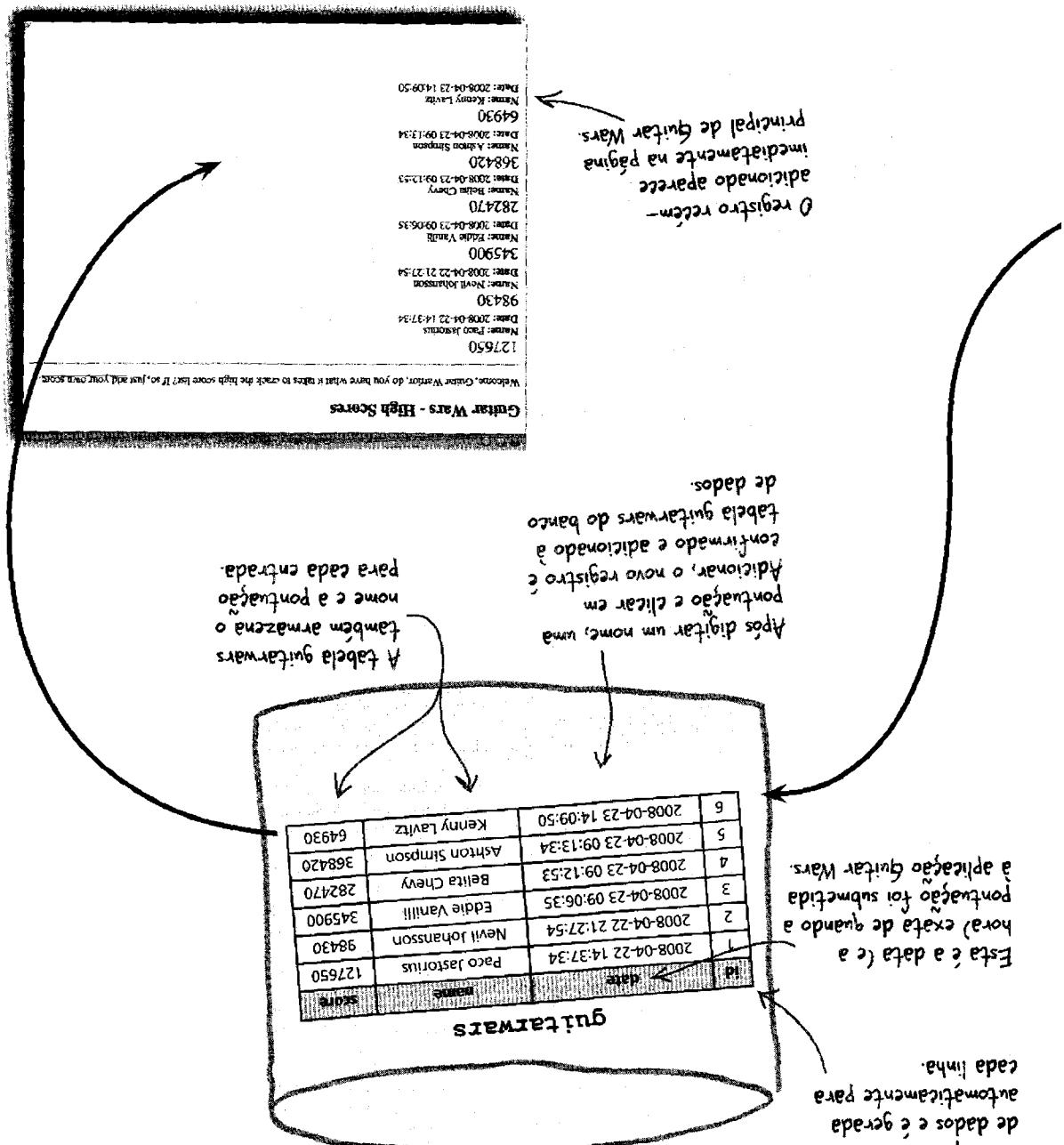


Este é Eddie,  
pseudoroqueiro  
e falsificador do  
Guitar Wars.

## A aplicação precisa armazenar imagens

Do jeito como está, a aplicação de recordes do Guitar Wars registra três tipos de informações: a data e hora de uma nova pontuação, o nome da pessoa que a está submetendo e a pontuação propriamente dita. Essas informações são digitadas através de um formulário que faz parte da interface de usuário da aplicação, e depois disso, são armazenadas em uma tabela chamada `guitarwars`, em um banco de dados MySQL..





comente o código



## Exercício

A aplicação dos recordes do Guitar Wars terá de ser modificada para acomodar a nova necessidade de se enviar arquivos gráficos. Circule e escreva comentários nas partes da aplicação que terão de se modificar para atender ao novo requisito.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>Guitar Wars - Maiores Pontuações</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
    <h2>Guitar Wars - Maiores Pontuações</h2>
    <p>Bem-vindo, intrépido guitarrista! Você é bom o suficiente para entrar na lista de recordes do Guitar Wars? Se for, clique aqui para adicionar a sua pontuação.</a>.</p>
    <br />

<?php
    // Conecta-se ao banco de dados
    $dbc = mysqli_connect('www.guitarwars.net', 'admin', 'rockit', 'gwdb');

    // Obtém os dados das pontuações a partir do MySQL
    $query = "SELECT * FROM guitarwars";
    $data = mysqli_query($dbc, $query);

    // Faz um loop através do array contendo os dados das pontuações,
    // formatando-os como HTML '<table>';
    while ($row = mysqli_fetch_array($data)) {
        // Exibe os dados das pontuações
        echo '<tr><td class="scoreinfo">';
        echo '<span class="score">' . $row['score'] . '</span><br />';
        echo '<strong>Name:</strong> ' . $row['name'] . '<br />';
        echo '<strong>Date:</strong> ' . $row['date'] . '</td></tr>';
    }
    echo '</table>';

    mysqli_close($dbc);
?>

</body>
</html>
```



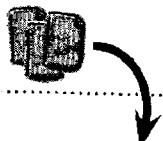
style.css

Este arquivo  
não precisa  
se modificar,  
portanto você  
não precisa se  
preocupar com  
ele.



index.php

Baixe isto!



O código-fonte completo da aplicação Guitar Wars está disponível para download no site da Alta Books: [www.altabooks.com.br](http://www.altabooks.com.br)

#	Date	Name	Score
1	2008-04-22 14:37:34	Paco Jastorius	127650
2	2008-04-22 21:27:54	Nevil Johansson	98430
3	2008-04-23 09:06:35	Eddie Vanilli	345900
4	2008-04-23 09:12:53	Belita Chevy	282470
5	2008-04-23 09:13:34	Ashton Simpson	368420
6	2008-04-23 14:09:50	Kenny Lavitz	64930

guitarwars

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>Guitar Wars - Adicione o seu Recorde</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
    <h2>Guitar Wars - Add Your High Score</h2>

<?php
if (isset($_POST['submit'])) {
    // Pega os dados em POST
    $name = $_POST['name'];
    $score = $_POST['score'];

    if (!empty($name) && !empty($score)) {
        // Conecta-se ao banco de dados
        $dbc = mysqli_connect('www.guitarwars.net', 'admin', 'rockit', 'gwdb');

        // Escreve os dados no banco
        $query = "INSERT INTO guitarwars VALUES (0, NOW(), '$name', '$score')";
        mysqli_query($dbc, $query);

        // Confirma êxito com o usuário
        echo '<p>Obrigado por adicionar o seu recorde!</p>';
        echo '<p><strong>Nome:</strong> ' . $name . '<br />';
        echo '<strong>Pontuação:</strong> ' . $score . '</p>';
        echo '<p><a href="index.php">&lt;&lt; Voltar para a lista dos recordes.</a></p>';

        // Limpa os dados da pontuação para limpar o formulário
        $Name = "";
        $Pontuação = "";

        mysqli_close($dbc);
    }
    else {
        echo '<p class="error">Por favor, insira todas as informações para
adicionar seu recorde.</p>';
    }
}

?>

<hr />
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
    <label for="name">Name:</label><input type="text" id="name" name="name"
    value="<?php if (!empty($name)) echo $name; ?>" /><br />
    <label for="score">Score:</label><input type="text" id="score" name="score"
    value="<?php if (!empty($score)) echo $score; ?>" />
    <hr />
    <input type="submit" value="Add" name="submit" />
</form>
</body>
</html>

```



addscore.php

## código guitar wars comentado



### Exercício Solução

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Guitar Wars - Maiores Pontuações</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<h2>Guitar Wars - Maiores Pontuações</h2>
<p>Welcome, Bem-Vindo, intrépido guitarrista! Você é bom o  
suficiente para entrar na lista de recordes do Guitar Wars? Se for,  
clique aqui para adicionar a sua pontuação.</a>.</p>
<br />

<?php
// Conecta-se ao banco de dados
$dbc = mysqli_connect('www.guitarwars.net',
'admin', 'rockit', 'gwdb');

// Obtém os dados das pontuações a partir de
$Query = "SELECT * FROM guitarwars";
$data = mysqli_query($dbc, $Query);

// Faz um loop através do array contendo os
// formatando-os como HTML '<table>';
while ($row = mysqli_fetch_array($data)) {
    // Exibe os dados das pontuações
    echo "<tr><td class='scoreinfo'>";
    echo "<span class='score'>" . $row['score'];
    echo "<strong>Name:</strong> " . $row['name'];
    echo "<strong>Date:</strong> " . $row['date'];
    echo "</td></tr>";

    mysqli_close($dbc);
?>
```

A imagem comprovando  
a pontuação precisa ser  
exibida na página principal.

A imagem precisa ser exibida index.php  
para o usuário, para se  
confirmar o êxito.

#### guitarwars

	date	name	score
1	2008-04-22 14:37:34	Paco Jastorius	127650
2	2008-04-22 21:27:54	Nevil Johansson	98430
3	2008-04-23 09:06:35	Eddie Vanilli	345900
4	2008-04-23 09:12:53	Belfita Chevy	282470
5	2008-04-23 09:13:34	Ashton Simpson	368420
6	2008-04-23 14:09:50	Kenny Lavitz	64930

A tabela precisa de uma  
nova coluna para armazenar o formulário precisa de  
o arquivo gráfico de cada uma tag <input> para  
a seleção da imagem.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Guitar Wars - Add Your High Score</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<h2>Guitar Wars - Adicione o seu Recorde</h2>
<?php
if (isset($_POST['submit'])) {
    // Pega os dados em POST
    $name = $_POST['name'];
    $score = $_POST['score'];

    if (!empty($name) & !empty($score)) {
        // Conecta-se ao banco de dados
        $dbc = mysqli_connect('www.guitarwars.net', 'admin', 'rockit',
'gwdb');

        // Escreve os dados no banco
        $Query = "INSERT INTO guitarwars VALUES (0, NOW(), '$name',
'$score')";
        mysqli_query($dbc, $Query);

        // Confirma êxito com o usuário
        echo "<p>Obrigado por adicionar o seu recorde!</p>";
        echo "<p><strong>Name:</strong> " . $name . "<br />";
        echo "<strong>Pontuação:</strong> " . $score . "</p>";
        echo "<p><a href='index.php'>&lt;&lt;/a> Voltar para a lista dos
recordes.</a></p>";

        // Limpa os dados da pontuação para limpar o formulário
        $name = "";
        $score = "";

        mysqli_close($dbc);
    } else {
        echo '<p class="error">Por favor, insira todas as informações para
adicionar seu recorde.</p>';
    }
}

<br />
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
<label for="name">Name:</label><input type="text" id="name"
name="name"
value="<?php if (!empty($name)) echo $name; ?>" /><br />
<label for="score">Pontuação:</label><input type="text" id="score"
name="score"
value="<?php if (!empty($score)) echo $score; ?>" />
<br />
<input type="submit" value="Add" name="submit" />
</form>
</body>
</html>
```

O arquivo da imagem  
precisa ser obtido a  
partir dos dados de  
POST.

Você deve usar a validação  
para certificar-se de  
que o campo do nome da  
imagem não esteja vazio.

A consulta SQL agora  
precisa inserir na  
tabela guitarwars o  
nome do arquivo que  
contém a imagem.

Esta consulta usa uma espécie  
de atalho, ao não especificar  
os nomes das colunas.

addscore.php

# Planejando os uploads de arquivos gráficos no Guitar Wars

Embora possa parecer que não é nada muito complicado adicionar suporte ao upload (envio) de arquivos gráficos ao Guitar Wars, a aplicação precisa ser modificada de várias maneiras. Por isso, é uma boa ideia ter um plano antes de mergulhar no código. Vamos esquematizar os passos necessários para atualizar o Guitar Wars e adicionar a funcionalidade que queremos.

## 1 Use ALTER para adicionar à tabela a coluna screenshot.

Em primeiro lugar vem o banco de dados, que precisa de uma nova coluna para armazenarmos o nome de cada arquivo gráfico contendo a pontuação do usuário. Uma vez que estamos planejando colocar todos os arquivos gráficos na mesma pasta, tudo o que precisamos armazenar no banco é o nome do arquivo (não é preciso incluir o caminho).



## 2

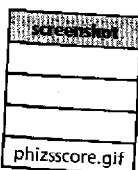
### Modifique o formulário de Adicionar Pontuação de modo que ele passe a usar um campo de input de arquivos para permitir o upload das imagens.

A página Adicionar Pontuação já tem um formulário para se adicionarem as pontuações, de modo que precisamos modificar esse formulário, incluindo nele um campo de input de arquivos. Esse campo funcionará conjuntamente com o navegador para apresentar ao usuário uma interface onde ele selecionará um arquivo a ser enviado.

Screen shot: Choose File  phizsscore.gif

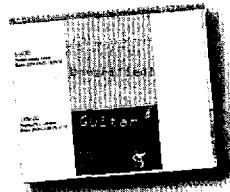
## 3 Escreva uma consulta INSERT para inserir o nome do arquivo gráfico na coluna screenshot da tabela.

O script Adicionar Pontuação que processa o formulário para a adição dos recordes também deve levar em consideração o novo campo de input de arquivos, e deve lidar com a inserção do nome do arquivo gráfico na coluna screenshot, quando inserir uma nova linha de dados de pontuação à tabela guitarwars.



## 4 Modifique a página principal de Guitar Wars de modo que ela passe a mostrar as imagens comprovantes dos recordes.

O último item da nossa lista de modificações envolve a página principal do Guitar Wars, index.php, que precisa ser modificada para exibir a imagem junto a cada pontuação exibida.



## **ALTERE sua tabela**

# O banco de dados das pontuações máximas precisa ser ALTERado

Além de uma série de ajustes no script PHP, a nova aplicação Guitar Wars precisa de uma nova coluna, na tabela guitarwars, para armazenar os nomes dos arquivos gráficos. É aí que entra o SQL, que oferece uma instrução chamada ALTER, capaz de modificar tabelas de tudo quanto é jeito, inclusive adicionando novas colunas de dados. Você usou a instrução ALTER, no capítulo anterior, para ajustar a tabela email\_list de Elmer, mas vamos recapitular aqui como esse comando funciona.

```
ALTER TABLE guitarwars DROP COLUMN score
```

A instrução **DROP COLUMN** elimina  
uma coluna inteira da tabela.

OK, talvez esse seja um exemplo perigoso, uma vez que ele revela como eliminar uma coluna inteira da tabela, com os dados e tudo. Mas é que certamente poderá haver uma situação em que você precise remover uma coluna inteira. É mais provável, porém, que você precise adicionar uma coluna, como é o caso aqui com o Guitar Wars. Isso é possibilitado por ADD COLUMN, que é uma das diversas alterações que você pode executar na tabela com ALTER.

A instrução **ALTER** é usada para mudar a estrutura do banco de dados

### **ADD COLUMN**

Adiciona uma coluna à tabela – basta especificar o nome da coluna e o seu tipo, após ADD COLUMN.

```
ALTER TABLE guitarwars  
ADD COLUMN age TINYINT
```

### **DROP COLUMN**

Exclui uma coluna (e quaisquer dados armazenados nela) de uma tabela – basta especificar o nome da coluna após DROP COLUMN.

```
ALTER TABLE guitarwars  
DROP COLUMN age
```

### **CHANGE COLUMN**

Muda o nome e o tipo de dados de uma coluna – basta especificar o nome atual da coluna, o novo nome e o novo tipo de dados, após CHANGE COLUMN.

```
ALTER TABLE guitarwars  
CHANGE COLUMN score high_score INT
```

### **MODIFY COLUMN**

Modifica o tipo de dados ou a posição de uma coluna dentro da tabela – basta especificar o nome da coluna o seu novo tipo de dados após MODIFY COLUMN. Para mudar a posição de uma coluna, especifique o nome dela e a sua posição exata (FIRST é a única opção aqui) ou uma posição relativa (AFTER para colocá-la após uma determinada coluna, sendo esta especificada pelo nome).

```
ALTER TABLE guitarwars  
MODIFY COLUMN date DATETIME AFTER  
age
```

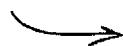
## Aponte seu lápis

Escreva uma instrução SQL para adicionar uma coluna chamada screenshot à tabela guitarwars. Certifique-se de fornecer à nova coluna um tipo de dados adequado. Depois, escreva outra consulta SQL, desta vez para verificar a estrutura da tabela e certificar-se de que a coluna foi adicionada com sucesso.

**guitarwars**

	screenshot

Escreva aqui a instrução  
para criar a coluna.



.....

.....

Escreva a outra  
instrução aqui.



.....

*aponte seu lápis solução*

**Aponte seu lápis  
Solução**

Escreva uma instrução SQL para adicionar uma coluna chamada screenshot à tabela guitarwars. Certifique-se de fornecer à nova coluna um tipo de dados adequado. Depois, escreva outra consulta SQL, desta vez para verificar a estrutura da tabela e certificar-se de que a coluna foi adicionada com sucesso.

A instrução ALTER adiciona a coluna screenshot à tabela guitarwars.

Uma vez que a coluna é nova, ela começa sua existência vazia (NULL) de quaisquer dados nas linhas.

A instrução ALTER não afeta quaisquer outros dados da tabela.

O nome da tabela a ser alterada  
vem depois de ALTER TABLE..

ALTER TABLE guitarwars

ADD COLUMN screenshot varchar(64)

**ADD COLUMN** indica que queremos alterar a tabela, adicionando a ela uma nova coluna.

O nome e o tipo de dados da nova coluna são especificados por último na consulta SQL - 64 caracteres são o suficiente para acomodar a maioria dos nomes de arquivos, mas você pode criar a coluna ainda maior do que isso se quiser.

## DESCRIBE guitarwars

Esta instrução exibe a estrutura da tabela, incluindo os nomes das colunas e os seus tipos de dados.

Terminamos o  
primeiro passo

**PRONTO** à  
1 Use ALTER para copiar a  
tabela a coluna screenshot.



## TEST DRIVE

### Adicione a coluna screenshot à tabela guitarwars.

Usando uma ferramenta MySQL, execute a instrução ALTER para adicionar a coluna screenshot à tabela guitarwars. Em seguida, emita a instrução DESCRIBE para dar uma olhada na estrutura da tabela e certificar-se de que a coluna foi adicionada.

Você pode construir a tabela guitarwars inicial baixando o código de exemplo para Guitar Wars e depois executando a consulta SQL que se encontra no arquivo guitarwars.sql.

A instrução DESCRIBE mostra a nova coluna screenshot.

```
File Edit Window Help QUB12
mysql> DESCRIBE email_list;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default |
+-----+-----+-----+-----+-----+
| id   | int(11) | NO   | PRI | NULL    |
| date | timestamp | NO   |     | CURRENT_TIMESTAMP |
| name | varchar(25) | NO   |     | NULL    |
| score | int(11) | NO   |     | NULL    |
| screenshot | varchar(64) | NO   |     | NULL    |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

não existem

### Perguntas Idiotas

**P:** As colunas criadas com ALTER precisam ser adicionadas ao final da tabela?

**R:** Não, elas podem ser adicionadas em qualquer lugar. Mas tenha em mente que a ordem das colunas dentro de uma tabela não tem tanta importância assim. Em outras palavras, você pode estruturar os resultados das suas consultas, de modo a organizar os dados de qualquer forma que quiser. Mas talvez goste da sensação de ordem estrutural criada pelo ordenamento específico das colunas, em cujo caso você poderá querer adicionar a coluna em um local exato. É possível fazer isso incluindo-se a palavra-chave FIRST a esta consulta ALTER. Ou então use AFTER para posicionar a nova coluna relativamente a uma já existente:

ALTER TABLE guitarwars  
ADD COLUMN age TINYINT  
AFTER name

Se você não especificar um lugar para a nova coluna ser adicionada, o padrão é no final da tabela.

**P:** O que acontece com as linhas de dados existentes no banco de dados após a adição da nova coluna screenshot?

**R:** Uma vez que a instrução ALTER só afeta a estrutura do banco, a nova coluna é criada vazia de dados nas linhas pré-existentes. Embora seja possível preencher a coluna screenshot de futuras linhas, as linhas pré-existentes todas terão uma coluna screenshot vazia.

**P:** Os nomes dos arquivos gráficos ainda poderão ser adicionados às linhas pré-existentes?

**R:** Sim, sem dúvida, e você usa a instrução UPDATE do SQL para fazê-lo. Não há nada que lhe impeça de enviar arquivos de imagens manualmente para o servidor, e depois usar UPDATE para vincular os nomes dos arquivos às pontuações existentes. Mas lembre-se de que a ideia aqui é permitir que o próprio usuário envie o arquivo gráfico, e portanto, faz sentido implementar a funcionalidade de upload das imagens. E os usuários poderão fazê-lo usando o script Adicionar Pontuação aprimorado que você está prestes a criar...

*adicone imagens com o formulário adicione pontuação*

## Como fazer para o usuário enviar a imagem?

Com a nova coluna adicionada ao banco de dados, estamos prontos para nos concentrarmos na funcionalidade que permitirá ao usuário enviar um arquivo gráfico. Mas como exatamente isso é possível? FTP? Telepatia? Isto na verdade nos leva de volta ao formulário Adicionar Pontuação, no qual poderemos usar um campo para permitir que o usuário selecione um arquivo gráfico a ser enviado.

The screenshot shows a web form titled "Guitar Wars - Add Your High Score". It has three input fields: "Name" with the value "Phiz Larston", "Score" with the value "186580", and a "Screen shot" field with a "Choose File" button. An arrow points from the "Choose File" button to the file name "phizsscore.gif" which is listed next to it. Below the fields is an "Add" button.

Os detalhes sobre o funcionamento deste botão são controlados pelo navegador e pelo sistema operacional nativo. Em geral, ele aciona uma caixa de diálogo do navegador, na qual o usuário pode navegar para encontrar o arquivo desejado no seu próprio disco rígido.

O formulário Adicionar Pontuação é o que permitirá aos usuários adicionar um novo registro à lista de recordes do Guitar Wars.

Após submeter-se o formulário, o arquivo gráfico binário é enviado ao servidor.

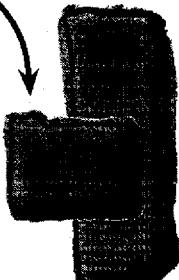


Assim, um campo de input ajuda o usuário a encontrar o arquivo a ser enviado. E depois? O campo também cuida da operação de fazer a imagem selecionada chegar até uma pasta do servidor, a partir de onde pode então ser exibida como parte da lista de recordes do Guitar Wars.

Esse campo de upload de arquivos do formulário é algum tipo de extensão esquisita do HTML? De forma alguma. A tag <input> do HTML tem suporte aos campos referentes a arquivos, e trabalha em conjunto com o PHP para permitir o upload dos arquivos. Mas antes de entrarmos nos detalhes do funcionamento do PHP, vamos dar uma olhada com mais atenção no próprio campo do formulário...

**Servidor web**

Uma pasta no servidor recebe o arquivo gráfico e o armazena.





## O Formulário Adicionar Pontuação em Detalhes

Este atributo instrui o formulário a usar um tipo especial de codificação, requerido para o upload de arquivos - ele afeta o modo como os dados POST são combinados e enviados após a submissão do formulário.

Estabelece um tamanho máximo para os arquivos enviados, neste caso 32KB (32.768 bytes).

Este formulário é autorreferente.

```
<form enctype="multipart/form-data" method="post" action="php echo $_SERVER['PHP_SELF']; ?&gt;&gt;</pre

```
<input type="hidden" name="MAX_FILE_SIZE" value="32768"/>
```



```
<label for="name">Nome:</label>
```



```
<input type="text" id="name" name="name" value="php if (!empty($name)) echo $name; ?&gt;" /&gt;</pre

```
<br />
```



```
<label for="score">Pontuação:</label>
```



```
<input type="text" id="score" name="score" value="php if (!empty($score)) echo $score; ?&gt;" /&gt;</pre

```
<br />
```



```
<label for="screenshot">Captura da Tela:</label>
```



```
<input type="file" id="screenshot" name="screenshot" />
```



```
<hr />
```



```
<input type="submit" value="Add" name="submit" />
```



```
</form>
```


```


```


```

Este é o campo de input propriamente dito, que em última instância, utiliza-se de uma caixa de diálogo do sistema operacional nativo para o usuário navegar até o arquivo e selecioná-lo.

**PRONTO**  
Mude que o formulário Adicionar Pontuação de modo que ele passe a usar um campo de input de arquivos para permitir o upload das imagens.

*armazenar arquivos de imagem no banco de dados*

## Insira o (nome do) arquivo gráfico no banco de dados

Simplesmente enviar um arquivo gráfico para o servidor através de um formulário não é suficiente. Temos de armazenar o nome do arquivo na nova coluna screenshot do banco de dados, para que a imagem possa ser acessada e exibida. Do jeito que está o script Adicionar Pontuação, já está inserindo novas pontuações na tabela guitarwars, usando a instrução SQL INSERT, mas esta não está levando em consideração a nova coluna screenshot:

```
INSERT INTO guitarwars VALUES (0, NOW(), '$name', '$score')
```

A coluna id é definida automaticamente através de AUTO\_INCREMENT - 0 (zero) é ignorado, embora a consulta realmente requeira um valor aqui.

Uma vez que esta instrução SQL está inserindo valores sem identificar colunas para eles, ela precisa incluir um valor para cada coluna. Mas nós acabamos de adicionar uma nova, o que significa que a consulta não funciona mais – está faltando um valor para a nova coluna screenshot. Assim, para adicionar um nome de arquivo gráfico ao banco como parte de uma nova linha de dados, precisamos adicionar um novo valor à instrução INSERT:

```
INSERT INTO guitarwars VALUES (0, NOW(), '$name', '$score', '$Screenshot')
```

As linhas de dados inseridas antes da criação da coluna screenshot não possuem um nome de arquivo gráfico.

| ID  | Date                | Name      | Score | Screenshot |
|-----|---------------------|-----------|-------|------------|
| 1   | 2014-07-10 10:00:00 | John      | 1000  |            |
| 2   | 2014-07-10 10:00:00 | Jane      | 1000  |            |
| 3   | 2014-07-10 10:00:00 | Bob       | 1000  |            |
| 4   | 2014-07-10 10:00:00 | Alice     | 1000  |            |
| 5   | 2014-07-10 10:00:00 | David     | 1000  |            |
| 6   | 2014-07-10 10:00:00 | Emily     | 1000  |            |
| 7   | 2014-07-10 10:00:00 | Frank     | 1000  |            |
| 8   | 2014-07-10 10:00:00 | Grace     | 1000  |            |
| 9   | 2014-07-10 10:00:00 | Hannah    | 1000  |            |
| 10  | 2014-07-10 10:00:00 | Ivan      | 1000  |            |
| 11  | 2014-07-10 10:00:00 | Jessica   | 1000  |            |
| 12  | 2014-07-10 10:00:00 | Karen     | 1000  |            |
| 13  | 2014-07-10 10:00:00 | Liam      | 1000  |            |
| 14  | 2014-07-10 10:00:00 | Mia       | 1000  |            |
| 15  | 2014-07-10 10:00:00 | Noah      | 1000  |            |
| 16  | 2014-07-10 10:00:00 | Olivia    | 1000  |            |
| 17  | 2014-07-10 10:00:00 | William   | 1000  |            |
| 18  | 2014-07-10 10:00:00 | Isabella  | 1000  |            |
| 19  | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 20  | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 21  | 2014-07-10 10:00:00 | Elijah    | 1000  |            |
| 22  | 2014-07-10 10:00:00 | Mason     | 1000  |            |
| 23  | 2014-07-10 10:00:00 | Alexis    | 1000  |            |
| 24  | 2014-07-10 10:00:00 | Madison   | 1000  |            |
| 25  | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 26  | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 27  | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 28  | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 29  | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 30  | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 31  | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 32  | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 33  | 2014-07-10 10:00:00 | James     | 1000  |            |
| 34  | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 35  | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 36  | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 37  | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 38  | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 39  | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 40  | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 41  | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 42  | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 43  | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 44  | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 45  | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 46  | 2014-07-10 10:00:00 | James     | 1000  |            |
| 47  | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 48  | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 49  | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 50  | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 51  | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 52  | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 53  | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 54  | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 55  | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 56  | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 57  | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 58  | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 59  | 2014-07-10 10:00:00 | James     | 1000  |            |
| 60  | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 61  | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 62  | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 63  | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 64  | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 65  | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 66  | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 67  | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 68  | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 69  | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 70  | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 71  | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 72  | 2014-07-10 10:00:00 | James     | 1000  |            |
| 73  | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 74  | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 75  | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 76  | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 77  | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 78  | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 79  | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 80  | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 81  | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 82  | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 83  | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 84  | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 85  | 2014-07-10 10:00:00 | James     | 1000  |            |
| 86  | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 87  | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 88  | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 89  | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 90  | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 91  | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 92  | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 93  | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 94  | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 95  | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 96  | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 97  | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 98  | 2014-07-10 10:00:00 | James     | 1000  |            |
| 99  | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 100 | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 101 | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 102 | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 103 | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 104 | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 105 | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 106 | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 107 | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 108 | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 109 | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 110 | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 111 | 2014-07-10 10:00:00 | James     | 1000  |            |
| 112 | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 113 | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 114 | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 115 | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 116 | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 117 | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 118 | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 119 | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 120 | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 121 | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 122 | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 123 | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 124 | 2014-07-10 10:00:00 | James     | 1000  |            |
| 125 | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 126 | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 127 | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 128 | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 129 | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 130 | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 131 | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 132 | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 133 | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 134 | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 135 | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 136 | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 137 | 2014-07-10 10:00:00 | James     | 1000  |            |
| 138 | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 139 | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 140 | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 141 | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 142 | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 143 | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 144 | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 145 | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 146 | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 147 | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 148 | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 149 | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 150 | 2014-07-10 10:00:00 | James     | 1000  |            |
| 151 | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 152 | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 153 | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 154 | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 155 | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 156 | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 157 | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 158 | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 159 | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 160 | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 161 | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 162 | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 163 | 2014-07-10 10:00:00 | James     | 1000  |            |
| 164 | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 165 | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 166 | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 167 | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 168 | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 169 | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 170 | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 171 | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 172 | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 173 | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 174 | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 175 | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 176 | 2014-07-10 10:00:00 | James     | 1000  |            |
| 177 | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 178 | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 179 | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 180 | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 181 | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 182 | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 183 | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 184 | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 185 | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 186 | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 187 | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 188 | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 189 | 2014-07-10 10:00:00 | James     | 1000  |            |
| 190 | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 191 | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 192 | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 193 | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 194 | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 195 | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 196 | 2014-07-10 10:00:00 | Levi      | 1000  |            |
| 197 | 2014-07-10 10:00:00 | Amelia    | 1000  |            |
| 198 | 2014-07-10 10:00:00 | Oliver    | 1000  |            |
| 199 | 2014-07-10 10:00:00 | Eliza     | 1000  |            |
| 200 | 2014-07-10 10:00:00 | Henry     | 1000  |            |
| 201 | 2014-07-10 10:00:00 | Abigail   | 1000  |            |
| 202 | 2014-07-10 10:00:00 | James     | 1000  |            |
| 203 | 2014-07-10 10:00:00 | Elizabeth | 1000  |            |
| 204 | 2014-07-10 10:00:00 | Matthew   | 1000  |            |
| 205 | 2014-07-10 10:00:00 | Charlotte | 1000  |            |
| 206 | 2014-07-10 10:00:00 | Benjamin  | 1000  |            |
| 207 | 2014-07-10 10:00:00 | Scarlett  | 1000  |            |
| 208 | 2014-07-10 10:00:00 | Lucas     | 1000  |            |
| 209 | 2014-07-10 10:00:00 | Levi      | 1000  |            |
|     |                     |           |       |            |

## Descubra o nome do arquivo enviado

Parece estar tudo certo com a consulta, mas nós ainda não sabemos qual é o nome do arquivo gráfico. É lógico assumir que o campo de input de arquivos do formulário oferece, de alguma forma, acesso ao nome do arquivo, mas como? A resposta reside em uma variável superglobal interna do PHP chamada `$_FILES`, que é semelhante à superglobal `$_POST` que já usamos para acessar dados do formulário. Como `$_POST`, `$_FILES` é um array, e dentro dele encontra-se não apenas o nome do arquivo enviado, mas também outras informações que poderão ser úteis.

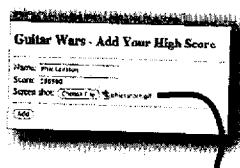
O formulário envia algumas informações úteis sobre o arquivo para o script PHP, através da variável superglobal `$_FILES`.



`phizsscore.gif`

Este é o arquivo gráfico sendo enviado através do campo de input de arquivos do formulário.

A superglobal interna `$_FILES` fornece acesso a informações sobre arquivos enviados.



`<input type="file" name="Screenshot" />`

`$_FILES['Screenshot']['name']`

`phizsscore.gif`

Nome do arquivo enviado.

`$_FILES['Screenshot']['type']`

`image/gif`

O tipo MIME do arquivo enviado, neste caso GIF.

`$_FILES['Screenshot']['size']`

`2242`

O tamanho (em bytes) do arquivo enviado.

`$_FILES['Screenshot']['tmp_name']`

`/tmp/phizS7o1kX`

A localização de armazenagem temporária do arquivo no servidor.

`$_FILES['Screenshot']['error']`

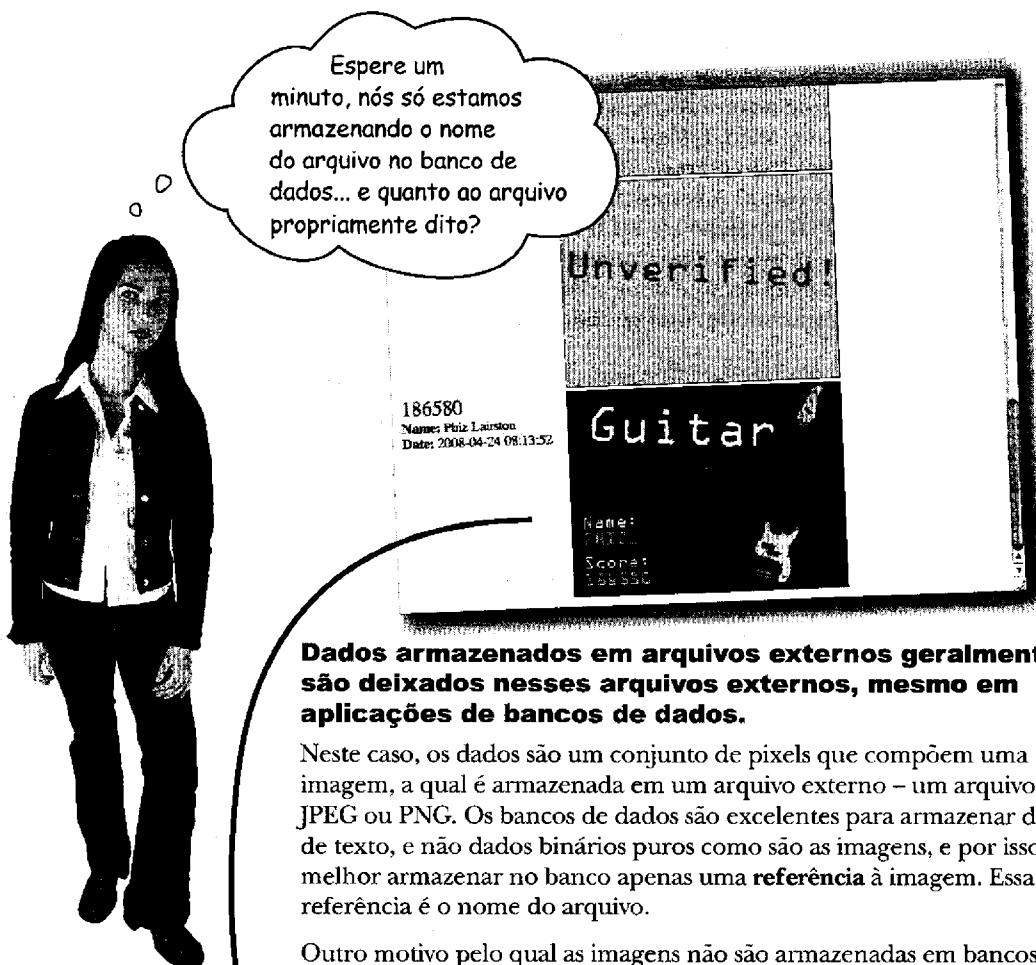
`0`

O código de erro para o upload do arquivo; O indica sucesso, outros valores indicam fracasso.

As outras informações disponibilizadas na variável `$_FILES` certamente são úteis, mas neste momento só precisamos do nome da imagem, que pode ser armazenado em uma variável local (`$Screenshot`) e usado na instrução SQL `INSERT`.

`$Screenshot = $_FILES['Screenshot']['name'];`

## **manter os dados de arquivo externo em arquivos externos**



**Dados armazenados em arquivos externos geralmente são deixados nesses arquivos externos, mesmo em aplicações de bancos de dados.**

Neste caso, os dados são um conjunto de pixels que compõem uma imagem, a qual é armazenada em um arquivo externo – um arquivo GIF, JPEG ou PNG. Os bancos de dados são excelentes para armazenar dados de texto, e não dados binários puros como são as imagens, e por isso, é melhor armazenar no banco apenas uma **referência** à imagem. Essa referência é o nome do arquivo.

Outro motivo pelo qual as imagens não são armazenadas em bancos de dados, nas aplicações web, é que seria muito mais difícil exibi-las usando-se código HTML. Lembre-se de que o código HTML referencia imagens a partir de arquivos externos usando os nomes dos arquivos. Assim, para gerar uma tag de imagem em HTML usa-se o nome do arquivo, e não os dados puros da imagem.

Para se colocar uma imagem em uma página web, só é preciso uma referencia ao arquivo.



O nome do arquivo gráfico.

A tag HTML <img> usa o nome para referenciar o arquivo gráfico armazenado no servidor web.



phizscore.gif



## Aponte seu lápis

A página principal do Guitar Wars (index.php) ainda não está exibindo as imagens de captura de tela que comprovam as pontuações. Complete o código para que ele passe a exibi-las.

```
<?php

// Conecta-se ao banco de dados
$dbc = mysqli_connect('www.guitarwars.net', 'admin', 'rockit', 'gwdb');

// Obtém os dados das pontuações a partir do MySQL
$query = ..... ;
$data = mysqli_query($dbc, $query);

// Faz um loop através do array contendo os dados das pontuações, formatando-os
// como HTML

echo '<table>';
while ($row = mysqli_fetch_array($data)) {
    // Exibe os dados das pontuações
    echo '<tr><td class="scoreinfo">';
    echo '<span class="score">' . $row['score'] . '</span><br />';
    echo '<strong>Name:</strong> ' . $row['name'] . '<br />';
    echo '<strong>Date:</strong> ' . $row['date'] . '</td>';
    if (is_file('.....') && filesize('.....') > 0) {
        echo '<td></td></tr>';
    }
    else {
        echo '<td></td></tr>';
    }
}
echo '</table>';

mysqli_close($dbc);
?>
```

**aponte seu lápis solução**



## Aponte seu lápis Solução

A página principal do Guitar Wars (index.php) ainda não está exibindo as imagens de captura de tela que comprovam as pontuações. Complete o código para que ele passe a exibi-las.

Em uma tentativa de ajudar a simplificar o código, não iremos usar "or die()" para produzir mensagens de erro e interromper o script quando uma função mysqli falhar.

Você poderá continuar utilizando esse recurso nas suas próprias aplicações; nós pararemos de usá-lo daqui em diante, apenas para manter o código o mais breve possível.

```
<?php
    // Conecta-se ao banco de dados
    $dbc = mysqli_connect ('www.guitarwars.net', 'admin', 'rockit', 'gwdb');

    // Obtém os dados das pontuações a partir do MySQL
    $query = "SELECT * FROM guitarwars";
    $data = mysqli_query ($dbc, $query);

    // Faz um loop através do array contendo os dados das pontuações, formatando-os
    // como HTML
    echo '<table>';
    while ($row = mysqli_fetch_array ($data)) {
        // Exibe os dados das pontuações
        echo '<tr><td class="scoreinfo">';
        echo '<span class="score">' . $row['score'] . '</span><br />';
        echo '<strong>Name:</strong>' . $row['name'] . '<br />';
        echo '<strong>Date:</strong>' . $row['date'] . '</td>';
        if (is_file( ... . $row['Screenshot'] ) && filesize( ... . $row['Screenshot'] ) > 0) {
            echo '<td></td></tr>';
        } else {
            echo '<td></td></tr>';
        }
    }
    echo '</table>';

    mysqli_close ($dbc);
?>
```

A instrução SQL que pede as pontuações não se modifica em nada!

Esta função verifica se o arquivo gráfico não está vazio.

Esta função verifica se o arquivo gráfico realmente existe.

A coluna screenshot do banco de dados armazena uma imagem para cada pontuação.

4

Modifique a página principal da Guitar Wars de modo que ela passe a mostrar as imagens comprovantes dos recordes.

**PRONTO**



## TEST DRIVE

### **Adicione um novo recorde ao Guitar Wars, incluindo a captura de tela.**

Caso ainda não o tenha feito, baixe o código de exemplo do Guitar Wars no site da Alta Books, [www.altabooks.com.br](http://www.altabooks.com.br). Está na pasta capítulo 05 (chapter 05). O código consiste da página principal (`index.php`), do script Adicionar Pontuação (`addscore.php`) e de uma folha de estilo (`style.css`).

Primeiro, você precisa modificar o script `addscore.php` de modo que o seu formulário Adicionar Pontuação passe a ter suporte ao upload de arquivos. Isso inclui adicionar novos campos ao formulário, ajustar a tag `<form>` e verificar se a variável `$Screenshot` não está vazia. Depois, incorpore ao script a nova consulta `INSERT` que insere pontuações.

Agora, vá ao script `index.php` e insira nele o código da página anterior, para que ele possa exibir a captura de tela referente a cada pontuação.

Envie todos esses arquivos para o seu servidor web e abra a página `addscore.php` em um navegador. Digite uma nova pontuação no formulário e clique em Submit. Em seguida, navegue até a página `index.php` e dê uma olhada na nova pontuação.

|        |                      |                           |
|--------|----------------------|---------------------------|
| 368420 | Name: Ashton Simpson | Date: 2008-04-23 09:13:34 |
| 64930  | Name: Kenny Lavitz   | Date: 2008-04-23 14:09:50 |
| 186580 | Name: Philz Lairston | Date: 2008-04-24 08:13:52 |

Algo não está certo! A imagem não está aparecendo junto com a nova pontuação.

### **PODER DO CÉREBRO**

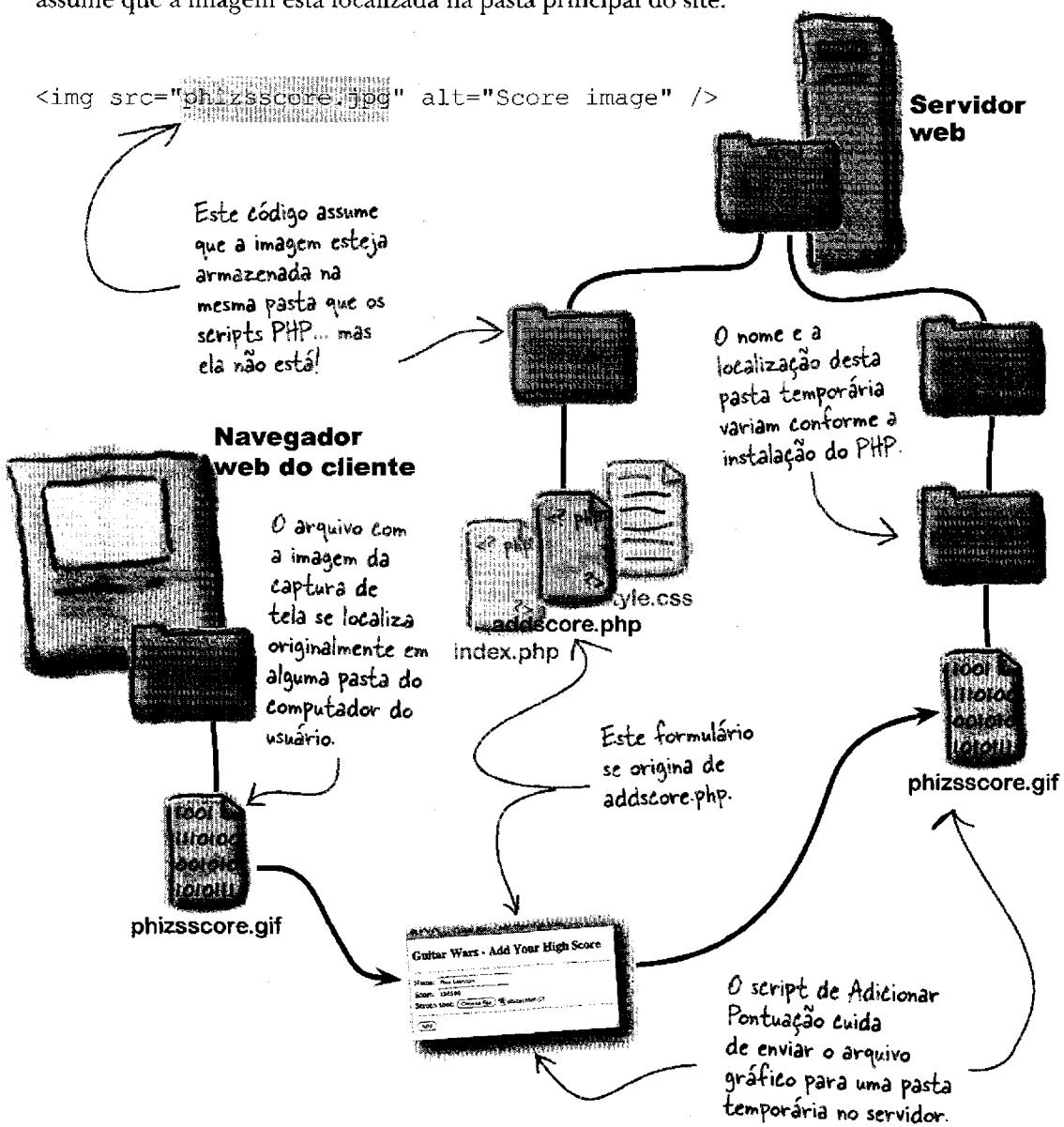
Por que você acha que a imagem com a captura da tela não está aparecendo junto com a nova pontuação? E quanto às pontuações que já estavam no banco de dados?

**arquivos enviados são armazenados em uma pasta temporária**

## Para onde foi o arquivo enviado?

O problema com a imagem enviada não aparecer é que nós assumimos que o arquivo seria enviado para a mesma pasta, no servidor, que a dos nossos scripts PHP. Ocorre que isso está simplesmente incorreto. O formulário de Adicionar Pontuação permite que o usuário selecione um arquivo do seu próprio computador, mas o arquivo é enviado para uma pasta temporária do servidor, na verdade. Essa pasta temporária é criada automaticamente no servidor, e em geral possui algum nome estranho, cheio de letras e números aleatórios.

Isso apresenta um problema para o nosso código <img> em index.php, porque ele assume que a imagem está localizada na pasta principal do site:





Armazenar imagens em uma criptica pasta temporária parece ser uma complicação desnecessária. Não é possível definirmos onde os arquivos enviados devem ser armazenados?

**Sim! O PHP lhe permite controlar onde os arquivos enviados serão armazenados.**

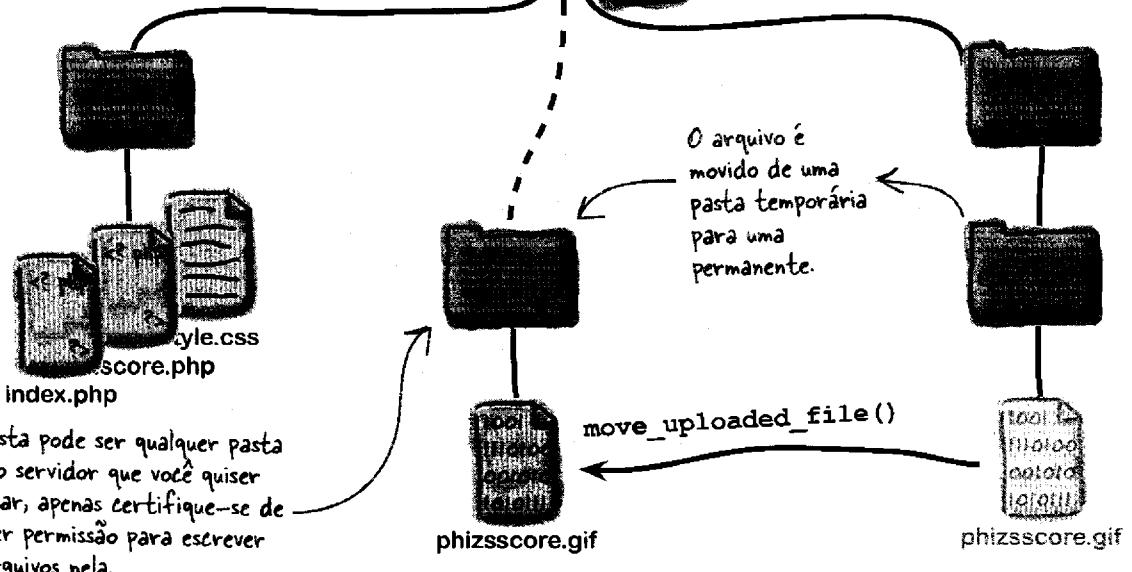
Porém, não é possível controlar, através do PHP, o local de armazenamento inicial dos arquivos enviados, motivo pelo qual essa localização é considerada como temporária. Mas você pode mover o arquivo para outro lugar, depois de ele ter sido enviado. A função PHP `move_uploaded_file()` aceita como parâmetros a localização-fonte e de destino do arquivo, e então realiza a transferência:

```
move_uploaded_file($FILES['screenshot']['tmp_name'], $target);
```

Esta é a localização-fonte do arquivo gráfico, incluindo o caminho e o nome do arquivo temporário.

Servidor Web

Esta é a localização de destino do arquivo gráfico, incluindo o caminho e o nome do arquivo permanente.



Esta pode ser qualquer pasta do servidor que você quiser usar, apenas certifique-se de ter permissão para escrever arquivos nela.

*não existem*  
**Perguntas Ídotas**

**P:** Não é possível modificar o lugar de armazenamento inicial dos arquivos enviados editando-se o arquivo `php.ini`?

**R:** Sim. O arquivo de inicialização do PHP (`php.ini`) pode ser usado para se modificar o lugar de armazenamento inicial dos arquivos enviados, através da opção `upload_tmp_dir`. Porém, se a sua aplicação estiver hospedada em um servidor virtual, você poderá não ter acesso a esse arquivo de inicialização, o que significa que, se quiser mover o arquivo enviado para uma outra pasta, terá de fazê-lo através de código PHP.

**P:** Por que a pasta de armazenamento inicial é chamada de “temporária”? Ela desaparece depois que o arquivo é retirado dela?

**R:** Não. A pasta é “temporária” no sentido de que não foi criada para servir como o lugar de armazenamento definitivo para os arquivos enviados. Você pode pensar nela como uma espécie de antessala, onde os arquivos enviados são armazenados até serem movidos para os seus lugares de armazenamento finais.

**P:** Por que eu não posso simplesmente deixar um arquivo na pasta temporária?

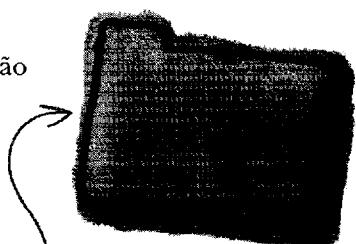
**R:** Você pode, em cujo caso terá de adicionar `$_FILES['screenshot']['tmp_name']` ao caminho da imagem, para se certificar de que ela possa ser encontrada na pasta temporária. Mas tenha em mente que você não pode controlar o nome ou a localização da pasta, em geral. Mais importante ainda, as pastas temporárias podem ser automaticamente esvaziadas, periodicamente, em alguns sistemas. Outro potencial problema é que a pasta temporária poderá não estar acessível ao público, e com isso, você não poderá referenciar os arquivos enviados a partir do seu código HTML, o que atrapalharia o funcionamento do Guitar Wars e da maioria das aplicações PHP. Movendo os arquivos enviados para outra pasta que não a de armazenamento temporário, você pode controlar cuidadosamente onde os arquivos ficarão, e como serão acessados.



### **Toda aplicação precisa de uma pasta images (imagens).**

OK, talvez dizer que “precisa”, possa ser um exagero, mas é importante que os pedaços das aplicações PHP fiquem o mais organizados possível, e uma maneira de fazê-lo é criando pastas para os diferentes componentes. Uma vez que as imagens são enviadas pelos usuários, elas não são algo sobre o que você normalmente terá controle direto, pelo menos em termos de nomes de arquivos e quantidades. Assim, é uma boa ideia armazená-las separadamente dos outros arquivos da aplicação.

Dito isto, precisamos de uma pasta `images`, onde os arquivos gráficos enviados para a aplicação `Guitar Wars` possam ser armazenados. Esta pasta pode servir também como lugar de armazenamento para quaisquer outras imagens que a aplicação use, caso surja a necessidade.



A pasta `images` na verdade não é muito mais importante que qualquer outra pasta, mas ela ajuda a organizar os arquivos gráficos em um só lugar.

**faça uma pasta para imagens enviadas**

## Crie um lar para os arquivos gráficos enviados

A pasta `images` é idêntica a qualquer outra do servidor, exceto pelo fato de que ela tem de ser colocada em algum lugar abaixo da pasta principal da aplicação. Na maioria dos casos, basta colocá-la diretamente como uma subpasta da principal, mas você pode criar uma hierarquia de pastas mais complexa, se quiser.

Criada a pasta `images` diretamente dentro da pasta principal do servidor, torna-se possível referenciar arquivos de dentro de scripts PHP, assim:

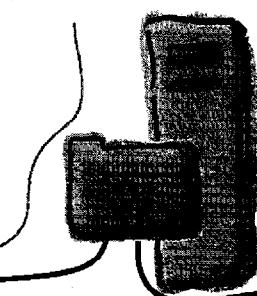
```
$target = GW_UPLOADPATH . "Screenshot";  
      ↓  
  images/phizsscore.gif
```

O nome do arquivo é concatenado ao caminho.

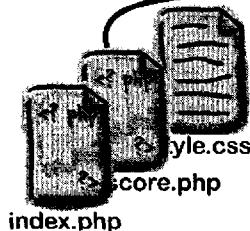
Esta é a pasta web da aplicação, onde os scripts PHP, incluindo `index.php`, ficam armazenados.

O caminho `$target` é criado a partir de uma nova constante que iremos adicionar ao script, chamada `GW_UPLOADPATH`, que armazena o caminho até a nossa pasta `images`. Assim como as variáveis, uma constante serve para armazenar algum dado. Mas o valor da constante não pode ser modificado, uma vez definido. O nome do arquivo gráfico, conforme digitado no formulário Adicionar Pontuação é então concatenado ao caminho até `images`.

Servidor web



A pasta `images` normalmente é colocada imediatamente dentro da pasta web.



Os arquivos gráficos enviados são movidos para a pasta `images`, onde podem ser exibidos através de tags HTML `<img>`.



Veja bem!

**Se a sua aplicação PHP estiver hospedada em qualquer outro lugar que não o seu computador local, você precisará usar FTP para criar a pasta `images`.**

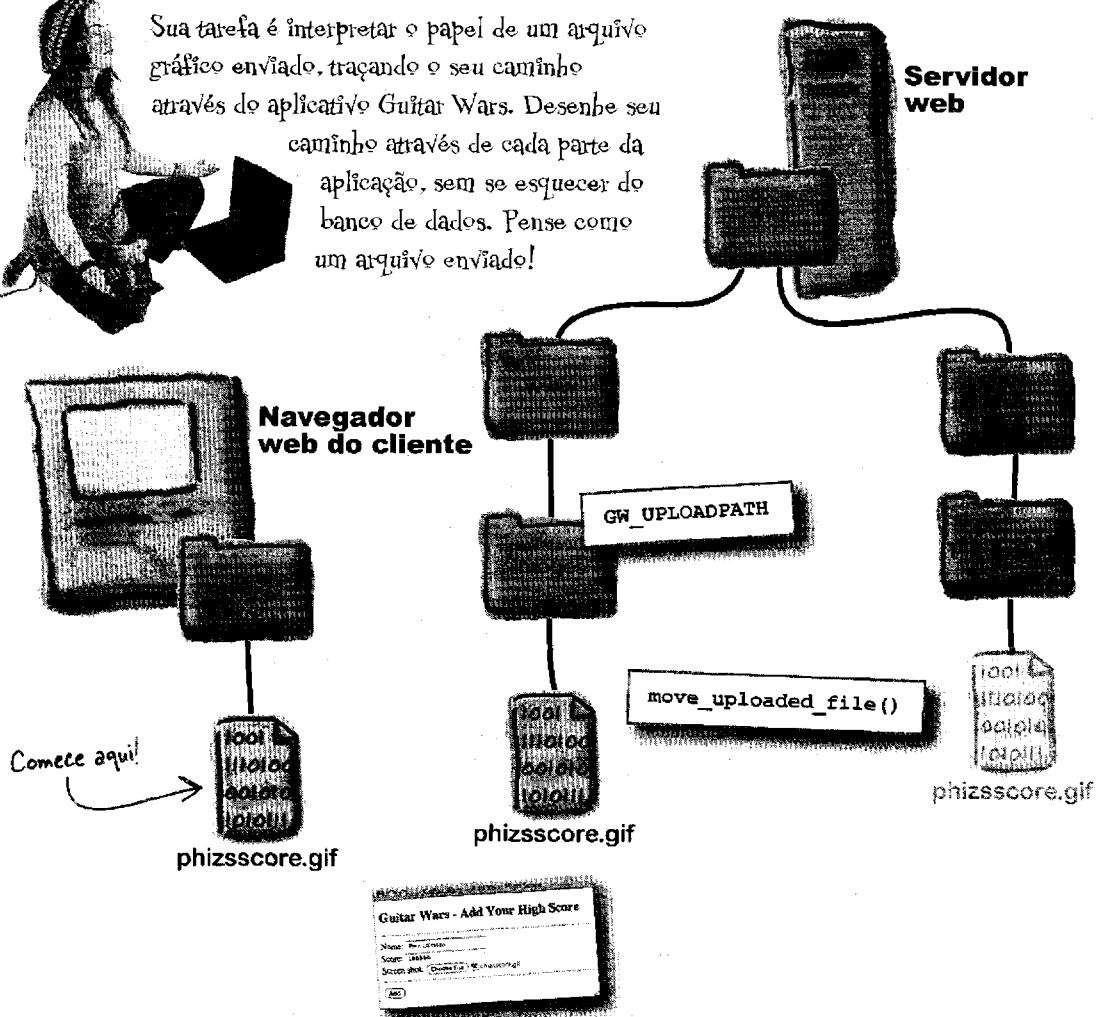
Use um programa de FTP para acessar o sistema de arquivos do seu site, e crie a pasta `images` diretamente dentro da pasta web da aplicação.

```
move_uploaded_file(  
  $FILES['Screenshot'][tmp_name'], $target);
```

## SINTA-SE como um arquivo gráfico enviado



Sua tarefa é interpretar o papel de um arquivo gráfico enviado, traçando o seu caminho através do aplicativo Guitar Wars. Desenhe seu caminho através de cada parte da aplicação, sem se esquecer do banco de dados. Pense como um arquivo enviado!



| # | date                | name          | score  | screenshot     |
|---|---------------------|---------------|--------|----------------|
| 7 | 2008-04-24 08:13:52 | Phiz Lairston | 186580 | phizsscore.gif |

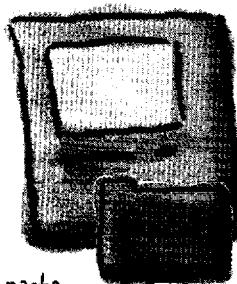
sinta-se como um arquivo gráfico enviado-solução

## SINTA-SE como um arquivo gráfico enviado - Solução

Sua tarefa é interpretar o papel de um arquivo gráfico enviado, traçando o seu caminho através do aplicativo Guitar Wars. Desenhe seu caminho através de cada parte da aplicação, sem se esquecer do banco de dados. Pense como um arquivo enviado!



Navegador web  
do cliente



Esta pasta fica no computador do usuário – você não tem controle nenhum sobre o nome dela, nem onde ela é armazenada, mas também nada disso lhe importa.

Você deve ser cuidadoso com o nome e a localização desta pasta, porque ela é usada em toda a aplicação Guitar Wars para armazenar e referenciar os arquivos gráficos enviados.

Primeiramente, o arquivo é enviado usando-se um campo de input do formulário.

`$screenshot = $_FILES['screenshot']['name'];`

`INSERT INTO guitarwars VALUES (0, NOW(), '$name', '$score', '$screenshot')`

Sim, este é um novo passo que não havíamos planejado – o seu projeto precisa ser flexível!

Caramba!

**PRONTO**  
the Move o arquivo gráfico da pasta temporária para uma pasta permanente criada para as imagens.



5

**guitarwars**

| #  | date                | name          | score  | Screenshot    |
|----|---------------------|---------------|--------|---------------|
| 7  | 2008-04-24 08:13:52 | Phiz Lairston | 186580 | phizscore.gif |
| 8  | 2008-04-24 08:13:52 | Phiz Lairston | 186580 | phizscore.gif |
| 9  | 2008-04-24 08:13:52 | Phiz Lairston | 186580 | phizscore.gif |
| 10 | 2008-04-24 08:13:52 | Phiz Lairston | 186580 | phizscore.gif |

addscore.php

0% ← → 100%

Após o arquivo ser enviado ao servidor e ser movido para o seu lugar de armazenamento final, seu nome é adicionado ao banco de dados.



## TEST DRIVE

**Dê aos arquivos gráficos enviados um lar permanente, com uma pasta própria para eles.**

Modifique o script addscore.php para que ele use a constante GW\_UPLOADPATH e armazene os arquivos gráficos no caminho para onde ela está apontando. Eis o pedaço do código que precisa ser modificado:

```
<?php
// Define as constantes do caminho e do tamanho máximo dos arquivos
define('GW_UPLOADPATH', 'images/');

if (isset($_POST['submit'])) {
    // Pega os dados em POST
    $name = $_POST['name'];
    $score = $_POST['score'];
    $screenshot = $_FILES['screenshot']['name'];

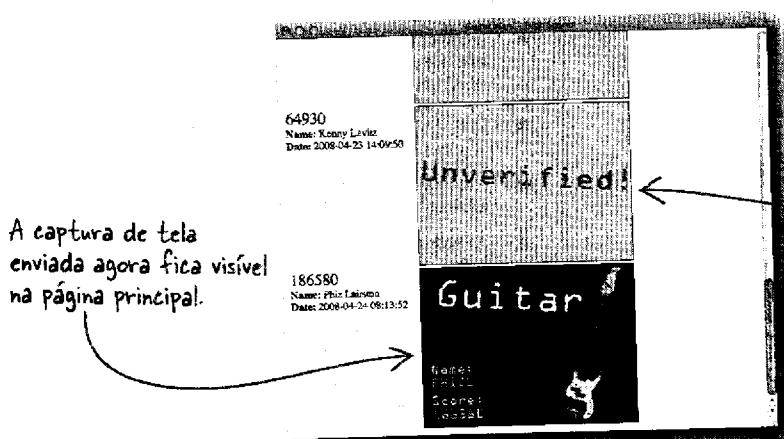
    if (!empty($name) && !empty($score) && !empty($screenshot)) {
        // Move o arquivo para a pasta-alvo
        $target = GW_UPLOADPATH . $screenshot;
        if (move_uploaded_file($_FILES['screenshot']['tmp_name'], $target)) {
            // Conecta-se ao banco de dados
            $dbc = mysqli_connect('www.guitarwars.net', 'admin', 'rockit', 'gwdb');

            // Escreve os dados no banco
            $query = "INSERT INTO guitarwars VALUES (0, NOW(), '$name', '$score', '$screenshot')";
            mysqli_query($dbc, $query);

            // Confirma êxito com o usuário
            echo '<p>Obrigado por adicionar o seu recorde!</p>';
            echo '<p><strong>Nome:</strong> ' . $name . '<br />';
            echo '<strong>Pontuação:</strong> ' . $score . '<br />';
            echo '</p>';
            echo '<p><a href="index.php">VOLTAR</a> | <a href="index.php?view=1">Voltar para a lista dos recordes.</a></p>';
        }
    }
}
```

addscore.php

O script index.php também é afetado pela constante GW\_UPLOADPATH. Não se esqueça de modificá-lo também. Após fazer essas modificações, envie os scripts para o seu servidor e tente adicionar uma nova pontuação novamente.



A imagem “unverified” (“não-verificada”) é exibida para pontuações antigas, que não têm uma captura de tela comprobatória.

## *não existem* Perguntas Ídiotas

**P:** Se o arquivo php.ini pode ser usado para se controlar o lugar de armazenamento dos arquivos enviados, por que é necessário mover os arquivos?

**R:** Porque nem sempre é possível modificar php.ini. Por exemplo, se você estiver criando a aplicação PHP em um servidor web virtual, é bastante provável que você não terá acesso para modificar as configurações de php.ini. E mesmo que possa modificar esse arquivo, você corre o risco de danificar sua aplicação caso precise movê-la para outro servidor. Em outras palavras, a aplicação ficaria dependente de um caminho controlado por php.ini, em vez de um caminho controlado pelo seu próprio código PHP.

**P:** Por que os usuários não podem digitar uma data, na aplicação Guitar Wars?

**R:** A data é uma parte importante do registro da pontuação, no sentido de que ela estabelece quando o recorde foi oficialmente publicado no site. Como ocorre com qualquer registro, a primeira pessoa a conseguir uma determinada pontuação recebe toda a glória. Em vez de confiar que o usuário nos dirá realmente quando foi que ele conseguiu a pontuação, podemos simplesmente usar a data e a hora do post como o registro oficial da pontuação. Isso elimina falsificações e emprega maior credibilidade à lista dos recordes. Os

usuários de uma aplicação tão competitiva estarão sempre procurando por um jeito de fraudá-la, portanto, você deve eliminar todas as possibilidades de fraude que puder encontrar!

Vale a pena ressaltar que a função NOW() usa a hora do servidor, que poderá não ser a mesma que a hora local do cliente. Isso não deverá ser um problema, porém, uma vez que todos os usuários estarão obrigados à mesma hora do servidor.

**P:** Não é possível que as pessoas escrevam em cima das imagens umas das outras, enviando arquivos gráficos com os mesmos nomes?

**R:** Sim. O problema tem a ver com o fato de que as imagens armazenadas no servidor usam exatamente o nome de arquivo fornecido pelo usuário no campo de upload do formulário. Assim, se dois usuários enviarem arquivos com o mesmo nome, a imagem do segundo usuário será escrita em cima da imagem do primeiro. Isso não é bom. Uma solução é adicionar alguma unicidade aos nomes dos arquivos no servidor. Uma forma simples de fazê-lo é adicionar a hora atual do servidor, em segundos, ao nome do arquivo, desta forma:

```
$target = GW_UPLOADPATH  
time() ←  
$Screenshot;
```

O resultado desse código é um nome 1221634560phizsscore.gif para o arquivo, em vez de phizsscore.gif, onde 1221634560 é a hora atual do servidor, expressa em segundos.

**P:** Nós poderíamos ter armazenado, no banco de dados do Guitar Wars, os dados propriamente ditos da imagem enviada?

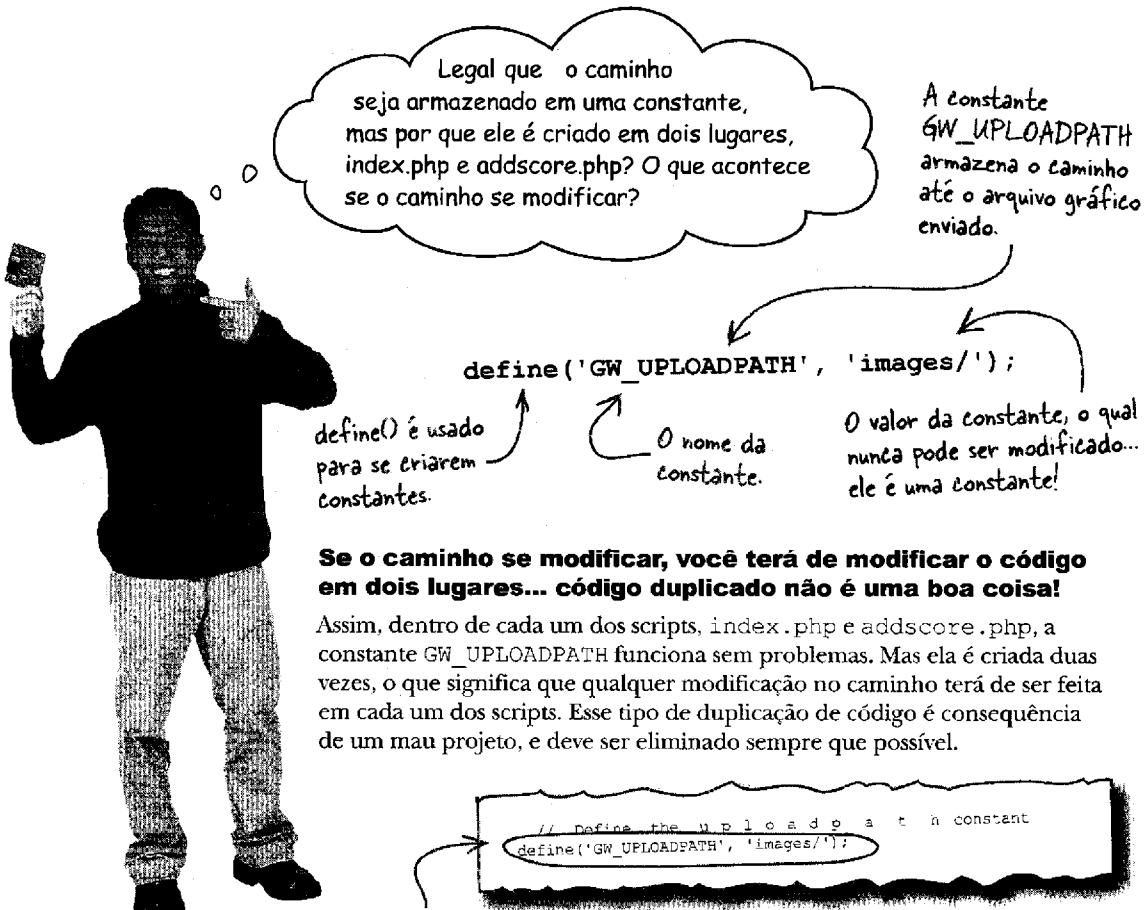
**R:** Sim. Os bancos de dados são bastante flexíveis, e lhe permitem armazenar dados binários dentro deles. Porém, o grande problema neste caso é que o Guitar Wars usa as imagens enviadas no código HTML, para que elas possam ser exibidas na página principal index.php. A tag HTML <img> foi criada para referenciar um arquivo gráfico armazenado no servidor, e não um conjunto de dados binários armazenados em um banco. Assim, mesmo que você alterasse a tabela guitarwars para armazenar dados binários, ainda se depararia com um desafio absurdo, ao tentar colocar os dados de volta em um formato que pudesse ser exibido usando-se código HTML.

*Não há nada de especial sobre a hora retornada pela função time(), exceto pelo fato de que ela gera números únicos... e o número retornado está sempre aumentando!*

## PONTOS DE BALA

- A instrução ALTER é usada para se modificar a estrutura de uma tabela de banco de dados MySQL, por exemplo para se adicionar uma nova coluna de dados.
- Com uma pequena ajuda do PHP e do MySQL, uma tag HTML <input> pode ser usada para se enviarem arquivos gráficos.
- A variável superglobal \$\_FILES é onde o PHP armazena informações sobre um arquivo enviado.
- A função padrão do PHP move\_uploaded\_file() lhe permite mover arquivos de um lugar para outro no servidor, e é essencial para o processamento dos arquivos enviados.
- A maioria das aplicações web beneficia-se de ter uma pasta images para armazenar as imagens usadas pela aplicação, especialmente aquelas enviadas pelos usuários.

**Os bancos de dados são ótimos para se armazenar dados de texto, mas em geral é melhor que eles referenciem dados binários de arquivos externos.**



```
define('GW_UPLOADPATH', 'images/');
```

`define()` é usado para se criarem constantes.

O nome da constante.

O valor da constante, o qual nunca pode ser modificado... ele é uma constante!

### Se o caminho se modificar, você terá de modificar o código em dois lugares... código duplicado não é uma boa coisa!

Assim, dentro de cada um dos scripts, `index.php` e `addscore.php`, a constante `GW_UPLOADPATH` funciona sem problemas. Mas ela é criada duas vezes, o que significa que qualquer modificação no caminho terá de ser feita em cada um dos scripts. Esse tipo de duplicação de código é consequência de um mau projeto, e deve ser eliminado sempre que possível.

```
// Define the upload path as a constant
define('GW_UPLOADPATH', 'images/');
```

The constant is stored twice, meaning it has to be maintained in two different places.

```
// Define the upload path and maximum file size constants
define('GW_UPLOADPATH', 'images/');
define('GW_MAXFILESIZE', 32768); // 32 KB
```

## PODER DO CÉREBRO

Para resolver o problema do código duplicado, precisamos armazenar a constante `GW_UPLOADPATH` em apenas um lugar. Você a armazenaria em `index.php` ou em `addscore.php`? Por quê?

*dados em scripts compartilhados com arquivos include*

## Dados compartilhados precisam ser compartilhados

Quando têm dados que devem ser compartilhados por diferentes scripts em uma aplicação, você precisa de uma forma de armazená-los em um só lugar, de onde cada script possa obtê-los para uso. Mas isso ainda não responde a questão de onde exatamente os dados devem ser postos...?

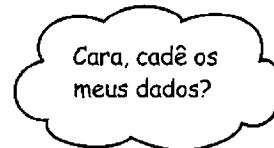
**Você poderia armazenar os dados apenas em index.php...**

```
// Define a constante do caminho da upload  
define('GW_UPLOADPATH', 'images/');
```

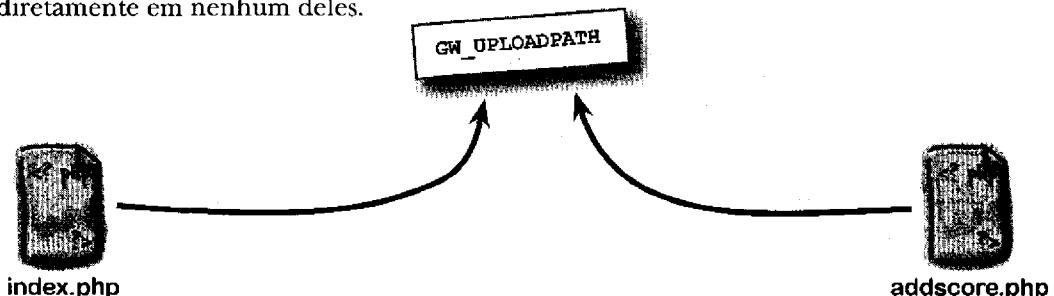


Os dados a serem compartilhados por scripts precisam ficar acessíveis a toda a aplicação, sem duplicações no código.

**...mas desse jeito os outros scripts não teriam acesso a eles.**



Assim, armazenar os dados a serem compartilhados em um script não funciona, porque os dados na verdade não estarão mais sendo compartilhados. A resposta está em conseguir, de algum modo, tornar os dados acessíveis a vários scripts, mas sem armazená-los diretamente em nenhum deles.

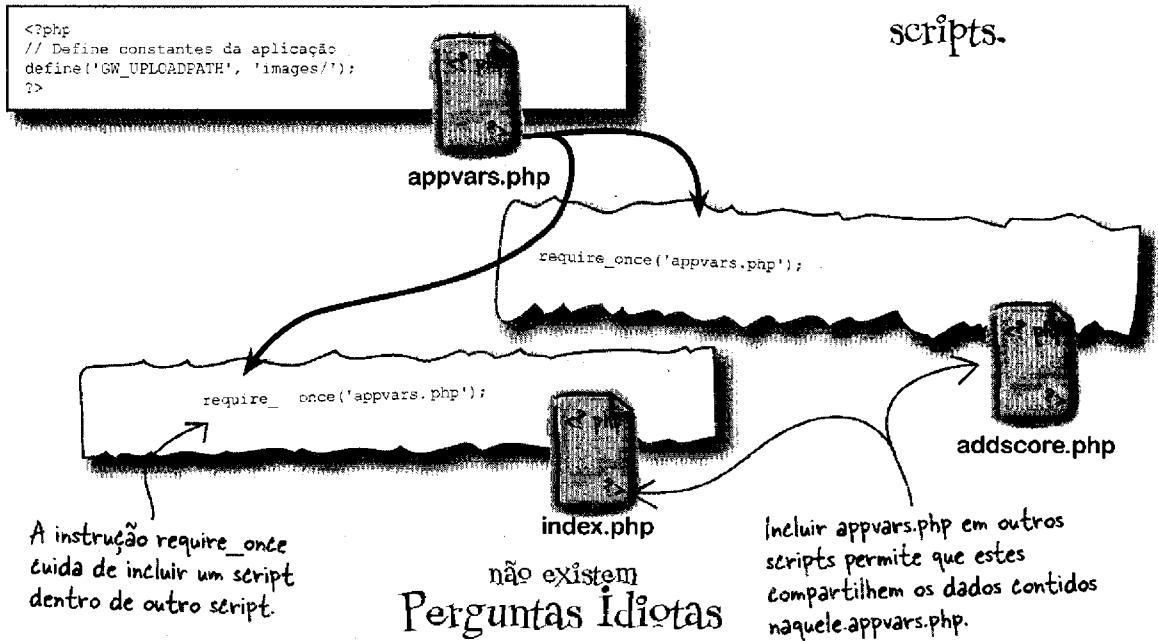


**Como tornar os dados acessíveis a ambos os scripts sem os armazenar em nenhum deles?**

A solução para o problema são os **arquivos include**, arquivos de código-fonte PHP que são inseridos em outros arquivos PHP conforme necessário.

# Serão necessários dados do script compartilhado

Os arquivos include são bastante poderosos, porque, após criá-los, você pode usá-los sempre que necessário em outros scripts, na prática compartilhando o código contido neles. A constante `GW_UPLOADPATH` pode ser colocada em um arquivo include para estabelecer um conjunto de “variáveis da aplicação”.



**P:** Ei, essas “variáveis” da aplicação não são na verdade constantes?

**R:** Em alguns casos, sim. Mas não há nenhum problema nisso. A ideia aqui não é se preocupar em diferenciar variáveis de constantes. Em vez disso, estamos apenas tentando estabelecer um lugar comum para o armazenamento de dados a serem compartilhados, dentro de uma aplicação. E esse lugar é um arquivo de script chamado `appvars.php`.

**P:** O código nos arquivos de script compartilhados limita-se a dados?

**R:** Não, de forma alguma. Qualquer código PHP pode ser colocado no seu próprio script e compartilhado usando-se a instrução `require_once`. Na verdade, é bastante comum que aplicativos compartilhem uma grande quantidade de códigos funcionais residentes em vários scripts. Não só é comum usar arquivos de scripts

compartilhados, como frequentemente fazer isso é uma ótima ideia, em termos de organização do código.

**P:** Por que a instrução PHP para se incluir códigos se chama `require_once` (“requisitar uma vez”)?

**R:** O nome arquivo `include` (“de inclusão”) vem de uma instrução PHP chamada `include`, a qual é bastante semelhante a `require_once`. A diferença é que `require_once` resulta em um erro caso o arquivo `include` não possa ser encontrado – `include`, por outro lado, não acusa um erro caso o arquivo `include` esteja faltando. Além disso, o “`once`” (“uma vez”) de `require_once` significa que o arquivo não poderá ser acidentalmente incluído mais de uma vez. Você poderá, às vezes, ver `include` sendo usado em vez de `require_once` para se incluir algum código que não seja tão importante, como por exemplo, código HTML puro que não esteja realizando algum propósito essencial. O PHP também tem as instruções `include_once` e `require`, que são variações de `require_once` e `include`.

Os arquivos include lhe permitem compartilhar código entre diversos scripts.

a instrução `require_once`

## Pense em `require_once` como um "inserir"

As inclusões não se limitam a um só arquivo PHP compartilhado, e podem aparecer em qualquer lugar do script que você queira usá-las. Você pode pensar em `require_once` com uma instrução `insert` ("inserir") que é substituída pelo conteúdo do script que ela referencia. No caso do Guitar Wars, as variáveis da conexão ao banco de dados poderiam também se beneficiar de serem movidas para um arquivo `include`. Assim, os conteúdos dos dois scripts compartilhados são inseridos diretamente em outros scripts, nos pontos onde forem requisitados.

```
require_once('appvars.php');
```

```
<?php
// Define constantes da aplicação
define('GW_UPLOADPATH', 'images/');
?>
```

appvars.php

Esta variável global armazena dados importantes para a aplicação, dos quais tanto `index.php` quanto `addscore.php` precisam.

```
require_once('connectvars.php');
```

```
<?php
// Define constantes de conexão do banco de dados
define('DB_HOST', 'www.guitarwars.net');
define('DB_USER', 'admin');
define('DB_PASSWORD', 'rockit');
define('DB_NAME', 'gwdb');
?>
```

connectvars.php

Em vez de duplicar as variáveis da conexão ao banco em cada script, nós podemos mover-las para um arquivo `include`, tornando-as compartilhadas.

A instrução `REQUIRE_ONCE` insere códigos de scripts compartilhados em outros scripts.

```

<?php
// Define as constantes da aplicação
define('GW_UPLOADPATH', 'images/');
// Define as constantes da conexão ao banco de dados
define('DB_HOST', 'www.guitarwars.net');
define('DB_USER', 'admin');
define('DB_PASSWORD', 'rockit');
define('DB_NAME', 'gwdb');

// Conecta-se ao banco de dados
$dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

// Obtém os dados das pontuações a partir do MySQL
$query = "SELECT * FROM guitarwars";
$data = mysqli_query($dbc, $query);

// Faz um loop através do array contendo os dados das
// pontuações, formatando-os como HTML
echo '<table>';
while ($row = mysqli_fetch_array($data)) {
    // Exibe os dados das pontuações
    echo '<tr><td class="scoreinfo">';
    echo '<span class="score">' . $row['score'] . '</span><br />';
    echo '<strong>Name:</strong> ' . $row['name'] . '<br />';
    echo '<strong>Date:</strong> ' . $row['date'] . '</td>';
    if (is_file(GW_UPLOADPATH . $row['Screenshot']) &&
        filesize(GW_UPLOADPATH . $row['Screenshot']) > 0) {
        echo '<td></td></tr>';
    } else {
        echo '<td></td></tr>';
    }
}
echo '</table>';

mysqli_close($dbc);
?>

```



## TEST DRIVE

**Crie dois arquivos include para o Guitar Wars, e depois compartilhe-os entre os outros scripts.**

Crie dois novos arquivos de texto, appvars.php e connectvars.php, e digite neles o código mostrado na página anterior. Em seguida, adicione instruções require\_once a index.php e addscore.php para que ambos os scripts compartilhados sejam incluídos. Envie todos os scripts para o seu servidor e teste o formulário Adicionar Pontuação e a página principal, para se certificar de que eles estejam funcionando com a nova e aprimorada estrutura organizacional com arquivos include.

Sensacional!  
Agora eu também  
tenho acesso aos dados  
compartilhados!



addscore.php

**PRONTO**  
Muito bom! Agora é só incluir o nome do arquivo enviado para uma constante, que será então compartilhada através de um arquivo include.



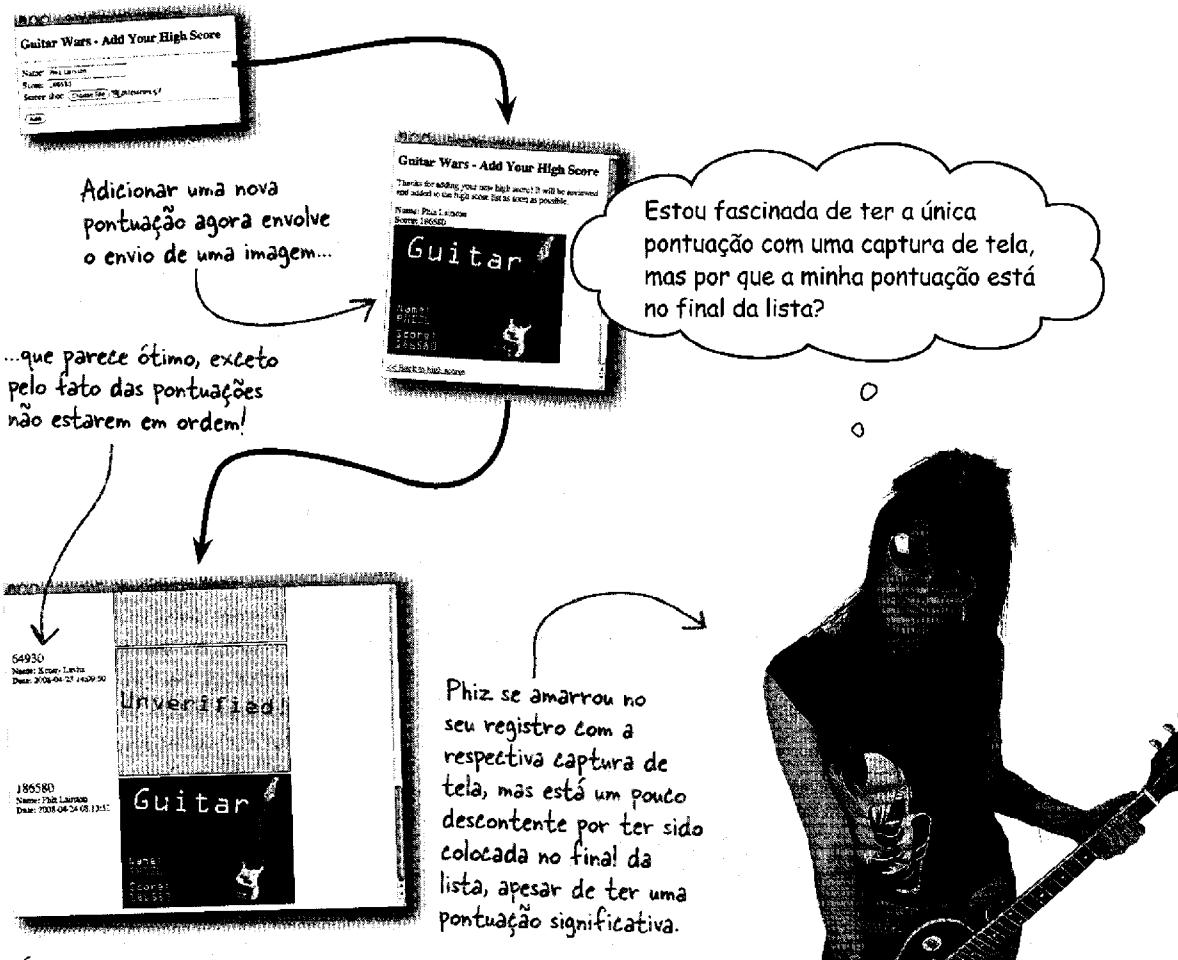
index.php

Opa, outro passo novo!  
Algumas coisas são  
difícies de planejar de  
antemão, portanto você  
terá de estar pronto  
para ajustar o seu  
projeto sempre  
que necessário.

## A instrução ORDER BY

# O mais importante para as pontuações máximas é ~~o timing a ordem~~

O Guitar Wars finalmente tem o suporte a imagens, permitindo que os usuários enviem capturas de telas para validar os seus recordes. Embora isso seja um grande aprimoramento na aplicação, ainda não resolveu o problema de que os usuários têm reclamado há algum tempo – a ordem das pontuações na página principal.



É verdade, as pontuações não estão em ordem. Elas estão sendo exibidas na ordem em que se encontram no banco de dados, que é totalmente arbitrária. Você nunca deve permitir que a ordem de exibição seja a mesma ordem em que os dados se encontram armazenados no banco, a não ser que ordem realmente não faça diferença. Neste caso, precisamos impor alguma ordem sobre os resultados da consulta. A instrução SQL ORDER BY torna isso possível.



# Ímas de Geladeira do PHP & MySQL

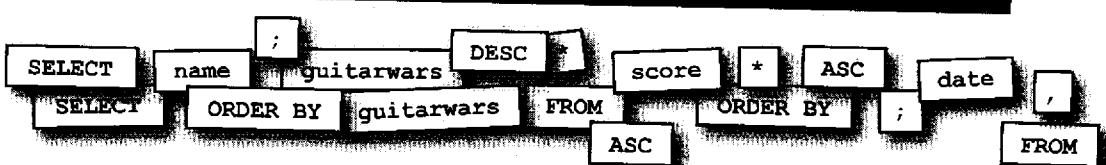
Veja se você consegue descobrir como ORDER BY funciona, usando os ímas abaixo para criar instruções SELECT ordenadas que resultem na saída abaixo. Circule também qual consulta você acha que representa a melhor solução para o Guitar Wars. Dica: ASC significa ASCendente e DESC significa DESCendente.

```
File Edit Window Help YYZ
mysql>.....
+-----+
| id | date           | name        | score      | screenshot   |
+-----+
3	2008-04-23 09:13:34	Ashton Simpson	368420	.
4	2008-04-23 09:12:53	Belita Chevy	282470	.
3	2008-04-23 09:06:35	Eddie Vanilli	345900	.
6	2008-04-23 14:09:50	Kenny Levitz	64930	.
2	2008-04-22 21:27:54	Nevil Johansson	98430	.
1	2008-04-22 14:37:34	Paco Jastorius	127650	.
7	2008-04-24 09:13:52	Phiz Lairston	186580	phizsscore.gif
+-----+
7 rows in set (0.0005 sec)
```

Aqui, os resultados da consulta  
são retornados em ordem numérica  
descendente, por pontuação, e depois  
em ordem ascendente, por data.

Aqui, os resultados da consulta são  
retornados em ordem alfabética  
ascendente, por nome.

```
File Edit Window Help YYZ
mysql>.....
+-----+
| id | date           | name        | score      | screenshot   |
+-----+
5	2008-04-23 09:13:34	Ashton Simpson	368420	.
3	2008-04-23 09:06:35	Eddie Vanilli	345900	.
4	2008-04-23 09:12:53	Belita Chevy	282470	.
7	2008-04-24 09:13:52	Phiz Lairston	186580	phizsscore.gif
1	2008-04-22 14:37:34	Paco Jastorius	127650	.
2	2008-04-22 21:27:54	Nevil Johansson	98430	.
6	2008-04-23 14:09:50	Kenny Levitz	64930	.
+-----+
7 rows in set (0.0005 sec)
```



## Ímas do php & MySQL solução



# Ímas de Geladeira do PHP & MySQL - Solução

Veja se você consegue descobrir como ORDER BY funciona, usando os ímas abaixo para criar instruções SELECT ordenadas que resultem na saída abaixo. Circule também qual consulta você acha que representa a melhor solução para o Guitar Wars. Dica: ASC significa ASCendente e DESC significa DESCendente.

```
File Edit Window Help YYZ
mysql> SELECT * FROM guitarwars ORDER BY name ASC ;
```

| ID | Date                | Name            | Score  | Screenshot     |
|----|---------------------|-----------------|--------|----------------|
| 5  | 2008-04-23 09:13:34 | Ashton Simpson  | 368420 |                |
| 4  | 2008-04-23 09:12:53 | Belita Chevy    | 282470 |                |
| 3  | 2008-04-23 09:06:35 | Eddie Vanilli   | 345900 |                |
| 6  | 2008-04-23 14:09:50 | Kenny Lavitz    | 64930  |                |
| 2  | 2008-04-22 21:27:54 | Nevil Johansson | 98430  |                |
| 1  | 2008-04-22 14:37:34 | Paco Jastorius  | 127650 |                |
| 7  | 2008-04-24 08:13:52 | Phiz Lairston   | 186580 | phizsscore.gif |

7 rows in set (0.0005 sec)

Aqui, os resultados da consulta são retornados em ordem numérica descendente, por pontuação, e depois em ordem ascendente, por data.

Aqui, os resultados da consulta são retornados em ordem alfabética ascendente, por nome.

Esta é a consulta de que precisamos para o Guitar Wars!

```
File Edit Window Help YYZ
mysql> SELECT * FROM guitarwars ORDER BY score DESC , date ASC ;
```

| ID | Date                | Name            | Score  | Screenshot     |
|----|---------------------|-----------------|--------|----------------|
| 5  | 2008-04-23 09:13:34 | Ashton Simpson  | 368420 |                |
| 3  | 2008-04-23 09:06:35 | Eddie Vanilli   | 345900 |                |
| 4  | 2008-04-23 09:12:53 | Belita Chevy    | 282470 |                |
| 7  | 2008-04-24 08:13:52 | Phiz Lairston   | 186580 | phizsscore.gif |
| 1  | 2008-04-22 14:37:34 | Paco Jastorius  | 127650 |                |
| 2  | 2008-04-22 21:27:54 | Nevil Johansson | 98430  |                |
| 6  | 2008-04-23 14:09:50 | Kenny Lavitz    | 64930  |                |

7 rows in set (0.000 sec)

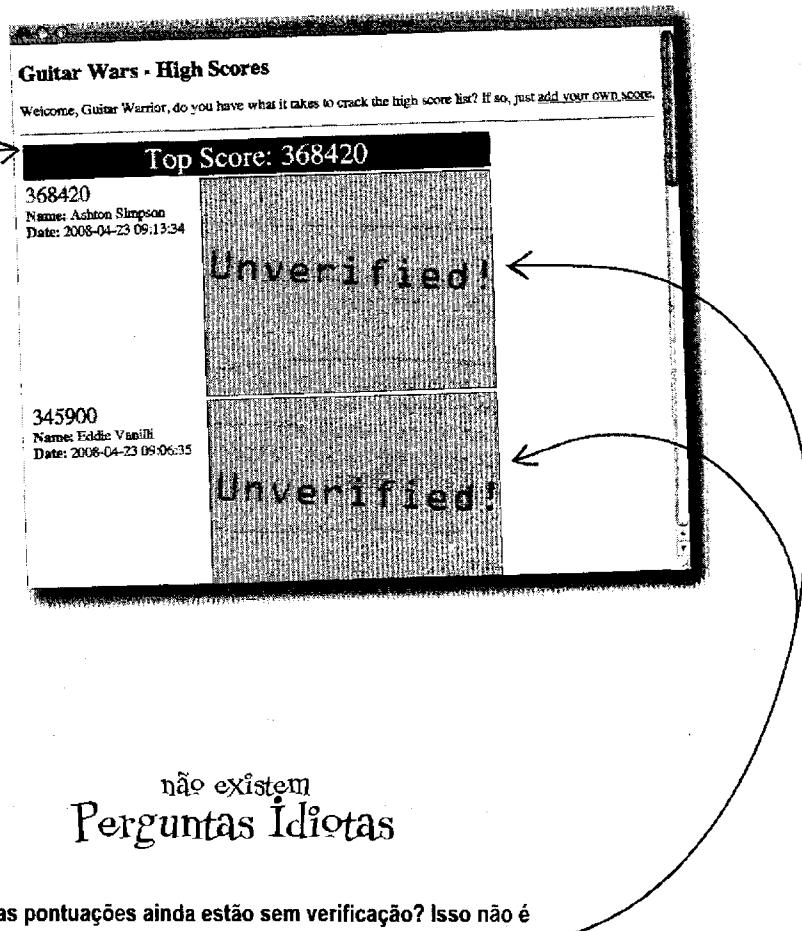
O ordenamento por data é secundário, e só se aplica caso haja duas pontuações idênticas, o que não ocorre aqui mas tem uma boa probabilidade de acontecer quando se tem um conjunto bastante grande de dados.

A vírgula é necessária para se separar os dois níveis de ordenamento.

## Homenageando o Guitar Warrior número um

Com a ordem das pontuações consertada, agora é possível fazer uma melhoria inesperada na lista dos recordes: colocar a maior pontuação de todas no topo da lista. O Guitar Warrior com a maior pontuação merece um cabeçalho dizendo claramente que ele é o número um, para que não haja dúvida... e para que os adversários saibam qual pontuação eles tem de superar.

Um cabeçalho  
saliente  
claramente a  
pontuação máxima,  
estabelecendo  
um alvo para os  
demais Guitar  
Warriors..



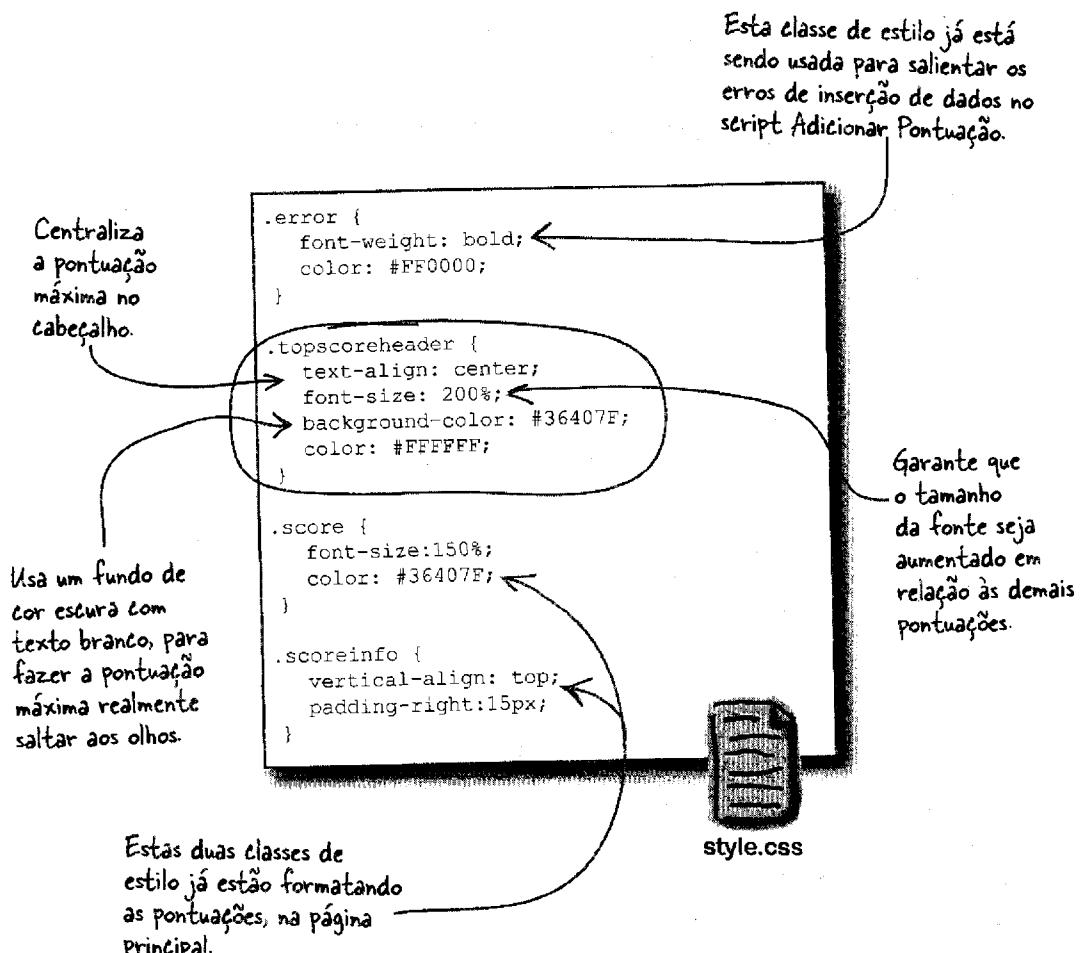
**P:** Muitas pontuações ainda estão sem verificação? Isso não é um problema?

**R:** É sim. Mas não nos impede de seguir adiante e incluir a funcionalidade de chamar a atenção para a pontuação máxima. A falta de verificação, na verdade, só significa que em algum momento nós teremos de limpar a lista, excluindo os registros sem comprovação. De fato, assim que terminarmos com esta funcionalidade de salientar a pontuação máxima, já passaremos para a questão das pontuações sem verificação.

*adicionando um pouco de css*

## Formate a pontuação máxima com HTML e CSS

A coisa mais importante sobre o cabeçalho que estamos incluindo é que ele será visto claramente acima de todas as demais pontuações da lista. Para isso, precisaremos de ajuda tanto do HTML quanto do CSS, para a parte visual. O cabeçalho será gerado como uma linha na tabela HTML, com um CSS especial aplicado a ela. Este estilo, `topscoreheader`, deve ser acrescentado à folha de estilo `style.css` do Guitar Wars.



O script `index.php` já está gerando uma tabela HTML contendo a lista das pontuações. Gerar um cabeçalho apenas para a pontuação máxima requer que isolemos o primeiro registro, o qual garantidamente será a maior pontuação, uma vez que a lista agora está ordenada. Um loop `while` é posto em ação, e nós precisamos de alguma forma de contar as pontuações e gerar o cabeçalho apenas para a primeira...



Complete o código do script `index.php` do Guitar Wars de modo que ele adicione um cabeçalho formatado para a pontuação máxima, usando o estilo CSS `topscoreheader`. Dica: Não se esqueça de que esse cabeçalho fará parte da tabela HTML contendo os recordes, a qual possui duas colunas.

```
...
// Faz um loop através do array contendo os dados das pontuações, formatando-
os como HTML
echo '<table>';
$i = 0;
while ($row = mysqli_fetch_array($data)) {
    // Exibe os dados das pontuações
    if (.....) {
        .....
        .....
    }
    echo '<tr><td class="scoreinfo">';
    echo '<span class="score">' . $row['score'] . '</span><br />';
    echo '<strong>Name:</strong> ' . $row['name'] . '<br />';
    echo '<strong>Date:</strong> ' . $row['date'] . '</td>';
    if (is_file(GW_UPLOADPATH . $row['Screenshot']) &&
        filesize(GW_UPLOADPATH . $row['Screenshot']) > 0) {
        echo '<td></td></tr>';
    }
    else {
        echo '<td></td></tr>';
    }
    .....
}
echo '</table>';
...
```



index.php

## exercício solução



Complete o código do script index.php do Guitar Wars de modo que ele adicione um cabeçalho formatado para a pontuação máxima, usando o estilo CSS topscoreheader. Dica: Não se esqueça de que esse cabeçalho fará parte da tabela HTML contendo os recordes, a qual possui duas colunas.

*fi é a variável que faz a conta das pontuações - nós podemos usá-la para isolar a primeira pontuação.*

...

```
// Faz um loop através do array contendo os dados das pontuações, formatando-os como HTML
echo '<table>';
$i = 0;
while ($row = mysqli_fetch_array($data)) {
    // Exibe os dados das pontuações
    if ( $i == 0 ) {
        echo '<tr><td colspan="2" class="topscoreheader">Top Score:</td>' . $row['score'] . '</td></tr>';
    }
    echo '<tr><td class="scoreinfo">';
    echo '<span class="score">' . $row['score'] . '</span><br />';
    echo '<strong>Name:</strong> ' . $row['name'] . '<br />';
    echo '<strong>Date:</strong> ' . $row['date'] . '</td>';
    if (is_file(GW_UPLOADPATH . $row['screenshot']) &&
        filesize(GW_UPLOADPATH . $row['screenshot']) > 0) {
        echo '<td></td></tr>';
    }
    else {
        echo '<td></td></tr>';
    }
    $i++;
}
echo '</table>';

Se fi for igual a 0, nós sabemos que esta é a primeira (ou seja, a maior) pontuação, e portanto, devemos gerar o código HTML para o cabeçalho.
```

*A classe de estilo topscoreheader encontra-se armazenada em style.css.*

*Incrementa o contador ao final do loop pelas pontuações - este código é o mesmo que fi = fi + 1;*



index.php



## TEST DRIVE

### Ordene as pontuações e dê destaque à maior delas.

Modifique o script `index.php` de modo que ele passe a usar a nova consulta `SELECT ordenada`, e depois, adicione o código que irá gerar o cabeçalho para a pontuação máxima. Envie o novo script para o seu servidor e abra-o no seu navegador, para ver a maior pontuação sendo exibida proeminenteamente.

Guitar Wars - High Scores

Welcome, Guitar Warrior, do you have what it takes to crack the high score list? If so, just add your own score.

| Top Score: 368420 |                      |                           |
|-------------------|----------------------|---------------------------|
| 368420            | Name: Ashton Simpson | Date: 2008-04-23 09:13:34 |
| 345900            | Name: Eddie Vanilli  | Date: 2008-04-23 09:06:35 |

Unverified!

Unverified!

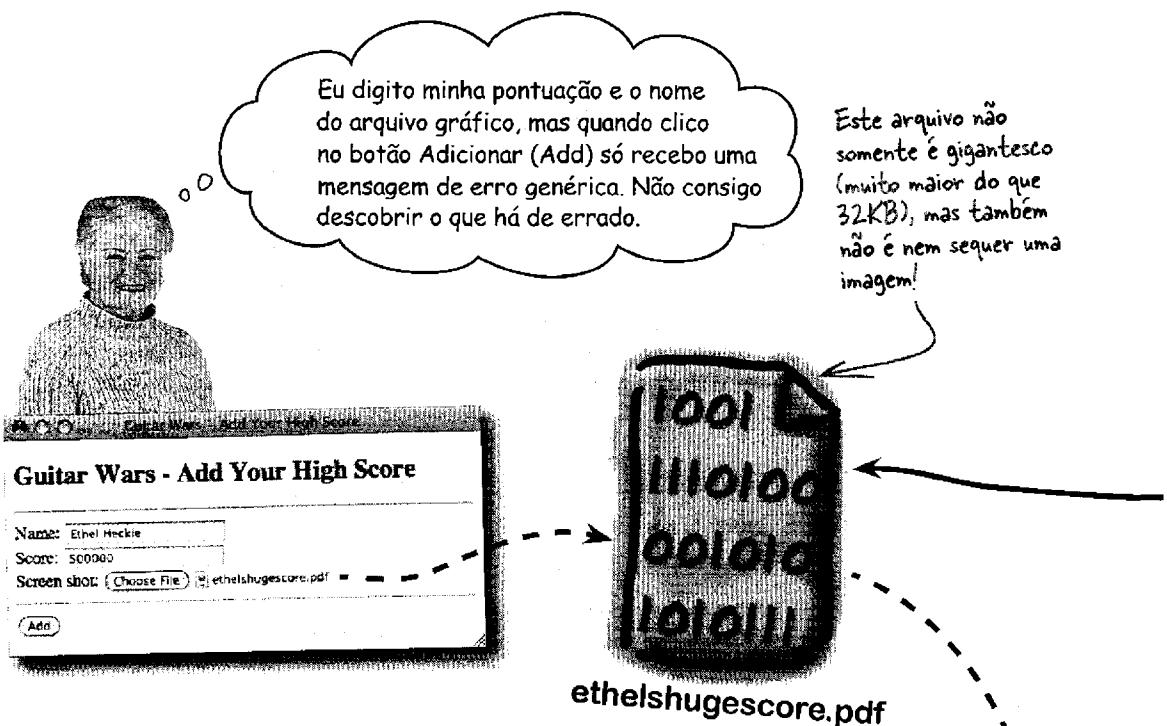
Legal que a ordem esteja correta... mas você sabe que qualquer uma dessas pontuações não-verificadas pode ser falsa.



### É verdade, ainda precisamos tratar das pontuações não-verificadas.

Mas uma coisa de cada vez. Parece que surgiu outro problema, que está impedindo as pessoas de enviarem os seus arquivos gráficos.

## adicionando restrição de tamanho para imagens

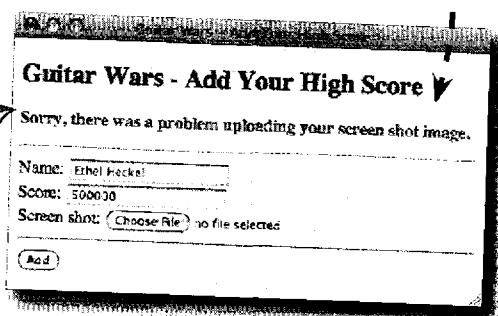


### O arquivo não só é gigantesco, ele nem é uma imagem!

Temos um problema: nosso formulário está rejeitando alguns arquivos sem dizer aos usuários por quê. Na verdade, é bom que ele esteja rejeitando arquivos, neste caso porque são grandes demais – lembre-se de que limitamos o tamanho dos arquivos a 32 KB, no código do formulário. Mas precisamos avisar ao usuário por que isso está acontecendo. E não apenas isso, nós também não queremos que os usuários enviem arquivos que não sejam imagens. Adicionar alguma validação ao formulário de Adicionar Pontuação permitirá que nós tenhamos maior controle sobre o modo como os arquivos são enviados.

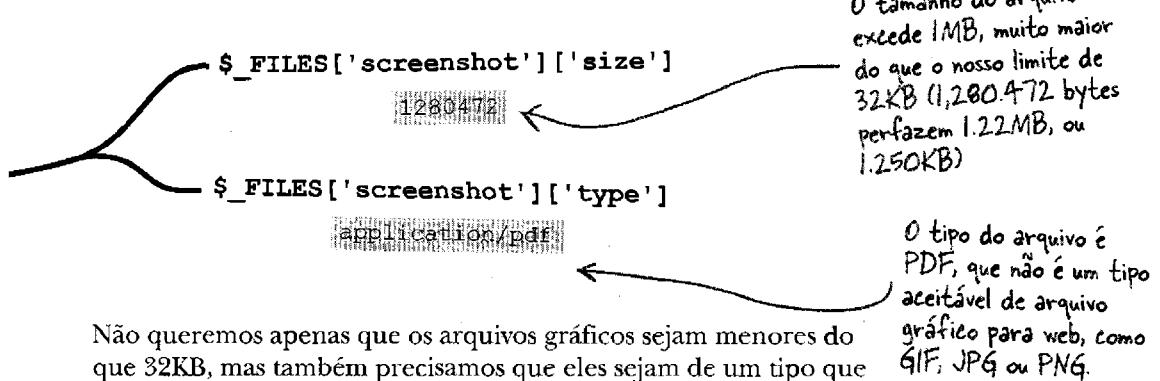
Assim, a validação no formulário addscore.php servirá a dois propósitos essenciais. Primeiramente, ela poderá aprimorar a prevenção contra uploads de arquivos grandes demais, fornecendo aos usuários uma notificação de que o arquivo não pode ter mais que 32 KB. E, em segundo lugar, ela poderá impedir que as pessoas enviem arquivos que não sejam imagens. O formulário precisará ter validação tanto para o tamanho quanto para o tipo do arquivo.

Esta mensagem de erro não diz muito ao usuário sobre o que houve de errado com a submissão do recorde.

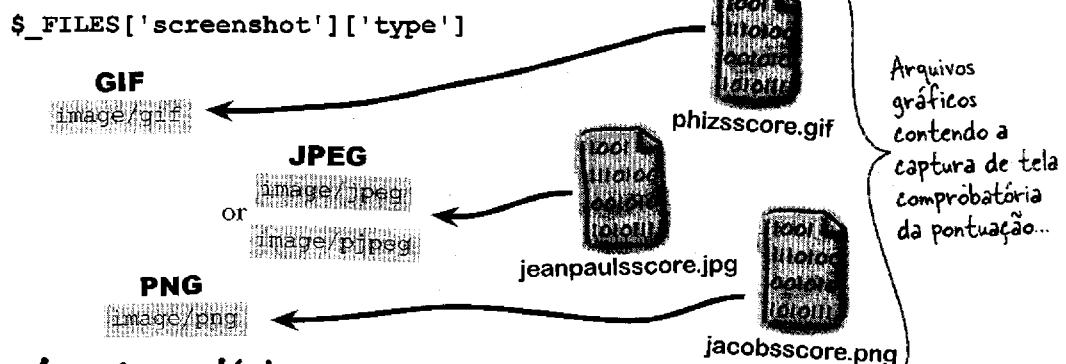


## São permitidas apenas imagens pequenas

Então, como exatamente nós verificamos o formulário Adicionar Pontuação e nos certificamos de que as imagens enviadas estarão obedecendo às limitações de tamanho e tipo? A resposta se encontra na variável superglobal `$_FILES`, da qual, se você se lembrar, nós obtivemos a localização de armazenamento temporário do arquivo enviado, para que pudéssemos movê-lo para a pasta `images`. Agora, iremos usá-la para obter o tamanho e o tipo MIME do arquivo.



Não queremos apenas que os arquivos gráficos sejam menores do que 32KB, mas também precisamos que eles sejam de um tipo que possa ser exibido como imagem web. Os seguintes tipos MIME são tipicamente usados para se representarem imagens web:



**Aponte seu lápis**

Escreva uma declaração `if` para verificar se um determinado arquivo é uma imagem, checando também se ele é maior do que 0 bytes e menor do que o determinado na constante `GW_MAXFILESIZE`. Assuma que o tamanho e o tipo do arquivo já tenham sido armazenados em variáveis chamadas `$screenshot_size` e `$screenshot_type`.

```
if ( _____ )
    _____
) (
```

*incorporando arquivo de validação no aplicativo*

 Aponte seu lápis  
Solução

Alguns navegadores usam este tipo MIME para reconhecer imagens JPEG.

Escreva uma declaração `if` para verificar se um determinado arquivo é uma imagem, checando também se ele é maior do que 0 bytes e menor do que o determinado na constante `GW_MAXFILESIZE`. Assuma que o tamanho e o tipo do arquivo já tenham sido armazenados em variáveis chamadas `$Screenshot_size` e `$Screenshot_type`.

```
para    if ((fscreenshot_type == 'image/gif') || (fscreenshot_type == 'image/jpeg') ||  
reconhecer  
imagens JPG,   (fscreenshot_type == 'image/pjpeg') || (fscreenshot_type == 'image/png')) && -  
              (fscreenshot_size > 0) && (fscreenshot_size <= GW_MAXFILESIZE))
```

```
<?php
// Define as constantes da aplicação
define('GW_UPLOADPATH', 'images/');
define('GW_MAXFILESIZE', 32768); // 32 KB
?>
```

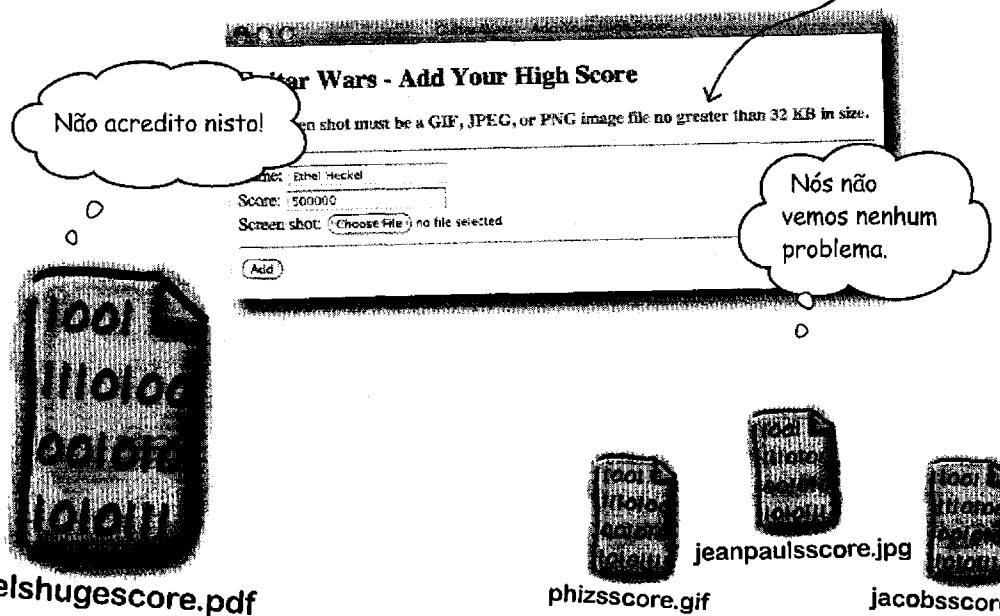
appvars.php

Uma vez que o tamanho máximo para o arquivo agora aparece em mais de um lugar no script Adicionar Pontuação, faz sentido armazená-lo na forma de uma constante.

A validação de arquivos torna a aplicação mais robusta

Um pouco de validação ajuda muito a tornar qualquer aplicação PHP mais intuitiva e fácil de usar, além de mais segura. Agora, uma mensagem de erro ajuda o usuário saber exatamente quais são as limitações impostas sobre os arquivos gráficos a serem enviados.

Uma mensagem de erro ajuda a explicar exatamente que tipos de arquivos podem ser enviados.



Uma vez que estamos tornando o script mais robusto, também é uma boa ideia verificar a superglobal `$_FILES` para nos certificarmos de que não houve um erro no upload.

Exibe uma mensagem de erro descriptiva, caso o arquivo seja do tipo errado, ou grande demais.

```

if (!empty($name) && !empty($score) && !empty($screenshot)) {
    if (($screenshot_type == 'image/gif') || ($screenshot_type == 'image/jpeg') ||
        ($screenshot_type == 'image/png') || ($screenshot_type == 'image/pjpeg')) &&
        ($screenshot_size > 0) && ($screenshot_size <= GW_MAXFILESIZE)) {
        if ($_FILES['screenshot']['error'] == 0)
            // Move o arquivo para a pasta-alvo
            $target = GW_UPLOADPATH . $screenshot;
            if (move_uploaded_file($_FILES['screenshot']['tmp_name'], $target)) {
                // Conecta-se ao banco de dados
                $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

                // Escreve os dados no banco
                $query = "INSERT INTO guitarwars VALUES (0, NOW(), '$name', '$score',
'$screenshot')";
                mysqli_query($dbc, $query);

                // Confirma êxito com o usuário
                echo '<p>= Obrigado por adicionar o seu recorde!</p>';
                echo '<p><strong>Name:</strong> ' . $name . '<br />';
                echo '<strong>Score:</strong> ' . $score . '<br />';
                echo '</p>';
                echo '<p><a href="index.php">&lt;&lt; Back to high scores</a></p>';

                // Limpa os dados da pontuação para limpar o formulário
                $name = "";
                $score = "";
                $screenshot = "";

                mysqli_close($dbc);
            }
        else {
            echo '<p class="error">Sorry, there... = Desculpe, houve um problema com o
arquivo que você tentou enviar.</p>';
        }
    }
    else {
        echo '<p class="error">O arquivo precisa ser um gráfico GIF, JPEG ou PNG com "menos
de ' . ' KB' . (GW_MAXFILESIZE / 1024) . ' KB de tamanho.</p>';
    }
}

// Tenta excluir o arquivo gráfico temporário
unlink($_FILES['screenshot']['tmp_name']);
}
else {
    echo '<p class="error">Por favor, digite todas as informações para adicionar sua
pontuação.</p>';
}

```

A função `unlink()` serve para apagar um arquivo do servidor. Nós suprimimos a sua mensagem de erro com `@`, no caso de o upload não ter se completado com sucesso.

O novo e aprimorado script Adicionar Pontuação agora tem a funcionalidade de verificação dos arquivos gráficos.



**test drive addscore.php**



---

## TEST DRIVE

---

### **Adicione a validação dos arquivos gráficos ao script Adicionar Pontuação.**

Modifique o script `addscore.php` de modo que ele passe a usar o novo código de validação dos arquivos gráficos. Envie o script para o seu servidor e teste o formulário Adicionar Pontuação com imagens válidas e inválidas (imagens grandes demais ou arquivos de outro tipo).

---

### *não existem* Perguntas Idiotas

**P:** Por que existem dois tipos MIME diferentes para imagens JPEG?

**R:** Essa é uma pergunta que pode ser mais bem respondida pelos fabricantes de navegadores, os quais, por algum motivo, decidiram usar tipos MIME diferentes para imagens JPEG. Para nos certificarmos de que a validação de arquivos JPEG funcionará na maior quantidade de navegadores possível, é necessário verificar ambos os tipos.

**P:** Por que é necessário verificar se os arquivos têm mais de 0 byte? Os arquivos gráficos não têm sempre mais do que 0 byte?

**R:** Na teoria, sim. Mas é tecnicamente possível criar um arquivo de 0 byte no servidor, caso o usuário especifique um arquivo que não existe no seu próprio computador. Apenas para se resguardar caso isso aconteça, `addscore.php` toma a atitude mais segura, qual seja a de verificar se o arquivo não está vazio.

**P:** Por que `GW_MAXFILESIZE` é colocado em `appvars.php`, uma vez que só é usado em `addscore.php`?

**R:** Embora seja verdade que `appvars.php` foi elaborado para armazenar dados que sejam compartilhados entre diversos scripts, ele também é um bom lugar para se armazenar quaisquer constantes. Neste caso, colocar `GW_MAXFILESIZE` em `appvars.php` facilita que a encontramos, caso você queira aumentar o limite de tamanho para os arquivos enviados.

**P:** Como funciona a linha de código que contém `@unlink()`?

**R:** A função interna do PHP `unlink()` apaga um arquivo do servidor, no nosso caso o arquivo temporário que foi enviado. Uma vez que é possível que o upload tenha falhado, e que não haja arquivo temporário, nós suprimimos quaisquer potenciais erros gerados por `unlink()` colocando um símbolo (`@`) antes dele. Você pode colocar `@` na frente de qualquer função PHP para suprimir o recurso de relatar erros dessa função.

E quanto às pontuações não verificadas?  
 Elas não desapareceram, se é que você não reparou.



### A lista das pontuações precisa ser limpa.

Com o upload dos arquivos gráficos aperfeiçoados graças à validação, não podemos continuar ignorando o problema das pontuações não verificadas. As novas pontuações, com imagens comprobatórias, não podem ficar atrás das pontuações antigas sem comprovação, as quais podem ou não ser legítimas. O Guitar Wars precisa de uma forma de remover pontuações antigas!

A pontuação máxima atual não está comprovada, o que não desperta muita confiança nos outros usuários..

**Guitar Wars - High Scores**

Welcome, Guitar Warlor, do you have what it takes to crack the high score list? If so, just add your own score.

| Top Score: 368420 |                      |                           |                    |
|-------------------|----------------------|---------------------------|--------------------|
| 368420            | Name: Ashton Simpson | Date: 2008-04-23 09:13:34 | <b>Unverified!</b> |
| 345900            | Name: Eddie Vanilli  | Date: 2008-04-23 09:06:35 | <b>Unverified!</b> |

**guitarwars**

| <b>id</b> | <b>date</b>         | <b>name</b>     | <b>score</b> | <b>scorefile</b> |
|-----------|---------------------|-----------------|--------------|------------------|
| 1         | 2008-04-22 14:37:34 | Paco Jastorius  | 127650       |                  |
| 2         | 2008-04-22 21:27:54 | Nevil Johansson | 98430        |                  |
| 3         | 2008-04-23 09:06:35 | Eddie Vanilli   | 345900       |                  |
| 4         | 2008-04-23 09:12:53 | Belita Chevy    | 282470       |                  |
| 5         | 2008-04-23 09:13:34 | Ashton Simpson  | 368420       |                  |
| 6         | 2008-04-23 14:09:50 | Kenny Lavitz    | 64930        |                  |
| 7         | 2008-04-24 08:13:52 | Phiz Lairston   | 186580       | phizsscore.gif   |

As pontuações sem imagens comprobatórias precisam ser removidas do banco de dados, para já!

Escreva aqui como você faria para limpar as pontuações não verificadas da lista:

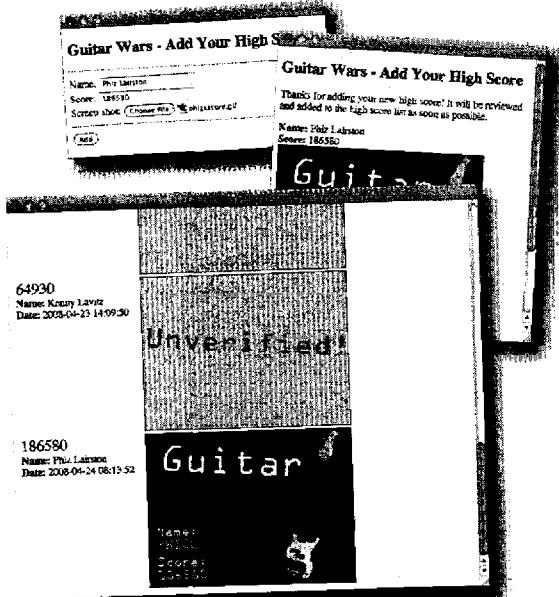
*adicionando uma página admin*

## Planeje uma página Admin

Uma vez que só precisamos remover algumas pontuações não verificadas do banco de dados, é perfeitamente razoável simplesmente acionar uma ferramenta SQL e remover linhas manualmente, com algumas consultas DELETE. Mas esta poderá não ser a última vez que tenha de remover uma pontuação, e não é divertido ter de usar consultas SQL manuais para fazer a manutenção de uma aplicação web. A ideia aqui é elaborar uma aplicação que possa ser mantida com o mínimo de trabalho possível.

O que precisamos é de uma página à qual somente o administrador do site tenha acesso, e que ele possa usar para remover pontuações... uma página Admin! Mas precisamos tomar muito cuidado na hora de distinguir quais partes do Guitar Wars são para o administrador e quais partes são para os usuários.

**Estas páginas são para os usuários:**



As aplicações web frequentemente incluem páginas para acesso público, bem como páginas admin apenas para os administradores fazerem a manutenção do site.

A página Adicionar Pontuação e a página principal do Guitar Wars são elaboradas para os usuários finais submeterem e visualizarem suas pontuações.

**Esta página é apenas para o administrador:**

A página Admin é elaborada apenas para uso do administrador do site – você não quer que os usuários finais possam sair por aí removendo pontuações.

| Guitar Wars - High Scores Administration   |                     |               |
|--|---------------------|---------------|
| Below is a list of all Guitar War high scores. Use this page to remove scores as needed. |                     |               |
| Achiles Stephan  | 2008-04-23 09:13:34 | 368470 Remove |
| Eddie Vanilla  | 2008-04-23 09:06:35 | 345500 Remove |
| Beth Chevy   | 2008-04-24 08:02:11 | 232470 Remove |
| Phil Lariviere   | 2008-04-24 08:13:52 | 184520 Remove |
| Dave Johnson   | 2008-04-24 08:02:11 | 127650 Remove |
| Nerf Jokinen   | 2008-04-24 08:02:11 | 98430 Remove  |
| Kenny Levine   | 2008-04-23 14:09:30 | 61930 Remove  |

Clicar em um link "Remove" apaga a respectiva pontuação.

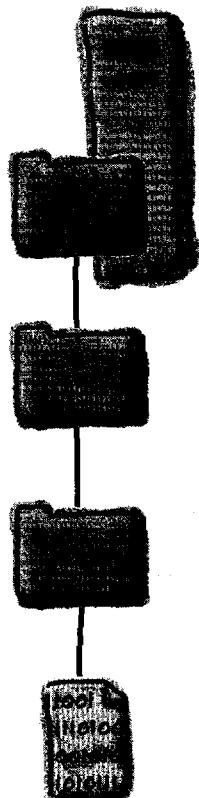


Escreva aqui o que os scripts Admin e de Remover Pontuação precisam fazer para implementarmos um recurso de remoção de pontuações no Guitar Wars. Em seguida, faça um esboço de como a remoção afetaria uma linha da tabela guitarwars e a sua respectiva imagem.



.....  
.....  
.....  
.....  
.....

### Servidor web



.....  
.....  
.....  
.....  
.....  
.....



removescore.php

**guitarwars**

| ID | Date                | Name            | Score  | Image         |
|----|---------------------|-----------------|--------|---------------|
| 1  | 2008-04-22 14:37:34 | Paco Jastorius  | 127650 |               |
| 2  | 2008-04-22 21:27:54 | Nevil Johansson | 98430  |               |
| 3  | 2008-04-23 09:06:35 | Eddie Vanilli   | 345900 |               |
| 4  | 2008-04-23 09:12:53 | Belfita Chevy   | 282470 |               |
| 5  | 2008-04-23 09:13:34 | Ashton Simpson  | 368420 |               |
| 6  | 2008-04-23 14:09:50 | Kenny Lavitz    | 64930  |               |
| 7  | 2008-04-24 08:13:52 | Phiz Lairston   | 186580 | phizscore.gif |

## exercício solução



Escreva aqui o que os scripts Admin e Remover Pontuação precisam fazer para implementarmos um recurso de remoção de pontuações no Guitar Wars. Em seguida, faça um esboço de como a remoção afetaria uma linha da tabela guitarwars e a sua respectiva imagem.

| Guitar Wars - High Scores Administration  |                     |               |
|---|---------------------|---------------|
| Below is a list of all Guitar Wars high scores. Use this page to remove scores as needed. |                     |               |
| Ashton Simpson  | 2008-04-23 09:13:34 | 368420 Remove |
| Eddie Vanilli   | 2008-04-23 09:06:35 | 345900 Remove |
| Belita Chevy  | 2008-04-24 08:02:11 | 282470 Remove |
| Phiz Lairston   | 2008-04-24 08:13:52 | 186580 Remove |
| Paco Jastorius  | 2008-04-24 08:02:11 | 127650 Remove |
| Nevil Johansson   | 2008-04-24 08:02:11 | 98430 Remove  |
| Kenny Lavitz  | 2008-04-23 14:09:50 | 64930 Remove  |

O script `admin.php` lista todas as linhas da tabela, cada uma com um link "Remove" ao lado, o qual repassa informações para o script Remover Pontuação.

admin.php

### Servidor web

O script `removescore.php` remove propriamente o registro do banco de dados, exclui o arquivo gráfico do servidor, e depois exibe uma mensagem de confirmação.

### Guitar Wars - Remove a High Score

Are you sure you want to delete the following high score?

Name: Ashton Simpson

Date: 2008-04-23 09:13:34

Score: 368420

Remove

### Guitar Wars - Remove a High Score

The high score of 368420 for Ashton Simpson was successfully removed.

<< Back to admin page

removescore.php

### guitarwars

| # | Date                | Name            | Score  | screenshot    |
|---|---------------------|-----------------|--------|---------------|
| 1 | 2008-04-22 14:37:34 | Paco Jastorius  | 127650 |               |
| 2 | 2008-04-22 21:27:54 | Nevil Johansson | 98430  |               |
| 3 | 2008-04-23 09:06:35 | Eddie Vanilli   | 345900 |               |
| 4 | 2008-04-23 09:12:53 | Belita Chevy    | 282470 |               |
| 5 | 2008-04-23 09:42:24 | Ashton Simpson  | 368420 |               |
| 6 | 2008-04-23 14:09:50 | Kenny Lavitz    | 64930  |               |
| 7 | 2008-04-24 08:13:52 | Phiz Lairston   | 186580 | phizscore.gif |

Embora esta linha em particular não tenha um arquivo gráfico associado, o script Remover Pontuação terá de estar pronto para excluir do servidor os arquivos de pontuações que os tenham associados.

## Gere links para remoção de pontuações na página Admin

Embora o script Remover Pontuação seja responsável pela remoção propriamente dita, nós precisamos de um script Admin que nos permita selecionar uma pontuação a ser eliminada. O script admin.php gera uma lista de pontuações, com links “Remove” para cada uma. Esses links repassam os dados referentes à respectiva pontuação para o script removescore.php.

&lt;?php

```
require_once('appvars.php');
require_once('connectvars.php');

// Conecta-se ao banco de dados
$dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
```



admin.php

```
// Obtém os dados das pontuações a partir do MySQL
$query = "SELECT * FROM guitarwars ORDER BY score DESC, date ASC";
$data = mysqli_query($dbc, $query);
```

```
// Faz um loop através do array contendo os dados das pontuações, formatando-os como HTML
echo '<table>';
while ($row = mysqli_fetch_array($data)) {
    // Exibe os dados das pontuações
    echo '<tr class="scorerow"><td><strong>' . $row['name'] . '</strong></td>';
    echo '<td>' . $row['date'] . '</td>';
    echo '<td>' . $row['score'] . '</td>';
    echo '<td><a href="removescore.php?id=' . $row['id'] . '&name=' . $row['name'] . '&date=' . $row['date'] . '&score=' . $row['score'] . '&Screenshot=' . $row['Screenshot'] . '">remove</a></td></tr>';
}
echo '</table>';

mysqli_close($dbc);
?>
```

A URL para o script Remover  
Pontuação está fazendo mais do que  
apenas criar um link para o script...  
está também repassando dados a ele!

Este código gera um link HTML para o  
script removescore.php, enviando a ele  
informações sobre a pontuação a ser excluída.

```
<a href="removescore.php?id=5&date=2008-04-23&2009:13:34&name=Ashton&20Simpson&score=368420&Screenshot=">
```

## Os scripts são capazes de comunicarem-se uns com os outros

Para o script Remover Pontuação poder excluir um registro, ele precisa saber qual. Mas isso é decidido no script Admin. Isso leva à pergunta "como o script Admin informa a Remover Pontuação qual registro deve ser excluído?" Essa comunicação entre os scripts é realizada empacotando-se os dados como parte de uma URL "Remove" para cada pontuação mostrada na página Admin. Se você analisar os detalhes da URL de uma determinada pontuação, perceberá que todos os dados referentes a ela se encontram lá.

A URL de um script pode ser usada para enviar dados na forma de uma requisição GET.

```
<a href="removescore.php?  
id=5&  
date=2008-04-23%2009:13:34&  
name=Ashton%20Simpson&  
score=368420&Screenshot="">Remove</a>
```

Cada informação tem um nome e um valor, e é separada de outros pares de nome/valor por um símbolo &.

A URL de "Remove" cria links para o script removescore.php, mas também inclui dados referentes à linha a ser excluída.

Guitar Wars - High Scores Administration		
Below is a list of all Guitar Wars high scores. Use this page to remove scores as needed.		
Ashton Simpson	2008-04-23 09:13:34	368420
Eddie Vanille	2008-04-23 09:06:35	345900
Bella Chevy	2008-04-24 08:02:11	282470
Plitz Lairiston	2008-04-24 08:13:52	186580
Paco Jastorius	2008-04-24 08:02:11	127650
Nevil Johansson	2008-04-24 08:02:11	98430
Kenny Lavitz	2008-04-23 14:09:30	64930

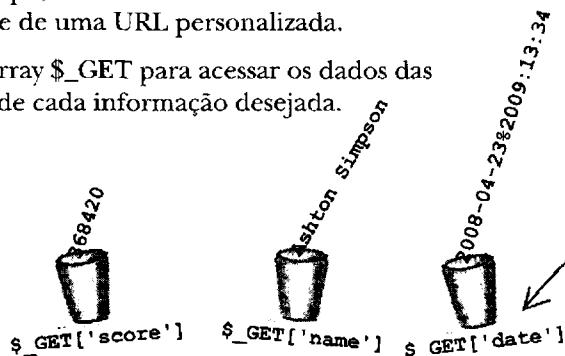
Clicar neste link não só abre o script Remover Pontuação, como também envia dados, na forma de uma requisição GET.

OK, então os dados são enviados através de uma URL, mas como exatamente o script Remover Pontuação põe as mãos nesses dados? Os dados enviados a um script através de uma URL ficam disponíveis na superglobal `$_GET`, que é um array bastante semelhante a `$_POST`. Empacotar dados em uma URL linkada é o mesmo que usar uma requisição GET em um formulário web. Em uma requisição GET tradicional do HTML, os dados do formulário são automaticamente enviados para o script que os processará, como parte da URL desse script. Estamos fazendo o mesmo aqui, ao criarmos manualmente a nossa própria requisição GET como parte de uma URL personalizada.

Assim como `$_POST`, para usar o array `$_GET` para acessar os dados das pontuações você precisa do nome de cada informação desejada.

O nome da informação é usado para acessá-la dentro do array `$_GET`.

A URL que leva a um script serve como uma forma de enviar dados importantes, como por exemplo, a ID de uma linha do banco de dados.



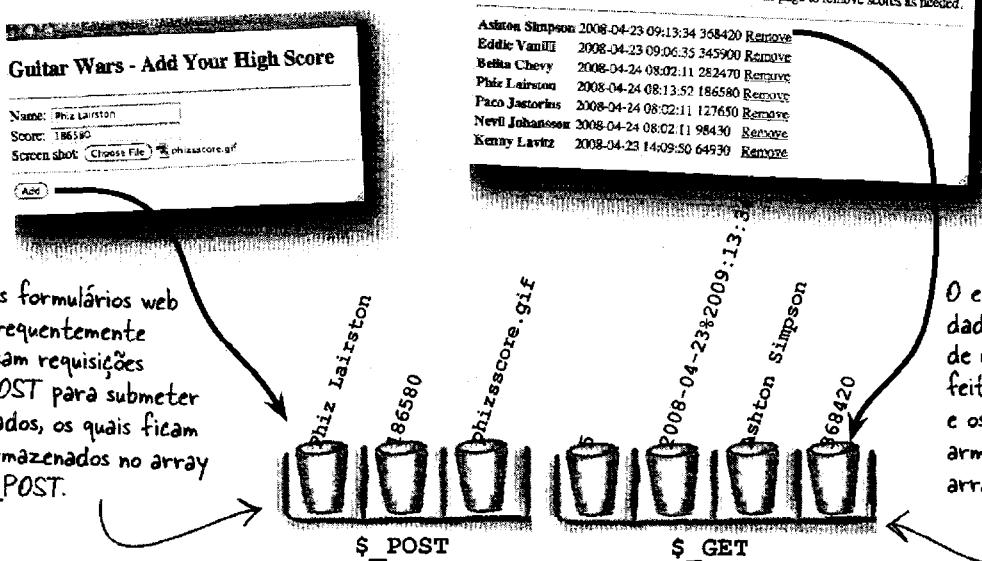


Não entendo por que tanto entusiasmo com GET. Por que não podemos passar os dados para o script usando POST? É assim que vimos fazendo até agora.

### **As requisições POST só podem ser iniciadas por um formulário, enquanto que as requisições GET podem ser empacotadas como URLs.**

Até o momento, nós sempre enviamos dados aos scripts a partir de um formulário, no qual o script encontrava-se listado como a ação para o botão Submit. Quando o usuário preenche o formulário e clica no botão Submit, os dados são empacotados e enviados para o formulário como uma requisição POST.

O problema é que a página Admin não usa um formulário para iniciar o script Remover Pontuação. Ela apenas estabelece links para o script, via uma URL. Assim, nós precisamos de uma forma de enviar dados a um script usando nada mais do que uma URL. É aqui que GET é particularmente útil, uma vez que fornece acesso a dados que estão empacotados em uma URL como parâmetros. De forma semelhante a POST, os dados que são enviados ao script através de uma requisição GET ficam disponíveis através de uma superglobal, mas esta é chamada de `$_GET` em vez de `$_POST`.



## Sobre GETs e POSTs

A diferença entre GET e POST não é apenas de formulário versus URL, uma vez que as requisições GET podem ser (e frequentemente são) usadas para se submeter dados de formulário, também. A distinção real entre GET e POST tem a ver com a intenção da requisição. GET é usada principalmente para se obter dados do servidor sem afetar nada nele. POST, por outro lado, geralmente envolve o envio de dados para o servidor, após o que o estado do servidor normalmente se modifica, de alguma forma, em resposta aos dados enviados.

Os dois tipos de requisições web, GET e POST, controlam o modo como você envia dados de e para scripts.

### POST

Usado para enviar dados ao servidor que causem, de alguma forma, uma mudança no estado deste, como por exemplo, a inserção de dados em um banco. Os dados ainda podem ser retornados em uma resposta. Ao contrário de GET, as requisições POST só podem ser feitas mediante a ação de um formulário web. Também ao contrário de GET, os dados enviados em uma requisição POST ficam escondidos.

### GET

Geralmente usada para obtenção de dados que não modifiquem nada no servidor. Para quantidades pequenas de dados, GET também é útil para enviá-los diretamente ao servidor, em uma URL. Ao contrário de POST, GET serve principalmente para enviar pequenas quantidades de dados.

## Perguntas Ídiotas

**P:** Já vi formulários web que usavam GET. Como isso funciona?

**R:** Tanto GET quanto POST têm o seu lugar, no que se refere a formulários web. Ao se criar um, o atributo method da tag <form> controla o modo como os dados são enviados, enquanto que o atributo action identifica o script que deverá receber e processar os dados:

```
<form method="post" action="addscore.php">
```

Quando o botão submit é clicado para submeter este formulário, o script addscore.php é executado, e os dados são enviados a ele através do array \$\_POST. Mas você poderia muito bem ter escrito a tag <form> desta maneira, em cujo caso os dados seriam enviados através do array \$\_GET:

```
<form method="get" action="addscore.php">
```

**P:** Ah, então não faz diferença se eu uso o método GET ou POST?

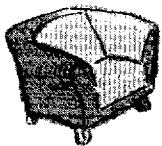
**R:** Errado. Faz bastante diferença. GET geralmente é usado para se obter dados de um servidor, sem modificar nada nele. Assim, GET é perfeito para formulários que

requerem informações do servidor sem alterar o seu estado, como por exemplo, ao selecionar linhas de um banco de dados. POST, por outro lado, é mais adequado para requisições que afetem o estado do servidor, como por exemplo, emitir consultas INSERT ou DELETE que modifiquem o banco de dados. Outra distinção entre GET e POST é que os dados enviados através de GET ficam visíveis em uma URL, enquanto que os dados de POST ficam escondidos e, portanto, um pouquinho mais seguros.

**P:** Como essa distinção entre GET e POST se reflete no envio de dados a um script através de uma URL?

**R:** Bem, antes de tudo, você só pode enviar dados a um script através de uma URL se estiver usando uma requisição GET, portanto POST é eliminado imediatamente. Além disso, uma vez que GET é usado unicamente com requisições que não alterem o estado do servidor, isso significa que você não deve fazer INSERTs, DELETE FROMs ou qualquer outra coisa que modifique o banco de dados em um script que esteja recebendo dados através da sua URL.

## Conversa Informal



Esta noite falando: **GET e POST**

### **GET:**

Mas então, fiquei sabendo que você anda dizendo que eu só presto para responder perguntas, mas não consigo fazer nada com as respostas. É verdade?

OK; é verdade que eu não fui elaborado para causar modificações no servidor, como apagar arquivos ou adicionar linhas ao banco de dados, mas isso não quer dizer que não sou importante.

É verdade, mas você fica permanentemente grudado com o seu amiguinho, o Formulário, enquanto que ele e eu somos apenas conhecidos. Eu tenho outras amizades, como por exemplo, URL.

Bem, então eu tenho uma pergunta para você. Como exatamente você age quando o seu ajudante, o Formulário, não está disponível. Você sabe que às vezes a Página não acha necessário se dar ao trabalho de requisitar os serviços do Formulário

Calma aí. Só estou ressaltando que, embora a minha função seja obter dados do servidor, eu sou razoavelmente flexível no modo como posso ser usado para isso.

Fico feliz de ouvi-lo. Foi um prazer conversar com você...

### **POST:**

Claro que é. Vamos encarar, você não tem nenhuma capacidade além da de pedir algo ao servidor.

Se você está dizendo. Tudo o que eu sei é que seria difícil fazer algo acontecer sem pessoas como eu, que realmente trabalham. Se o servidor ficasse sempre no mesmo estado, seria tudo bem chato na aplicação.

Então você acha que o seu “círculo de amizades” de alguma forma compensa a sua incapacidade de agir? Não sei, não.

Olha, o Formulário é meu amigo, e há muito tempo eu me comprometi a não fazer nenhuma requisição sem ele. Pode julgar a minha lealdade se quiser, mas eu não vou trair meu amigo!

Ok, admito que você tem razão nisso.

*como removescore.php vai funcionar*

## GET, POST e a remoção de pontuações

Nós já sabemos que a remoção de pontuações no Guitar Wars começa com um link “Remove” na página Admin, o qual aponta para o script Remover Pontuação. Nós sabemos também que os dados podem ser enviados através da URL do link para o script. Mas temos um problema: uma requisição GET não deve fazer modificações no servidor, como apagar uma pontuação. Uma possível solução é não mudar nada no servidor... ainda. E se o script Remover Pontuação, inicialmente, exibisse uma página de confirmação antes de realmente excluir a pontuação do banco de dados?

The screenshot shows a web page titled "Guitar Wars - Remove a High Score". It asks the user if they are sure they want to delete the following high score. Below this is a table with a single row of data: Name: Ashton Simpson, Date: 2008-04-23 09:13:34, and Score: 368420. At the bottom is a form with two radio buttons labeled "Yes" and "No", and a "Submit" button. A link "[<< Back to admin page](#)" is at the bottom left.

Uma página de confirmação dá ao usuário uma chance de confirmar a remoção, em vez de simplesmente executá-la instantaneamente.

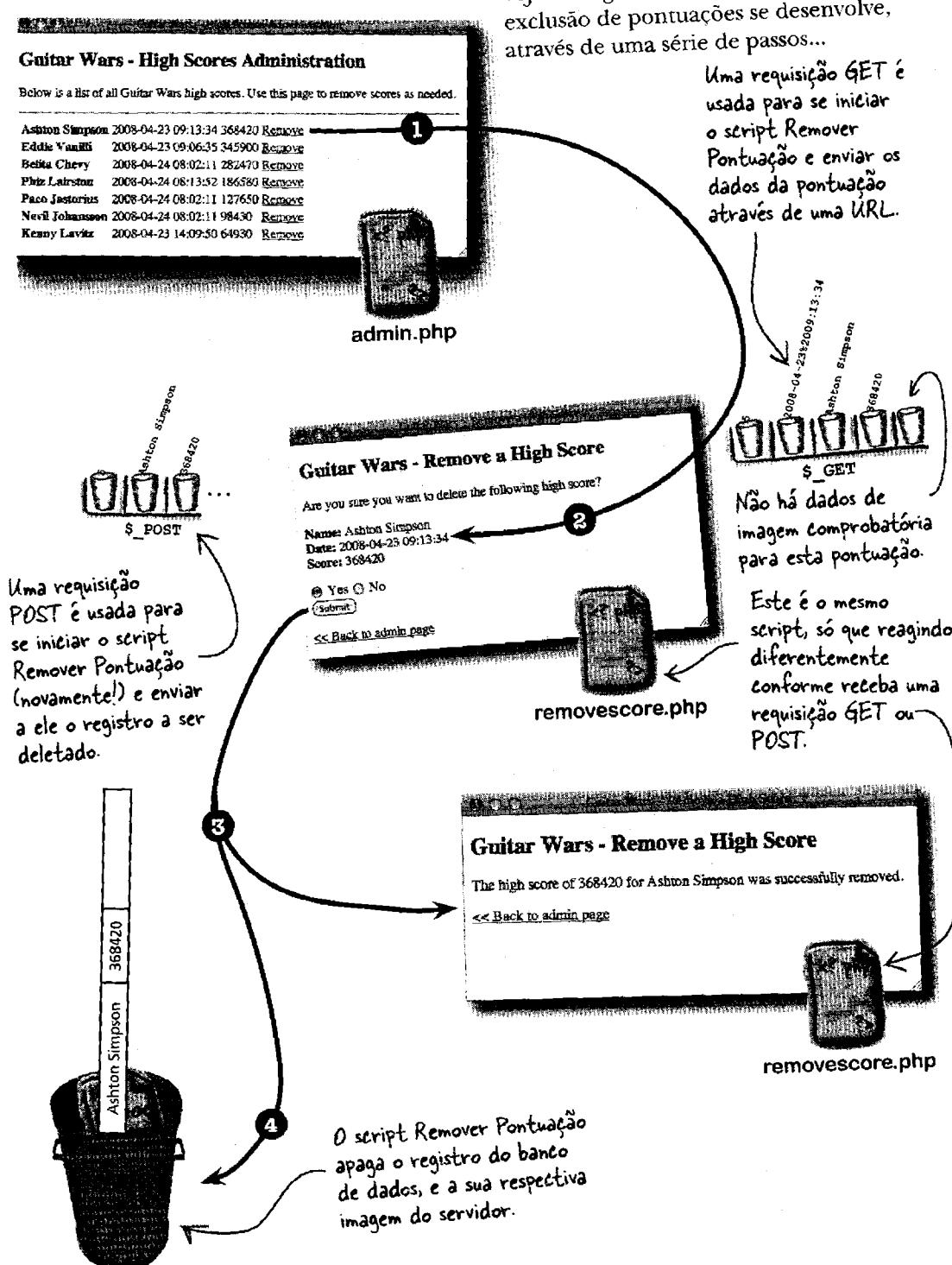
É totalmente possível, e até útil em alguns casos, que o mesmo script responda tanto a requisições GET quanto POST.

A página de confirmação mostra o registro a ser removido, junto com um simples formulário de Yes/No. Selecionar Yes e clicar no botão Submit resulta em a pontuação ser removida, enquanto que escolher No cancela a operação.

Pensando em termos de GETs e POSTs, o script Remover Pontuação pode exibir a página de confirmação como uma resposta à requisição GET do script Admin. E uma vez que a própria confirmação é um formulário, ela pode emitir a sua própria requisição POST quando submetida. Se o formulário for autorreferente, o mesmo script (removescore.php) poderá processar o POST e executar a remoção. Eis os passos desse processo:

- 1 O script Remover Pontuação é iniciado, através de uma requisição GET, pelo usuário ao clicar no link “Remove” da página Admin.**
- 2 O script Remover Pontuação usa os dados armazenados no array `$_GET` para gerar um formulário de confirmação da exclusão.**
- 3 O script Remover Pontuação é iniciado novamente, desta vez através de uma requisição POST do usuário que está submetendo o formulário de confirmação.**
- 4 O script Remover Pontuação apaga o registro selecionado do banco de dados, e também apaga do servidor o arquivo gráfico contendo a imagem comprobatória.**

Vejamos agora como o processo de exclusão de pontuações se desenvolve, através de uma série de passos...



<sup>não existem</sup>  
**Perguntas Idiotas**

**P:** Como é possível o mesmo script processar tanto requisições GET quanto POST?

**R:** A resposta tem a ver com o modo como o script é chamado. No caso do script Remover Pontuação, ele é chamado de duas formas diferentes. A primeira é quando o usuário clica em um link “Remove” da página Admin, em cujo caso uma URL o leva até o script. Uma vez que os dados são empacotados dentro da URL, essa é considerada uma requisição GET. Ela faz o script gerar um formulário web, cuja ação aponta de volta para o mesmo script Remover Pontuação. Assim, quando o usuário submete o formulário, o script é chamado pela segunda vez. Mas, ao contrário da primeira, não há nenhuma URL contendo dados, e, portanto, nenhuma requisição GET. Em vez disso, os dados da pontuação a ser removida são enviados através de uma requisição POST e ficam, portanto, disponíveis no array \$\_POST.

**P:** Então a forma como o script é chamado é o que realmente determina o que ele faz?

**R:** Sim! Quando o script vê que alguns dados foram enviados através de uma URL, na forma de uma requisição GET, ele sabe que deve exibir um formulário de confirmação, em vez de apagar qualquer coisa do banco de dados. Assim, os dados enviados no array \$\_GET são usados apenas dentro da página de confirmação, e não têm nenhum efeito duradouro no servidor. Quando o script vê que há dados sendo enviados através de uma requisição POST, ele sabe que pode apagar os dados do banco. Assim, ele usa o array \$\_POST para acessar os dados e criar uma consulta DELETE FROM que apaga a pontuação. E uma vez que a maioria das pontuações tem um respectivo arquivo gráfico armazenado no servidor, o script também apaga esse arquivo.

## Isole a maior pontuação para ser removida

Com o processo de remoção explicado, podemos agora concentrar nossa atenção no lado do banco de dados. O script Remover

Pontuação é responsável por excluir um registro de pontuação, o que implica excluir uma linha do banco. Como você deve se lembrar, a instrução SQL DELETE FROM nos permite apagar linhas. Mas, para fazê-lo, precisamos primeiramente achar a linha. Isso é feito colocando-se uma cláusula WHERE na consulta que usa DELETE FROM. Por exemplo, esta consulta SQL apaga a linha cuja coluna name está definida como 'Ashton Simpson':

Esta consulta apaga linhas cuja coluna name contenha o texto 'Ashton Simpson'.

```
DELETE FROM guitarwars WHERE name = 'Ashton Simpson'
```

ID	date	name	score	screenshot
1	2008-04-22 14:37:34	Paco Jastorius	127650	
2	2008-04-22 21:27:54	Nevil Johansson	98430	
3	2008-04-23 09:06:35	Eddie Vanilli	345900	
4	2008-04-23 09:12:53	Belita Chevy	282470	
5	2008-04-23 09:13:34	Ashton Simpson	368420	
6	2008-04-23 14:09:50	Kenny Lavitz	64930	
7	2008-04-24 08:13:52	Phiz Lairston	186580	phizsscore.gif

O nome da tabela é obrigatório com DELETE FROM, para que a instrução saiba de que tabela você está apagando dados.

O nome do usuário é o critério usado para se apagar a pontuação.

Há, porém, um problema com esta consulta. Em um mundo de milhões de Guitar Warriors, existe a possibilidade de haver mais de um Ashton Simpson. Esta consulta não apaga somente uma linha, ela exclui todas as linhas que tenham o nome 'Ashton Simpson'. A consulta precisa de mais informações para apagar a linha certa:

Ao se especificar a pontuação junto com o nome, a operação fica muito mais precisa.

```
DELETE FROM guitarwars WHERE name = 'Ashton Simpson' AND score = 368420!
```

ID	date	name	score	screenshot
1	2008-04-22 14:37:34	Paco Jastorius	127650	
2	2008-04-22 21:27:54	Nevil Johansson	98430	
3	2008-04-23 09:06:35	Eddie Vanilli	345900	
4	2008-04-23 09:12:53	Belita Chevy	282470	
5	2008-04-23 09:13:34	Ashton Simpson	368420	
6	2008-04-23 14:09:50	Kenny Lavitz	64930	
7	2008-04-24 08:13:52	Phiz Lairston	186580	phizsscore.gif

O operador AND modifica a consulta, de modo que tanto o nome quanto a pontuação precisam corresponder ao que está sendo pedido.

Agora que tanto o nome quanto a consulta precisam corresponder ao que está sendo pedido, a chance de se apagar acidentalmente mais de uma linha diminui consideravelmente.

*colocando um LIMIT no seu DELETE*

## Controle o quanto você pode remover com LIMIT

Usar as colunas de nome e de pontuação como base para se excluir uma linha é bom... mas não é suficiente. O desenvolvimento de aplicações tem que se pautar pela minimização dos riscos a qualquer custo, e ainda há um pequeno risco de se apagar mais de uma linha que tenham o mesmo nome e a mesma pontuação. A solução é forçar a consulta a só apagar uma linha, não importa o que aconteça. A cláusula LIMIT faz isso acontecer:

```
DELETE FROM guitarwars WHERE name = 'Ashton Simpson' AND score =  
'368420' LIMIT 1
```

Para ter o máximo de segurança,  
coloque um limite sobre o número de  
linhas que podem ser apagadas.

O número apóis LIMIT diz ao MySQL o número máximo de linhas que podem ser apagadas – neste caso, uma. Assim, fica garantido que nunca apagaremos mais de uma linha com esta consulta. Mas e se houver dois Ashton Simpsons com a mesma pontuação? É claro que este é um cenário improvável, mas às vezes vale a pena considerar cenários extremos ao se pensar no melhor projeto possível para uma aplicação.

*guitarwars*

<b>id</b>	<b>date</b>	<b>name</b>	<b>score</b>	<b>scratcheshot</b>
1	2008-04-22 14:37:34	Paco Jastorius	127650	
2	2008-04-22 21:27:54	Nevil Johansson	98430	
3	2008-04-23 09:06:35	Eddie Vanilli	345900	
4	2008-04-23 09:12:53	Belita Chevy	282470	
5	2008-04-23 09:13:34	Ashton Simpson	368420	
6	2008-04-23 14:09:50	Kenny Lavitz	64930	
7	2008-04-24 08:13:52	Phiz Lairston	186580	phizsscore.gif
...				
523	2008-11-04 10:03:21	Ashton Simpson	368420	ashtonsscore.jpg

Duas linhas com  
exatamente o mesmo  
nome e a mesma  
pontuação apresentam  
um problema para  
a nossa consulta  
DELETE.

Escreva abaixo o que acontece com esta tabela quando a instrução DELETE acima é executada. Como você poderia certificar-se de que a pontuação correta seja apagada?

.....

.....

.....

Faria alguma diferença usar a ID da pontuação na cláusula WHERE da consulta DELETE FROM? Isso poderia ajudar a garantir que estaremos apagando a pontuação certa, não?



### Sim, poderia! A ID da pontuação é a maneira perfeita de se isolar o registro para ser apagado.

A unicidade é uma das principais vantagens de se criar chaves primárias para as suas tabelas. A coluna id da tabela guitarwars é a chave primária e, portanto, é única para cada pontuação. Usando essa coluna na cláusula WHERE da consulta DELETE FROM, nós eliminamos toda a dúvida sobre qual registro será excluído. Eis uma nova consulta, desta vez usando a coluna id para nos ajudar:

```
DELETE FROM guitarwars WHERE id = 5
```



Apagar dados com base em uma chave primária ajuda a garantir precisão ao se isolar à linha certa para ser excluída.

Havendo a confiança de que a coluna id é de fato a chave primária, este código apagará de forma segura apenas uma linha. Mas e se não tiver sido você quem criou o banco de dados, e talvez a unicidade não tenha sido devidamente implementada. Nesse caso, uma cláusula LIMIT poderia ainda ser útil. O raciocínio é este: sempre que você quiser afetar apenas uma linha, deixe isso claro na consulta.

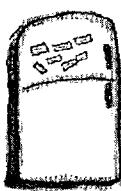
```
DELETE FROM guitarwars WHERE id = 5 LIMIT 1
```



A cláusula LIMIT diz explicitamente que a consulta não pode apagar mais de uma linha.

Nunca é má ideia ser bastante explícito com o que você quer que seja feito em uma consulta, e neste caso, LIMIT adiciona uma medida adicional de segurança à consulta DELETE.

O script removescore.php quase pronto



## Ímas de Geladeira PHP e MySQL

O script removescore.php está quase pronto, mas ainda faltam algumas partes importantes do código. Use os ímas para preencher as lacunas no código e implementar no Guitar Wars a capacidade de eliminar pontuações indesejadas.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title>Guitar Wars - Remover uma Pontuação</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
    <h2>Guitar Wars - Remove a High Score</h2>

<?php

    ..... ('appvars.php');

    ..... ('connectvars.php');

if (isset($_GET['id'])) && isset($_GET['date']) && isset($_GET['name']) &&
    isset($_GET['score']) && isset($_GET[ ..... ]) {

    // Pega os dados em GET
    $id = $_GET['id'];
    $date = $_GET['date'];
    $name = $_GET['name'];
    $score = $_GET['score'];

    ..... = $_GET[ ..... ];

}

else if (isset($_POST['id']) && isset($_POST['name']) && isset($_POST['score'])) {
    // Pega os dados em POST

    ..... = $_POST[ ..... ];

    $name = $_POST['name'];
    $score = $_POST['score'];
    $screenshot = $_POST ['screenshot'];
}

else {
    echo '<p class="error">= Desculpe, nenhuma pontuação foi especificada para ser
removida.</p>';
}

if (isset($_POST['submit'])) {

    if ($_POST['confirm'] == ..... ) {

        // Exclui o arquivo gráfico do servidor
        unlink(GW_UPLOADPATH . $screenshot);

        // Conecte no banco de dados
        $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
    }
}
```

```

// Exclui os dados da pontuação do banco

$query = "..... guitarwars WHERE ..... LIMIT .....

";
mysqli_query($dbc, $query);
mysqli_close($dbc);

// Confirma êxito com o usuário
echo '<p>A pontuação de ' . ' para ' . ' foi removida com sucesso.';
}
else {
    echo '<p class="error">A pontuação não foi removida.</p>';
}

}

else if (isset( ..... ) && isset( ..... ) && isset( ..... ) &&
isset($score) && isset($screenshot)) {
echo '<p>Tem certeza de que deseja apagar a pontuação abaixo?</p>';
echo '<p><strong>Nome: </strong>' . $name . '<br /><strong>Data: </strong>' . $date .
'<br /><strong>Pontuação: </strong>' . $score . '</p>';
echo '<form method="post" action="removescore.php">';
echo '<input type="radio" name="confirm" value="Yes" /> Yes ';
echo '<input type="radio" name="confirm" value="No" checked="checked" /> No <br />';
echo '<input type="submit" value="Submit" name="submit" />';

echo '<input type="hidden" name= ..... value="" ..... . '" />';

echo '<input type="hidden" name="name" value="' . $name . '" />';
echo '<input type="hidden" name="score" value="' . $score . '" />';
echo '<input type="hidden" name="screenshot" value="'.$screenshot.'" />';

}

echo '<p><a href= ..... >&lt;&lt; Voltar para a página admin</a></p>';

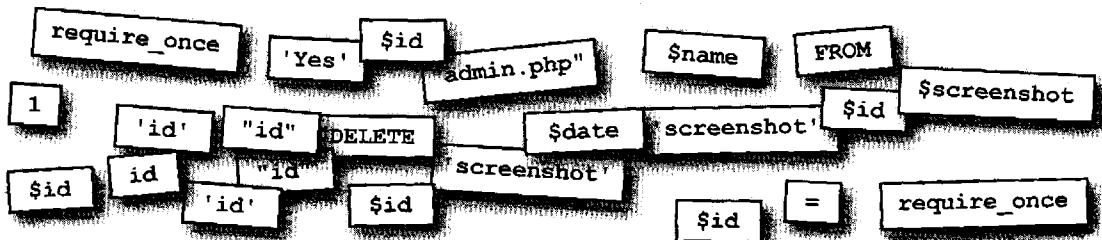
?>

</body>
</html>

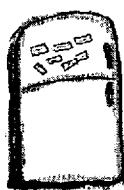
```



removescore.php



## o script removescore.php finalizado



# Ímas de Geladeira PHP e MySQL - Solução

O script removescore.php está quase pronto, mas ainda faltam algumas partes importantes do código. Use os ímas para preencher as lacunas no código e implementar no Guitar Wars a capacidade de eliminar pontuações indesejadas.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title>Guitar Wars - Remover uma Pontuação</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
    <h2>Guitar Wars - Remove a High Score</h2>
    <?php
        require_once ('appvars.php');
        require_once ('connectvars.php');

        if (!isset($_GET['id'])) && !isset($_GET['date']) && !isset($_GET['name']) &&
            !isset($_GET['score']) && !isset($_GET[ 'screenshot' ]) {
            // Pega os dados em GET
            $id = $_GET['id'];
            $date = $_GET['date'];
            $name = $_GET['name'];
            $score = $_GET['score'];

            $screenshot = $_GET[ 'screenshot' ];
        }
        else if (isset($_POST['id']) && isset($_POST['name']) && isset($_POST['score'])) {
            // Pega os dados em POST
            $id = $_POST[ 'id' ];
            $name = $_POST['name'];
            $score = $_POST['score'];
            $screenshot = $_POST['screenshot'];
        }
        else {
            echo '<p class="error">Desculpe, nenhuma pontuação foi especificada para ser
removida</p>';
        }

        if (isset($_POST['submit'])) {
            if ($_POST['confirm'] == 'Yes') {
                // Exclui o arquivo gráfico do servidor
                @unlink(GW_UPLOADPATH . $screenshot);
            }
        }

        // Conecte ao banco de dados
        $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
    
```

Inclua os arquivos de script compartilhados, mas use required\_once, uma vez que eles são essenciais para a remoção das pontuações.

O script reage diferentemente conforme a requisição recebida seja GET ou POST.

A diretiva PHP de supressão de erros, @, impede que erros sejam relatados. Faz sentido usar essa opção com unlink(), uma vez que poderemos estar tentando excluir um arquivo que não existe. Nesse caso, não queremos que o usuário receba uma mensagem de erro.

Este script pode ser usado para se remover quaisquer pontuações, de modo que o arquivo gráfico enviado precisa ser excluído também.

A coluna id é usada como critério na consulta DELETE, onde se usa também um LIMIT de apenas uma linha.

```
// Exclui os dados da pontuação do banco
$query = "DELETE FROM guitarwars WHERE id = $id LIMIT 1";
mysqli_query($dbc, $query);
mysqli_close($dbc);

// Confirma êxito com o usuário
echo '<p>A pontuação de ' . ' para ' . ' foi removida com sucesso.'';
}
else {
    echo '<p class="error">A pontuação não foi removida.</p>';
}

else if (isset( $id ) && isset( $name ) && isset( $date ) &&
isset($score) && isset($screenshot)) {
    echo '<p>Tem certeza de que deseja apagar a pontuação abaixo?</p>';
    echo '<p><strong>Nome: </strong>' . $name . '<br /><strong>Data: </strong>' . $date
        . '<br /><strong>Pontuação: </strong>' . $score . '</p>';
    echo '<form method="post" action="removescore.php">';
    echo '<input type="radio" name="confirm" value="Yes" /> Yes ';
    echo '<input type="radio" name="confirm" value="No" checked="checked" /> No <br />';
    echo '<input type="submit" value="Submit" name="submit" />';
    echo '<input type="hidden" name="id" value="' . $id . '" />';
    echo '<input type="hidden" name="name" value="' . $name . '" />';
    echo '<input type="hidden" name="score" value="' . $score . '" />';
    echo '<input type="hidden" name="screenshot" value="' . $screenshot . '" />';
    echo '</form>';

    echo '<p><a href=' . "admin.php" . '>&lt;&lt; Voltar para a página admin</a></p>';
}

?>

</body>
</html>
```

Forneca um link de volta à Página Admin, para melhor navegação.

**\$id**  
"id"  
**'id'**  
Sobraram  
alguns ímas.

Alguns campos escondidos do formulário são usados para se armazenarem os dados da pontuação, para que eles sejam enviados como parte da requisição POST.

Nós não usamos \$\_SERVER['PHP\_SELF'] aqui porque isso incluiria quaisquer dados que tenham sido enviados através da URL, como um GET. Nós queremos nos certificar de que nenhum dado GET seja enviado junto com este formulário - apenas dados POST.

removescore.php  
O formulário de confirmação só é exibido se todas estas variáveis estiverem definidas.

*o test drive final do aplicativo guitar wars*



## TEST DRIVE

### Adicione os scripts de Remover Pontuação e Admin ao Guitar Wars, para que seja possível excluir pontuações.

Crie dois novos arquivos de texto, `removescore.php` e `admin.php`, e insira neles o código que acabamos de completar. Envie os novos scripts para o seu servidor, e depois abra o script Admin no seu navegador. Clique no link "Remove" de uma pontuação que você queira excluir, e depois confirme a remoção na página Remover Pontuação. Volte à página Admin para se certificar de que o registro foi apagado, e depois vá para a página principal do Guitar Wars (`index.php`), para ver lá o efeito da modificação.

A nova página Remover Pontuação cuida de confirmar e fazer a remoção das pontuações indesejadas.

Guitar Wars - High Scores Administration  
Below is a list of all Guitar War high scores. Use this page to remove scores as needed.  
Actions: Simpson 2008-04-23 07:15:46 366420 Remove; Eddie Van Halen 2008-04-23 09:02:53 343000 Remove; Phil Lesh 2008-04-24 08:20:11 283470 Remove; Paul Kantner 2008-04-24 08:21:11 12750 Remove; Neal Johnson 2008-04-24 08:21:11 98451 Remove; Kanga Lark 2008-04-23 11:09:53 64950 Remove

A nova página Admin fornece links para se removerem pontuações não verificadas.

Esta cidade só tem espaço para um super roqueiro, e esse sou eu!

Guitar Wars - Remove a High Score  
Are you sure you want to delete the following high score?  
Name: Action Simpson  
Date: 2008-04-23 09:12:34  
Score: 366420  
Delete  
Cancel

Guitar Wars - Remove a High Score  
The high score of Action for Action Simpson was successfully removed.  
Back to index

A página principal do Guitar Wars agora só mostra pontuações comprovadas.

Este é o pequeno Jacob, Guitar Warrior prodígio.



Os Guitar Warriors legítimos estão felizes, agora que só há pontuações comprovadas.

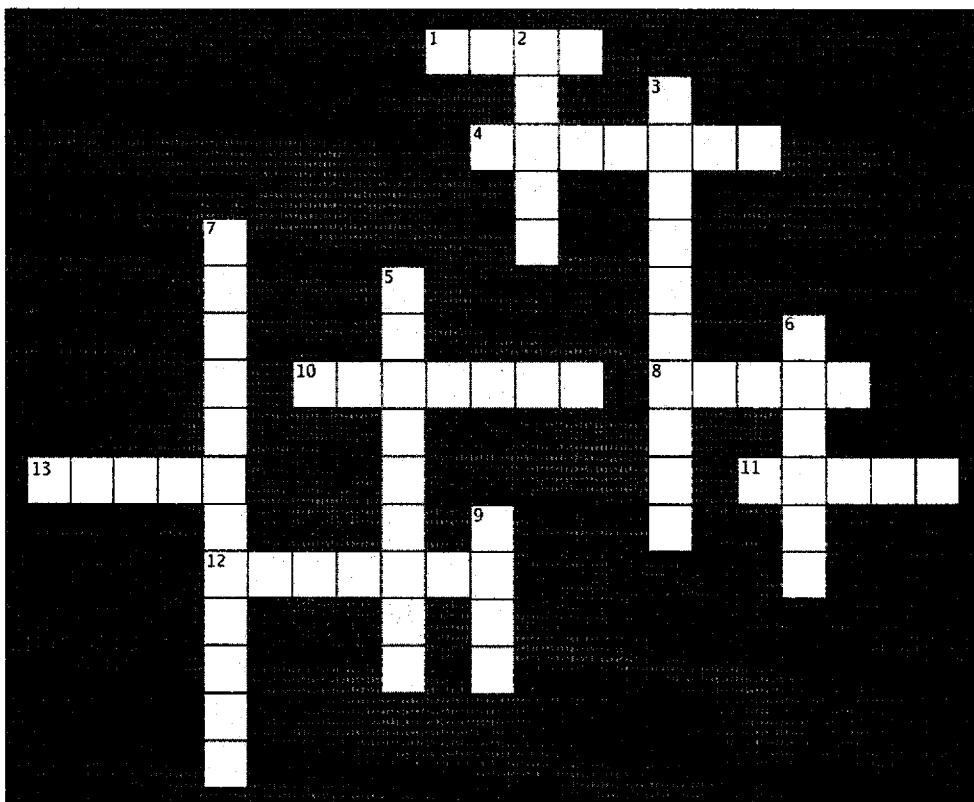
Guitar Wars - High Scores  
Welcome, Guitar Warrior, do you have what it takes to check the high score list? If so, just click here.  
Top Score: 389740  
389740 Name: Jacob Schaeffer Date: 2008-04-25 11:49:23  
Guitar  
Name: Jacob Schaeffer Date: 2008-04-25 11:49:23  
186580 Name: Paul Kantner Date: 2008-04-24 08:13:22  
Guitar  
Name: Paul Kantner Date: 2008-04-24 08:13:22

As pontuações não verificadas, ou seja, aquelas sem imagens comprobatórias, foram removidas do sistema.



# Palavras Cruzadas PHP & MySQL

Cansado de enviar arquivos gráficos? Que tal então enviar um pouco de conhecimento para um monte de quadrinhos?



## Horizontais

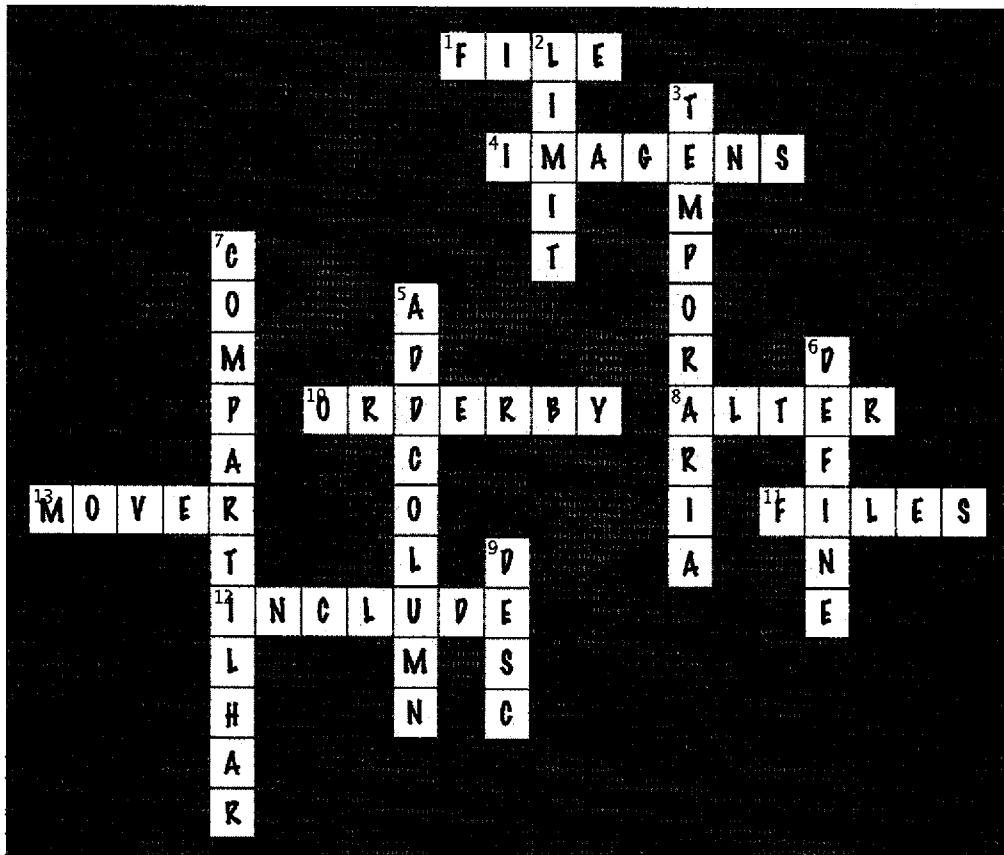
1. O atributo type da tag <input> deve ser definido como isto, para que um campo do formulário aceite upload de arquivos.
4. Geralmente é uma boa ideia armazenar as imagens enviadas à aplicação em uma pasta... .
8. Esta instrução SQL é usada para se modificar a estrutura de uma tabela.
10. Esta instrução SQL é usada para se colocar os resultados de uma consulta em uma determinada ordem.
11. Informações sobre os arquivos enviados são armazenadas na variável superglobal \$... .
12. Esta instrução PHP é usada para se inserir código de outro script.
13. É uma boa ideia fazer isto com um arquivo recém-enviado.

## Verticais

2. Para impedir que uma instrução DELETE FROM apague mais de uma linha, use esta instrução.
3. Quando um arquivo é enviado através de um formulário, ele é colocado em uma pasta ... no servidor.
5. Ao alterar uma tabela, este comando SQL cuida de adicionar uma nova coluna.
6. Esta instrução PHP é usada para se criar uma constante.
7. Os arquivos include são bastante úteis para se ... dados entre vários scripts.
9. Esta instrução SQL é usada como parte de outra instrução para ordenar resultados de consultas em ordem descendente.



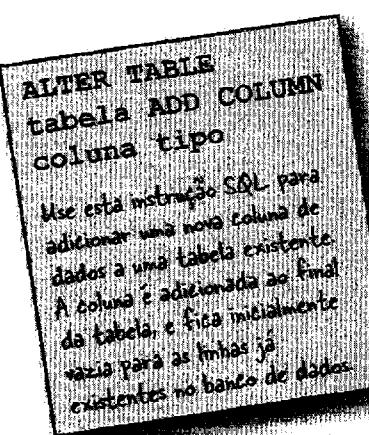
# Palavras Cruzadas PHP & MySQL - Solução





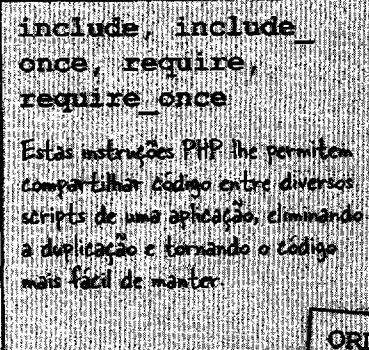
## Sua Caixa de Ferramentas PHP & MySQL

**Fique à vontade para fazer uma reverência virtual. Você não só é adorado pelos guitarristas virtuais do mundo todo, como também adquiriu novas habilidades em PHP e MySQL: alterar a estrutura de tabelas, lidar com uploads de arquivos, controlar a ordem de dados e remover dados.**



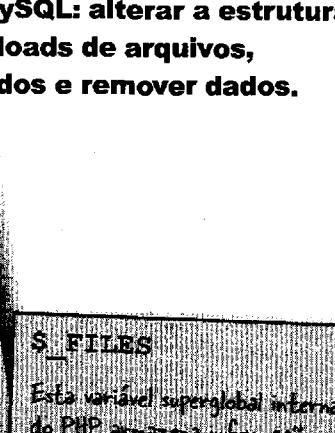
### ALTER TABLE tabela ADD COLUMN coluna tipo

Use esta instrução SQL para adicionar uma nova coluna de dados a uma tabela existente. A coluna é adicionada ao final da tabela, e fica implicitamente vazia para as linhas já existentes no banco de dados.



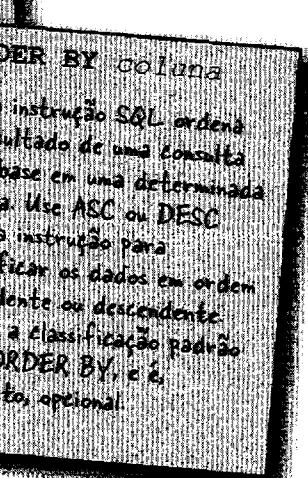
### include, include\_ once, require, require\_once

Estas instruções PHP lhe permitem compartilhar código entre diversos scripts de uma aplicação, eliminando a duplicação e tornando o código mais fácil de manter.



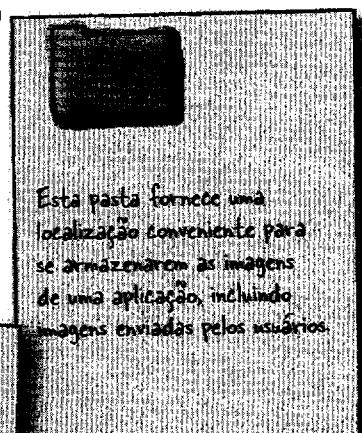
### S\_FILES

Esta variável superglobal interna do PHP armazena informações sobre arquivos que tenham sido enviados através de um formulário. Você pode usá-la para determinar o nome do arquivo, a sua localização de armazenamento temporário, o tamanho e o tipo do arquivo, entre outras coisas.

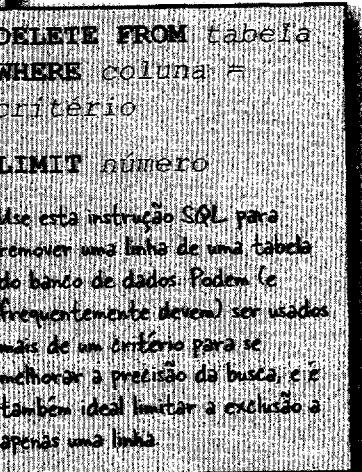


### ORDER BY coluna

Esta instrução SQL ordena o resultado de uma consulta com base em uma determinada coluna. Use ASC ou DESC após a instrução para classificar os dados em ordem ascendente ou descendente. ASC é a classificação padrão para ORDER BY, e é, portanto, opcional.



Esta pasta fornece uma localização conveniente para se armazenarem as imagens de uma aplicação, incluindo imagens entradas pelos usuários.



### DELETE FROM tabela WHERE coluna = critério

### LIMIT número

Use esta instrução SQL para remover uma linha de uma tabela do banco de dados. Podem (e frequentemente devem) ser usados mais de um critério para se melhorar a precisão da busca, e é também ideal limitar a exclusão a apenas uma linha.



## Presuma que Estão Todos Querendo Te Pegar

É só subir aqui, bagunçar um pouco a fiação telefônica, e esta cidadezinha não vai nem saber o que a atingiu.

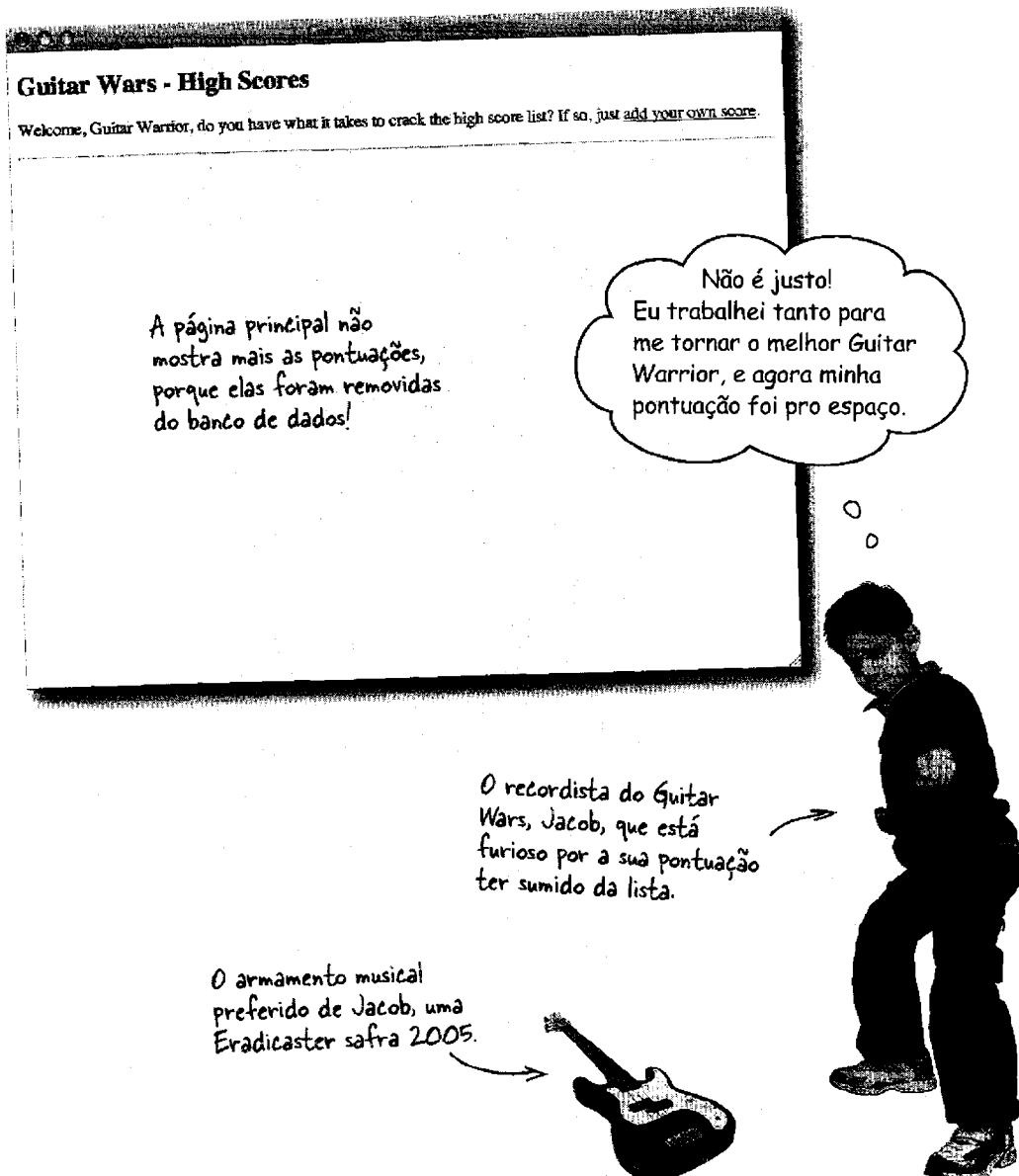


**Seus pais estavam certos: não fale com estranhos.** Ou pelo menos não confie neles. Na pior das situações, não dê a eles as chaves para acessar os dados da sua aplicação, presumindo que eles não vão fazer nada de errado. Vivemos num mundo cruel, e você não pode partir do princípio que todos são confiáveis. Na verdade, como desenvolvedor de aplicações web você precisa ser metade incrédulo e metade adepto de teorias de conspiração. Sim, as pessoas em geral são más e elas definitivamente querem te pegar! OK, talvez isso seja um pouco de exagero, mas é muito importante levar a segurança a sério e elaborar as suas aplicações de forma que elas fiquem protegidas contra qualquer um que pretenda causar danos.

*guitar wars* está sendo hackeado

## O dia em que a música morreu

Parece que os quinze minutos de fama do nosso roqueiro-virtual-prodígio se acabaram: a pontuação de Jacob no Guitar Wars sumiu, junto com todas as outras. Parece que há alguma força do mal em ação, bagunçando o aplicativo e impedindo os Guitar Warriors de terem os seus placares publicados. Guitarristas virtuais insatisfeitos são usuários insatisfeitos, e a consequência última disso é um desenvolvedor de aplicações insatisfeito... você!



## Para onde foram as pontuações?

Nós sabemos que a página principal do Guitar Wars está vazia, mas será que isso significa que o banco de dados também está vazio? Uma consulta SELECT pode responder essa questão:

A consulta SELECT revela que a tabela guitarwars está completamente vazia—todas as pontuações se foram!

```
File Edit Window Help f6Was9
mysql> SELECT * FROM guitarwars;
+----+-----+-----+-----+
| id | date | name | score | screenshot |
+----+-----+-----+-----+
0 rows in set (0.0005 sec)
```

De algum modo, todas as linhas de dados foram apagadas do banco Guitar Wars. Seria possível que alguém talvez esteja usando o nosso script Remover Pontuação para o mal? Nós temos de proteger as pontuações!



### Aponte seu lápis

Faça um círculo em torno das técnicas, dentre as apresentadas a seguir, que você poderia usar para proteger as pontuações do Guitar Wars contra hackers, e escreva em baixo o porquê.



**Proteger com senha a página Admin, de modo que somente quem souber a senha (você!) possa remover pontuações.**



**Criar um sistema de registro dos usuários, e só conceder a alguns (de preferência só a você!) os privilégios administrativos.**



**Verificar o endereço IP do computador que está tentando acessar a página Admin, e só permitir alguns endereços (de preferência apenas o seu próprio!).**



**Eliminar totalmente o recurso de remover pontuações.**

*protegendo as pontuações do guitar wars?*



## Aponte seu lápis

### Solução

Faça um círculo em torno das técnicas, dentre as apresentadas a seguir, que você poderia usar para proteger as pontuações do Guitar Wars contra hackers, e escreva em baixo o porquê.



**Proteger com senha a página Admin, de modo que somente quem souber a senha (você!) possa remover pontuações.**

Proteger com senha a página Admin é uma solução boa e rápida, porque não é muito complicada e fornece ao site uma segurança razoável rapidamente.

Todas as técnicas foram circuladas, porque todas elas são capazes de resolver o problema, embora algumas sejam mais viáveis que outras.



**Criar um sistema de registro dos usuários, e só conceder a alguns (de preferência só a você!) os privilégios administrativos.**

Um sistema de registro dos usuários, com privilégios administrativos limitados, é uma ótima solução, mas envolve bastante planejamento e programação... o Guitar Wars precisa de segurança agora!!



**Verificar o endereço IP do computador que está tentando acessar a página Admin, e só permitir alguns endereços (de preferência apenas o seu próprio!).**

Verificar o IP funciona, mas torna o site dependente do endereço IP do seu computador, o qual poderá se modificar.



**Eliminar totalmente o recurso de remover pontuações.**

Remover o recurso certamente resolve este problema específico, mas lembre-se: o recurso foi adicionado originalmente, no capítulo anterior, para tornar a manutenção do site mais fácil.

## Segurança contra os bárbaros

Uma maneira simples e direta de se garantir rapidamente a segurança do Guitar Wars é usar autenticação HTTP para proteger com senha a página Admin. Na prática, será pedido um nome e uma senha, mas a ideia teórica é exigir do administrador uma determinada informação secreta antes de permitir que ele tenha acesso a recursos restritos da aplicação, como por exemplo, os links para remoção de pontuações.

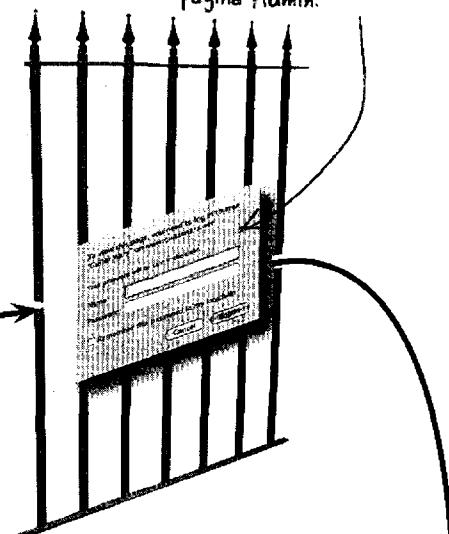
Quando uma página é protegida usando-se autenticação HTTP, aparece uma janela pop-up pedindo o nome do usuário e a senha, para que seja concedido o acesso à página protegida. No caso do Guitar Wars, você pode limitar o acesso à página Admin a qualquer número de pessoas que quiser, possivelmente até a apenas você!

A autenticação HTTP oferece uma forma simples de se fazer a segurança de uma página que use PHP.

A janela de autenticação HTTP agora se interpõe entre os usuários e a página Admin.



As pontuações do banco de dados agora estão protegidas, uma vez que o acesso à página Admin foi restringido.



Os links "Remove" da página Admin agora só estão disponíveis para o administrador do Guitar Wars.

guitarwars

ID	Data	Jogador	Pontuação	Arquivo
14	2008-05-01 20:36:07	Belita Chevy	282470	belitascore.gif
15	2008-05-01 20:36:45	Jacob Scurcherson	385740	jacobsscore.gif
16	2008-05-01 20:37:02	Nevil Johansson	918430	nevilscore.gif
17	2008-05-01 20:37:23	Paco Jastorius	127650	pacoscore.gif
18	2008-05-01 20:37:40	Phiz Lairston	166580	phizscore.gif
19	2008-05-01 20:38:00	Kenny Lavitz	64930	kennyscore.gif
20	2008-05-01 20:38:23	Jean Paul Jones	243360	jeanpaulscore.gif

Guitar Wars - High Scores Administration

Show a list of all Guitar Wars high scores. Use the page to remove scores as desired.		
Jacob Scurcherson	2008-05-01 20:36:45	385740 Registered
Belita Chevy	2008-05-01 20:36:07	282470 Registered
Jean Paul Jones	2008-05-01 20:38:23	64930 Registered
Phiz Lairston	2008-05-01 20:37:40	166580 Registered
Paco Jastorius	2008-05-01 20:37:23	127650 Registered
Nevil Johansson	2008-05-01 20:37:02	918430 Registered
Kenny Lavitz	2008-05-01 20:36:00	385740 Registered

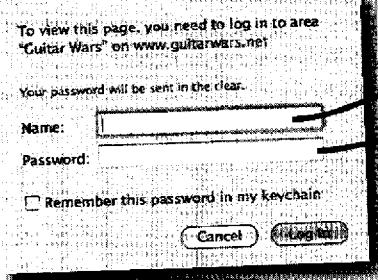
## usando autenticação HTTP

# Protegendo a página Guitar Wars Admin

A autenticação HTTP funciona assim: quando um usuário tenta acessar uma página protegida, como a nossa página Admin, eles recebem uma janela que lhes pede um nome de usuário e uma senha.

O navegador web usa uma janela como esta para requisitar um nome e senha, antes de permitir o acesso a uma página protegida.

Para manter as coisas simples, a senha não é criptografada.



Esta variável superglobal do PHP armazena o nome do usuário digitado na janela de autenticação.

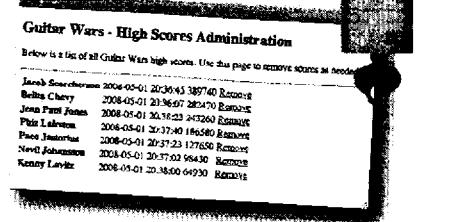
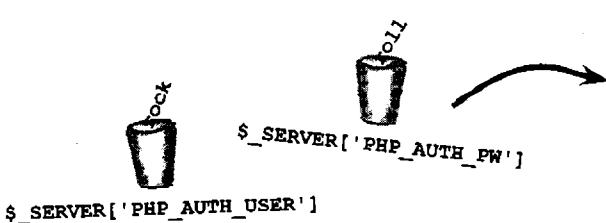
`$_SERVER['PHP_AUTH_USER']`

`$_SERVER['PHP_AUTH_PW']`

Esta variável armazena a senha digitada na janela de autenticação.

O PHP entra em cena através do seu acesso ao nome e à senha digitados pelo usuário. Eles ficam armazenados na superglobal `$_SERVER`, que é semelhante a outras superglobais que você já usou (`$_POST`, `$_FILES`, etc.). Um script PHP pode analisar o nome e a senha digitados pelo usuário e decidir se ele deve receber acesso à página protegida. Digamos que só vamos permitir acesso se o nome do usuário for "rock" e a senha for "roll". Eis aqui o modo como o acesso à página Admin é concedido:

A página Admin só fica acessível se tiverem sido digitados o nome e a senha corretos.



não existem  
**Perguntas Idiotas**

**P:** A autenticação HTTP é realmente segura?

**R:** Sim e não. Depende do que você está tentando conseguir com a segurança. Nada é 100% seguro, portanto nós estamos sempre falando de níveis de segurança. Para se protegerem pontuações no Guitar Wars, a autenticação HTTP fornece um nível razoável de segurança. Você poderia adicionar criptografia à senha para ter um nível ainda maior. Porém, mesmo isso provavelmente não seria suficiente em uma aplicação que envolvesse dados mais sensíveis, como por exemplo, dados financeiros.

**P:** O que acontece se o nome e a senha forem digitados incorretamente?

**R:** O navegador emite um pequeno choque elétrico através do mouse. Brincadeira, não acontece nada tão dramático assim. Em geral, é exibida uma mensagem para avisar o usuário de que ele está tentando acessar uma página segura, a qual aparentemente não é da conta dele. Em última instância, é você quem escolhe o que essa mensagem deverá dizer.

**P:** A autenticação HTTP requer tanto o nome do usuário quanto a senha? E se eu só quiser usar a senha?

**R:** Não é obrigatório usar ambos. Se você quiser usar apenas uma senha, basta verificar só a variável global `$_SERVER['PHP_AUTH_PW']`. Falaremos mais sobre como essa variável é verificada, em instantes....

**P:** Como exatamente você protege uma página com autenticação HTTP?  
Chamando-se uma função PHP?

**R:** Exato. A autenticação HTTP envolve o estabelecimento de uma linha de comunicação entre o navegador e o servidor, através de cabeçalhos HTTP. Você pode pensar no cabeçalho como sendo uma pequena conversa entre o navegador e o servidor. Ambos usam os cabeçalhos frequentemente para se comunicarem fora do contexto do PHP, mas o PHP lhe permite enviar um cabeçalho, que é o modo como a autenticação HTTP funciona. Estamos prestes a aprender muito mais sobre os cabeçalhos e o seu papel na autenticação HTTP com PHP.

 **PODER DO  
CÉREBRO**

Quando é que a autenticação da página deve realmente ocorrer?

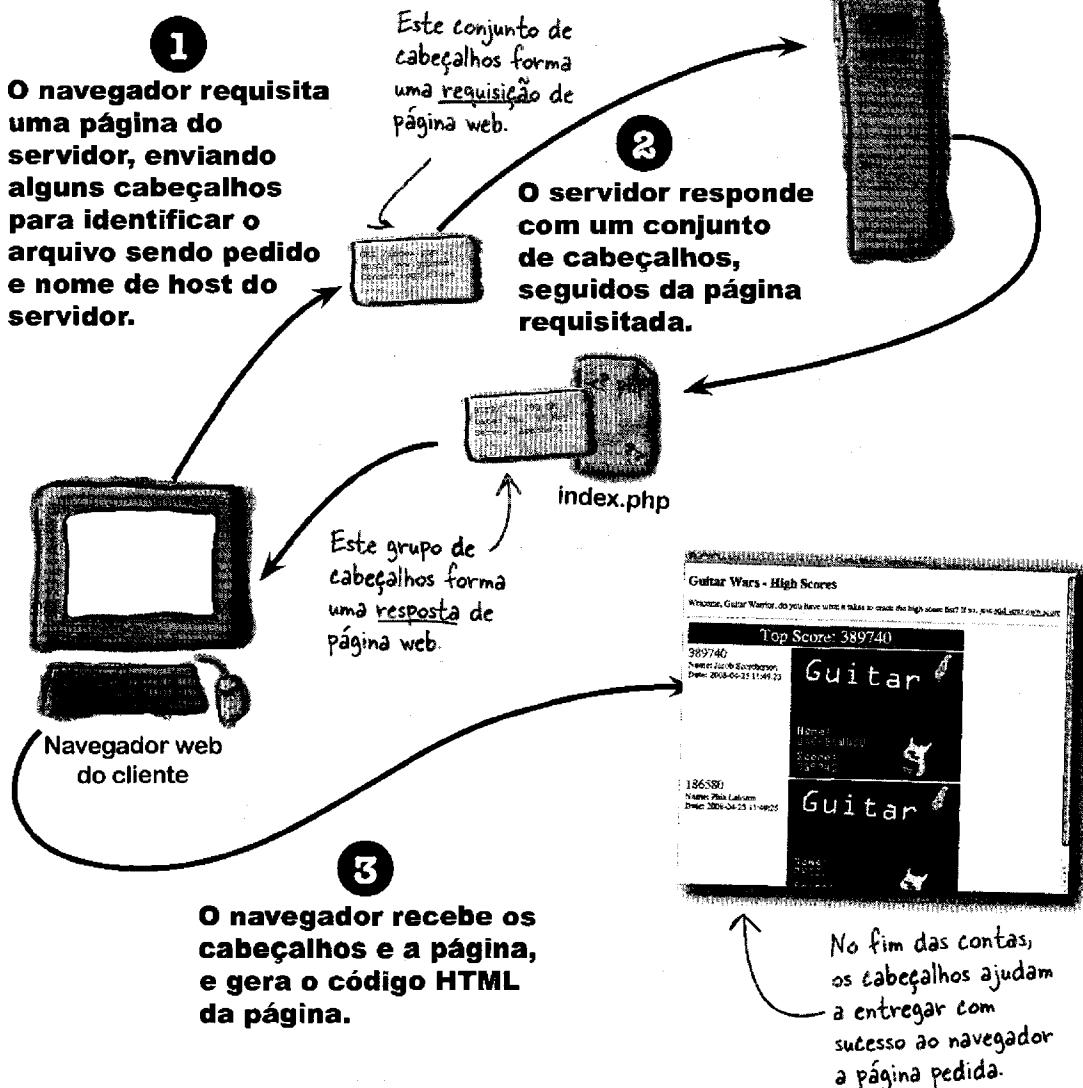
## A autenticação HTTP exige cabeçalhos

A ideia por trás da autenticação HTTP é que o servidor esconda uma página protegida, e instrua o navegador a pedir um nome e senha ao usuário. Se o usuário os digitar corretamente, o navegador vai em frente e fornece a página. O diálogo entre navegador e servidor se realiza através dos cabeçalhos, que são pequenas mensagens de texto com instruções específicas sobre o que está sendo pedido ou entregue.

Os cabeçalhos na verdade são usados sempre que você visita uma página web, e não apenas quando a autenticação é requerida. Eis o modo como uma página normal, não protegida, é enviada pelo servidor para o navegador com a ajuda dos cabeçalhos:

Toda página web é entregue com a ajuda dos cabeçalhos.

### Servidor web





## Anatomia de um Cabeçalho

Os cabeçalhos controlam precisamente como e que tipo de informação é enviada do navegador para o servidor web, e vice-versa. O cabeçalho frequentemente consiste de um par de nomes/valor que identifica uma determinada informação, como por exemplo, o tipo de conteúdo da página (HTML). Um determinado grupo de cabeçalhos é enviado ao servidor como parte de uma requisição web, e depois outro grupo é retornado ao navegador como parte da resposta. Vamos dar uma olhada com mais detalhes nestes grupos de cabeçalhos, para descobrir exatamente o que é enviado, à medida que o cliente e o servidor comunicam-se um com o outro.

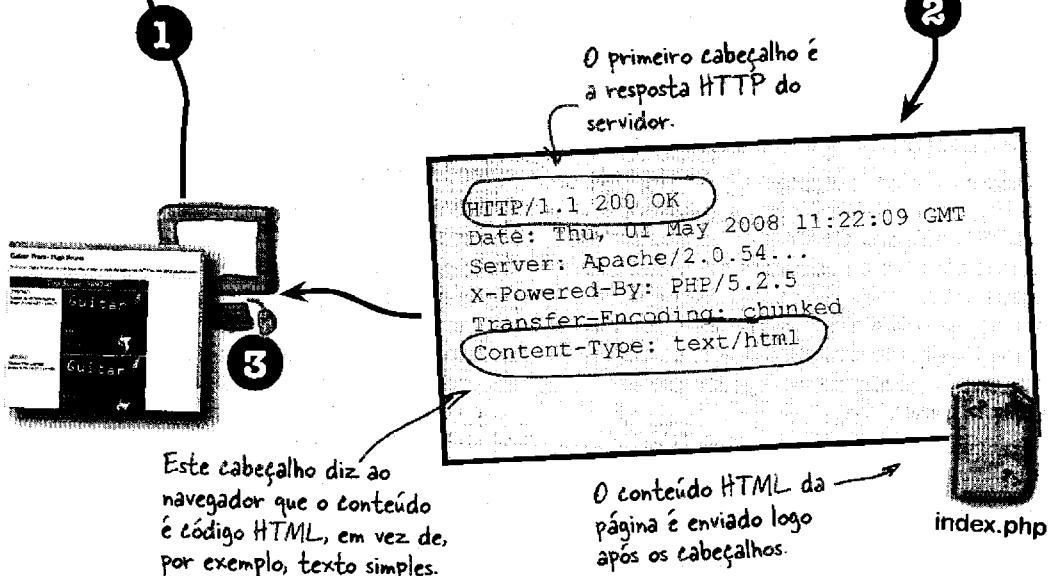
A maioria dos cabeçalhos consiste de um par de nomes/valor separado por dois pontos.

Este cabeçalho especifica o navegador que está fazendo a requisição.

```

GET /index.php HTTP/1.1
Host: www.guitarwars.net
Connection: close
User-Agent: Mozilla/5.0...
Accept-Charset: ISO-8859-1...
Cache-Control: no
Accept-Language: de,en;q=0.7...
  
```

O primeiro cabeçalho não é um par de nomes/valor – é a requisição GET da página.



Os cabeçalhos nos são importantes, no que diz respeito ao Guitar Wars, porque eles fornecem o mecanismo que interrompe a entrega de uma página do servidor, requerendo que o usuário digite um nome e uma senha para liberar a página. Em outras palavras, é preciso trabalhar nos cabeçalhos retornados pelo servidor para proteger uma página com autenticação HTTP.



## Tudo Sobre o Cabeçalho

**Na entrevista desta semana: por que tanto falatório?**

**Use a Cabeça!**: Você parece estar recebendo bastante atenção no que diz respeito a autenticar páginas web. Isso se justifica ou você está apenas tendo os seus quinze minutos de fama virtual?

**Cabeçalho**: Não, isso se justifica sim. Você está se esquecendo de que eu tenho um papel vital na entrega de qualquer página web existente. Assim, acho que posso dizer que a web nem sequer funcionaria se eu não estivesse em cena. Eu vou durar muito mais do que quinze minutos, mesmo que as pessoas não apreciem o meu trabalho.

**Use a Cabeça!**: Então qual é exatamente esse papel que você desempenha?

**Cabeçalho**: Você tem que entender que navegadores e servidores web não são pessoas, portanto eles não podem simplesmente ligar um para o outro ou enviar uma mensagem de texto.

**Use a Cabeça!**: SÉRIO?!

**Cabeçalho**: Sim, eu sei que é um choque, mas as máquinas simplesmente não se comunicam do mesmo jeito que as pessoas. Mas os navegadores e os servidores ainda precisam falar uns com os outros, e eles o fazem através de mim.

**Use a Cabeça!**: E como isso acontece?

**Cabeçalho**: Quando alguém digita uma URL ou clica em um link de uma página, o navegador cria uma requisição GET e a envia ao servidor. Essa requisição é empacotada em uma série de cabeçalhos, cada um dos quais contendo informações sobre a requisição. Eles contêm informações como o nome e o host da página sendo pedida; o tipo de navegador que está fazendo o pedido, etc.

**Use a Cabeça!**: Não consigo ver o que há de tão importante nisso.

**Cabeçalho**: Bem, quando você vai a um café, você acha importante dizer ao atendente que quer um cappuccino grande sem creme?

**Use a Cabeça!**: Claro que sim, o atendente precisa saber o que é que eu quero.

**Cabeçalho**: É a mesma coisa, aqui. O navegador diz ao servidor o que ele quer, empacotando a requisição e enviando-a em cabeçalhos.

**Use a Cabeça!**: Interessante. Mas ouvi dizer que os servidores também são capazes de enviar cabeçalhos. Eu pensava que os servidores só enviassem páginas web.

**Cabeçalho**: Ah, boa pergunta. A minha importância no outro lado da comunicação é a mesma, porque o servidor precisa fazer mais do que apenas jogar um monte de conteúdo em cima do navegador. O navegador poderia não ter a menor ideia do que fazer com esse conteúdo, se não lhe fosse passadas mais algumas informações.

**Use a Cabeça!**: Como por exemplo...?

**Cabeçalho**: O tipo do conteúdo seria um exemplo. Essa é provavelmente a informação mais importante, mas o servidor envia também outras coisas, como o tamanho do conteúdo, a data e a hora da entrega, e assim por diante.

**Use a Cabeça!**: Quando é que a página propriamente dita é enviada?

**Cabeçalho**: O conteúdo, seja ele código HTML, dados PDF ou dados de imagens como um GIF ou JPEG, segue logo depois que o servidor me envia ao navegador.

**Use a Cabeça!**: OK; estou começando a entender como você funciona em relação às páginas normais. Mas e quanto a esse negócio de autenticação?

**Cabeçalho**: Eu desempenho o mesmo papel em uma página autenticada que nas páginas normais, exceto pelo fato de que eu também cuido de informar ao navegador que página precisa ser autenticada. Dessa forma, o navegador poderá pedir ao usuário as informações de autenticação.

**Use a Cabeça!**: Ou seja, um nome de usuário e uma senha?

**Cabeçalho**: Exatamente. E depois, fica a cargo do código PHP no servidor decidir se o nome e a senha são válidos, em cujo caso o servidor pode enviar o restante da página.

**Use a Cabeça!**: Fascinante. Obrigado pelos esclarecimentos.

**Cabeçalho**: Sem problemas. Isso faz parte do meu trabalho.

## Controle os cabeçalhos com PHP

Com o PHP, você pode controlar cuidadosamente os cabeçalhos enviados pelo servidor ao navegador, abrindo a possibilidade de realizar tarefas como a autenticação HTTP. A função interna `header()` é usada para se enviar um cabeçalho, do servidor para o navegador, de dentro de um script PHP.

```
header('Content-Type: text/html');
```

A função `header()` envia imediatamente o cabeçalho, e deve ser chamada antes de qualquer conteúdo propriamente dito ser enviado ao navegador. Esse é um requerimento bastante estrito – se mesmo um único caractere ou espaço for enviado antes do cabeçalho, o navegador o rejeitará com um erro. Por esse motivo, as chamadas à função `header()` devem preceder qualquer código HTML dentro de um script PHP:

Mesmo um espaço que viesse antes da tag <?php causaria um erro neste script de exemplo.

Os espaços dentro das tags <?php ?> não é problema, porque eles não são repassados para o navegador.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
      lang="en" >
```

```
    ...
</html>
```

A função `header()` lhe permite criar e enviar um cabeçalho a partir de um script PHP.

O servidor envia este cabeçalho ao navegador para ser processado, antes de tentar enviar qualquer parte do conteúdo HTML da página.

Todo esse negócio de cabeçalhos é fascinante, mas como nós os usamos na prática, para proteger as páginas com autenticação?



como a autenticação de cabeçalho funciona?

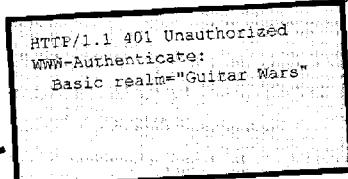
## Autenticando com cabeçalhos

Para se autenticar a página Admin do Guitar Wars mediante cabeçalhos, é preciso criar um conjunto bastante específico deles – dois, na verdade –, que façam o navegador saber que deve pedir um nome e uma senha ao usuário antes de entregar a página. Esses dois cabeçalhos são gerados através de código PHP no script Admin, e controlam a entrega da página para o navegador.

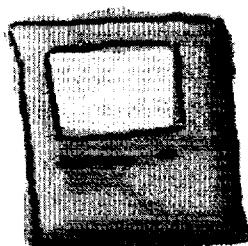
Servidor web



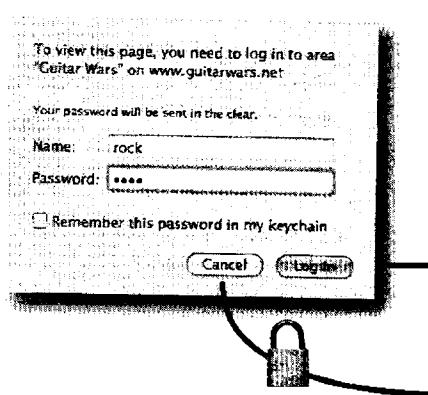
Os cabeçalhos de autenticação HTTP são enviados do servidor para o navegador.



Antes de enviar o conteúdo da página admin.php para o navegador, o servidor processa quaisquer cabeçalhos da página.

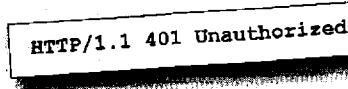


O navegador, então, pede que o usuário digite um nome e uma senha.



Navegador web do cliente  
São necessários dois cabeçalhos específicos para se requerer a autenticação de uma página Web.

Este cabeçalho pede ao navegador para tentar autenticar o usuário, pedindo-lhe um nome e uma senha.



Este cabeçalho informa ao navegador que o usuário não está autorizado a ver a página.

WWW-Authenticate: Basic realm="Guitar Wars"

O "basic realm" (domínio básicos) é apenas uma expressão usada para identificar individualmente esta autenticação em particular – ela aparece na janela de autenticação.

Após processar os cabeçalhos, o navegador espera a resposta do usuário, através da janela de autenticação. O navegador responde com ações completamente diferentes, dependendo do que o usuário fizer...

**Guitar Wars - High Scores Administration**

Below is a list of all Guitar Wars high scores. Use this page to remove scores as needed.

User	Date	Time	Score	Action
Jacob Scorcherson	2008-05-01	20:36:45	389740	<a href="#">Remove</a>
Belta Chevy	2008-05-01	20:36:07	282470	<a href="#">Remove</a>
Jean Paul Jones	2008-05-01	20:38:23	243260	<a href="#">Remove</a>
Phiz Lairston	2008-05-01	20:37:40	186580	<a href="#">Remove</a>
Paco Jastorius	2008-05-01	20:37:23	127650	<a href="#">Remove</a>
Nevil Johansson	2008-05-01	20:37:02	98430	<a href="#">Remove</a>
Kenny Lavitz	2008-05-01	20:38:00	64930	<a href="#">Remove</a>

Se o usuário digitar um nome e senha corretos, e clicar em Log In, o servidor envia o conteúdo HTML da página admin.php para o navegador. Este exibe a página Admin, e o usuário poderá então remover pontuações, da mesma forma como na versão não protegida anterior.

O conteúdo HTML da página Admin é entregue depois que se digita um nome e uma senha corretos.

Eis aqui o realm (domínio)!!

Se o nome e a senha do usuário forem submetidos e forem incorretos, a janela de autenticação apenas aparece para usuário de novo.

To view this page, you need to log in to area "Guitar Wars" on www.guitarwars.net

Your password will be sent in the clear.

Name:

Password:

Remember this password in my keychain!

[Cancel](#) [Login](#)

Se o usuário digitou um nome e senha incorretos, e clicar em Login In, o servidor diz para o navegador pedir ao usuário de novo o login. O navegador continuará este processo enquanto o usuário tentar combinações incorretas de nome e senha. Em outras palavras, se eles não souberem o nome e a senha, o único jeito de sair é clicar em CANCELAR.

**Guitar Wars**

Sorry, you must enter a valid user name and password to access this page.

(Sorry, you... = Desculpe, você deve digitar um usuário e senha válida para acessar esta página.)

Se o usuário clicar em CANCELAR para sair da autenticação, o servidor envia para o navegador uma página com uma mensagem de acesso negado e nada mais — a página admin.php não foi enviada. A mensagem de acesso negado é controlada pelo código PHP no script admin.php que está associado com os cabeçalhos. Este código chama a função PHP exit() para mostrar a mensagem e imediatamente sair do script:

```
exit('<h2>Guitar Wars</h2>Sorry, you must enter a
valid '
      'user name and password to access this page.'');
```

A aplicação tem a oportunidade de finalizar o script e exibir uma mensagem de erro personalizada, caso o usuário cancele o processo de autenticação.

*finalizando o código de autenticação*



## ímas de Geladeira PHP

No script Admin do Guitar Wars, estão faltando diversos pedaços importantes do código PHP que fornece a autenticação HTTP. Use os ímas para completar o código abaixo, e use cabeçalhos para tornar a página Admin segura. Dica: alguns ímas poderão ser usados mais de uma vez.

```
<?php / Nome do usuário e senha para autenticação  
..... = 'rock';  
..... = 'roll';  
  
if (!isset( ..... ) ||  
    !isset( ..... ) ||  
    ($_SERVER['PHP_AUTH_USER'] != ..... ) || ($_SERVER['PHP_AUTH_PW'] != ..... )) {  
    // Nome do usuário/senha incorretos, então enviar os cabeçalhos de autenticação  
    ..... ('HTTP/1.1 401 Unauthorized');  
    ..... ('WWW-Authenticate: Basic realm= ..... ');  
    ..... ('<h2>Guitar Wars</h2>Desculpe, você deve digitar um'.  
        'válida para acessar esta página.');
```

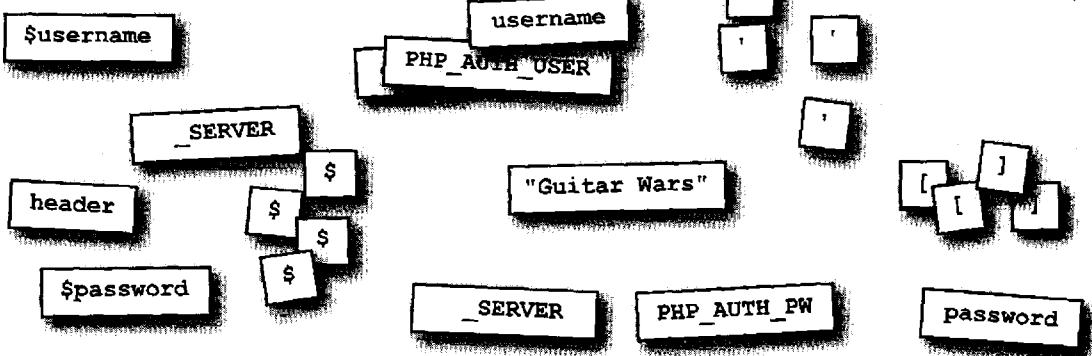
}

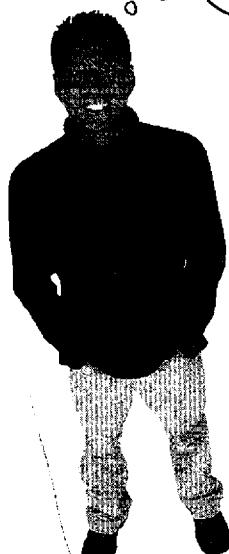
?>

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">  
...  
</html>
```



admin.php





Será que é possível enviar outros tipos de cabeçalhos usando-se PHP?

### Sim, é possível... os cabeçalhos não servem apenas para segurança.

Embora a autenticação seja o principal caso em que os cabeçalhos se fazem necessários, eles são bastante flexíveis e são capazes de fazer outras coisas interessantes. Basta chamar a função `header()` com o par de nome/valor apropriado, desta forma:

O navegador é redirecionado à página `about.php`, após receber este cabeçalho.

```
<?php
header('Location: http://www.guitawars.net/about.php');
?>
```

Neste caso, temos um cabeçalho de localização, o qual redireciona da página atual para outra, chamada `about.php`, no mesmo site do `Guitar Wars`. Aqui, nós usamos um cabeçalho semelhante para redirecionar para a página `about.php` depois de cinco segundos:

O navegador é redirecionado para a página About depois de 5 segundos.

```
<?php
header('Refresh: 5; url=http://www.guitawars.net/about.php');
echo 'Em 5 segundos, você será levado para a página About.';
```

E neste caso, nós temos um cabeçalho de recarregamento, uma vez que ele recarrega a página após ter-se passado um determinado período de tempo. Você frequentemente vê a URL nesse tipo de cabeçalho referenciar a página atual, para que ela se recarregue periodicamente.

Um último caso é o chamado cabeçalho de tipo de conteúdo, porque ele controla o tipo do conteúdo sendo entregue pelo servidor. Como exemplo, você pode forçar uma página a ser exibida como texto puro, em vez de HTML, usando o seguinte cabeçalho ao chamar a função `header()`:

**Veja bem!**

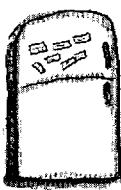
Os cabeçalhos precisam ser a primeira coisa enviada ao navegador, em um arquivo PHP.

Devido ao fato de os cabeçalhos serem enviados antes de qualquer conteúdo, é extremamente importante não permitir que um espaço sequer apareça fora do código PHP, antes de ser chamada a função `header()` no script PHP.

```
<?php
header('Content-Type: text/plain');
echo 'Este <strong>texto</strong> na verdade não aparecerá
em negrito.';
```

O conteúdo é entregue ao navegador como texto simples.

Neste exemplo, o texto enviado ao navegador é exibido exatamente como mostrado, sem nenhuma formatação especial. Em outras palavras, o servidor está dizendo ao navegador para não exibir o conteúdo como HTML, e portanto, as tags HTML são exibidas literalmente, como texto.



## Ímas de Geladeira PHP - Solução

No script Admin do Guitar Wars, estão faltando diversos pedaços importantes do código PHP que fornece a autenticação HTTP. Use os ímas para completar o código abaixo, e use 0 para tornar a página Admin segura. Dica: alguns ímas poderão ser usados mais de uma vez.

```
<?php // Nome do usuário e senha para autenticação
$username = 'rock'; ← O nome do usuário e a senha
$password = 'roll'; ← são armazenados em variáveis,
                      no início do script.

if (!isset($_SERVER['PHP_AUTH_USER']) || !isset($_SERVER['PHP_AUTH_PW'])) {
    if (!$_SERVER['PHP_AUTH_USER'] != $username || !$_SERVER['PHP_AUTH_PW'] != $password) {
        // Nome de usuário/senha incorretos, então enviar os cabeçalhos de autenticação
        header('HTTP/1.1 401 Unauthorized');
        header('WWW-Authenticate: Basic realm=' . "Guitar Wars");
        exit('<h2>Guitar Wars</h2>Desculpe, você deve digitar um usuário e senha válidos para acessar esta página.');
    }
}
?>

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
...
</html>
```

O nome e a senha digitados pelo usuário são verificados aqui.

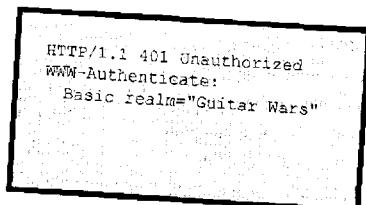
Nenhum código é entregue ao navegador até que os cabeçalhos tenham sido enviados e processados.

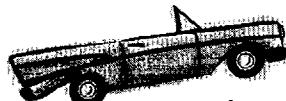
A superglobal `$_SERVER` dá acesso ao nome e à senha digitados pelo usuário na janela de autenticação.

A duas chamadas à função `header()` resultam em estes cabeçalhos serem enviados ao navegador.



admin.php





## TEST DRIVE

### Adicione autorização HTTP ao script Admin.

Modifique o script admin.php de modo que ele passe a usar autenticação HTTP, para que só você tenha acesso a ele. Envie o script para o seu servidor, e depois abra-o no seu navegador. Experimente digitar um nome e senha errados primeiro, para ver como o acesso é restringido.

Um nome de usuário e uma senha agora impedem o acesso não autorizado à página Admin.

Pontuações não podem ser removidas sem autorização

Os Guitar Warriors estão felizes por a aplicação agora estar segura!

### não existem Perguntas Ídiotas

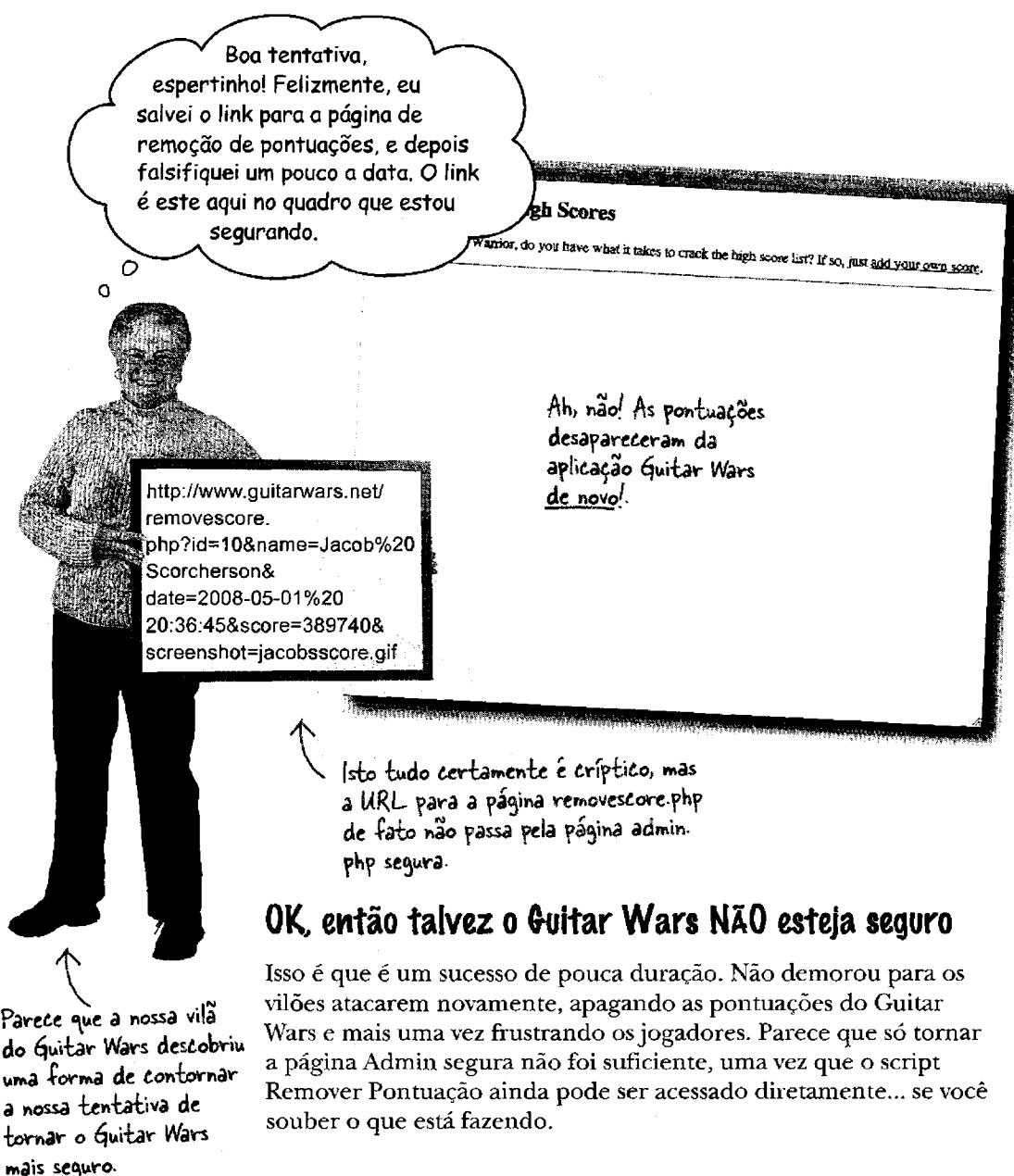
**P:** Quando exatamente a função exit() é chamada no script Admin do Guitar Wars?

**R:** Embora a função exit() apareça no código PHP logo abaixo das duas chamadas a header(), ela só é chamada se o usuário sair da janela de autenticação, clicando no botão Cancel. Se a autenticação falhar, o servidor não continua a execução para além das duas chamadas a header(). Em vez disso, ele re-envia os cabeçalhos e tenta novamente. Somente se o usuário clicar em Cancel é que o servidor chega à função exit(), em cujo caso ele envia o conteúdo da chamada à função, e nada mais. Se a autenticação tiver sucesso, exit() não é chamada, porque o script nunca chega à parte de dentro da declaração if – o código dentro de if só é executado se não existirem um nome e senha definidos, ou se eles tiverem sido digitados incorretamente.

**P:** O ":" de uma autenticação HTTP tem algum propósito real?

**R:** Sim, ele define uma zona de segurança que é protegida por um determinado nome de usuário e senha. Uma vez que o nome e senha tenham sido digitados corretamente, em um determinado realm, o navegador irá se lembrar deles, e não continuará a exibir a janela de autenticação para cabeçalhos subsequentes no mesmo realm. Em outras palavras, os realms permitem que o navegador se lembre que você já entendeu aos requerimentos de segurança para um determinado conjunto de páginas – basta especificar o mesmo realm para os cabeçalhos de autenticação dessas páginas.

## outro problema de segurança

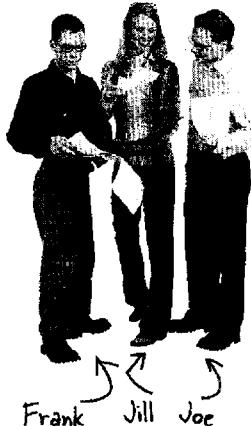


## OK, então talvez o Guitar Wars NÃO esteja seguro

Isso é que é um sucesso de pouca duração. Não demorou para os vilões atacarem novamente, apagando as pontuações do Guitar Wars e mais uma vez frustrando os jogadores. Parece que só tornar a página Admin segura não foi suficiente, uma vez que o script Remover Pontuação ainda pode ser acessado diretamente... se você souber o que está fazendo.

Escreva abaixo como você acha que poderíamos resolver este novo problema, evitando que as pontuações sejam apagadas:

.....  
.....  
.....



Frank Jill Joe

Nós precisamos tornar seguro o script Remover Pontuação, e tenho quase certeza que podemos simplesmente usar a autenticação HTTP novamente.

**Joe:** Faz sentido, já que funcionou bem para a página Admin.

**Frank:** É verdade. Então tudo o que temos de fazer é colocar o mesmo código de cabeçalho de autorização no script Remover Pontuação, e pronto. Certo?

**Jill:** Sim, isso certamente vai funcionar. Mas eu me preocupo em replicar todo aquele código de autorização em dois lugares. O que acontecerá se, no futuro, nós tivermos de adicionar outra página que precise ser protegida? Nós replicamos o código mais uma vez?

**Joe:** É; código duplicado é definitivamente um problema.

Principalmente porque há um nome de usuário e uma senha que todos os scripts precisam compartilhar. Se algum dia quisermos mudá-los, teremos de fazer a modificação em cada um dos scripts protegidos

**Frank:** Já sei! Que tal colocar as variáveis \$username e \$password em um arquivo include próprio, e depois compartilhá-lo entre os scripts protegidos? Nós poderíamos até mesmo colocá-las no arquivo include appvars.php das variáveis da aplicação

**Joe:** Eu gostei do raciocínio, mas essa solução só lida com uma pequena parte do problema de duplicação de código. Lembre-se, nós estamos falando de um pedaço de código relativamente grande.

```
<?php
// Nome do usuário e senha para autenticação
$username = 'rock';
$password = 'roll';

if (!isset($_SERVER['PHP_AUTH_USER']) || !isset($_SERVER['PHP_AUTH_PW']) ||
    ($_SERVER['PHP_AUTH_USER'] != $username) || ($_SERVER['PHP_AUTH_PW'] != $password)) {
    // Nome do usuário/senha incorretos, então enviar os cabeçalhos de autenticação
    header('HTTP/1.1 401 Unauthorized');
    header('WWW-Authenticate: Basic realm="Guitar Wars"');
    exit('<h2>Guitar Wars</h2>Desculpe, você precisa digitar uma senha válida para acessar
esta página.');
}
?>

<html>
```

**Jill:** Vocês dois têm razão, e é por isso que eu acho que precisamos de um novo arquivo include para armazenar todo o código de autorização, e não apenas as variáveis \$username e \$password.

**Frank:** Ah, e nós podemos simplesmente incluir esse script em qualquer página que quisermos proteger com autorização HTTP.

**Joe:** Isso mesmo! Só precisamos nos certificar de sempre incluí-lo em primeiro lugar, uma vez que ele depende dos cabeçalhos para todo o trabalho de autorização HTTP.

admin.php

## *criando authorize.php*

# Crie um script Autorizar

Nós já temos todo o código de que precisamos para um novo script, Autorizar; é só uma questão de passar o código de admin.php para um novo arquivo (authorize.php), e substituir o código original por uma instrução require\_once.

Estamos retirando este  
código de admin.php  
para o colocarmos no seu  
próprio arquivo de script,  
authorize.php.

```
<?php
// Nome do usuário e senha para autenticação
$username = 'rock';
$password = 'roll';

if (!isset($_SERVER['PHP_AUTH_USER']) || !isset($_SERVER['PHP_AUTH_PW']) ||
    ($_SERVER['PHP_AUTH_USER'] != $username) || ($_SERVER['PHP_AUTH_PW'] != $password)) {
    // Nome de usuário/senha incorretos, então enviar os cabeçalhos de autenticação
    header('HTTP/1.1 401 Unauthorized');
    header('WWW-Authenticate: Basic realm="Guitar Wars"');
    exit('<h2>Guitar Wars</h2>Desculpe, você precisa digitar uma senha válida para acessar esta página...');

?>
```



admin.php

```

<?php
// Nome do usuário e senha para autenticação
$username = 'rock';
$password = 'roll';

if (!isset($_SERVER['PHP_AUTH_USER']) || !isset($_SERVER['PHP_AUTH_PW']) ||
    $_SERVER['PHP_AUTH_USER'] != $username) || ($_SERVER['PHP_AUTH_PW'] != $password)) {
    // Nome do usuário/senha incorretos, então enviar os cabeçalhos de
    // autenticação ('HTTP/1.1 401 Unauthorized');
    header('WWW-Authenticate: Basic realm="Guitar Wars"');
    exit('<h2>Guitar Wars</h2>Desculpe, você precisa digitar uma senha válida para acessar
esta página.');
}
?>

```

Uma vez que o script Autorizar é compartilhado, é garantido que as duas páginas tenham o mesmo realm de autenticação, o que significa que elas compartilharão o mesmo nome de usuário e a mesma senha.

authorize.php

O script Autorizar compartilhado é incluído logo no início deste script, uma vez que ele chama a função header().

```

<?php
require_once('authorize.php');

<html>

```

```

<?php
require_once('authorize.php');

<html>

```

O código de autenticação no script Admin é substituído por uma única linha de código PHP.

admin.php

removescore.php

## PONTOS DE BALA

- Os scripts PHP podem usar cabeçalhos para controlar o modo como o servidor entrega conteúdo web ao navegador.
- A função PHP header() é usada para se enviarem cabeçalhos ao navegador, os quais podem ser usados para se redirecionar uma página, controlar o tipo de conteúdo da página ou requisitar a autenticação da página.
- Quando os cabeçalhos são enviados ao navegador usando-se a função header(), as chamadas a header() devem ser feitas antes de qualquer conteúdo ser enviado.
- Quando uma página é protegida com autenticação HTTP, o nome e a senha digitados pelo usuário ficam armazenados na superglobal \$\_SERVER.
- O “basic realm” de uma autenticação HTTP é uma zona de segurança que se associa a um determinado nome de usuário e senha, permitindo que várias páginas tenham a sua segurança feita conjuntamente.
- A função exit() do PHP finaliza um script PHP, impedindo que qualquer código após ela seja executado ou enviado de qualquer maneira ao navegador.

*segurança não existem perguntas idiotas*

## *não existem Perguntas Idiotas*

**P:** Ainda não entendo totalmente como a Ethel conseguiu contornar a segurança do Guitar Wars. O que foi que ela fez?

**R:** Ela se aproveitou da fraqueza inerente a se proteger apenas uma página (Admin), quando o recurso de remoção de pontuações na verdade usa duas páginas (Admin e Remover Pontuação). A página Admin apresenta uma série de links Remove, vinculados à página Remover Pontuação. Os detalhes sobre qual pontuação deve ser removida são passados na URL, permitindo que o script Remover Pontuação os acesse através da superglobal `$_GET`. Se você conseguisse construir uma URL legítima para a página Remover Pontuação, poderia remover pontuações sem jamais passar pela página Admin. Foi isso que Ethel fez.

**P:** Mas como ela sabia a estrutura da URL para a página Remover Pontuação?

**R:** Ela é bastante engenhosa, mas não é preciso ser gênio para fazer isso. Lembre-se de que ela mencionou que salvou o link para a página Remover Pontuação, quando o site inteiro estava desprotegido. Bem, um link salvo nos favoritos é apenas uma URL, e ela pôde usá-la para construir uma URL para acessar diretamente a página Remover Pontuação sem ter de passar pela página Admin.

**P:** OK, mas as pontuações foram reinseridas depois do último ataque. Isso não significa que as URLs antigas não deveriam funcionar mais, uma vez que as datas estariam diferentes?

**R:** Sim, é uma observação muito boa. Mas lembre-se, Ethel é bastante esperta. Ela pode facilmente olhar na página principal do Guitar Wars, anotar as novas datas, e então colocá-las na URL antiga para remover as novas pontuações sem qualquer problema. É importante nunca subestimar a habilidade que pessoas determinadas têm de fazer engenharia reversa nos seus scripts PHP, para explorar fraquezas do sistema.

**P:** Tudo bem, então a proteção das duas páginas, Admin e Remover Pontuação, consegue parar a Ethel. Mas isso não torna muito difícil remover pontuações legitimamente?

**R:** Não, de forma alguma. Sem a ajuda dos realms, certamente seria complicado remover as pontuações legitimamente, porque você teria de digitar o nome do usuário e a senha separadamente para as páginas Admin e Remover Pontuação. Mas lembre-se de que foi estabelecido um realm (domínio) que é o mesmo em ambas as páginas, significando que elas estarão dentro da mesma zona de segurança. É uma vez que você tenha passado pela janela de autenticação para uma página de um dado realm, o nome e a senha são lembrados para todo o realm. O resultado final é que digitar com sucesso o nome e a senha apenas uma vez já é suficiente para liberar ambas as páginas.



*Nunca subestime a  
habilidade que pessoas  
determinadas tem  
de fazer engenharia  
reversa nos seus  
scripts PHP, para  
explorar fraquezas do  
sistema.*



# TEST DRIVE

## Crie o script Autorizar e o inclua nos scripts Admin e Remover Pontuação, para torná-los seguros.

Crie um arquivo de texto chamado authorize.php, e digite nele o código para o script Autorizar. Depois, modifique o script admin.php para que ele inclua o script Autorizar em vez do código de autenticação HTTP propriamente dito. Adicione a mesma instrução require\_once ao início do script removescore.php, para que ele também fique protegido pela autenticação HTTP.

Envie todos os scripts para o seu servidor e depois experimente abrir o script Remover Pontuação diretamente no seu navegador. Talvez você precise limpar algumas sessões de autenticação HTTP prévias, no seu navegador, para que ele lhe peça novamente o nome e a senha – a maioria dos navegadores memorizam os domínios de autenticação, para que você não tenha de ficar redigitar o nome e a senha.

Agora é preciso digitar um nome de usuário e uma senha tanto para a página Admin quanto para a Remover Pontuação.

**Guitar Wars - High Scores**

http://www.guitarwars.net/removescore.php?  
id=10&  
name=Jacob%20Schorcherson&  
date=2008-05-01%2020:36:45&  
score=389740&  
screenshot=jacobsscore.gif

↑ Esta URL evita a página Admin e acessa a página Remover Pontuação diretamente.

A página Remover Pontuação é protegida, independentemente de como o usuário chega a ela.

**PODER DO CÉREBRO**

Você consegue pensar em outras maneiras pelas quais a aplicação Guitar Wars esteja em risco?

um fracasso de pontuação falsa

## Guitar Wars Episódio II: Ataque dos Clones de Pontuação

Lamentavelmente, a felicidade no universo do Guitar Wars não durou muito, porque pontuações fantasmas estão aparecendo por toda a parte, no lugar das pontuações legítimas... o que incita o ódio em todo o universo Guitar Wars. Parece que é perfeitamente possível danificar a lista de pontuações do Guitar Wars sem remover pontuações. Mas como?

**Guitar Wars - High Scores**

Welcome, Guitar Warrior. do you have what it takes to crack the high score list? If so, just add your own score.

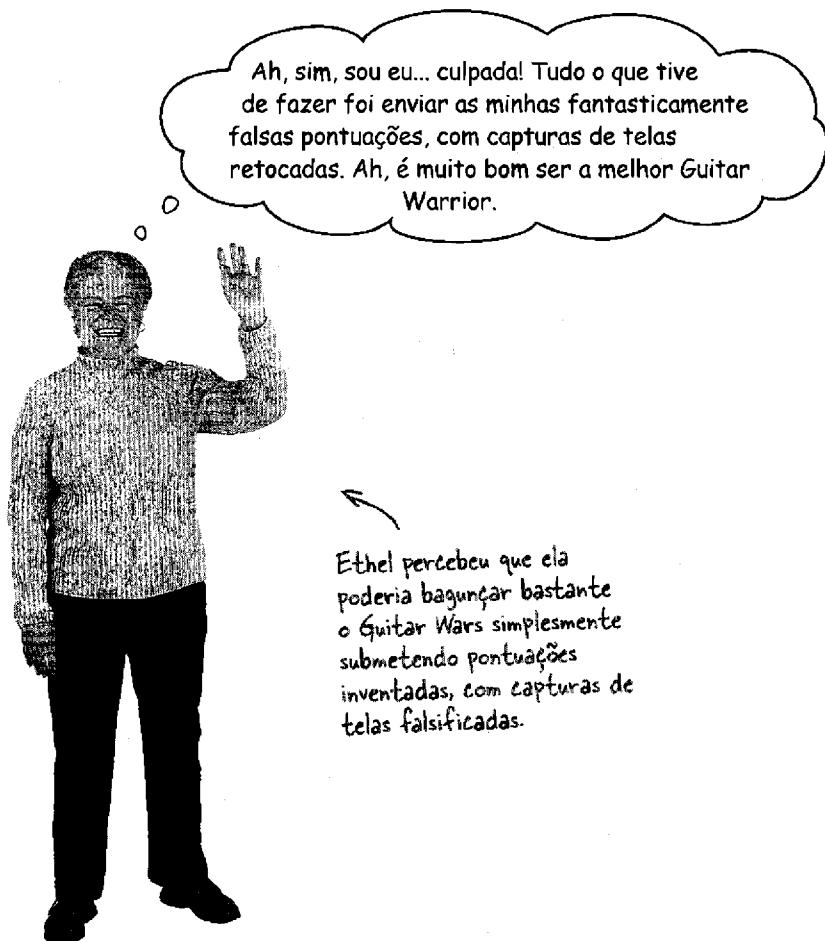
**Top Score: 500000**

Rank	Date	Name	Score	Image
1	2008-05-02 14:02:54	Ethel Heckel	500000	
2	2008-05-01 20:36:45	Jacob Scorcherson	389740	
3	2008-05-01 20:37:02	Nevil Johansson	98430	
4	2008-05-01 20:37:23	Paco Jastorius	127650	
5	2008-05-01 20:37:40	Phiz Lairston	186580	
6	2008-05-01 20:38:00	Kenny Lavitz	64930	
7	2008-05-01 20:38:23	Jean Paul Jones	243260	
8	2008-05-01 21:14:56	Lddy Gee	308710	
9	2008-05-01 21:15:17	T-Bone Taylor	354190	
10	2008-05-02 14:02:54	Ethel Heckel	500000	

O recorde de Ethel é claramente suspeito, devido à captura de tela mal falsificada, e ao fato de que ela "por acaso" conseguiu uma pontuação de exatamente 500.000.

## Subtração por adição

Até agora, vimos assumindo que qualquer pontuação submetida com uma captura de tela era considerada legítima. Agora, é razoavelmente seguro dizer que nem sempre é o caso! E está bem claro quem é o culpado...



**Escreva abaixo como você resolveria o problema de as pessoas conseguirem inserir pontuações falsificadas na aplicação Guitar Wars:**

.....  
.....  
.....

## A segurança requer intervenção humana

Mesmo neste mundo moderno em que vivemos, às vezes, um ser humano real, capaz de pensar, é imbatível. No caso em questão, é difícil ter uma solução melhor, para analisar uma determinada informação e avaliar se ela é válida ou não, do que uma pessoa real. Estamos falando da moderação, na qual um ser humano é encarregado de aprovar o conteúdo enviado a uma aplicação web, antes de esse conteúdo ser liberado para o público em geral.

Com a moderação, cada nova pontuação é adicionada ao banco de dados, mas não aparece para o público enquanto o moderador não a aprovar.

A modificação em Admin adiciona um link "Aprova" a cada nova pontuação, para que ela possa ser aprovada.

Guitar Wars - Add Your High Score

Name: Pez Luv  
Score: 322710  
Screen shot:  no file selected

Simplesmente adicionar uma nova pontuação não mais a adiciona automaticamente à lista que pode ser vista pelo público.

Score	Action
500000	<u>Remove / Approve</u>
389740	<u>Remove</u>

Guitar Wars - High Scores Administration

Name	Date	Score	Action
Taylor Taylor	2008-05-01 14:02:51	500000	Remove / Approve
Jacob Hernandez	2008-05-01 14:02:47	389740	Remove
Taylor Taylor	2008-05-01 14:02:47	389740	Remove
Pez Luv	2008-05-01 20:34:20	322710	Remove
Biff Jack	2008-05-02 20:27:54	31450	Remove
Laddy Gee	2008-05-02 21:14:49	30770	Remove
John Taylor	2008-05-01 20:34:07	28470	Remove
Zane Paul Jones	2008-05-01 20:34:23	24720	Remove
Phil Lefebvre	2008-05-01 20:34:24	17450	Remove
Paco Jimenez	2008-05-01 20:34:25	17700	Remove
West Johnson	2008-05-01 20:34:25	17700	Remove
Alamy Latvia	2008-05-01 20:34:50	9450	Remove

Vai tentando passar qualquer documento falsificado, digo, pontuação falsa, por mim. Eu sou muito atento e raramente cometo erros.

O Guitar Wars realmente poderia se beneficiar de alguma moderação. É claro que ainda é possível alguém fraudar cuidadosamente uma captura de tela, e talvez fazer passar uma pontuação falsa pelo moderador humano. Mas não será fácil, e isso não muda o fato de que a moderação é um ótimo método de desencorajar os fraudadores. Tenha em mente que fazer a segurança de uma aplicação PHP depende basicamente de uma boa prevenção.

Nosso destemido moderador do Guitar Wars... ele nunca se deparou com uma pontuação na qual confiasse totalmente.

A moderação humana é uma forma excelente de se melhorar a integridade do conteúdo submetido por usuários.

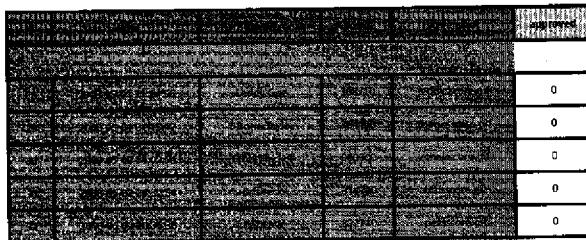


# Planeje moderação no Guitar Wars

Adicionar um recurso de moderação humana ao Guitar Wars é significativo porque isso afeta diversas partes da aplicação. O banco de dados precisa ser modificado, precisa ser criado um novo script para executar a aprovação, a página Admin precisa adicionar um link "Aprovar" a cada pontuação e, finalmente, a página principal precisa ser modificada de modo a exibir só as pontuações aprovadas. Com todas as modificações assim envolvidas, é importante ter um plano e executar uma modificação de cada vez.

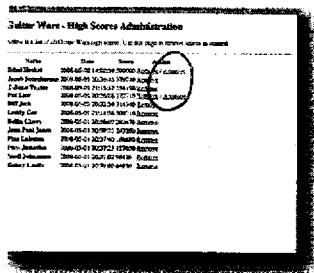
## 1 Use ALTER para adicionar uma coluna approved à tabela.

Vamos começar pelo banco de dados, o qual precisa de uma nova coluna para guardar a informação de se uma pontuação foi ou não aprovada.



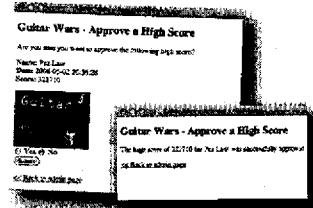
## 3 Modifique a página Admin para incluir um link "Aprovar" para as pontuações que ainda tenham de ser aprovadas.

O script Aprovar Pontuação fica no back-end e não deverá normalmente ser acessado diretamente. Em vez disso, ele é acessado através de links "Aprovar" gerados e exibidos na página Admin – somente as pontuações não aprovadas terão o link "Aprovar" ao lado delas.



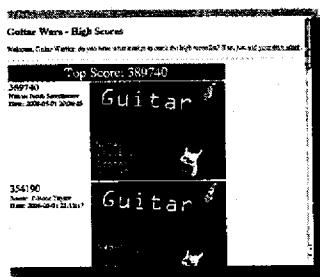
## 2 Crie um script Aprovar Pontuação para realizar a tarefa de aprovar uma nova pontuação (definir a coluna approved como 1).

Com o banco de dados pronto para acomodar a aprovação das pontuações, você precisa de um script para realizar essa tarefa. Este script Aprovar Pontuação é responsável por procurar uma determinada pontuação no banco e modificar a coluna approved dela.



## 4 Modifique a consulta da página principal para só exibir pontuações aprovadas.

O último passo é se certificar de que tudo o que estamos modificando seja refletido na exibição da página principal da aplicação. Assim, a página principal é modificada para só exibir as pontuações que tenham sido aprovadas – sem essa modificação, todas as outras seriam inúteis.



adicionando uma coluna APPROVED para a tabela guitarwars

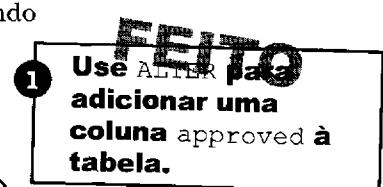
## Abra espaço para aprovações com ALTER

Para adicionar a nova coluna approved à tabela guitarwars, é preciso usar uma vez a instrução ALTER TABLE, que é um comando SQL que já vimos anteriormente.

```
ALTER TABLE guitarwars  
ADD COLUMN approved TINYINT
```

O tipo de dados BOOL do MySQL é um outro nome para TINYINT, portanto você pode usar qualquer um dos dois.

A nova coluna approved é do tipo TINYINT, e usa 0 para indicar uma pontuação não aprovada ou 1 para indicar uma pontuação aprovada. Assim, todos os novos registros devem começar tendo o valor 0, para indicar que são inicialmente não aprovados.



Espere um minuto. Não sei se você pode simplesmente ir adicionando uma coluna ao banco de dados sem modificar o script Adicionar Pontuação - ele não terá de inserir dados à nova coluna, com INSERT?



**É verdade, uma nova coluna implica um novo valor na consulta INSERT do script Adicionar Pontuação.**

É importante não perder de vista o fato de que uma aplicação PHP é um cuidadoso arranjo de diversas partes e pedaços: um banco de dados que consiste de tabelas com linhas e colunas, código PHP, código HTML, e geralmente código CSS. Nem sempre fica imediatamente claro que, para modificar uma parte, é preciso modificar outra. Adicionar a nova coluna approved na tabela guitarwars também exige a modificação da consulta INSERT no script Adicionar Pontuação:

```
INSERT INTO guitarwars  
VALUES (0, NOW(), '$name', '$score', '$screenshot', 0)
```

Todas as linhas das novas pontuações têm approved definida como 0... não aprovadas!

#	date	name	score	screenshot	approved
...					
30	2008-05-02 14:02:54	Ethel Heckel	500000	ethelsscore.gif	0
31	2008-05-02 20:32:54	Biff Jeck	314340	biffsscore.gif	0
32	2008-05-02 20:36:38	Pez Law	322710	pezsscore.gif	0

Quando uma nova linha é adicionada, a sua coluna approved é definida como 0, para que ela, de início, seja não aprovada.



## Aponte seu lápis

O script Aprovar Pontuação tem estrutura semelhante ao Remover Pontuação, exceto pelo fato de que a sua tarefa é aprovar pontuações. Complete o código do script Aprovar Pontuação abaixo, certificando-se de fazer a página segura, e de só aprovar a pontuação apropriada, com base em dados enviados através de uma URL.

```
<?php
....;
?>
...
<?php
    require_once('appvars.php');
    require_once('connectvars.php');
    ...
    if (isset($_POST['submit'])) {
        if (.....) {
            // Conecta-se ao banco de dados
            $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
            // Aprova a pontuação, definindo a coluna approved do banco de dados
            $query = "UPDATE guitarwars SET .....";
            mysqli_query($dbc, $query);
            mysqli_close($dbc);

            // Confirma o êxito com o usuário
            echo .....
        }
        else {
            echo .....
        }
    }
    ...
    echo '<p><a href=".....">&lt;&lt; Voltar à página Admin</a></p>';
?>
...
```

*o script para aprovar pontuação completo*

## Aponte seu lápis Solução

O script Aprovar Pontuação tem estrutura semelhante ao Remover Pontuação, exceto pelo fato de que a sua tarefa é aprovar pontuações. Complete o código do script Aprovar Pontuação abaixo, certificando-se de fazer a página segura, e de só aprovar a pontuação apropriada, com base em dados enviados através de uma URL.

```
<?php  
require_once('authorize.php')  
?  
...  
<?php  
require_once('appvars.php');  
require_once('connectvars.php');  
...  
if (isset($_POST['submit'])) {  
    if (.....  
        if ($_POST['confirm'] == 'Yes') {  
            // Conecta-se ao banco de dados  
            $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);  
            // Aprova a pontuação, definindo a coluna approved do banco de dados  
            $query = "UPDATE guitarwars SET approved = 1 WHERE id = ?";  
            mysqli_query($dbc, $query);  
            mysqli_close($dbc);  
            // Confirma o êxito com o usuário  
            echo '<p>A pontuação de ' . $score . ' para ' . $name . ' foi aprovada com sucesso.';  
        }  
    }  
    else {  
        echo '<p class="error">Desculpe, houve um problema para aprovar a pontuação.</p>';  
    }  
    ...  
    echo '<p><a href="admin.php" ">&lt;&lt; Voltar à página Admin</a></p>';  
?  
...  
}
```

*Tudo o que você precisa fazer para garantir a segurança da página Aprovar Pontuação é incluir o script Autorizar, mas isso deve ser a primeira coisa no script, uma vez que o processo depende dos cabeçalhos.*

**2 Crie um script Aprovar Pontuação para realizar a tarefa de aprovar uma nova pontuação (definir a coluna approved como 1).**

*A ID precisa ser verificada, para que a aprovação seja feita.*

*approved = 1 WHERE id = ?*

*Definir a coluna approved como 1 aprova a pontuação.*

*Confirme a aprovação com o usuário, mostrando a pontuação aprovada e o nome.*

*É importante informar ao usuário quando uma pontuação não puder ser aprovada, de forma semelhante a como os outros scripts do Guitar Wars relatam os erros.*

*Forneca um link de volta à página Admin, para facilitar a navegação.*

não existem  
Perguntas Idiotas

**P:** Por que não é necessário informar o nome do arquivo gráfico para se aprovar uma pontuação?

**R:** Porque o processo de se aprovar uma pontuação só exige as informações necessárias para se achar uma determinada linha na tabela, e então aprová-la. Isso significa que você só precisa informar o estritamente necessário para achar a linha em questão. A data, o nome e a pontuação são suficientes para se especificar uma determinada linha e definir a sua coluna approved como 1.

**P:** Parece pouco informativo usar 0 e 1 na coluna approved. Não há outras maneiras de se representar essa informação?

**R:** Sim. O tipo de dados ENUM do MySQL, que significa "enumerado", lhe permite criar uma coluna com uma lista restrita de possíveis valores. Assim, em vez de adicionar a coluna approved como um TINYINT que deverá ter o valor 0 ou 1, você poderia adicioná-la como um ENUM que pudesse ter apenas os valores 'yes' e 'no', desta forma:

```
ALTER TABLE guitarwars
ADD COLUMN approved ENUM('yes',
                           'no')
```

### Aponte seu lápis

Os dados das pontuações usados para aprová-las, no script Aprovar Pontuação, são passados através de links "Aprovar" que são gerados no script Admin. Complete o código do script Admin para que ele gere esses links.

```
// Faz um loop através do array contendo os dados das pontuações,
formatando-os como HTML
...
echo '<table>';
echo '<tr><th>Name</th><th>Date</th><th>Score</th><th>Action</th></tr>';
while ($row = mysqli_fetch_array($data)) {
    // Exibe os dados das pontuações
    echo '<tr class="scorerow"><td><strong>' . $row['name'] . '</strong></td>';
    echo '<td>' . $row['date'] . '</td>';
    echo '<td>' . $row['score'] . '</td>';
    echo '<td><a href="removescore.php?id=' . $row['id'] . '&date=' .
        $row['date'] . '&name=' . $row['name'] . '&score=' . $row['score'] . '&
        screenshot=' . $row['screenshot'] . '">Remove</a>';
    if (.....) {
        echo .....
        .....
        .....
    }
    echo '</td></tr>';
}
echo '</table>';
```

Dica: só as pontuações não aprovadas é que devem ter um link "Aprovar".

## gerando links de aprovação



# Aponte seu lápis

## Solução

Os dados das pontuações usados para aprová-las, no script Aprovar Pontuação, são passados através de links "Aprovar" que são gerados no script Admin.

Complete o código do script Admin para que ele gere esses links.

```
// Faz um loop através do array contendo os dados das pontuações,  
formatando-os como HTML  
  
echo '<table>';  
echo '<tr><th>Name</th><th>Date</th><th>Score</th><th>Action</th></tr>';  
while ($row = mysqli_fetch_array($data)) {  
  
    // Exibe os dados das pontuações  
    echo '<tr class="scorerow"><td><strong>' . $row['name'] . '</strong></td>';  
    echo '<td>' . $row['date'] . '</td>';  
    echo '<td>' . $row['score'] . '</td>';  
    echo '<td><a href="removescore.php?id=' . $row['id'] . '&  
&date=' . $row['date'] . '&  
' . $row['name'] . '&score=' . $row['score'] . '&  
' . $row['Screenshot'] . '">Remove</a>;  
    if (.....$row['approved'] == '0'.....) {  
        echo ' / <a href="approvescore.php?id=' . $row['id'] . '&  
&date=' . $row['date'] . '&  
' . $row['name'] . '&score=' . $row['score'] . '&  
' . $row['Screenshot'] . '">Approve</a>;  
    }  
    echo '</td></tr>';  
}  
echo '</table>';  
...
```

Verifica se a pontuação está aprovada ou não, antes de gerar o link "Aprovar".

Gera o link "Aprovar", de modo tal que a id, a data, o nome, a pontuação e o nome do arquivo gráfico são passados na URL.

O link "Aprovar" liga a página Admin à página Aprovar Pontuação.

Guitar Wars - High Scores Administration

Select a list of Guitar War high scores. Use this page to remove scores as needed.

Name	Date	Score	Action
Daniel Mitchell	2008-05-02 14:05:54	94000	<a href="#">Remove</a>
Paul Seachrist	2008-05-02 14:05:54	94000	<a href="#">Remove</a>
T. Michael Peller	2008-05-02 21:15:45	84710	<a href="#">Remove</a>
Pete Law	2008-05-02 20:27:40	82210	<a href="#">Remove</a>
Bill Jack	2008-05-02 20:28:24	81400	<a href="#">Remove</a>
Lenny Gau	2008-05-01 21:14:57	80770	<a href="#">Remove</a>
Julian Clegg	2008-05-01 21:14:57	80770	<a href="#">Remove</a>
James Jones	2008-05-01 21:14:57	80770	<a href="#">Remove</a>
Pete Lakin	2008-05-01 20:27:40	76500	<a href="#">Remove</a>
Pete Johnson	2008-05-01 20:27:40	127650	<a href="#">Remove</a>
Neill Johnson	2008-05-01 20:27:40	84900	<a href="#">Remove</a>
Kenny Lenitz	2008-05-01 20:28:00	84900	<a href="#">Remove</a>

Guitar Wars - Approve a High Score

Are you sure you want to approve the following high score?

Name: Pete Law  
Date: 2008-05-02 21:15:45  
Score: 82210

Yes  No

Guitar Wars - Approve a High Score

The high score of 82210 for Pete Law was successfully approved.  
Click here to return to the list.

**FEITO**  
Modifique a página Admin para incluir um link "Aprovar" para as pontuações que ainda tenham de ser aprovadas.

## Pontuações não aprovadas não são dignas

Agora, já temos toda a infraestrutura necessária para o recurso de moderação na aplicação Guitar Wars. Só falta o último passo, que é alterar a página principal de modo que ela mostre apenas as pontuações aprovadas. Para isso, teremos de ajustar a consulta SQL SELECT de modo tal que ela só obtenha os registros cuja coluna approved esteja definida como 1 (aprovada). Isso é feito por meio de uma instrução WHERE.

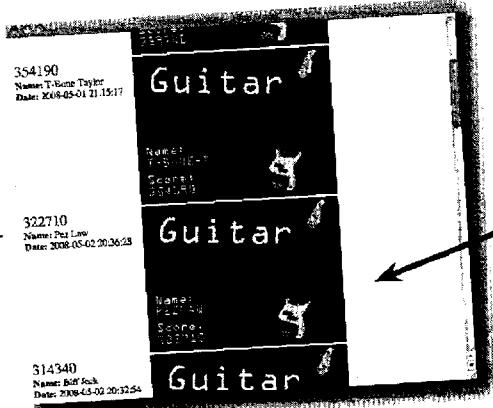
```
SELECT * FROM guitarwars
WHERE approved = 1
ORDER BY score DESC, date ASC
```

A adição da instrução WHERE à esta consulta elimina quaisquer pontuações que não tenham sido aprovadas, o que inclui todas as novas inserções. Isso dá ao moderador a chance de analisá-las e decidir se elas devem ser removidas ou liberadas (aprovadas) para visualização pública.

Use WHERE para selecionar linhas com base no valor de uma determinada coluna.

Se a coluna approved estiver definida com algo diferente de 1, a pontuação não será exibida.

#	date	name	score	screenshot	approved
28	2008-05-01 21:14:56	Leddy Gee	308710	leddysscore.gif	1
29	2008-05-01 21:15:17	T-Bone Taylor	354190	tbonesscore.gif	1
30	2008-05-02 14:02:54	Ethel Heckel	500000	ethelsscore.gif	0
31	2008-05-02 20:32:54	Biff Jeck	314340	biffsscore.gif	1
32	2008-05-02 20:36:38	Pez Law	322710	pezsscore.gif	1



Agora, somente as pontuações aprovadas aparecem na página principal (index.php).

Modifique a consulta da página principal para só exibir pontuações aprovadas.

FEITO



## **TEST DRIVE**

### **Crie o script Aprovar e ajuste o restante da aplicação Guitar Wars para usá-lo.**

Usando alguma ferramenta MySQL, emita a consulta ALTER para adicionar a nova coluna approved à tabela guitarwars. Em seguida, modifique a consulta INSERT do script addscore.php para inserir um 0, na coluna approved, para novas linhas de dados.

Agora, crie um arquivo de texto chamado aprovescore.php e digite nele o código para o script Aprovar Pontuação. Em seguida, modifique o script admin.php de modo que ele inclua um link "Aprovar" para pontuações que ainda precisem ser aprovadas. Finalmente, modifique a consulta SELECT de index.php para que ela mostre apenas as pontuações aprovadas.

Envie todos os scripts para o seu servidor, e abra a página principal do Guitar Wars no seu navegador. Tome nota de quais pontuações estão visíveis, e depois abra a página Admin. Clique em um dos links "Aprovar" e realize a operação de aprovar o registro. Depois, volte à página principal para ver se esse registro agora está aparecendo.

**Guitar Wars - High Scores Administration**

Below is a list of all Guitar Wars high scores. Use this page to remove scores as needed.

Name	Date	Score	Action
Ethel Heckel	2008-05-02 14:02:54	500000	Remove / Approve
Jacob Scorcherson	2008-05-01 20:36:45	389740	Remove
T-Bone Taylor	2008-05-01 21:15:17	354190	Remove
Pez Law	2008-05-02 20:36:28	322710	Remove / Approve
Biff Jeck	2008-05-02 20:32:54	314340	Remove
Leddy Gee	2008-05-01 21:14:56	308710	Remove
Belita Chevy	2008-05-01 20:36:07	282470	Remove
Jean Paul Jones	2008-05-01 20:38:23	242260	Remove
Phiz Lefstrom	2008-05-01 20:37:40	186580	Remove
Paco Jostarius	2008-05-01 20:37:23	127650	Remove
Nevil Johansson	2008-05-01 20:37:02	98430	Remove
Kenny Lavitz	2008-05-01 20:38:00	64930	Remove

*Os novos links "Aprovar" da página Admin fornecem acesso à página Aprovar Pontuação, onde os registros podem ser aprovados individualmente.*

**Guitar Wars - Approve a High Score**

Are you sure you want to approve the following high score?

Name: Pez Law  
Date: 2008-05-02 20:36:28  
Score: 322710

**Guitar**

Yes  No  
[Submit](#)

Após a operação ter sido concluída com sucesso, uma mensagem de confirmação é exibida.

**Guitar Wars - Approve a High Score**

The high score of 322710 for Pez Law was successfully approved.

[« Back to admin page](#)

Um simples formulário requer confirmação antes de realmente aprovar o registro.

354190  
Name: T-Bone Taylor  
Date: 2008-05-01 21:15:17

**Guitar**

Name:  
Score:  
322710

322710  
Name: Pez Law  
Date: 2008-05-02 20:36:28

**Guitar**

Name:  
Score:  
322710

314340  
Name: Biff lock  
Date: 2008-05-02 20:32:54

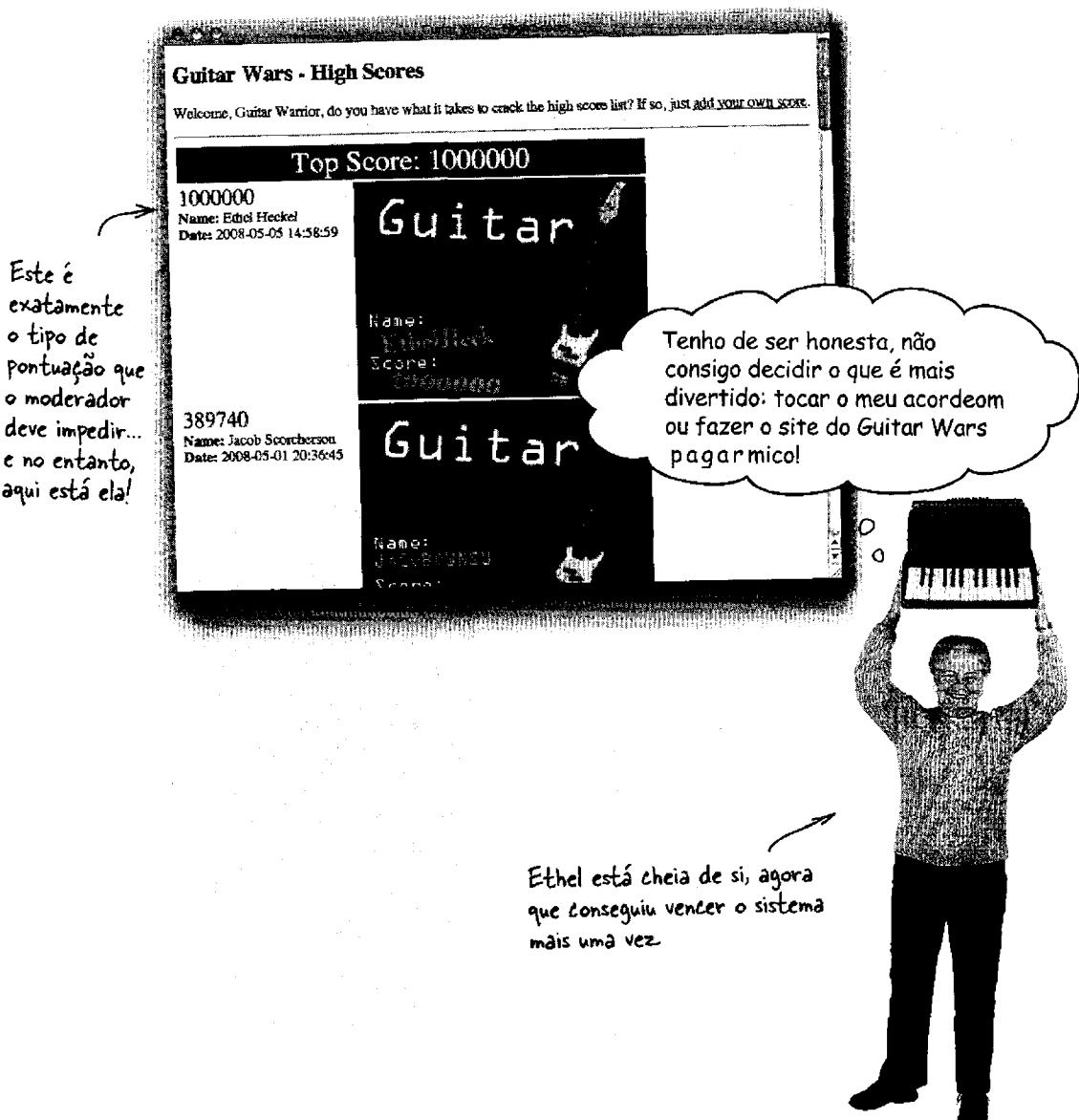
**Guitar**

A recém-aprovada pontuação agora aparece na página principal do Guitar Wars.

*ethel marca de novo*

## O hack de um milhão de pontos

A versão moderada do Guitar Wars representa uma melhoria significativa na segurança, mas está longe de ser completamente à prova de balas. Parece que a nossa infiltradora conseguiu achar outra fraqueza no sistema, e de alguma forma está conseguindo fazer as suas pontuações passarem pelo moderador. Ethel precisa ser detida, permanentemente, para que possamos restaurar a confiança no universo do Guitar Wars.



# Tudo em moderação...?

Embora o moderador saiba, sem sombra de dúvida, que ele não aprovou a pontuação submetida por Ethel, está claro que essa pontuação chegou ao banco, com a coluna approved definida como 1. Nós sabemos que o script Adicionar Pontuação define a coluna approved como 0 para novas pontuações, porque acabamos de modificar a consulta INSERT desse script. Algo simplesmente não está certo!

O moderador do  
Guitar Wars não  
consegue entender  
o que aconteceu.

Como é  
possível? Eu sei que não  
aprovei essa pontuação. Um  
milhão de pontos!



<b>id</b>	<b>date</b>	<b>name</b>	<b>score</b>	<b>screenshot</b>	<b>approved</b>
21	2008-05-01 20:36:07	Belita Chevy	282470	belitasscore.gif	1
22	2008-05-01 20:36:45	Jacob Scorcherson	389740	jacobsscore.gif	1
23	2008-05-01 20:37:02	Nevil Johansson	98430	nevillsscore.gif	1
24	2008-05-01 20:37:23	Paco Jastorius	127650	pacosscore.gif	1
25	2008-05-01 20:37:40	Phiz Lairston	186580	phizsscore.gif	1
26	2008-05-01 20:38:00	Kenny Lavitz	64930	kennysscore.gif	1
27	2008-05-01 20:38:23	Jean Paul Jones	243260	jeanpaulsscore.gif	1
28	2008-05-01 21:14:56	Leddy Gee	308710	leddysscore.gif	1
29	2008-05-01 21:15:17	T-Bone Taylor	354190	tbonesscore.gif	1
31	2008-05-02 20:32:54	Biff Jeck	314340	biffsscore.gif	1
32	2008-05-02 20:36:38	Pez Law	322710	pezsscore.gif	1
33	2008-05-05 14:58:59	Ethel Heckel	1000000	ethelsscore2.gif	1



Como você acha que o post falso de Ethel conseguiu passar  
pelo moderador?

↑  
Esta pontuação  
não foi aprovada  
pelo moderador,  
e no entanto,  
a sua coluna  
approved está  
definida como 1,  
o que resulta em  
ela ser exibida.

tente o golpe da Ethel



## EXERCITANDO O CÉREBRO

Ocorre que o hack de um milhão de pontos de Ethel não teve nada a ver com o formulário Aprovar Pontuação. Foi descoberto que a invasão dela se baseou apenas no formulário Adicionar Pontuação. Segue abaixo os dados exatos que Ethel digitou no formulário Adicionar Pontuação; digite os mesmos dados no seu próprio formulário e adicione o registro. O que você acha que está acontecendo?

Não se esqueça do  
espaço depois de --,  
aqui.

Ethel Heckel

1000000', 'ethelsscore2.gif', 1) --

The screenshot shows a web page titled "Guitar Wars - Add Your High Score". It has fields for "Name:" and "Score:", both currently empty. Below these is a "Screen shot:" field with a "Choose File" button. At the bottom is an "Add" button. Arrows from the text above point to the "Name:" field and the "Score:" field.

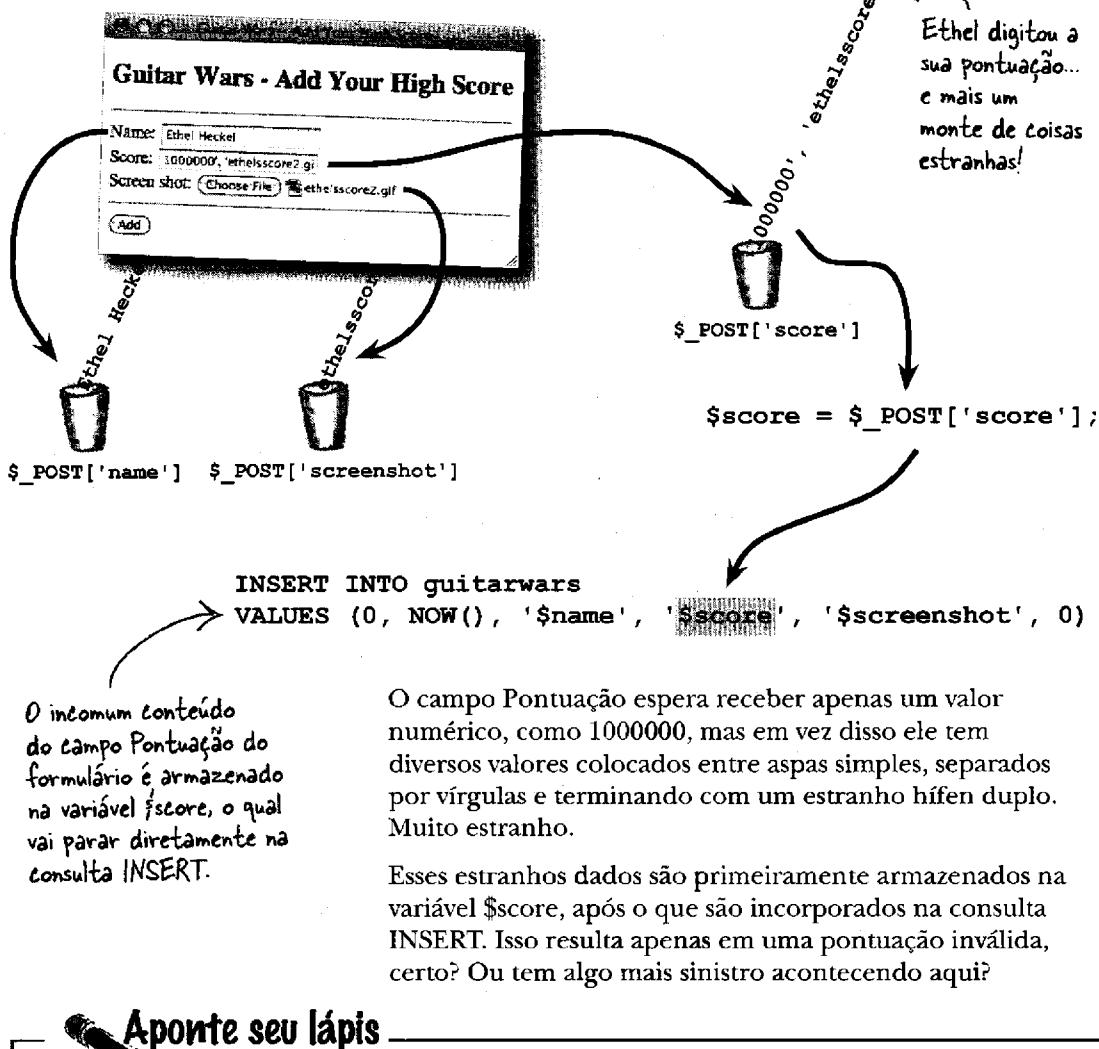
Este pode ser qualquer arquivo  
gráfico GIF ou JPEG com menos  
de 32KB.

ethelsscore2.gif



## Como exatamente ela fez isso?

Para entender o que está acontecendo neste inteligente tipo de ataque a formulários, vamos rastrear o fluxo dos dados enquanto eles viajam através do script Adicionar Pontuação.



### Aponte seu lápis

Usando os dados exatos que foram mostrados na página anterior, escreva aqui a consulta SQL inteira do ataque de um milhão de pontos. Certifique-se de substituir as variáveis da consulta pelos dados propriamente ditos. Escreva também o que você acha que está acontecendo.

.....  
.....

como uma injeção de SQL funciona

## Aponte seu lápis Solução

INSERT INTO guitarwars

VALUES (0, NOW(), 'Ethel Heckel', '1000000', 'ethelsscore2.gif', 1) -- ', 'ethelsscore2.gif', 0)

De alguma forma,  
Ethel criou a sua  
própria versão da  
consulta, a qual  
está substituindo  
a consulta original.

Que consulta estranha. O nome do  
arquivo gráfico aparece duas vezes,  
e não sei o que significa o duplo hífen...  
essa consulta funciona?

Uma vez que a coluna  
approved é a última na  
estrutura do banco de dados,  
ela está sendo forçada a ter  
um valor de 1... aprovado!



## Enganando o MySQL com comentários

O verdadeiro culpado pelo ataque de um milhão de pontos de Ethel, por incrível que pareça, são os comentários SQL. O hífen duplo (--) é usado, em SQL, para desativar o restante de uma linha de código SQL, transformando-o em um comentário. É preciso haver um espaço depois do hífen duplo para que ele funcione (--) , mas tudo o que vier depois do espaço é ignorado. Agora, dê outra olhada na consulta de Ethel, levando em conta esse novo conhecimento.

INSERT INTO guitarwars

VALUES (0, NOW(), 'Ethel Heckel', '1000000', 'ethelsscore2.gif', 1) -- ',  
~~'ethelsscore2.gif'~~

Faz mais sentido agora? Na prática, o comentário apagou o restante do código SQL, para que ele não gerasse um erro, permitindo que a versão da consulta criada por Ethel pudesse chegar ao sistema sem percalços. O resultado final é uma pontuação instantaneamente aprovada, a qual o moderador nunca teve a chance de analisar.

O comentário -- faz com  
que o restante da linha de  
código SQL seja ignorado.

-- ,  
↑

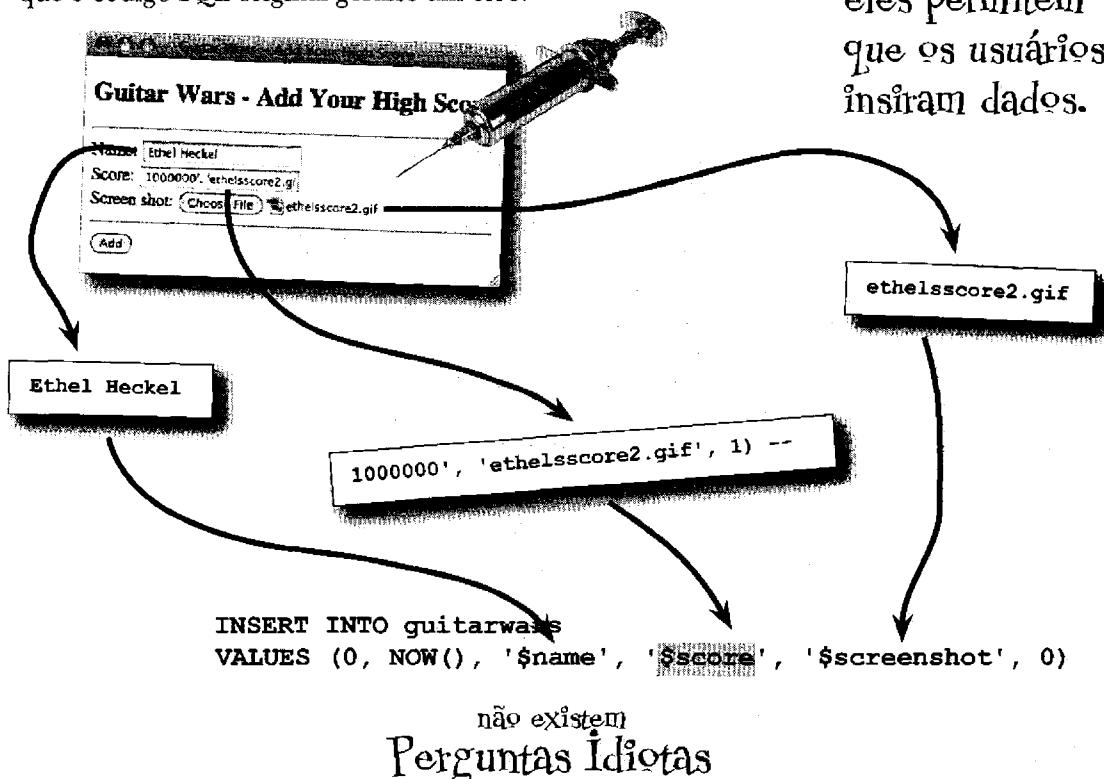
Ethel enganou o  
sistema, fazendo-o  
aprovar a sua  
pontuação.

id	date	name	score	screenshot	approved
33	2008-05-05 14:58:59	Ethel Heckel	1000000	ethelsscore2.gif	1

## O formulário Add Score recebeu uma injeção de SQL.

O ataque de Ethel é conhecido como injeção de SQL, e envolve um truque extremamente sorrateiro, no qual os dados do formulário são usados para se modificar a operação fundamental de uma consulta. Assim, em vez de o campo do formulário apenas fornecer uma informação, como um nome ou uma pontuação, ele altera a própria consulta SQL subjacente. No caso do Guitar Wars, a injeção de SQL de Ethel usou o campo Pontuação como um meio de não somente fornecer a pontuação, mas também o nome do arquivo gráfico, o valor para approved e um comentário no final, para evitar que o código SQL original gerasse um erro.

Os campos do formulário são um ponto fraco na segurança das aplicações Web, porque eles permitem que os usuários insiram dados.



**P:** Existem outros tipos de comentários em SQL, fora --?

**R:** Sim. Outra variante de comentário envolve o uso de # em vez de --, mas tem obviamente o mesmo efeito de anular qualquer código SQL até o final da linha que contém o comentário. O SQL também oferece suporte a comentários multilinhas, semelhantes àqueles do PHP no sentido que você coloca o código comentado (desativado) entre /\* e \*/.

**P:** O ataque de injeção SQL de Ethel ainda teria funcionado se a coluna approved não fosse a última da tabela?

**R:** Não, e essa é uma questão realmente importante. A consulta INSERT que foi usada depende da ordem padrão das colunas na tabela. Colocar 1 no final da consulta funcionou por acaso, porque approved era a última coluna, aparecendo imediatamente depois da coluna screenshot.

## Proteja seus dados contra injeções de SQL

A fraqueza que as injeções de SQL realmente exploram reside nos campos de formulários que não são validados para impedir a inserção de caracteres perigosos. "Caracteres perigosos" são todos aqueles que tenham o potencial de modificar a natureza de uma consulta SQL, como vírgulas, aspas ou caracteres de comentários como --. Até mesmo espaços ao final de algum dado podem se revelar perigosos. É fácil eliminar espaços com a função interna do PHP trim() – basta passar todos os dados de formulários através da função trim() antes de usá-los em uma consulta SQL.

```
$name = trim($_POST['name']);  
$score = trim($_POST['score']);  
$screenshot = trim($_FILES['screenshot']['name']);
```

A função trim() se  
livra de quaisquer  
espaços que apareçam  
antes ou depois dos  
dados deste formulário.

Mas os espaços não são o único problema. Ainda existem as vírgulas, aspas, caracteres de comentários e assim por diante. Portanto, além de retirar os espaços em branco, nós também precisamos de uma forma de encontrar e neutralizar outros caracteres problemáticos. O PHP nos auxilia com outra função interna, mysqli\_real\_escape\_string(), a qual faz escape de caracteres potencialmente perigosos, para que eles não possam afetar o modo como uma consulta se executa. Esses caracteres ainda poderão aparecer como dados em campos do formulário, eles só não serão capazes de interferir nas consultas.

Juntas, as funções trim() e mysqli\_real\_escape\_string() fornecem uma sólida linha de defesa contra injeções de SQL.

As injeções SQL  
podem ser evitadas  
processando-se  
devidamente os  
dados de formulários.

A função mysqli\_real\_escape\_string() converte caracteres perigosos em um formato com escape, o qual não é capaz de afetar as consultas SQL.

```
$name = mysqli_real_escape_string($dbc, trim($_POST['name']));  
$score = mysqli_real_escape_string($dbc, trim($_POST['score']));  
$screenshot = mysqli_real_escape_string($dbc, trim($_FILES['screenshot']['name']));
```

mysqli\_real\_escape\_string()  
é considerada uma função  
de bancos de dados, e é por  
isso que ela exige que você  
lhe forneça uma variável de  
conexão a banco de dados, da  
mesma forma que aquela usada  
ao se submeter consultas.

Processar os três campos do formulário Guitar Wars com as funções trim() e mysqli\_real\_escape\_string() reduz em muito as chances de um outro ataque de injeção SQL. Mas elas não são o suficiente – talvez haja uma forma de tornar a própria consulta menos vulnerável...

## Um INSERT mais seguro (com parâmetros)

Além da fraqueza dos campos do formulário, a injeção de SQL de Ethel também explorou o fato de a coluna approved vir depois da coluna screenshot na estrutura do banco de dados. Foi assim que ela conseguiu hackear o sistema apenas adicionando 1 ao final de INSERT, e fazendo-o ser colocado na coluna approved. O problema é que a consulta INSERT é estruturada de forma tal que ela precisa inserir dados em todas as colunas, o que aumenta desnecessariamente o risco.

O ideal é que não precisássemos definir as colunas id e approved, uma vez que elas podem ficar com os valores padrões.

```
INSERT INTO guitarwars
VALUES (0, NOW(), '$name', '$score', '$Screenshot', 0)
```

Uma consulta INSERT pode ser escrita de forma a especificar exatamente quais valores vão para quais colunas.

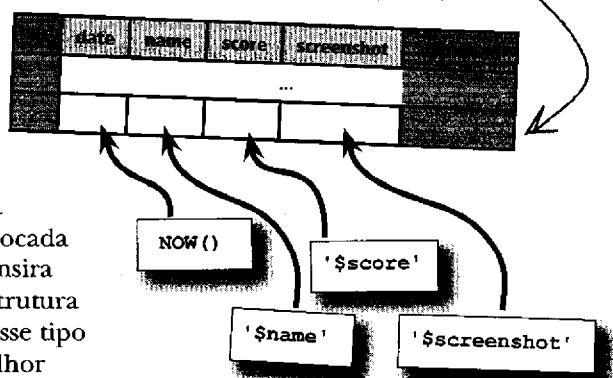
Quando dados são inseridos em uma tabela desta forma, a ordem deles precisa bater com a ordem das colunas na estrutura da tabela. Assim, a quinta informação será colocada na coluna screenshot porque esta é a quinta coluna da tabela. Mas na verdade não é necessário inserir explicitamente as colunas id ou approved, uma vez que id é autoincrementada e approved deve sempre ser 0. Uma abordagem melhor é inserir apenas os dados explicitamente requeridos em uma nova pontuação. As colunas id e approved poderão então usar os padrões de AUTO\_INCREMENT e 0, respectivamente.

Nós precisamos de uma consulta INSERT reestruturada, que espere uma lista de colunas antes da lista de dados, com um pareamento exato. Isso elimina o risco de a coluna approved ser definida – ela não faz mais parte da consulta. Se esse tipo de consulta parece familiar, é porque você a usou diversas vezes em outros exemplos.

```
INSERT INTO guitarwars ([id], [name], [score], [Screenshot])
VALUES (NOW(), '$name', '$score', '$Screenshot')
```

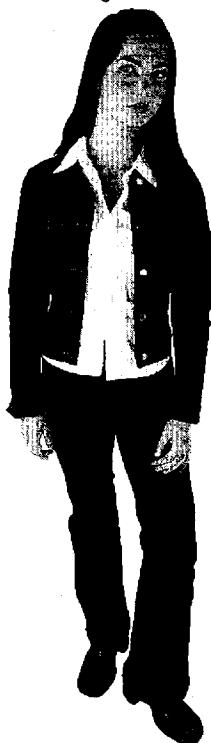
Nada pode ser inserido na coluna approved, porque ela não está listada como parte da consulta.

A coluna id pode ser deixada de fora, uma vez que ela se autoincrementa de qualquer forma.



Esta versão da consulta INSERT informa exatamente em qual coluna deve ser colocada qual informação, permitindo que você insira dados sem ter de se preocupar com a estrutura subjacente da tabela. Na verdade, usar esse tipo de INSERT é considerado como um melhor estilo de programação, porque os dados são inseridos exatamente onde você quiser, em vez de o destino deles ser ditado pelo layout da tabela.

## **o comando DEFAULT**



Espere um pouco. Esta é a primeira vez que ouço falar de valores padrões em tabelas MySQL. Isso é realmente possível?

**Não só é possível, como é uma ótima ideia especificar valores DEFAULT para as colunas sempre que possível.**

O comando SQL DEFAULT é o que lhe permite especificar um valor padrão para uma coluna. Se a coluna tiver um valor padrão, você não precisa defini-la em consultas INSERT, e pode ter a certeza de que ela automaticamente adotará o valor padrão. Isso é perfeito para a coluna approved da tabela guitarwars. Agora nós só precisamos modificar a tabela mais uma vez, para definir o valor padrão de approved como 0 (não aprovado).

Uma vez que a coluna approved já existe nesta instrução ALTER TABLE, nós temos de usar MODIFY COLUMN em vez de ADD COLUMN.

em vez de ADD

ALTER TABLE guitarwars  
MODIFY COLUMN approved TINYINT  
**DEFAULT 0**

DEFAULT faz a coluna approved ser assinalada automaticamente com 0, a menos que um INSERT especifique de outra forma.

Você precisa especificar o tipo da coluna – garanta que seja o mesmo da 1<sup>a</sup>vez que adicionou a coluna.

Com a coluna approved agora alterada para adotar um valor padrão, a nova e melhorada consulta INSERT do script Adicionar Pontuação pode inserir pontuações sem sequer mencionar a coluna approved. Isto é um bom design, uma vez que não há necessidade de inserir explicitamente um valor que pode ser padronizado, e ainda adiciona uma pequena dose de segurança, ao não expor a coluna approved a potenciais ataques.

# A validação de formulários nunca é inteligente demais

Um último passo para se minimizar o risco de ataques de injeção SQL envolve a validação do formulário no script Adicionar Pontuação. Antes de verificar se o tipo ou o tamanho do arquivo gráfico estão dentro dos limites estabelecidos pela aplicação, os três campos do formulário são verificados para garantir que não estejam vazios.

```
if (!empty($name) && !empty($score) && !empty($screenshot)) {  
    ...  
}
```

Não há nada de errado com o código atual, mas o processo de garantir a segurança de uma aplicação frequentemente exige que você dê um passo além do habitual. Uma vez que o campo Pontuação espera um número, faz sentido verificar não apenas se existe um valor, mas se o valor existente é numérico. A função PHP `is_numeric()` faz exatamente isso, retornando `true` se o valor passado a ela for um número ou `false` caso contrário. Ela está sempre fazendo os trabalhos triviais, como por exemplo, verificar se foi recebido um número quando um número era o que você esperava, que no fim das contas vão tornar a sua aplicação tão segura quanto possível.

```
is_numeric(465730)  
    ↑ Verdadeiro  
        ↓ is_numeric('one million!')  
            ↑ Falso.  
                ↓ is_numeric(0)  
                    ↓ is_numeric($score)
```

Esta declaração  
if verifica se  
todos os campos do  
formulário estão  
preenchidos.

Verdadeiro ou falso,  
dependendo de se o  
usuário digitou ou não  
um número no campo  
Pontuação.

Sempre que possível, exija que os dados de formulário estejam no formato que você especificou.



**Exercício**

Reescreva a declaração if de validação do formulário Adicionar Pontuação, de modo que ela use a função `is_numeric()` para permitir apenas pontuações numéricas.

**test drive o novo addscore.php**



### Exercício

### SOLUÇÃO

```
if (!empty($_name) && is_numeric($_score) && !empty($_screenshot)) {
```

```
}
```



## TEST DRIVE

### Aprimore o tratamento dos dados do formulário no script Adicionar Pontuação.

Ajuste a atribuição de dados do formulário às variáveis, no script addscore.php, de modo que as funções trim() e mysqli\_real\_escape\_string() sejam usados para limpar os dados do formulário. Depois, modifique a consulta INSERT para que ela especifique tanto os nomes das colunas quanto os valores, eliminando-se a necessidade de fornecer valores para as colunas id e approved. Modifique também a declaração if que valida os campos do formulário para que ela verifique se a pontuação é numérica.

Finalmente, use uma ferramenta MySQL para rodar a consulta ALTER que estabelece o padrão de 0 para a coluna approved.

Envie o novo script Adicionar Pontuação para o seu servidor, navegue até ele, e depois tente realizar o mesmo ataque de injeção de SQL novamente.

Agora o campo  
Pontuação do  
formulário só  
aceitará números, e  
nada mais.

Guitar Wars - Add Your High Score

Please enter all of the information to add your high score.

Name: Ethel Heckel

Score: 1000000<sup>1</sup>.ethelsscore2.g

Screen shot: Choose File no file selected

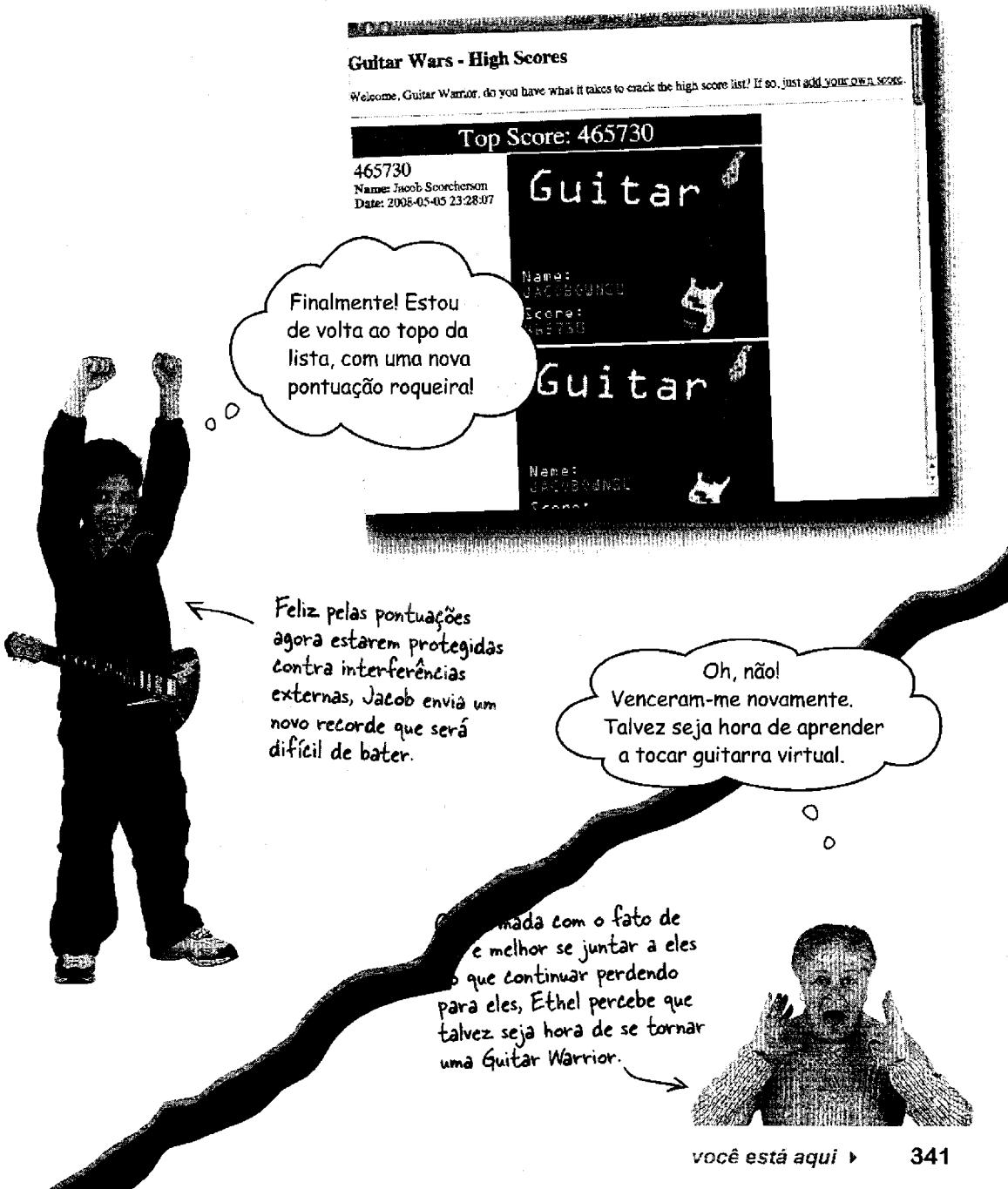
Add

É claro que esta  
mensagem de erro  
poderia ser um pouco mais  
específica, mas ela atende  
às nossas necessidades  
sem precisarmos adicionar  
muito mais programação  
ao script.

A validação de formulários é  
um tópico que vai muito além da  
segurança de bancos de dados. O  
Capítulo 10 revisitará a validação  
com muito mais detalhes...

## Cessar fogo!

Parece que a determinação de Ethel de interferir nas pontuações do Guitar Wars finalmente foi quebrada, graças às melhorias que tornaram a aplicação imune a injecções de SQL. O verdadeiro campeão do Guitar Wars respondeu inserindo um novo recorde.



# Sua Caixa de Ferramentas PHP & MySQL

**Além de aprimorar a aplicação de recordes do Guitar Wars, você adquiriu várias novas ferramentas e técnicas. Vamos rever as mais importantes.**

## header()

Esta função interna do PHP é usada para se enviar um cabeçalho do servidor para o navegador, permitindo que você realize tarefas tais como redirecionar a página, especificar um determinado tipo de conteúdo ou executar autenticação HTTP.

## S\_SERVER

Entre outras coisas, esta superglobal interna do PHP armazena o nome e a senha digitados pelo usuário ao tentar acessar uma página que requer autenticação HTTP. Você pode verificar se esses dados batem com o esperado, para proteger páginas que precisam ser restrinvidas.

## is\_numeric()

Esta função interna do PHP verifica se um determinado valor é um número. É útil para checar se um campo numérico do formulário realmente recebeu um valor numérico.

## trim(), mysqli\_real\_escape\_string()

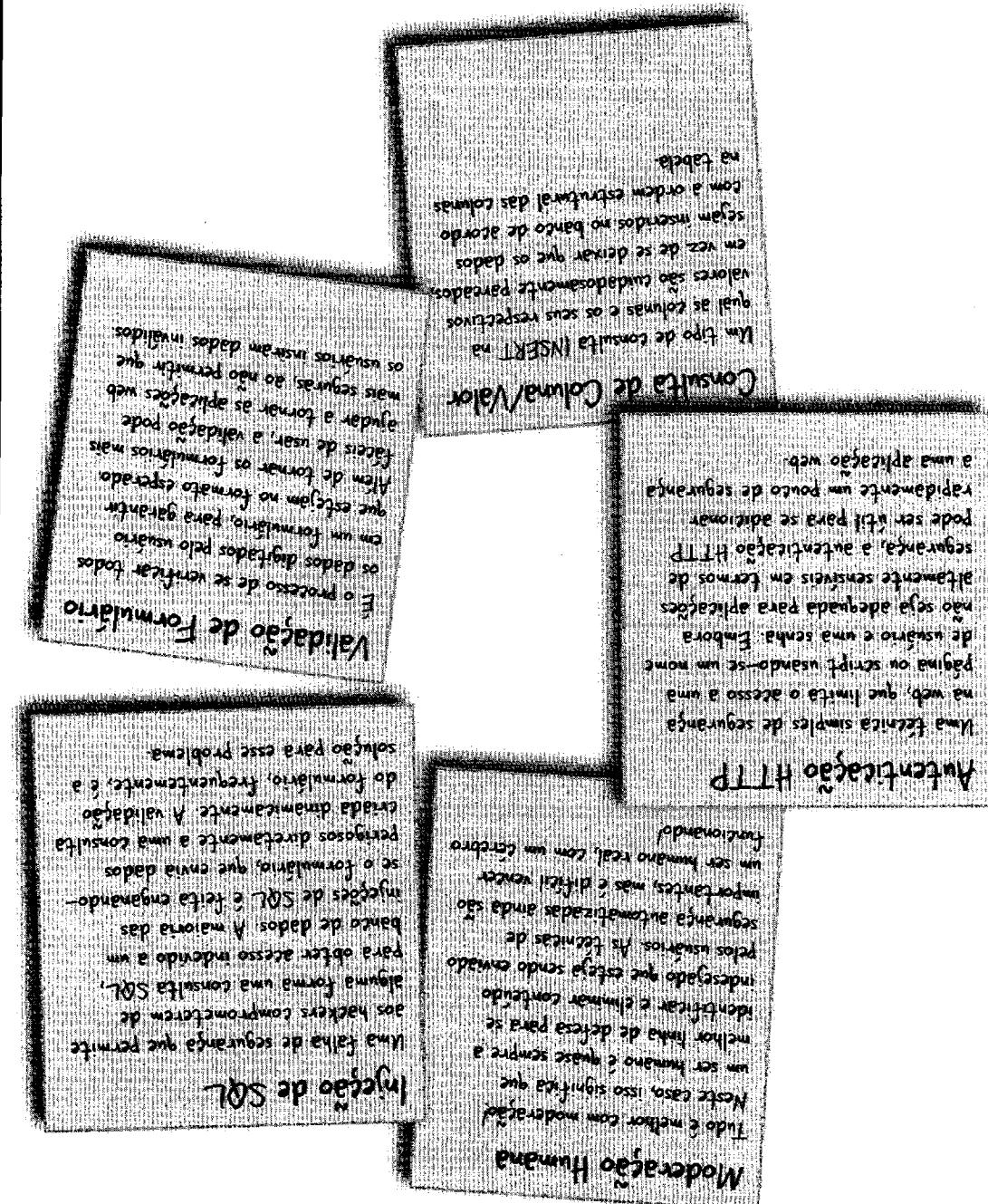
Estas duas funções internas do PHP são úteis para se processar dados de formulários e evitá-las que caracteres problemáticos interfiram em consultas SQL. A primeira função retira espaços em branco, enquanto que a segunda faz escape de caracteres especiais.

## exit()

Esta função interna faz um script PHP se interromper imediatamente. Uma vez que o script encontra a função exit(), nenhum outro código PHP subsequente é executado, e nenhum código HTML subsequente é enviado ao navegador.

## DEFAULT value

Esta instrução SQL estabelece o valor padrão de uma coluna da tabela. Se uma nova linha for adicionada e a coluna não for definida, ela adotará o valor padrão.





## 7 Criando Aplicações Web Personalizadas

# Lembra de Mim?

Desculpe, qual é o seu nome? Johnson, certo. Bem, não estou encontrando nenhum registro seu, Sr. Jackson. Tem certeza que o senhor contratou a garantia estendida para a sua célula de armazenamento criogênico? Ah, entendo, então o senhor está ligando de dentro da célula neste exato momento. Desculpe, qual é o seu nome mesmo?



**Ninguém gosta de ser esquecido, especialmente usuários de aplicações web.** Se a aplicação tiver que trabalhar a noção de "comunidade", ou seja, se for planejado que os usuários interajam com a aplicação de uma forma pessoal, então ela terá de se lembrar dos usuários. Você detestaria ter que se apresentar novamente à sua família a cada vez que entrasse em casa. Não precisa fazê-lo porque seus parentes possuem uma coisa maravilhosa chamada memória. Mas aplicações web não se lembram das pessoas automaticamente – é preciso que um desenvolvedor web esperto use as ferramentas disponíveis (PHP e MySQL, talvez?) para criar aplicações web personalizadas que realmente sejam capazes de se lembrar dos usuários.

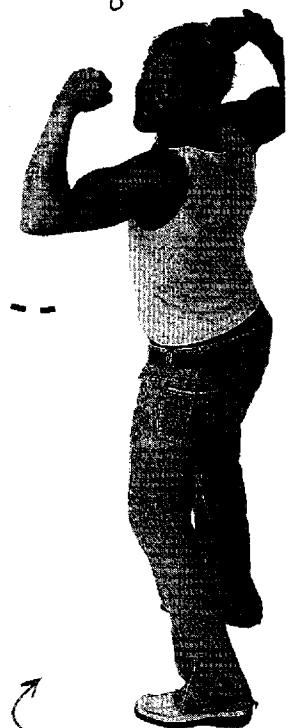
*uma boa incompatibilidade é difícil de encontrar*

## Dizem que os opositos se atraem

É uma história antiga: garoto conhece garota, garota acha que o garoto é completamente insensível, garoto acha que a garota é maluca, mas no fim das contas as diferenças se transformam na atração, e eles acabam vivendo felizes para sempre. Essa história é a inspiração do novo e inovador site de encontros mis-match.net. O Mismatch – "Desencontro" – usa a teoria de que "os opositos se atraem" para combinar encontros entre as pessoas com base nas suas diferenças.

O problema é que o Mismatch ainda precisa de um desenvolvedor web para terminar de construir o sistema. É aí que você entra. Milhões de corações solitários estão esperando ansiosamente que você termine a aplicação... não os decepcione!

Repare  
só nestes  
músculos!



Sidney adora reality shows, yoga e sushi, e espera conseguir um desencontro ideal.

Não vejo a hora de encontrar o meu par imperfeito!

Johan Nettles  
Male (Masculino)  
1981-11-03  
Athens, GA

**As aplicações web pessoais dependem fundamentalmente das informações dos usuários, portanto estes deverão ter a capacidade de acessar a aplicação em um nível pessoal.**



Sidney Kelsow  
Female (feminino)  
1984-07-19  
Tempe, AZ

Johan adora luta livre profissional, levantamento de pesos e Spam, e tem interesse em qualquer uma que responda ao seu anúncio.

Os usuários do Mismatch precisam ser capazes de interagir com o site em um nível pessoal. Isso significa, entre outras coisas, que eles precisarão de perfis pessoais, nos quais inserirão informações sobre si mesmos, as quais poderão compartilhar com outros usuários do Mismatch: sexo, data de nascimento, localização, etc.

## O negócio do Mismatch são os dados pessoais

Portanto, o negócio do Mismatch é estabelecer conexões através de dados pessoais. Essas conexões devem ocorrer dentro de uma,, cada um dos quais devendo ser capaz de interagir com o site e gerenciar os seus próprios dados pessoais. Uma tabela chamada mismatch\_user será usada para guardar as informações dos usuários do Mismatch.

Este é o banco de dados do Mismatch.

Dentro do banco de dados, a tabela mismatch\_user armazena os usuários e as suas informações pessoais.

mismatch_user								
user_id	join_date	first_name	last_name	gender	birthdate	city	state	picture
1	2008-04-17 09:43:11	Sidney	Kelsow	F	1984-07-19	Tempe	AZ	sidneypic.jpg
11	2008-05-23 12:24:06	Johan	Nettles	M	1981-11-03	Athens	GA	johanpic.jpg

Cada linha da tabela mismatch\_user contém dados pessoais referentes a um usuário.

Mismatch - View Profile

Mismatch - Edit Profile

Username: sidneyk

Last name: Kelsow  
Gender: Female  
Birthdate: 1984-07-19  
Location: Tempe, AZ

Picture:

Would you like to edit your profile?

Personal Information

First name: Sidney  
Last name: Kelsow  
Gender: Female  
Birthdate: 1984-07-19  
City: Tempe  
State: AZ  
Picture: Choose File sidneypic.jpg

Save Profile

As páginas Editar e Ver Perfil precisam saber qual perfil devem acessar.

Sidney Kelsow  
Female(Feminino)  
1984-07-19  
Tempe, AZ

Além de visualizar o perfil, os usuários do Mismatch podem editá-lo usando a página Editar Perfil. Mas há um problema: a aplicação necessita saber qual perfil ela deve editar. A página precisa de algum modo de saber qual é o usuário que a está acessando.

PODER DO CÉREBRO

Como o Mismatch poderia personalizar a página Editar Perfil para cada usuário do sistema?

## O Mismatch precisa de logins para os usuários

A solução para o problema de acesso aos dados pessoais no Mismatch envolve o uso de logins, o que significa que os usuários precisam de um meio de fazer login na aplicação. Isso dá ao Mismatch a capacidade de fornecer acesso a informações personalizadas para cada usuário. Por exemplo, o usuário logado só poderia editar o seu próprio perfil, mas poderia também ter acesso de visualização aos perfis dos outros usuários. Os logins são a chave para a personalização no aplicativo Mismatch.

Um login geralmente requer duas informações: um nome de usuário e uma senha.

Os logins permitem que as aplicações web tratem os usuários de forma pessoal.

### Nome do Usuário (Username)

A tarefa do nome do usuário é fornecer uma identificação para cada usuário individual dentro do sistema. Os usuários poderão acessar e potencialmente comunicarem-se uns com os outros através dos seus nomes no sistema.

jnettles sidneyk  
↓  
Os nomes de usuário geralmente consistem de caracteres alfanuméricos, e são escolhidos pelo próprio usuário.

A combinação de nome de usuário e senha permite ao usuário fazer login na aplicação Mismatch e acessar dados pessoais, por exemplo, para editar o seu perfil.

sidneyk  
\*\*\*\*\*  
↓  
O nome e a senha são tudo de que a aplicação precisa para saber quem é o usuário.

Quando um usuário faz login, a aplicação torna-se capaz de lembrar quem é ele, e de fornecer uma experiência personalizada.

### Senha (Password)

A senha é responsável por fornecer um determinado grau de segurança ao login dos usuários, o que ajuda a proteger os seus dados pessoais. Para fazer o login, o usuário precisa digitar tanto o nome quanto a senha.

\*\*\*\*\*

As senhas são dados extremamente sensíveis, e não devem jamais ficar visíveis em uma aplicação, nem mesmo dentro do banco de dados.

A página Editar Perfil agora indica que o usuário está logado.

You are logged in as sidneyk.

### Mismatch - Edit Profile

#### Personal Information

First name: Sidney

Last name: Kelsow

Gender: Female

Birthdate: 1984-07-19

# Elabore um plano para os logins dos usuários

Adicionar o suporte a logins ao Mismatch não é tarefa fácil, e é importante estabelecer exatamente o que precisará ser feito antes de começarmos a escrever código e executar consultas ao banco de dados. Nós sabemos que existe uma tabela a qual armazena os usuários, então a primeira coisa a fazer é modificá-la para que ele armazene as informações de login. Também precisaremos de uma forma de permitir que os usuários digitem os seus dados de login, e isso precisará ser integrado de alguma forma ao restante da aplicação Mismatch, para que páginas como Editar Perfil só fiquem acessíveis depois de um login com sucesso. Eis aqui os passos que estabelecemos até o momento:

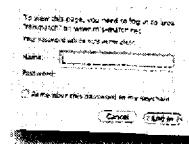
## 1 Usar ALTER para adicionar colunas username e password à tabela.

O banco de dados agora precisa de novas colunas para armazenar os dados de login – nome e senha – de cada usuário.



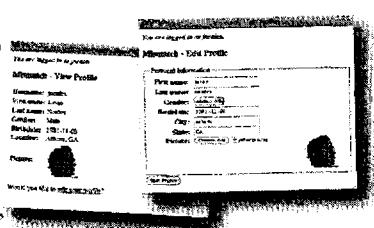
## 2 Criar um novo script de Login que peça ao usuário para digitar seu nome e senha.

O formulário de Login é o que protegerá, em última instância, as páginas pessoais, porque é ele que pede um nome e uma senha válidos. Essas informações precisam ser digitadas corretamente para que Mismatch possa exibir dados específicos do usuário. Assim, o script precisa limitar o acesso a páginas personalizadas, de forma tal que elas não possam ser vistas sem um login válido.



## 3 Conectar o script de Login ao restante da aplicação Mismatch.

As páginas Editar Perfil e Ver Perfil da aplicação Mismatch só devem ficar acessíveis para os usuários logados. Assim, precisamos nos certificar de que os usuários tenham de fazer login, através do script apropriado, antes de receberem acesso a essas páginas.



## *configuração do mismatch*



### **Antes de seguir adiante, tire um momento para testar a aplicação Mismatch e pegar o jeito de como ela funciona.**

Baixe todo o código para a aplicação Mismatch no site da Alta Books, em [www.altabooks.com.br](http://www.altabooks.com.br). Envie todo o código para o seu servidor web, exceto os arquivos .sql, os quais contêm instruções SQL para criar as tabelas necessárias. Certifique-se de executar a instrução de cada arquivo .sql em uma ferramenta MySQL, para que você tenha as tabelas iniciais das quais irá precisar para o Mismatch.

Quando estiver tudo pronto, vá até a página index.php no seu navegador web, e experimente a aplicação. Tenha em mente que as páginas Ver Perfil e Editar Perfil inicialmente não estarão funcionando, uma vez que elas dependem dos logins dos usuários, funcionalidade essa que estamos justamente implementando agora.

Estes dois links levam a partes personalizadas da aplicação.

Mismatch - Where opposites attract!

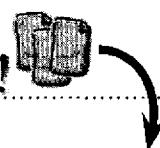
View Profile | Edit Profile

Latest members:

	Johan
	Paul
	Diana
	Jason
	Belita

A página principal do Mismatch lhe permite ver o nome e a foto dos usuários mais recentes, porém nada mais, se você não estiver logado.

Faça o Download!



O código-fonte completo da aplicação Mismatch está disponível para download no site da Alta Books:

[www.altabooks.com.br](http://www.altabooks.com.br)

# Preparando o banco de dados para os logins

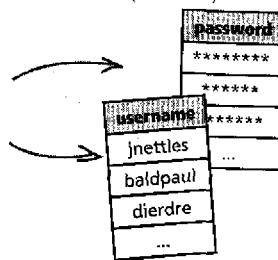
OK, de volta à construção. A tabela mismatch\_user já está fazendo um bom trabalho de armazenar informações do perfil de cada usuário, mas ainda fica a dever no que se refere às informações de login. Mais especificamente, faltam na tabela colunas para se armazenar o nome e a senha de cada usuário.

A tabela mismatch\_user precisa de colunas para armazenar o nome e a senha de login dos usuários.

mismatch_user									
user_id	join_date	first_name	last_name	gender	birthdate	city	state	picture	

Tantos os dados dos nomes quanto os das senhas consistem de texto puro, portanto, é possível usar o familiar tipo de dados VARCHAR do MySQL para ambas as colunas. Porém, ao contrário de alguns outros dados dos usuários, você não pode permitir jamais que o nome ou a senha fiquem vazios (NULL).

As colunas username e password contêm dados de texto simples, mas nunca devem ficar vazias.



não existem

## Perguntas Idiotas

P: Por que eu não posso simplesmente usar user\_id, em vez de username, para identificar os usuários?

R: Você até pode, se quiser. De fato, o propósito de user\_id é justamente fornecer um modo eficiente de se identificar individualmente as linhas referentes a cada usuário. Porém, IDs numéricas tendem a ser difíceis de memorizar, e os usuários realmente preferem escolher os seus próprios nomes ao se cadastrarem em aplicações web personalizadas. Assim, trata-se mais de uma decisão de usabilidade permitir que Johan faça login como "jnettles" em vez de "11". Ninguém quer ser diminuído a um mero número!



## Aponte seu lápis

Poucas pessoas gostariam de ter de lembrar uma senha com mais de 16 caracteres!

Termine de escrever uma instrução SQL que adicione as colunas username e password à tabela posicionada conforme abaixo, sendo que username precisa ser capaz de armazenar 32, password 16 caracteres e nenhuma delas deve permitir dados NULL.

mismatch\_user

user_id	username	password	join_date	first_name	last_name	gender	birthdate	city	state	picture

aponte seu lápis solução



## Aponte seu lápis Solução

Termine de escrever uma instrução SQL que adicione as colunas username e password à tabela posicionada conforme abaixo, sendo que username precisa ser capaz de armazenar 32, password 16 caracteres e nenhuma delas deve permitir dados NULL.

ALTER TABLE é usada para adicionar novas colunas a uma tabela existente.

→ ALTER TABLE mismatch\_user ADD username VARCHAR(32) NOT NULL AFTER user\_id, ADD password VARCHAR(16) NOT NULL AFTER username.

A instrução AFTER controla, na tabela, onde as novas colunas serão adicionadas.

A coluna username é adicionada primeiro, portanto, você pode referenciá-la aqui.

mismatch\_user

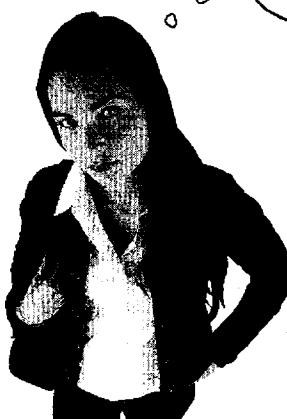
user_id	username	password	join_date	first_name	last_name	gender	birthdate	city	state	picture

A posição das colunas em uma tabela não necessariamente faz diferença, embora possa ser mais adequado, do ponto de vista organizacional, posicionar as colunas mais importantes primeiro.

FEITO

1 Usar ALTER para adicionar colunas username e password à tabela.

É claro que você não vai simplesmente armazenar a senha no banco do jeito que ela está... não é preciso criptografar as senhas antes de armazená-las?



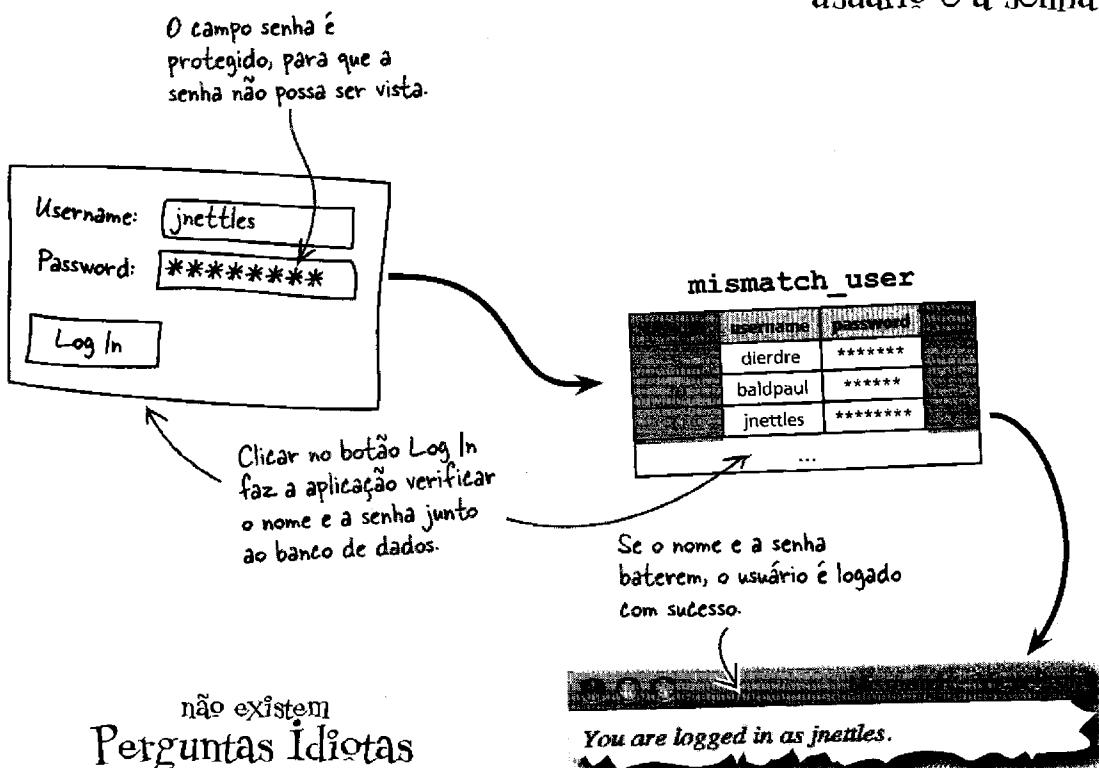
**Boa observação... as senhas precisam ser criptografadas.**

A criptografia no Mismatch implica converter a senha em um formato irreconhecível, ao ser armazenada no banco de dados. Qualquer aplicação com suporte a login dos usuários precisa criptografar as senhas, para que os usuários possam ter a certeza de que as suas senhas estão sãs e salvas. Não é aceitável expor as senhas nem mesmo dentro do banco de dados. Assim, nós precisamos de uma forma de criptografar as senhas antes de inseri-las na tabela mismatch\_user. Porém, antes disso, a criptografia não nos será muito útil se não tivermos uma forma

# Construindo uma interface de usuário para o login

Com o banco de dados alterado para armazenar os dados de login dos usuários, ainda precisamos de uma forma que permita que eles digitem os dados e realmente façam login na aplicação. Esta interface de login precisa consistir de campos de edição de texto para o nome e para a senha, bem como de um botão para executar o processamento do login.

Uma operação de log-in requer uma interface para o nome de usuário e a senha



## não existem Perguntas Idiotas

P: O que fica realmente armazenado no banco de dados não são asteriscos, certo?

R: Certo. Os asteriscos mostrados em um campo de senha simplesmente servem para fornecer segurança visual, impedindo que alguém olhe por sobre os seus ombros enquanto você digita a senha. Quando o formulário é submetido, é a própria senha que é enviada, e não os asteriscos. É por isso que é importante a senha ser criptografada antes de ser inserida no banco.



Caso esteja preocupado sobre como os usuários poderão fazer login, se ainda não atribuímos nomes e senhas a eles... relaxe.

Nós iremos criar os nomes e as senhas em um instante. Por enquanto, é importante elaborar os alicerces do login, mesmo que ainda tenhamos de realizar mais tarefas para que tudo funcione a contento.

## a função sha()

# Criptografe as senhas com SHA()

A interface de login é bem simples, mas ainda não nos detivemos na necessidade de criptografar a senha. O MySQL oferece uma função chamada SHA() que aplica um algoritmo criptográfico a uma string de texto. O resultado é uma string criptografada que possui exatamente 40 caracteres hexadecimais, independentemente do tamanho da senha original. Assim, a função na verdade gera um código de 40 caracteres que representa individualmente a senha.

Uma vez que SHA() é uma função do MySQL, e não uma função PHP, você pode chamá-la como parte de uma consulta que insira uma senha na tabela. Por exemplo, este código insere um novo usuário na tabela mismatch\_user, criptografando a senha com SHA() durante a operação.

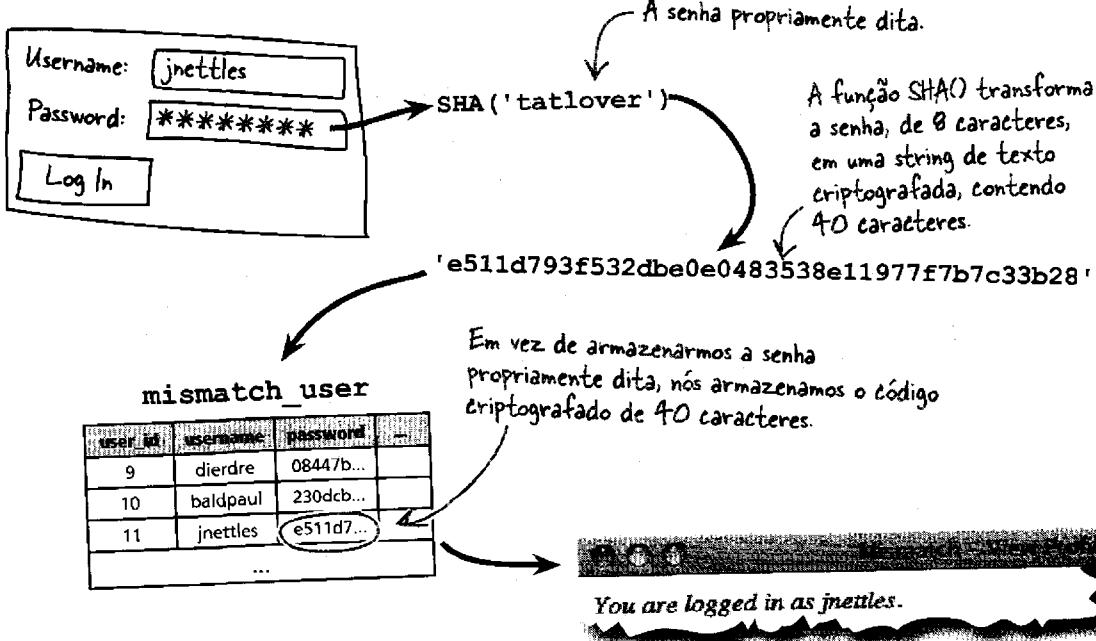
```
INSERT INTO mismatch_user  
    (username, password, join_date) VALUES ('jnettles', SHA('tatlover'), NOW())
```

A função SHA() criptografa a senha na forma de um código de 40 caracteres hexadecimais, o qual é armazenado na coluna password da tabela mismatch\_user.

A mesma função SHA() trabalha do outro lado da equação do login, verificando se a senha digitada pelo usuário bate com aquela que foi armazenada criptografada no banco de dados.

A função SHA() do MySQL criptografa um determinado texto, transformando-o em um código de 40 caracteres.

↑  
Esta é a senha real, na forma como foi digitada no respectivo campo do formulário.



## Descriptografando Comparando senhas

Uma vez que você tenha criptografado uma determinada informação, o instinto natural é pensar que ela deverá ser descriptografada em algum ponto. Mas a função SHA() é de mão única – ela apenas criptografa, sem poder voltar atrás. Isso foi feito de caso pensado, para preservar a segurança dos dados criptografados – mesmo que um hacker invada o seu banco de dados e roube todas as senhas, ele não conseguirá decodificá-las. Assim, como é possível fazer login do usuário se você não pode decodificar a sua senha?

Você não precisa saber a senha original do usuário para saber se ele a digitou corretamente na tela de login. Isso é porque SHA() gera o mesmo código de 40 caracteres, desde que você forneça a ela a mesma string de texto. Assim, você pode simplesmente criptografar a senha digitada pelo usuário e comparar o código resultante com o valor presente na coluna password da tabela mismatch\_user. Isso pode ser feito com um consulta SQL, que tenta selecionar a linha referente a um usuário com base na senha.

Esta é a senha digitada pelo usuário para fazer o login.

```
SELECT * FROM mismatch_user
WHERE password = SHA('tatlover')
```

A função SHA() é chamada para criptografar a senha, de modo que ela pode aparecer na cláusula WHERE.

Esta consulta SELECT seleciona todas as linhas da tabela mismatch\_user cuja coluna password bata com a senha digitada, 'tatlover' neste caso. Uma vez que estamos comparando versões criptografadas da senha, não é necessário saber a senha original. Uma consulta para realmente fazer o login do usuário usa SHA(), mas ela precisaria também fazer SELECT na ID do usuário, como veremos em instantes.

## Abrindo espaço para a senha criptografada

A função SHA() apresenta um problema para o Mismatch, uma vez que as senhas criptografadas acabam tendo 40 caracteres de extensão, mas a nossa recém-criada coluna password só armazena até 16 caracteres. Precisamos de um ALTER para ampliar a coluna password, de modo que ela possa armazenar as senhas criptografadas.

```
ALTER TABLE mismatch_user
CHANGE password password VARCHAR(40) NOT NULL
```

O tamanho da coluna password é modificado para 40, para caberem as senhas criptografadas.

A função SHA() fornece apenas a operação criptografar – não é possível decodificar dados que tenham sido criptografados.

não existem

## Perguntas Idiotas

**Q:** O que significa o acrônimo SHA()?

**A:** Significa Secure Hash Algorithm, ou "Algoritmo de Hash Seguro". O "hash" é um termo usado em programação de computadores, que se refere a uma string única, de extensão fixa, a qual representa individualmente uma string de texto. No caso de SHA(), o hash é a string de texto criptografada, com 40 caracteres hexadecimais, a qual representa de forma única a senha original.

**Q:** Existem outras formas de se criptografar senhas?

**A:** Sim. O MySQL oferece uma outra função semelhante a SHA() chamada MD5(), que executa um tipo parecido de criptografia. Mas o algoritmo SHA() é considerado um pouco mais seguro do que MD5(), portanto é melhor usar SHA(). O PHP também oferece funções equivalentes (sha1() e md5()), para o caso de você precisar fazer alguma criptografia em código PHP, em vez de dentro de uma consulta SQL.

*modificando mismatch\_user*



## TEST DRIVE

### **Adicione as colunas username e password à tabela mismatch\_user, e depois teste-as.**

Usando alguma ferramenta MySQL, execute a instrução ALTER para adicionar as colunas username e password à tabela mismatch\_user.

```
ALTER TABLE mismatch_user ADD username VARCHAR(32) NOT NULL AFTER user_id,  
ADD password VARCHAR(16) NOT NULL AFTER username
```

Mas a nossa coluna password precisa ser capaz de conter uma string de 40 caracteres, portanto emita ALTER mais uma vez para abrir um espaço maior para os dados.

```
ALTER TABLE mismatch_user  
CHANGE password password VARCHAR(40) NOT NULL
```

Agora, para testar as novas colunas, vamos fazer um INSERT de um novo usuário.

```
INSERT INTO mismatch_user  
(username, password, join_date) VALUES ('jimi', SHA('heyjoe'), NOW())
```

*Não se esqueça de criptografar a senha, chamando SHA().*

Para certificar-se de que a senha foi de fato criptografada no banco de dados, dê uma olhada nela, executando um SELECT no novo usuário.

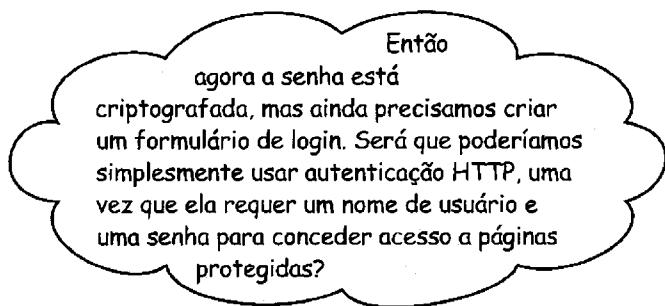
```
SELECT password FROM mismatch_user WHERE username = 'jimi'
```

E, finalmente, você pode simular uma verificação de login fazendo um SELECT no nome do usuário e usando a função SHA() com a senha em uma cláusula WHERE.

*Par um login bem-sucedido, esta deve ser a mesma senha usada quando se insere a linha.*

```
SELECT username FROM mismatch_user WHERE password = SHA('heyjoe')
```

```
File Edit Window Help OppositesAttract  
mysql> SELECT username FROM mismatch_user WHERE password = SHA('heyjoe');  
+-----+  
| username |  
+-----+  
| jimi |  
+-----+  
1 row in set (0.000 sec)  
  
Sómente um usuário tem  
a senha que bate com o  
código criptografado.
```



### **Sim! A autenticação HTTP certamente pode ser usada como um sistema simplificado de login dos usuários.**

Se você se lembrar da aplicação Guitar Wars do capítulo anterior, a autenticação HTTP foi usada para se restringir o acesso a determinadas partes da aplicação, pedindo um nome e uma senha ao usuário. De modo geral, essa é a mesma funcionalidade requerida pelo Mismatch, exceto pelo fato de que agora nós temos um banco de dados inteiro contendo combinações possíveis de nome de usuário/senha, em vez de apenas um nome e uma senha. Os usuários do Mismatch poderiam usar a mesma janela de autenticação HTTP; porém, cada usuário irá digitar o seu próprio nome e a sua própria senha.



To view this page, you need to log in to area "Mismatch" on www.mis-match.net  
Your password will be sent in the clear.

Name:

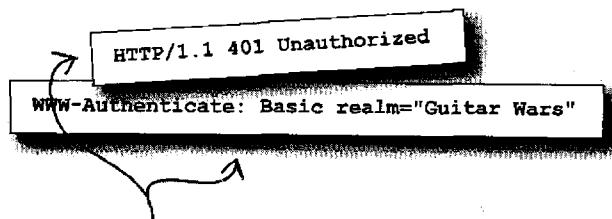
Password:

Remember this password in my keychain

A janela de autenticação HTTP padrão, que tem aparência diferente conforme o navegador, pode servir como uma interface de login simplificada.

## Autorizando usuários com HTTP

Como foi mostrado no Guitar Wars, é preciso que sejam enviados dois cabeçalhos para se restringir o acesso a uma determinada página através da janela de autenticação HTTP. Esses cabeçalhos têm o efeito de pedir que o usuário digite um nome e uma senha para obter acesso à página Admin do Guitar Wars.



Estes dois cabeçalhos precisam ser enviados para que o acesso a uma página seja restrito mediante autenticação HTTP.

O envio dos cabeçalhos para autenticação HTTP resulta em duas linhas de código PHP – uma chamada à função header() para cada cabeçalho sendo enviado.

```
header('HTTP/1.1 401 Unauthorized');  
header('WWW-Authenticate: Basic realm="Mismatch"');
```

Este é o realm da autenticação, que tem efeito sobre toda a aplicação.

A autenticação HTTP requer que nós enviamos dois cabeçalhos.

É necessário digitar um nome e uma senha para se acessar páginas restritas na aplicação Guitar Wars.

O usuário só poderá ver e usar esta página se fornecer o nome e a senha corretos.

The screenshot shows a web browser window with a login form. At the top, a message says: "To view this page, you need to log in to the area 'Guitar Wars' on www.guitarwars.net". The form has fields for "Name" (containing "rock") and "Password" (containing "\*\*\*\*\*"). A note below the fields says: "Your password will be sent in the clear". There is a checkbox for "Remember this password in my keychain". Below the form, a list of names is shown with their last login times and "Remove" links: Ashton Sim 2008-04-24 08:02:11, Eddie Vanil, Bella Chev, Phiz Lairst, Paolo Jastorni 2008-04-24 08:02:11, Nevil Johansson 2008-04-24 08:02:11, Kenny Lavitz 2008-04-23 14:09:50. Each entry has a "Remove" link next to it.



**Exercício**  
Eis o script de login.

Circule as diferentes partes da aplicação Mismatch que sejam impactadas pelo script de login (`login.php`) e pelo seu uso da autenticação HTTP para controlar o acesso. Em seguida, escreva abaixo de que modo essas partes são afetadas.

login.php

viewprofile.php

index.php

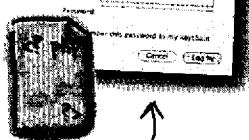
editprofile.php

mismatch_user	

## exercício solução

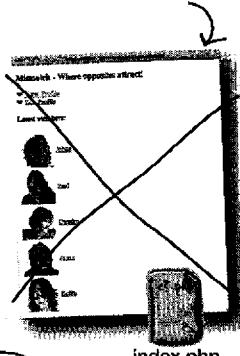


### Exercício Solução

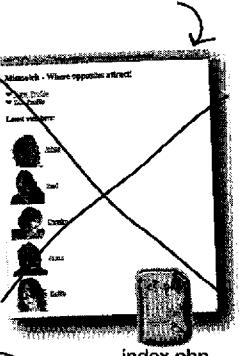


login.php

Quando um usuário faz login, o seu nome e a sua senha são verificados em relação ao conteúdo do banco de dados, para garantir que pertençam a um usuário registrado.



mismatch\_user



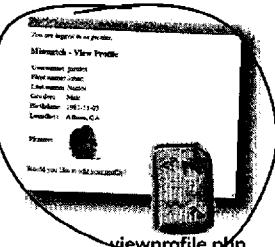
index.php

Se não for encontrada uma linha cujos dados batam com o nome e a senha fornecidos, o script de login exibe uma mensagem de erro e impede o acesso.

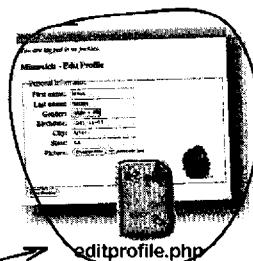
Circule as diferentes partes da aplicação Mismatch que sejam impactadas pelo script de login (login.php) e pelo seu uso da autenticação HTTP para controlar o acesso. Em seguida, escreva abaixo de que modo essas partes são afetadas.

A home page não tem nenhum papel direto nos logins dos usuários, porque ela precisa ficar acessível a todos.

A visualização e a edição dos perfis ficam restritas, significando que apenas os usuários que fizerem login poderão acessar essas páginas.



viewprofile.php



editprofile.php

A página Editar Perfil não só utiliza o script de login para restringir o acesso, como também precisa do nome do usuário para determinar qual é o perfil que deverá ser editado.

## não existem Perguntas Ídiotas

**P:** Por que não é necessário incluir a home page na exigência de login dos usuários?

**R:** Porque a home page é o primeiro lugar que o usuário acessa ao visitar o site, e é importante que os visitantes possam olhar o site antes de exigir deles um login. Assim, a home page serve ao mesmo tempo como uma propaganda e um ponto de partida – propaganda para os visitantes e ponto de partida para os usuários registrados, os quais precisam fazer login para acessarem qualquer ponto mais profundo da aplicação.

**P:** Os usuários logados podem ver o perfil de qualquer outro usuário?

**R:** Sim. A ideia é que os perfis fiquem visíveis para todos os usuários que fizerem login, mas permaneçam invisíveis aos visitantes não registrados. Em outras palavras, você precisa ser um membro do Mismatch para ver o perfil de outro usuário.

**P:** Como a criptografia da senha afeta a autenticação HTTP?

**R:** Existem duas questões em jogo aqui: transmitir a senha e armazenar a senha. A função SHA() do MySQL concentra-se em armazenar uma senha, de forma segura e criptografada, no banco de dados. O banco não está preocupado com a forma como você transmitem a senha originalmente, de modo que este tipo de criptografia não tem impacto nenhum sobre a autenticação HTTP. Porém, poder-se-ia argumentar que a criptografia deveria também ser realizada durante a transmissão da senha, quando a janela de autenticação HTTP a submete ao servidor. Esse tipo de criptografia está fora do escopo deste capítulo e, em todo caso, só é necessária quando se está lidando com dados extremamente sensíveis.

# Fazendo login dos usuários com autenticação HTTP

O script de login (login.php) é responsável por pedir um nome e uma senha do usuário, usando cabeçalhos de autenticação HTTP, por obter os respectivos valores de nome e senha da superglobal `$_SERVER`, e por verificar se eles batem com os dados contidos em `mismatch_user` antes de conceder acesso a uma página restrita.

```
<?php
    require_once('connectvars.php');

    if (!isset($_SERVER['PHP_AUTH_USER']) || !isset($_SERVER['PHP_AUTH_PW'])) {
        // O nome/senha não foram digitados, portanto, enviar os cabeçalhos de autenticação headers
        header('HTTP/1.1 401 Unauthorized');
        header('WWW-Authenticate: Basic realm="Mismatch"');
        exit('<h3>Mismatch</h3>Desculpe, você deve digitar seu nome e senha para fazer login e acessar esta página.');
    }

    // Conecta-se ao banco de dados
    $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

    // Obtém os dados de login digitados pelo usuário
    $user_username = mysqli_real_escape_string($dbc, trim($_SERVER['PHP_AUTH_USER']));
    $user_password = mysqli_real_escape_string($dbc, trim($_SERVER['PHP_AUTH_PW']));

    // Procura o nome e a senha no banco de dados
    $query = "SELECT user_id, username FROM mismatch WHERE username = '$user_username' AND password = SHA1('$user_password')");
    $data = mysqli_query($dbc, $query);

    if (mysqli_num_rows($data) == 1) {
        // O login foi bem-sucedido, portanto, definir as variáveis de ID e nome do usuário
        $row = mysqli_fetch_array($data);
        $user_id = $row['user_id'];
        $username = $row['username'];
    } else {
        // O nome/senha estão incorretos, portanto, enviar os cabeçalhos de autenticação
        headers('HTTP/1.1 401 Unauthorized');
        header('WWW-Authenticate: Basic realm="Mismatch"');
        exit('<h2>Mismatch</h2>Desculpe, você deve digitar seu nome e senha para fazer login e acessar esta página.');
    }

    // Confirma o login bem-sucedido.
    echo(<p class="logado">Você está logado como <b>$username</b>!</p>);
?>
```

**Se o nome e a senha não tiverem sido digitados, envie os cabeçalhos de autenticação para pedir que o usuário digite esses dados.**

**Obtém o nome e a senha digitados pelo usuário.**

**Realiza uma consulta para ver se existe alguma linha cujos dados de nome e senha criptografados batam com aqueles digitados.**

**Se for encontrada uma linha, significa que o login foi bem-sucedido e podemos definir as variáveis `$user_id` e `$username`.**

**Chegando a este ponto, está tudo bem, portanto confirme o login bem-sucedido.**

**FEITO**  
Criar um novo script de Login que peça ao usuário para digitar seu nome e senha.



**test drive mismatch**



## TEST DRIVE

### Crie o novo script de login, e o inclua nos scripts Ver Perfil e Editar Perfil.

Crie um arquivo de texto chamado login.php e digite nele o código para o script de login (ou baixe o script no site da Alta Books, [www.altabooks.com.br](http://www.altabooks.com.br)). Em seguida, adicione algum código PHP no início dos scripts viewprofile.php e editprofile.php para incluir o novo script de login.

Envie todos os scripts ao seu servidor web, e em seguida, abra a página principal do Mismatch em um navegador. Clique no link Ver Perfil ou Editar Perfil para fazer login e acessar as páginas personalizadas. É claro que isso só funcionará se você já tiver adicionado um usuário, com respectivos nome e senha, ao banco de dados.

Estes dois links levam às páginas protegidas, as quais chamam o script de login caso o usuário não esteja logado.

Mismatch - Where opposites attract!

View Profile  
Edit Profile

Latest members:

- Johan
- Paul
- Djendre
- Jason
- Brilita

Esta senha é criptografada com SHA() e comparada com aquela presente no banco de dados, para determinar se o login deve ser permitido.

To view this page, you need to log in to area "Mismatch" on [www.mis-mismatch.net](http://www.mis-mismatch.net). Your password will be sent in the clear.

Name: jnetties  
Password:   Remember this password in my keychain

O script de login usa autenticação HTTP para impedir acesso não autorizado às páginas Ver Perfil e Editar Perfil.

A home page não é protegida pelo script de login, mas serve de ponto de partida para se navegar até níveis mais profundos da aplicação.

Qualquer página do Mismatch que precisar exigir login só precisará incluir o script login.php no início do seu código.

You are logged in as jnetles.

### Mismatch - View Profile

Username: jnetles  
First name: Johan  
Last name: Nettles  
Gender: Male  
Birthdate: 1981-11-03  
Location: Athens, GA

Picture:



Would you like to [edit your profile?](#)

Ambas as páginas reconhecem o login com uma confirmação que é fornecida pelo script de login.

Se o nome e a senha forem válidos, o usuário é logado, e o restante da página pode ser enviado.

```
<?php require_once('login.php');  
?  
<html>  
<head>  
<title>Mismatch - View Profile</title>  
<link rel="stylesheet" type="text/css" href="style.css" />  
</head>  
<body><h3>Mismatch - View Profile</h3>
```

DB\_USER, DB\_PASSWORD, DB\_NAME);  
de database

viewprofile.php

O script de login é a primeira coisa a ser incluída nos scripts. Ver Perfil e Editar Perfil, para proteger as respectivas páginas.

```
<?php require_once('login.php');  
?  
<html>  
<head>  
<title>Mismatch - Edit Profile</title>  
<link rel="stylesheet" type="text/css" href="style.css" />  
</head>
```

### Mismatch - Edit Profile

Personal Information	
First name:	<input type="text" value="johan"/>
Last name:	<input type="text" value="Nettles"/>
Gender:	<input checked="" type="radio"/>
Birthdate:	<input type="text" value="1981-11-03"/>
City:	<input type="text" value="Athens"/>
State:	<input type="text" value="GA"/>
Picture:	<input type="button" value="Choose File"/> 

[Save Profile](#)

DB\_USER, DB\_PASSWORD, DB\_NAME);

editprofile.php

Cada usuário recebe a sua própria experiência personalizada do Mismatch.

3

**FEITO**  
Conectar o script de Login ao restante da aplicação Mismatch.

*mismatch precisa de um formulário de cadastro*

Ruby adora filmes de terror, resolver quebra-cabeças e comida apimentada, mas neste exato momento, ela odeia o Mismatch por não permitir que se cadastre e use o sistema.



Eu adoraria fazer login e começar a editar meu perfil, mas não consigo descobrir como me cadastrar.

**Os usuários novos do Mismatch precisam de uma forma de se cadastrarem.**

O novo script de login faz um bom trabalho ao usar a autenticação HTTP para permitir que os usuários façam login. O problema é que os usuários não têm nenhuma forma de se cadastrarem – fazer login é difícil quando você ainda nem sequer tem um nome e uma senha. O Mismatch precisa de um formulário de cadastramento, o qual permita que novos usuários se juntem ao site, criando um nome e uma senha para eles.

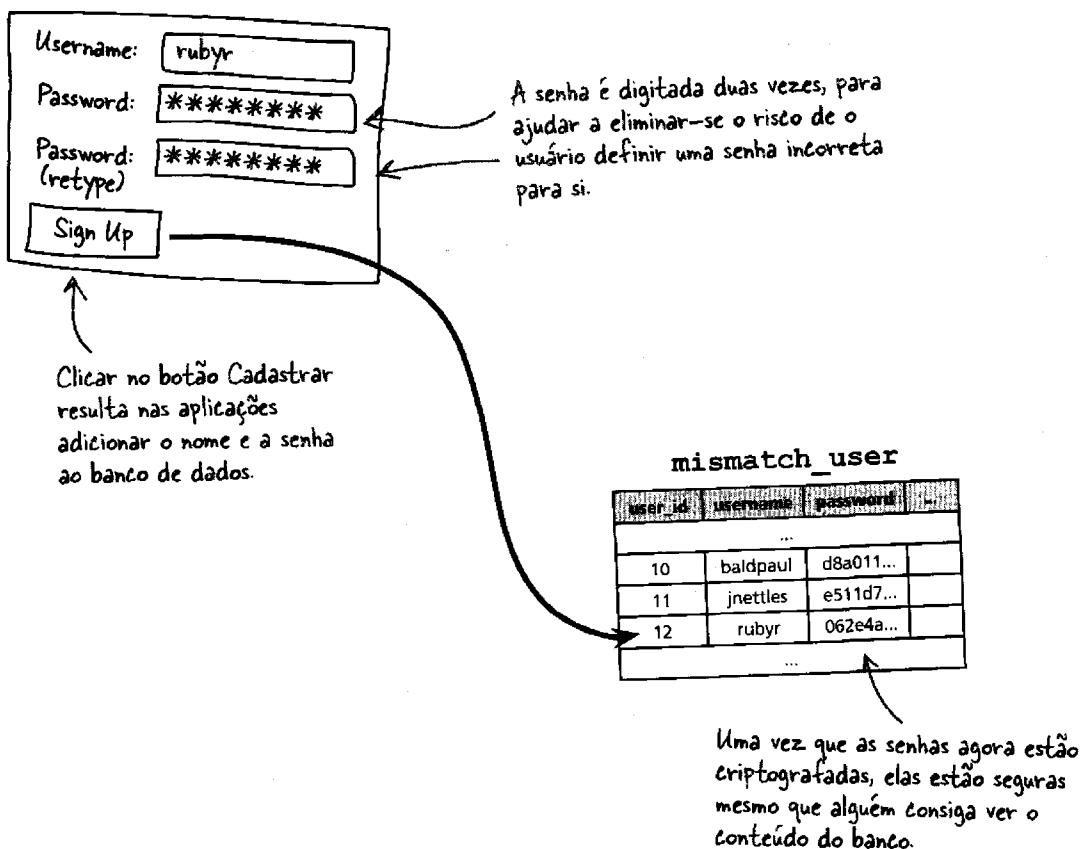
**Nome de Usuário?**

**Senha?**

# Um formulário para novos usuários se cadastrarem

Como este novo formulário de Cadastro deve ser? Nós sabemos que ele precisa permitir que o usuário digite o nome e a senha desejados... mais alguma coisa? Uma vez que o usuário está estabelecendo a sua senha com o formulário e as senhas em formulários web geralmente são escondidas com asteriscos por motivo de segurança, é uma boa ideia ter dois campos para a senha. Dessa forma, o usuário digita a senha duas vezes, apenas para nos certificarmos de que não houve nenhum erro de digitação.

Assim, a tarefa da página de cadastramento é obter o nome e a senha do usuário, certificar-se de que o nome ainda não esteja sendo usado por ninguém, e finalmente adicionar o novo usuário ao banco mismatch\_user.



Um problema potencial com o script de cadastro é o usuário tentar se inscrever com um nome que já existe. O script precisa ser esperto o suficiente para capturar esse problema e fazer o usuário digitar um nome diferente. Assim, a tarefa da página de cadastro é obter o nome e a senha do usuário, certificar-se de que o nome ainda não esteja sendo usado por ninguém, e finalmente, adicionar o novo usuário ao banco mismatch\_user.



## signup.php finalizando



# Ímas de Geladeira do PHP & MySQL

O script de cadastro do MySQL usa um formulário personalizado para pedir ao usuário que escolha o seu nome e senha desejados. O problema é que o script encontra-se incompleto. Use os ímas abaixo para finalizá-lo, de modo que novos usuários possam se cadastrar e se juntar à comunidade Mismatch.

Eis o formulário de cadastro.

### Mismatch - Sign Up

Please enter your username and desired password to sign up to Mismatch.

#### Registration Info

Username:	rubyr
Password:	.....
Password (retype):	.....

Sign Up

```
<?php
require_once('appvars.php');
require_once('connectvars.php');

// Conecta-se ao banco de dados
$dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

if (isset($_POST['submit'])) {
    // Obtém os dados do perfil a partir de POST

        ..... = mysqli_real_escape_string($dbc, trim($_POST['.....']));
        ..... = mysqli_real_escape_string($dbc, trim($_POST['.....']));
        ..... = mysqli_real_escape_string($dbc, trim($_POST['.....']));

    if (!empty($username) && !empty($password1) && !empty($password2) &&
        (..... == .....)) {

        // Certifica-se de que ninguém já tenha se registrado com o mesmo nome

        $query = "SELECT * FROM mismatch_user WHERE username = '.....'";
        $data = mysqli_query($dbc, $query);
        if (mysqli_num_rows($data) == 0) {
            // O nome do usuário é único, inserir os dados no banco
            $query = "INSERT INTO mismatch_user (username, password, join_date) VALUES ";
            " ('.....', SHA('.....'), NOW())";
            mysqli_query($dbc, $query);

        // Confirme o sucesso com o usuário
        echo '<p>A sua conta foi criada com sucesso. Agora você pode fazer login e '. 
        '<a href="editprofile.php">editar seu perfil</a>.</p>';

        mysqli_close($dbc);
        exit();
    }
}
```

Não se esqueça que você tem de fazer escape de qualquer apóstrofo que apareça dentro de aspas simples.

```

else {
    // Já existe uma conta com este nome, exibir mensagem de erro.
    echo '<p class="error">Já existe uma conta com este nome. Por favor, escolha outro. ' .
        'nome.</p>';
    ..... = "";
}

}
else {
    echo '<p class="error">Você deve digitar todos os dados de login, incluindo a senha ' .
        'desejada duas vezes.</p>';
}
}

mysqli_close($dbc);
?>

<p>Por favor, digite seu nome de usuário e senha desejados para se cadastrar no Mismatch.</p>
<form method="post" action=<?php echo $_SERVER['PHP_SELF']; ?>>>
<fieldset>
    <legend>Informações de Registro</legend>
    <label for="username">Nome de Usuário:</label>

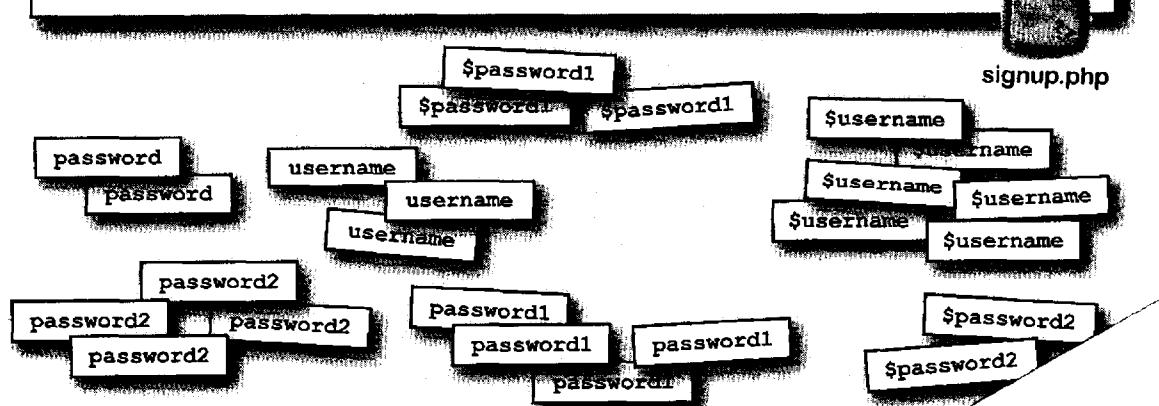
    <input type="text" id="....." name="....."
           value=<?php if (!empty(.....)) echo .....; ?>><br />

    <label for=".....">Senha:</label>

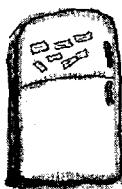
    <input type="....." id="....." name="....." /><br />
    <label for=".....">Senha (digite novamente):</label>
    <input type="....." id="....." name="....." /><br />

</fieldset>
<input type="submit" value="Cadastrar" name="submit" />
</form>

```



*signup.php completo*



# Ímas de Geladeira do PHP & MySQL - Solução

O script de cadastro do MySQL usa um formulário personalizado para pedir ao usuário que escolha o seu nome e senha desejados. O problema é que o script encontra-se incompleto. Use os ímas abaixo para finalizá-lo, de modo que novos usuários possam se cadastrar e se juntar à comunidade Mismatch.

Eis o formulário de cadastro.

```

<?php
require_once('appvars.php');
require_once('connectvars.php');

// Conecte-se ao banco de dados
$dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

if (isset($_POST['submit'])) {
    // Obtemos dados do perfil a partir de POST

    $username = mysqli_real_escape_string($dbc, trim($_POST['username']));
    $password1 = mysqli_real_escape_string($dbc, trim($_POST['password1']));
    $password2 = mysqli_real_escape_string($dbc, trim($_POST['password2']));

    if (!empty($username) && !empty($password1) && !empty($password2) &&
        ($password1 == $password2)) { ← Certifique-se de que nenhum dos campos do formulário esteja vazio e que as duas senhas digitadas sejam idênticas.

        // Certifique-se que ninguém já tenha se registrado com o mesmo nome

        $query = "SELECT * FROM mismatch_user WHERE username = '$username'"; ← Realize uma consulta para ver se existem linhas batendo com o nome de usuário digitado.

        $data = mysqli_query($dbc, $query);
        if (mysqli_num_rows($data) == 0) {
            // O nome do usuário é único, inserir os dados no banco
            $query = "INSERT INTO mismatch_user (username, password, join_date) VALUES ";
            $query .= "('$username', SHA('$password1'), NOW())"; ← Se não houver nenhuma linha, então o usuário é novo, então podemos fazer o INSERT.

            mysqli_query($dbc, $query); ← Qualquer uma das senhas poderia ser usada aqui, uma vez que elas devem ser idênticas para que chegemos a este ponto.

            // Confirma o sucesso com o usuário
            echo '<p>A sua conta foi criada com sucesso. Agora você pode fazer login e <a href="editprofile.php">editar seu perfil/a>.</p>';

            mysqli_close($dbc);
            exit();
        }
    }
}

```

Mismatch - Sign Up

Please enter your username and desired password to sign up to Mismatch.

Registration Info

Username:	abv
Password:	*****
Retype:	*****

```

else {
    // Já existe uma conta com este nome, exibir mensagem de erro
    echo '<p class="error">Já existe uma conta com este nome. Por favor, escolha outro'.
    nome</p>';
}

$username = "";
} else {
    echo '<p class="error">Você deve digitar todos os dados de login, incluindo a senha ' .
        'desejada duas vezes.</p>';
}
}

mysqli_close($dbc);
?>

<p>Por favor, digite seu usuário e senha desejados para se cadastrar no Mismatch.</p>
<form method="post" action=<?php echo $_SERVER['PHP_SELF']; ?>>>
<fieldset>
    <legend>Informações de Registro</legend>
    <label for="username">Nome de usuário:</label>
    <input type="text" id="username" name="username" value=<?php if (!empty($_POST['username'])) echo $_POST['username'] ?>" /><br />
    <label for="password1">Password:</label>
    <input type="password" id="password1" name="password1" /><br />
    <label for="password2">Password (retype):</label>
    <input type="password" id="password2" name="password2" /><br />
</fieldset>
<input type="submit" value="Sign Up" name="submit" />
</form>

```

## não existem Perguntas Idiotas

signup.php

**P:** Por que não poderíamos simplesmente usar autenticação HTTP para o cadastro de novos usuários?

**R:** Porque o propósito do script de cadastro não é restringir o acesso a determinadas páginas. A sua tarefa é permitir que o usuário escolha um nome e uma senha, e então adicioná-los ao banco de dados. Certamente é possível usar a janela de autenticação HTTP como formulário de input para o nome e a senha, mas usar a funcionalidade de autenticação apenas para cadastrar um novo usuário seria como usar uma espingarda para matar um mosquito. É melhor criar um formulário personalizado para os cadastros – assim você obterá o benefício da verificação dupla da senha, para se evitar erros de digitação.

**P:** Então o script de cadastro faz login dos usuários depois que eles se cadastram?

**R:** Não. E o motivo tem a ver principalmente com o fato de que o script de login já lida com a tarefa de logar o usuário, portanto não há necessidade de duplicar esse código no script de cadastro. Em vez disso, o script de cadastro apresenta um link para a página Editar Perfil, onde o usuário presumivelmente desejará ir após se cadastrar. E uma vez que ainda não estará logado, a janela de login lhe será apresentada em resposta à tentativa de acessar a página Editar Perfil. Assim, o script de cadastro leva o usuário até a janela de login através da página Editar Perfil, em vez de fazer o seu login automaticamente.



## TEST DRIVE

### Adicione a funcionalidade de cadastro ao Mismatch.

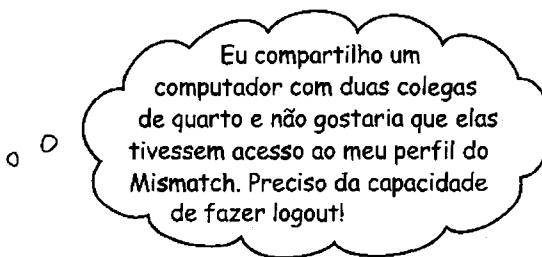
Crie um arquivo de texto chamado signup.php e digite nele o código do script de cadastro (ou baixe o script no site da Alta Books, [www.altabooks.com.br](http://www.altabooks.com.br)). Em seguida, modifique o script login.php, adicionando-lhe links para o script de cadastro para os usuários que não conseguirem fazer o login.

Envie os scripts ao seu servidor web, e em seguida, abra a página Cadastrar no seu navegador. Cadastre-se como um novo usuário e depois faça login. Em seguida, edite e visualize o seu perfil para confirmar que o cadastro e o login funcionaram corretamente. A aplicação agora tem aquele toque pessoal que estava faltando.

The diagram illustrates the user flow from registration to login and profile editing:

- Mismatch - Sign Up:** A registration form is shown with fields for Username (Ruby), Password, and Password (Retype). An arrow points from this screen to the login screen.
- Login Screen:** A login form is shown with fields for Name (Ruby) and Password. A checkmark is selected for "Remember this password in my keychain". A callout bubble states: "A autenticação HTTP é usada para logar Ruby com base nas informações que ela forneceu no seu cadastro."
- Mismatch - Edit Profile:** A profile edit form is shown with fields for First name (Ruby), Last name (Reagan), Gender (Male), Birthdate (1972-09-18), City (Cupertino), State (CA), and Picture (Choose file). A callout bubble says: "Legal! Agora eu posso fazer login no Mismatch e editar e visualizar meu perfil pessoal."
- Mismatch - View Profile:** A view profile screen shows the user's information: Username: ruby, First name: Ruby, Last name: Reagan, Gender: Male, Birthdate: 1972-09-18, Location: Cupertino, CA, and a picture placeholder. A callout bubble says: "O perfil de Ruby só fica acessível depois do login."
- User Silhouette:** A silhouette of a person wearing glasses and a beanie is shown at the bottom right, looking at the screen.

*mismatch também precisa permitir que os usuários façam logout*



**Os sites de comunidades precisam permitir que os usuários façam logout, para que os seus dados pessoais não possam ser acessados por terceiros em um computador compartilhado.**

Permitir que os usuários façam logout pode parecer algo extremamente simples, mas apresenta um problema relativamente sério quando se usa a autenticação HTTP. O problema é que esta serve para ser executada uma vez para uma determinada página ou conjunto de páginas – ela só é resetada quando o navegador é fechado. Em outras palavras, o usuário nunca é "deslogado" de uma página web autenticada via HTTP até que o navegador seja fechado ou que o usuário manualmente finalize a sessão de autenticação HTTP. Esta última opção é mais fácil de executar em alguns navegadores (Firefox, por exemplo) do que em outros (Safari).

*Uma vez feito o login, você permanece logado até fechar o navegador.*

You are logged in as sidneyk.

Mismatch - Edit Profile

Personal Information

First name:	Sidney	Name:	sidneyk
Last name:	Kelso	Password:	*****
Gender:	Female	Your password will be sent in the clear.	
Birthdate:	1984-07-19	Remember this password in MY keychain	
City:	Tempe	Cancel	Login
State:	AZ	<input type="checkbox"/>	
Picture:	<input type="button" value="Choose file"/> sidneypic.jpg		

*Um recurso de logout permitiria a Sidney controlar cuidadosamente o acesso ao seu perfil pessoal.*

Embora a autenticação HTTP apresente uma forma simples e útil de estabelecer os logins dos usuários na aplicação Mismatch, ela não fornece nenhum controle para fazer o logout do usuário. Nós precisamos tanto do recurso de lembrar dos usuários quanto de permitir que eles saiam do sistema quando quiserem.



Não seria um sonho se nós pudéssemos nos lembrar do usuário sem mantê-lo logado para sempre? Ou sou uma romântica PHP incurável?

## Às vezes você só precisa de um cookie

O problema originalmente resolvido pela autenticação HTTP tem duas faces: há a questão de se limitar o acesso a certas páginas, e há a questão de lembrar que o usuário digitou informações sobre si mesmo. O segundo problema é o mais complexo, porque ele envolve a capacidade de a aplicação se lembrar de quem o usuário é ao longo de diversas páginas (scripts). O Mismatch resolve essa questão verificando o nome e a senha armazenados na superglobal `$_SERVER`. Assim, nós podemos tirar proveito do fato de que o PHP armazena o nome e a senha da autenticação HTTP em uma superglobal que persiste ao longo de diversas páginas.

To view this page, you need to log in to area "Mismatch" on www.mis-mismatch.net.  
Your password will be sent in the clear.

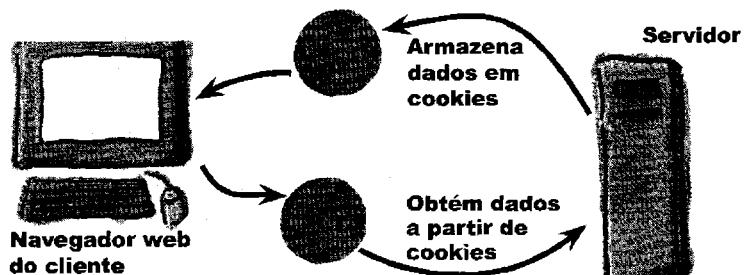
Name: sidneyk  
Password: \*\*\*\*\*  
 Remember this password in my keychain  
 Cancel  Logout

Os cookies lhe permitem armazenar no cliente pequenos pedaços de dados de forma persistente, significando que eles podem continuar existindo mesmo depois de finalizado o script... e podem também ser apagados a qualquer momento!

A autenticação HTTP armazena dados persistentemente no cliente, mas não permite que você os apague quando termina de usá-los.

A superglobal `$_SERVER` armazena o nome e a senha persistentemente.

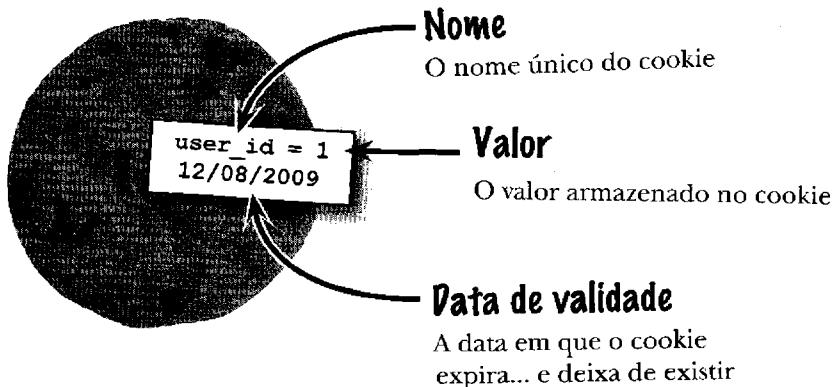
Mas não podemos mais nos darmos ao luxo de usar a autenticação HTTP, porque ela não tem suporte a logouts. Assim, precisamos procurar outro meio de implementar a persistência do usuário ao longo de diversas páginas. Uma possível solução são os cookies, pedaços de dados armazenados pelo navegador no computador do usuário. Os cookies são bastante parecidos com variáveis PHP, exceto pelo fato de que eles continuam existindo depois que você fecha o navegador, desliga o computador, etc. E o que é mais importante, os cookies podem ser apagados, significando que você pode eliminá-los quando tiver terminado de armazenar dados, por exemplo, quando o usuário indicar que deseja fazer logout.



Os dados dos cookies são armazenados no computador do usuário pelo seu navegador web. Você tem acesso aos dados através de código PHP, e o cookie é capaz de persistir ao longo não só de várias páginas (scripts), mas até mesmo ao longo de várias sessões do navegador. Assim, fechar o navegador não faz o usuário se deslogar automaticamente do Mismatch. Isso não é problema para nós, porque podemos apagar um cookie a qualquer momento, a partir de código de script, o que torna possível oferecer um recurso de logout. Podemos dar aos usuários controle totalmente sobre em que momento eles querem sair do sistema.

## Do que é feito o cookie?

O cookie armazena um único dado, sob um nome único, praticamente da mesma forma que uma variável em PHP. Ao contrário das variáveis, o cookie pode ter um prazo de validade. Quando a data de validade é atingida, ele é destruído. Assim, os cookies não são exatamente imortais – eles apenas vivem por mais tempo do que variáveis PHP. Você pode criar um cookie sem prazo de validade, em cujo caso ele se comporta do mesmo jeito que uma variável PHP – é destruído assim que o navegador é fechado.



Os cookies lhe permitem armazenar uma string de texto sob um determinado nome, mais ou menos como uma variável de texto PHP. O que torna os cookies tão poderosos é o fato de poderem viver por mais tempo do que os dados de um script, especialmente em situações em que a aplicação consiste de várias páginas que precisam se lembrar de alguns dados, por exemplo, informações de login.



Definir o prazo de validade de um cookie com uma data bastante futura ajuda a torná-lo mais permanente.

Quando você não fornece nenhum prazo de validade, o cookie é apagado assim que o navegador for fechado.

Assim, o Mismatch pode imitar a persistência fornecida pela superglobal `$_SERVER` configurando dois cookies – um para o nome do usuário e outro para a senha. Mas nós não precisamos realmente manter a senha, poderia ser mais útil armazenar a ID do usuário em vez disso.

### não existem Perguntas Idiotas

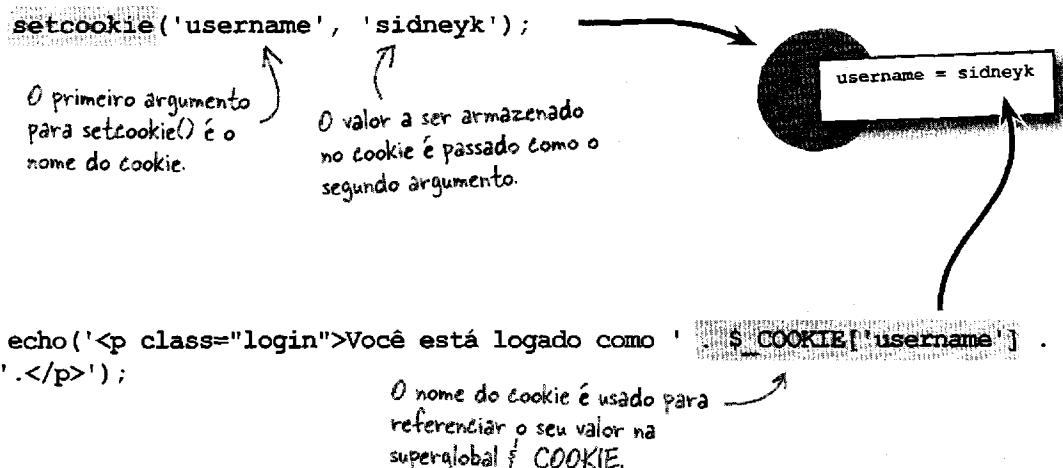
**P:** Qual é a grande vantagem dos cookies serem persistentes? Os dados armazenados em um banco de dados MySQL não são também persistentes?

**R:** Sim, os dados de um banco certamente são persistentes. Na verdade, são tecnicamente muito mais persistentes do que um cookie, porque não há nenhum prazo de validade envolvido – depois que você coloca dados em um banco, eles ficam lá até que você os remova explicitamente. A questão real, no que se refere aos cookies e a persistência é a conveniência. Nós não precisamos armazenar para sempre a ID atual ou o nome do usuário só para permitir que ele acesse seu perfil; só precisamos de uma forma rápida de saber quem ele é. O que realmente precisamos é de persistência temporária, o que pode parecer paradoxal até você considerar o fato de que precisamos que os dados permaneçam existindo por mais tempo do que uma página (persistência), mas não eternamente.

A função setcookie()

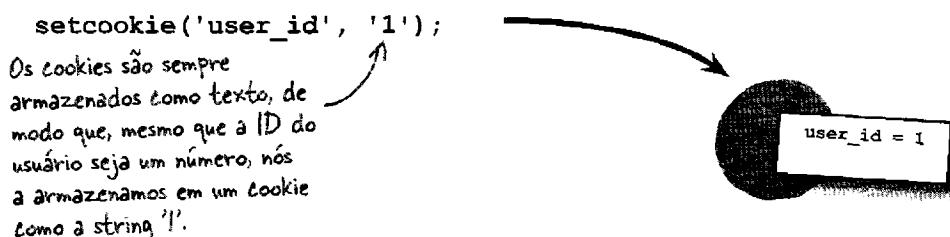
## Use cookies com PHP

O PHP fornece acesso a cookies através de uma função chamada setcookie() e uma superglobal chamada \$\_COOKIE. A função setcookie() é usada para se definir o valor e a data de validade (opcional) de um cookie, e a superglobal \$\_COOKIE é usada para se obter o valor de um cookie.



A vantagem de se definir um cookie é que os seus dados persistem ao longo de vários scripts, portanto nós podemos nos lembrar do nome do usuário sem ter de pedir que ele o informe a cada vez que passar de uma página para outra na aplicação. Mas não se esqueça, precisamos armazenar a ID do usuário em um cookie, uma vez que ela serve de chave primária para consultas ao banco de dados.

A função  
setcookie() do  
PHP lhe permite  
armazenar dados  
em cookies.



A função `setcookie()` também aceita um terceiro argumento opcional, o qual define o prazo de validade do cookie, que é a data na qual este é automaticamente apagado. Se você não especificar um prazo de validade, como no exemplo acima, o cookie expira automaticamente assim que o navegador for fechado.

## Aponte seu lápis

Ajustar o Mismatch para usar cookies envolve mais do que apenas escrever um novo script de logout. Primeiramente, nós temos de rever o script de login e modificá-lo para usar cookies em vez da autenticação HTTP. Circule e comente as partes do código de login que você acha que precisa modificar para acomodar os cookies.

```
<?php
require_once('connectvars.php');

if (!isset($_SERVER['PHP_AUTH_USER']) || !isset($_SERVER['PHP_AUTH_PW'])) {
    // O nome/senha não foram digitados, portanto enviar os cabeçalhos de
    autenticação header('HTTP/1.1 401 Unauthorized');
    header('WWW-Authenticate: Basic realm="Mismatch"');
    exit('<h3>Mismatch</h3>Desculpe, você deve digitar seu nome e senha para fazer'.
        'login e acessar esta página. Se ainda não for um usuário registrado, por favor'.
        '<a href="signup.php">cadastre-se</a>');
}

// Conecta-se ao banco de dados
$dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

// Obtém os dados de login digitados pelo usuário
$user_username = mysqli_real_escape_string($dbc, trim($_SERVER['PHP_AUTH_USER']));
$user_password = mysqli_real_escape_string($dbc, trim($_SERVER['PHP_AUTH_PW']));

// Procura o nome e a senha no banco de dados
$query = "SELECT user_id, username FROM mismatch_user WHERE username = ".
    "'$user_username' AND password = SHA('$user_password')";
$data = mysqli_query($dbc, $query);

if (mysqli_num_rows($data) == 1) {
    // O login foi bem-sucedido, portanto definir as variáveis de ID e nome do usuário
    $row = mysqli_fetch_array($data);
    $user_id = $row['user_id'];
    $username = $row['username'];
}
else {
    // O nome/senha estão incorretos, portanto enviar os cabeçalhos de autenticação
    headers
    header('HTTP/1.1 401 Unauthorized');
    header('WWW-Authenticate: Basic realm="Mismatch"');
    exit('<h2>Mismatch</h2>Desculpe, você deve digitar um nome e '.
        'senha válidos para fazer login e acessar esta página'.
        'Se ainda não for um usuário registrado, por favor,'.
        '<a href="signup.php">cadastre-se</a>');
}

// Confirma o login bem-sucedido.
echo('<p class="login">Você está logado como ' . $username . '.</p>');
?>
```



login.php



## Aponte seu lápis Solução

Ajustar o Mismatch para usar cookies envolve mais do que apenas escrever um novo script de logout. Primeiramente, nós temos de rever o script de login e modificá-lo para usar cookies em vez da autenticação HTTP. Circule e comente as partes do código de login que você acha que precisa modificar para acomodar os cookies.

Nós precisamos verificar a existência de um cookie para ver se o usuário está logado ou não.

Em vez de obter o nome e a senha a partir de uma janela de autenticação, nós precisamos usar um formulário com dados POST.

```
<?php
require_once('connectvars.php');

if (!isset($_SERVER['PHP_AUTH_USER']) || !isset($_SERVER['PHP_AUTH_PW'])) {
    // O nome/senha não foram digitados. Portanto enviar os cabeçalhos de
    // autenticação header('HTTP/1.1 401 Unauthorized');
    header('WWW-Authenticate: Basic realm="Mismatch"');
    exit('<h3>Mismatch</h3>Desculpe, você deve digitar seu nome e senha para fazer
    login e acessar esta página. Se ainda não for um usuário registrado, por favor,
    <a href="signup.php">cadastre-se</a>.');
}

// Não precisamos mais enviar cabeçalhos
// de autenticação HTTP.

// Conecta-se ao banco de dados
$dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

// Obtém os dados de login digitados pelo usuário
$user_username = mysqli_real_escape_string($dbc, trim($_SERVER['PHP_AUTH_USER']));
$user_password = mysqli_real_escape_string($dbc, trim($_SERVER['PHP_AUTH_PW']));

// Procura o nome e a senha no banco de dados
$query = "SELECT user_id, username FROM mismatch_user WHERE username = "
    . "'$user_username' AND password = SHA('$user_password')";
$data = mysqli_query($dbc, $query);

if (mysqli_num_rows($data) == 1) {
    // O login foi bem-sucedido, portanto definir as variáveis de ID e nome do usuário
    $row = mysqli_fetch_array($data);
    $user_id = $row['user_id'];
    $username = $row['username'];
} else {
    // O nome/senha estão incorretos, portanto enviar os cabeçalhos de autenticação headers
    header('HTTP/1.1 401 Unauthorized');
    header('WWW-Authenticate: Basic realm="Mismatch"');
    exit('<h2>Mismatch</h2>Desculpe, você deve digitar um nome e '
        . 'senha válidos para fazer login e acessar esta página'.
        'Se ainda não for um usuário registrado, por favor,'.
        '<a href="signup.php">cadastre-se</a>.');
}

// Confirma o login bem-sucedido.
echo('<p class="login">Você está logado como ' . $username . '.</p>');
?>
```

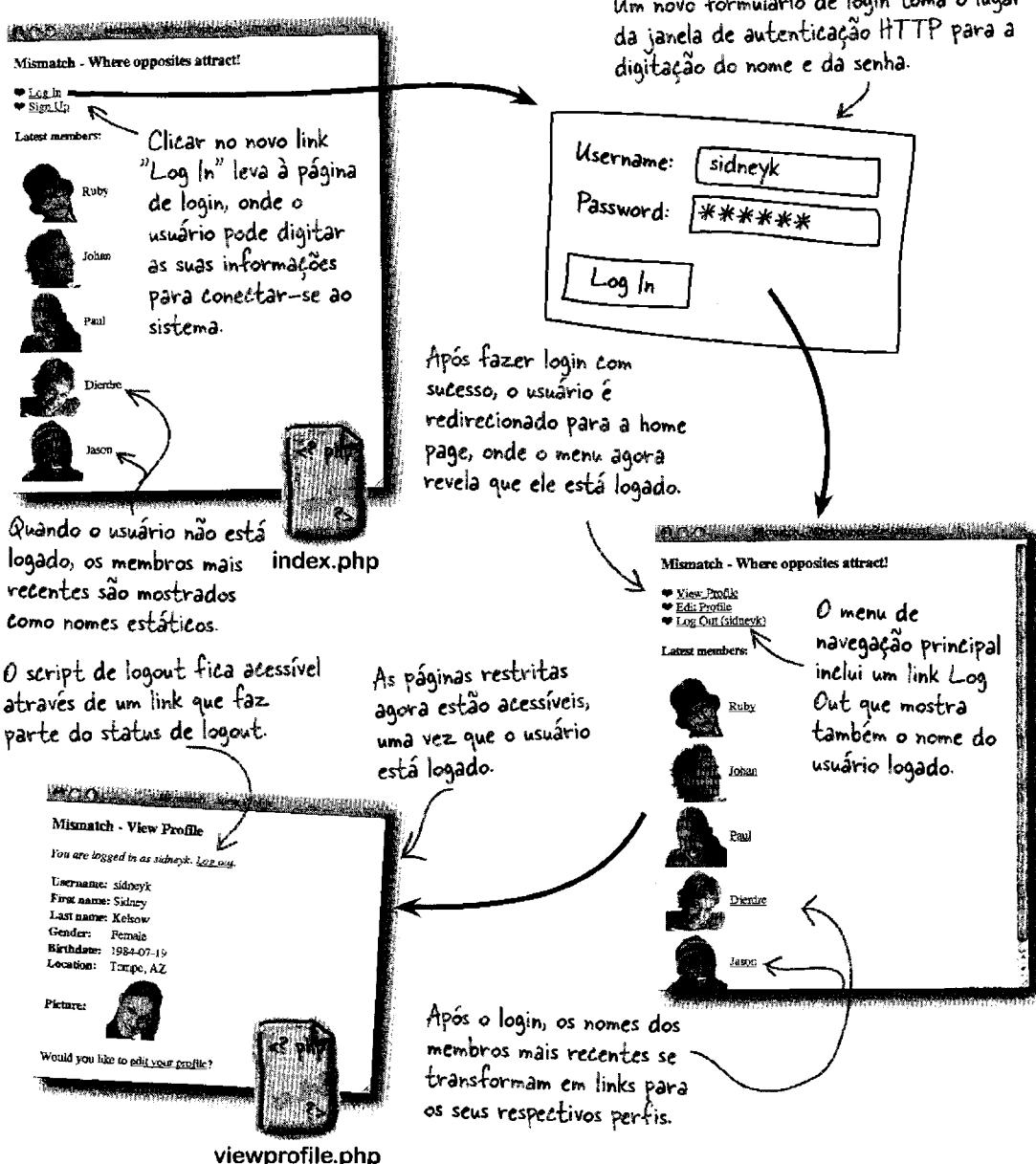
Uma vez que não iremos usar a janela de autenticação HTTP para digitação do nome e da senha, temos de criar um formulário de login HTML para o usuário fornecer essas informações.



login.php

## Repensando o fluxo dos logins

Usar cookies em vez de autenticação HTTP para os logins do Mismatch envolve mais do que apenas repensar o armazenamento dos dados dos usuários. E quanto à interface do login? O login baseado em cookies precisará fornecer o seu próprio formulário, uma vez que não poderá usar a janela de autenticação para a digitação de nome e senha. Nós não só precisamos criar esse formulário, como também temos de pensar em como ele modifica o fluxo da aplicação, à medida que os usuários fazem login e acessam outras páginas.



## Um login baseado em cookies

A nova versão do script de login, que usa cookies para implementar a persistência, é um pouco mais complexa do que a sua predecessora, uma vez que ela precisa fornecer o seu próprio formulário para a digitação do nome do usuário e da senha. Mas é também mais poderosa, uma vez que oferece a funcionalidade de logout.

```
<?php
    require_once('connectvars.php');
    $error_msg = '';
    if (!empty($error_msg)) {
        // Limpa a mensagem de erro
        $error_msg = '';
    }
    // Se o usuário não estiver logado, tenta fazer o login
    if (!isset($_COOKIE['user_id'])) {
        if (!isset($_POST['username'])) {
            // Conecta-se ao banco de dados
            $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
            // Obtém os dados de login digitados pelo usuário
            $user_username = mysqli_real_escape_string($dbc, trim($_POST['username']));
            $user_password = mysqli_real_escape_string($dbc, trim($_POST['password']));
            if (!empty($user_username) && !empty($user_password)) {
                // Procura o nome e a senha no banco de dados
                $query = "SELECT user_id, username FROM mismatch_user WHERE username = '$user_username' AND password = SHA('$user_password')";
                $data = mysqli_query($dbc, $query);
                if (mysqli_num_rows($data) == 1) {
                    // O login foi bem-sucedido, portanto definir os cookies de ID e nome do usuário e redirecionar para a home page
                    $row = mysqli_fetch_array($data);
                    setcookie('user_id', $row['user_id']);
                    setcookie('username', $row['username']);
                    $home_url = "http://$_SERVER[HTTP_HOST]/dirname($_SERVER['PHP_SELF'])";
                    header("Location: $home_url");
                } else {
                    // O nome/senha estão incorretos, portanto definir uma mensagem de erro
                    $error_msg = 'Desculpe, você deve digitar um nome e senha válidos para fazer login.';
                }
            } else {
                // O nome/senha não foram digitados, portanto definir uma mensagem de erro
                $error_msg = 'Desculpe, você deve digitar seu nome e senha para fazer login.';
            }
        }
    }
    // Redirecione o usuário para a home page do Mismatch após o login bem-sucedido.
    // Defina a variável da mensagem de erro caso algo esteja errado com os dados de login.
    // O script de login agora é uma página web completa, portanto ele requer todos os elementos HTML padrões.
    <html>
        <head>
            <title>Mismatch - Log In</title>
            <link rel="stylesheet" type="text/css" href="style.css" />
        </head>
        <body>
            <h3>Mismatch - Log In</h3>Continua na página do lado...
    </body>
</html>
```

As mensagens de erro agora são armazenadas em uma variável e exibidas, se necessário, mais adiante no script.

Eis o novo formulário de login.

Verifique o cookie user\_id para ver se o usuário está logado.

Se o usuário não estiver logado, verificar se ele submeteu dados de login.

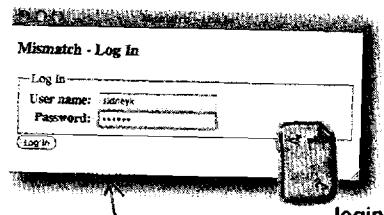
Os dados digitados pelo usuário agora vêm de POST, e não de uma janela de autenticação.

Faça login do usuário definindo os cookies user\_id e username.

Redirecione o usuário para a home page do Mismatch após o login bem-sucedido.

Defina a variável da mensagem de erro caso algo esteja errado com os dados de login.

O script de login agora é uma página web completa, portanto ele requer todos os elementos HTML padrões.



login.p

```

<?php
    // Se o cookie estiver vazio, exibir mensagem de erro (se houver) e o formulário de
    // login; caso contrário, confirmar login
    if(empty($_COOKIE['user_id'])) {
        echo "<p class='error'>".SESSION_MSG."</p>";
    } else {
        echo "<form method='post' action='<?php echo $_SERVER['PHP_SELF']; ?>'>
            <fieldset>
                <legend>Log In</legend>
                <label for='username'>Nome de Usuário:</label>
                <input type='text' id='username' name='username'
                    value='<?php if(!empty($_user['username'])) echo $_user['username']; ?>'><br/>
                <label for='password'>Senha:</label>
                <input type='password' id='password' name='password' />
            </fieldset>
            <input type='submit' value='Log In' name='submit' />
        </form>
    }
    <?php
}
else {
    // Confirma o login bem-sucedido.
    echo "<p class='logado'>Você está logado como " . $_COOKIE['username'] . ".</p>";
}
?>
</body>
</html>

```

Tudo que vem antes desta chave ainda faz parte da primeira cláusula if.

Finalize o código HTML para completar a página web de login.

Estes dois campos do formulário são usados para o usuário digitar o nome e a senha.

Se o usuário estiver logado neste ponto, informe-o disso.

## não existem Perguntas Ídiotas

**P:** Por que é necessário armazenar tanto a ID quanto o nome do usuário em cookies?

**R:** Uma vez que essas duas informações identificam individualmente os usuários dentro do banco de dados do Mismatch, você poderia usar apenas uma das duas com o propósito de manter a aplicação sabendo quem é o usuário logado. Porém, user\_id é uma referência melhor (mais eficiente) no que diz respeito ao banco de dados, porque é uma chave primária numérica. Por outro lado, user\_id é pouco informativa e não tem nenhum significado para o usuário, portanto o username é útil para o usuário saber que está logado, pois ele vê o seu próprio nome na página. Uma vez que ocorre de várias pessoas compartilharem o mesmo computador, é importante não só

informar ao usuário que ele está logado, mas também qual é o nome que está logado.

**P:** Então por que não armazenar também a senha em um cookie, como parte dos dados de login?

**R:** A senha só é importante para verificar inicialmente se o usuário realmente é quem ele diz ser. Uma vez que o processo de login verifica a senha, não há motivo para mantê-la em jogo. Além disso, senhas são dados muito sensíveis, portanto é uma boa ideia evitar armazená-las temporariamente sempre que possível.

**P:** Parece que o formulário do script de login está dentro da declaração if? Isso é possível?

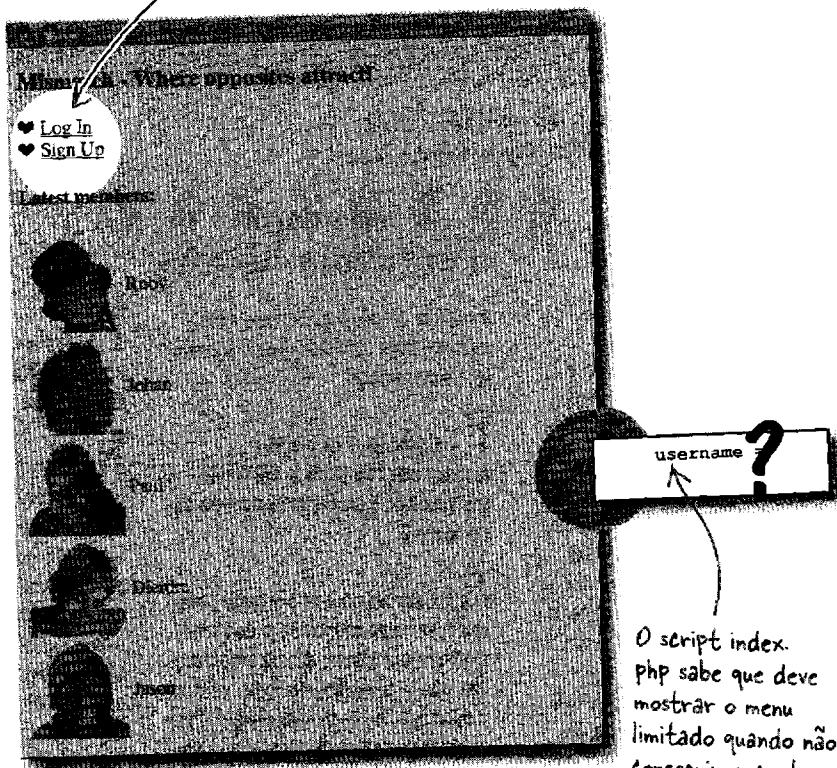
**R:** Sim. Na verdade, é bem comum ver código PHP "dividido" em torno de código HTML, como acontece no script de login. Só porque você fechou uma seção de código PHP com ?, não quer dizer que a lógica do código está fechada. Quando você abre outra seção de código PHP com <?php, a lógica continua exatamente de onde parou. No script de login, o formulário HTML é contido dentro da primeira ramificação de if, enquanto que a ramificação else aparece depois do código do formulário. Dividir código PHP com código HTML, desta forma, evita que você tenha de gerar o formulário com um monte de instruções echo confusas.

## Navegando na aplicação Mismatch

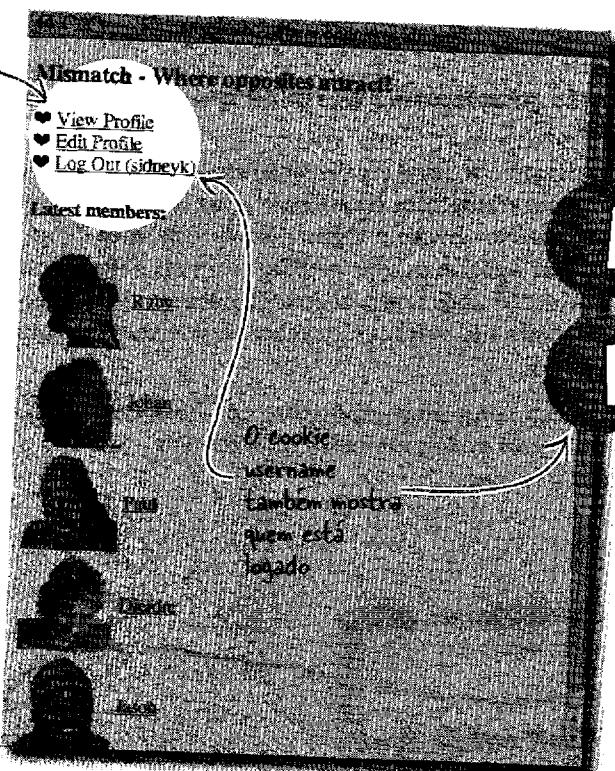
O novo script de login modifica o fluxo da aplicação Mismatch, passando a requerer um menu simples que apareça na home page (index.php). Esse menu é importante porque fornece acesso às diferentes partes principais da aplicação, atualmente as páginas Ver Perfil e Editar Perfil, bem como a possibilidade de os usuários fazerem login, se cadastrarem e fazerem logout, dependendo dos seus estados de login atuais. O fato de que o menu se modifica com base no estado de login do usuário é significativo, e em última análise, é o que dá ao menu o seu poder e a sua utilidade.

Um menu diferente é  
mostrado dependendo de  
se o cookie username está  
definido ou não.

Este menu aparece  
quando o usuário  
não está logado,  
dando a ele a  
oportunidade de  
fazer login ou se  
cadastrar.



O menu é gerado por código PHP dentro do script index.php, e esse código usa a superglobal `$_COOKIE` para procurar o cookie `username` e ver se o usuário está logado ou não. O cookie com a ID do usuário poderia também ter sido usado, mas como o nome do usuário é o mostrado no menu, faz mais sentido usá-lo.



O cookie user\_id não é usado pelos diferentes menus, mas ainda é importante para a persistência do usuário no Mismatch.

O cookie determina qual menu é exibido

O cookie  
username  
também mostra  
que o usuário  
está logado

Menu para  
usuários logados.

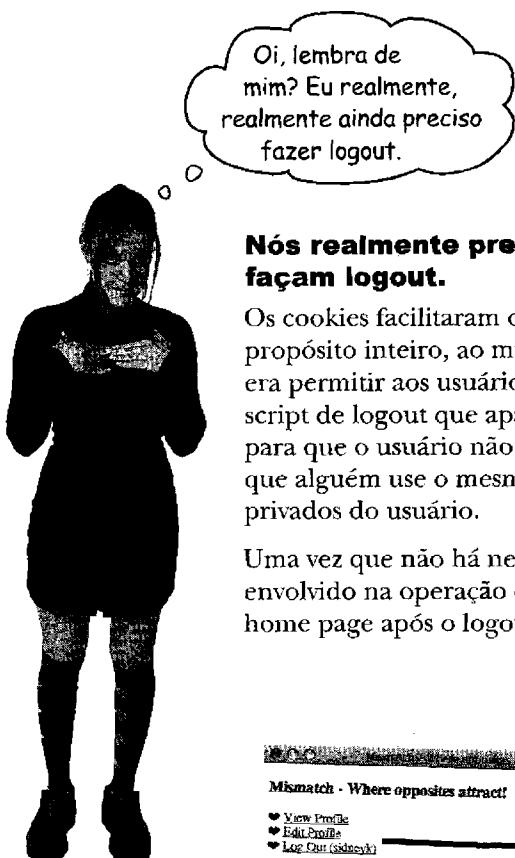
[View Profile](#)  
[Edit Profile](#)  
[Log Out \(' . \\$COOKIE\['username'\] . \)](#)

```
// Gera o menu de navegação
if (isset($_COOKIE['username'])) {
    echo '&#10084; <a href="viewprofile.php">Ver Perfil</a><br />';
    echo '&#10084; <a href="editprofile.php">Editar Perfil</a><br />';
    echo '&#10084; <a href="logout.php">Log Out (' . $_COOKIE['username'] . ')';
} else {
    echo '&#10084; <a href="login.php">Log In</a><br />';
    echo '&#10084; <a href="signup.php">Cadastrar-se</a>';
}
```

Os símbolos de coraçãozinhos ao lado de cada item do menu são possibilidades por este código HTML, que é suportado na maioria dos navegadores.

Menu para  
visitantes  
(usuários ainda  
não logados).

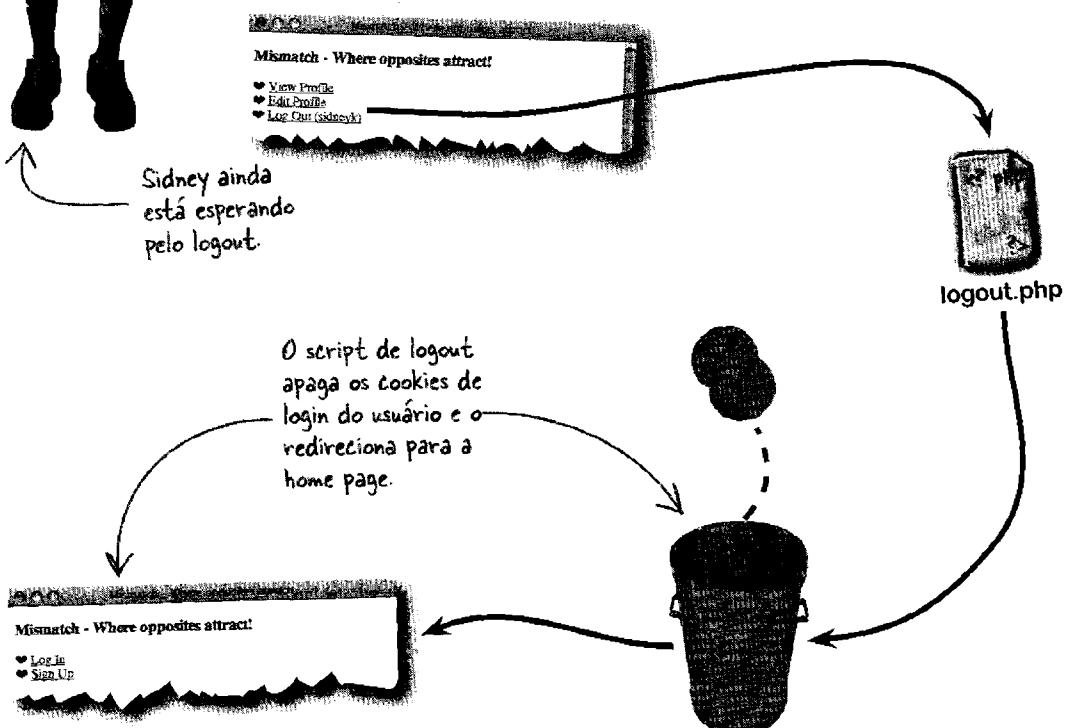
## **log out de usuários deletando cookies**



### **Nós realmente precisamos permitir que os usuários façam logout.**

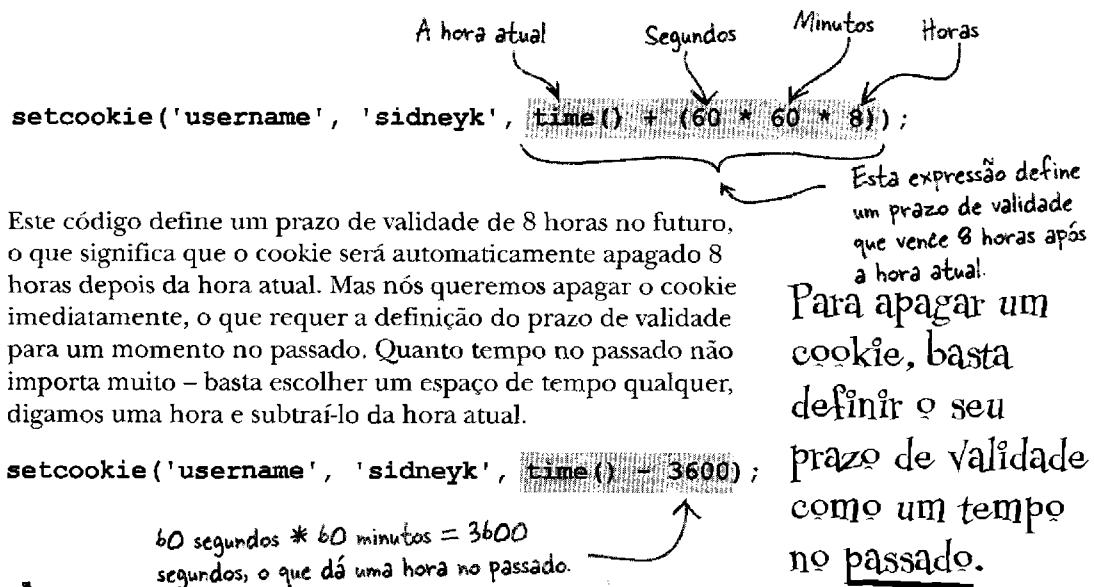
Os cookies facilitaram o login e a navegação no Mismatch, mas o nosso propósito inteiro, ao mudarmos da autenticação HTTP para os cookies, era permitir aos usuários fazerem logout. Nós precisamos de um novo script de logout que apague os dois cookies (user ID e username), para que o usuário não mais tenha acesso à aplicação. Isso impede que alguém use o mesmo computador mais tarde e acesse os dados privados do usuário.

Uma vez que não há nenhum componente de interface de usuário envolvido na operação de logout, basta redirecionar o usuário para a home page após o logout.



## Fazer logout significa apagar cookies

Fazer o logout de um usuário envolve apagar os dois cookies que mantêm registro do usuário. Isso é feito chamando-se a função `setcookie()`, e passando-se a ela uma data de validade que faça os cookies serem apagados na data em questão.



Estão faltando algumas partes do código do script de logout do Mismatch. Escreva o código que está faltando, certificando-se de apagar os cookies antes da página de logout ser redirecionada para a home page.

```
<?php
    // Se o usuário estiver logado, apagar o cookie para fazer o logout
    if ( ..... ) {
        // Apaga os cookies user ID e username, definindo os seus prazos de validade como
        // uma hora atrás (3600)
        ..... ;
        .....
    }

    // Redireciona para a home page
    $home_url = 'http://'. $_SERVER['HTTP_HOST'] . dirname($_SERVER['PHP_SELF']) .
    ..... ;
    header('Location: ' . $home_url);

?>
```

o script completo logout.php



## Exercício Solução

```
<?php  
    // Se o usuário estiver logado, apagar o cookie para fazer o logout  
    if (isset($_COOKIE['user_id'])) {  
        // Apaga os cookies user ID e username, definindo os seus prazos de validade como  
        // uma hora atrás (3600)  
        setcookie('user_id', '', time() - 3600);  
        setcookie('username', '', time() - 3600);  
        // Redireciona para a home page do  
        // Mismatch, que é construída como  
        // uma URL absoluta.  
        // Redireciona para a home page  
        $home_url = 'http://'. $_SERVER['HTTP_HOST'] . dirname($_SERVER['PHP_SELF']) .  
        '/index.php';  
        header('Location: ' . $home_url);  
    }  
?>
```

Só faz logout do usuário se este já estiver logado.

Define cada cookie como vencendo uma hora atrás, para que eles sejam apagados pelo sistema.

Um cabeçalho de localização resulta em o navegador redirecionar para outra página.

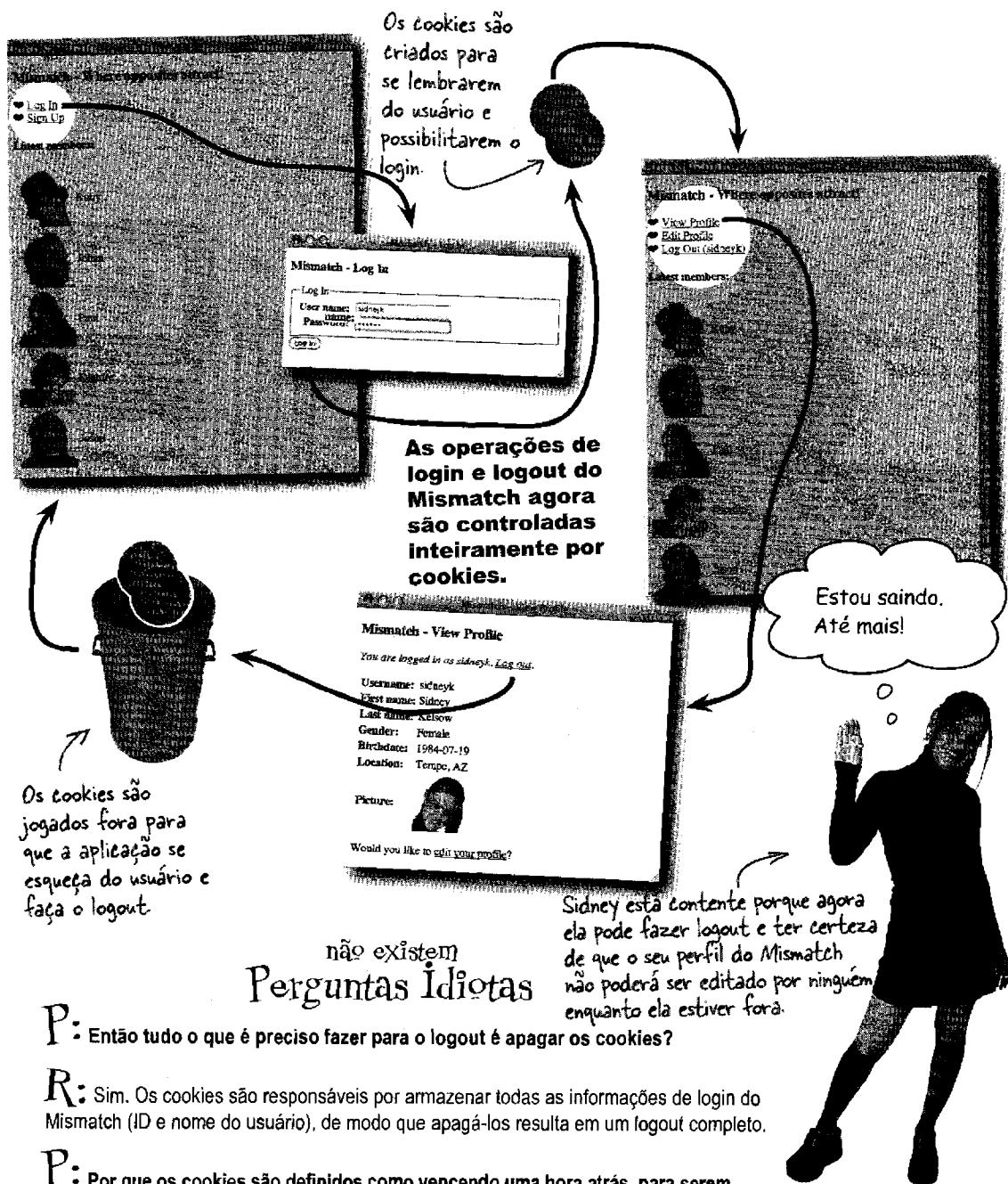


## TEST DRIVE

### Use cookies para adicionar a funcionalidade de logout ao Mismatch.

Modifique os scripts do Mismatch para que eles usem cookies para permitir que os usuários façam login e logout (ou baixe os scripts do site da Alta Books, www.altabooks.com.br). Deverão ser feitas modificações nos scripts index.php, login.php, logout.php, editprofile.php e viewprofile.php. As modificações nos dois últimos são poucas e envolvem principalmente modificar as referências das variáveis globais \$user\_id e \$username para que elas passem a usar a superglobal \$\_COOKIE.

Envie os scripts para o seu servidor web e depois abra a página principal do Mismatch (index.php) em um navegador web. Tome nota do menu de navegação e depois clique no link "Log In" para entrar no sistema. Repare em como o script de login leva você de volta à página principal, enquanto que o menu se modifica para refletir o seu status de logado. Agora clique em "Log Out" para pulverizar os cookies e sair do sistema.



## não existem Perguntas Idiotas

P: Então tudo o que é preciso fazer para o logout é apagar os cookies?

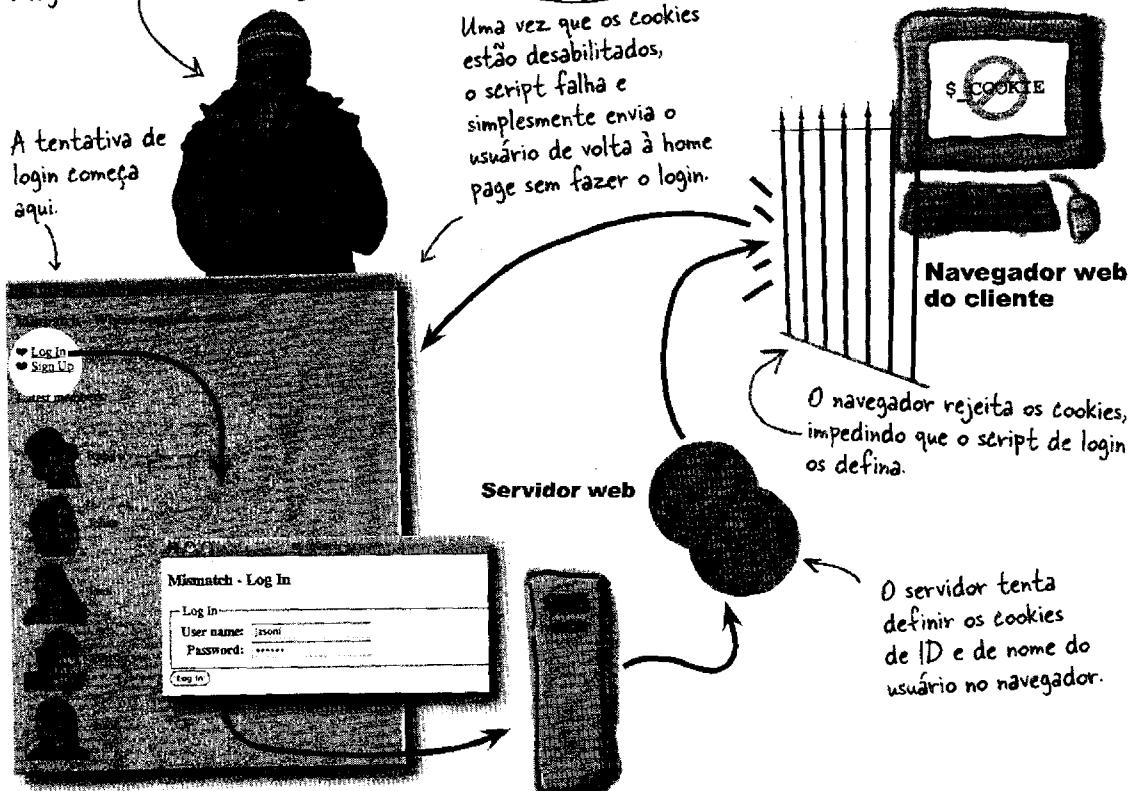
R: Sim. Os cookies são responsáveis por armazenar todas as informações de login do Mismatch (ID e nome do usuário), de modo que apagá-los resulta em um logout completo.

P: Por que os cookies são definidos como vencendo uma hora atrás, para serem apagados? Existe algo significativo sobre o espaço de tempo de uma hora?

R: Não. O cookie é automaticamente apagado pelo navegador assim que o seu prazo de validade vence. Assim, a operação de apagar um cookie imediatamente exige que se defina o prazo de validade como vencendo em algum momento do passado. Uma hora (3600 segundos) é apenas um espaço de tempo arbitrário, escolhido para indicar consistentemente que estamos apagando um cookie.

## armazenando dados do usuário no servidor, ao invés no cliente

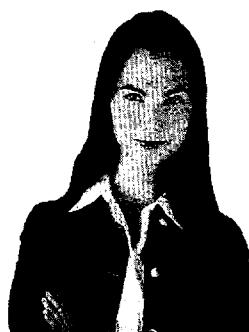
Este é o usuário do Mismatch Jason, fã de escaladas, de piercings e de Howard Stern. Ele desabilitou os cookies no seu navegador, o que representa um problema para o login.



Quem se importa com o Jason? Não é verdade que a maioria das pessoas tem os cookies habilitados?

**Sim, mas as aplicações web devem ser o mais acessíveis ao maior número de pessoas possível.**

Algumas pessoas simplesmente não se sentem à vontade para usar cookies, então elas optam pela segurança de desabilitá-los. Sabendo disso, vale a pena tentar acomodar os usuários que não terão a opção de usar cookies para o login. Mas tem mais: acontece que há outra opção que usa o servidor para armazenar os dados de login, em vez de usar o cliente. E uma vez que os nossos scripts já estão rodando no servidor, faz sentido armazenar também dados de login nele.

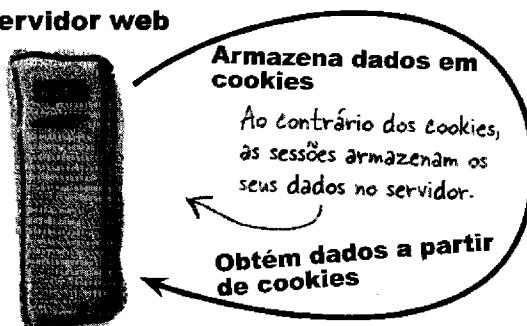


## As sessões não dependem do cliente

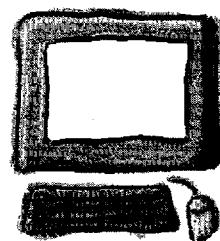
Os cookies são coisinhas poderosas, mas têm suas limitações, como por exemplo, ficarem sujeitos a limitações que vão além do seu controle. Mas e se não tivéssemos de depender do navegador? E se pudéssemos armazenar dados diretamente no servidor? As sessões fazem justamente isso, e lhe permitem armazenar pedaços individuais de informações da mesma forma que os cookies, mas os dados são armazenados no servidor em vez de no cliente. Isso coloca os dados das sessões fora dos limites impostos pelo navegador aos cookies.

As sessões lhe permitem armazenar persistentemente pequenos pedaços de dados no servidor, independentemente do cliente.

### Servidor web



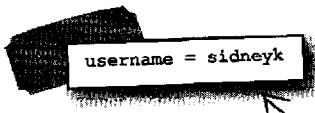
O navegador não interfere diretamente na armazenagem dos dados de sessões, uma vez que tudo fica no servidor.



### Navegador web do cliente

Uma vez que os dados de sessões são armazenados no servidor, eles são mais seguros e mais confiáveis do que dados armazenados em cookies.

As sessões armazenam dados em variáveis de sessão, que são os equivalentes lógicos dos cookies no servidor. Quando você coloca dados em uma variável de sessão usando código PHP, eles são armazenados no servidor. Você pode então acessar os dados na variável de sessão a partir de código PHP, e eles continuam persistentes ao longo de várias páginas (scripts). Como no caso dos cookies, você pode apagar uma variável de sessão a qualquer momento, o que nos possibilita continuarmos a oferecer o recurso de logout com código baseado em sessões.



Certamente existe alguma desvantagem, certo? Mais ou menos. Ao contrário dos cookies, as sessões não oferecem tanto controle sobre o tempo durante o qual a variável de sessão armazenará os dados. As variáveis de sessão são automaticamente destruídas assim que a sessão termina, o que em geral, coincide com o ato de o usuário fechar o navegador. Assim, embora as variáveis de sessão não sejam armazenadas no navegador, elas são afetadas indiretamente por ele, uma vez que são apagadas quando a sessão do navegador finaliza.

O usuário não é capaz de apagar dados de sessões manualmente com o seu navegador, o que pode ser um problema quando se usam cookies.

Não há um prazo de validade associado com as variáveis de sessão, porque elas são automaticamente apagadas quando a sessão termina.

as funções `session_start()` e `session_destroy()`

## A vida e a obra das sessões

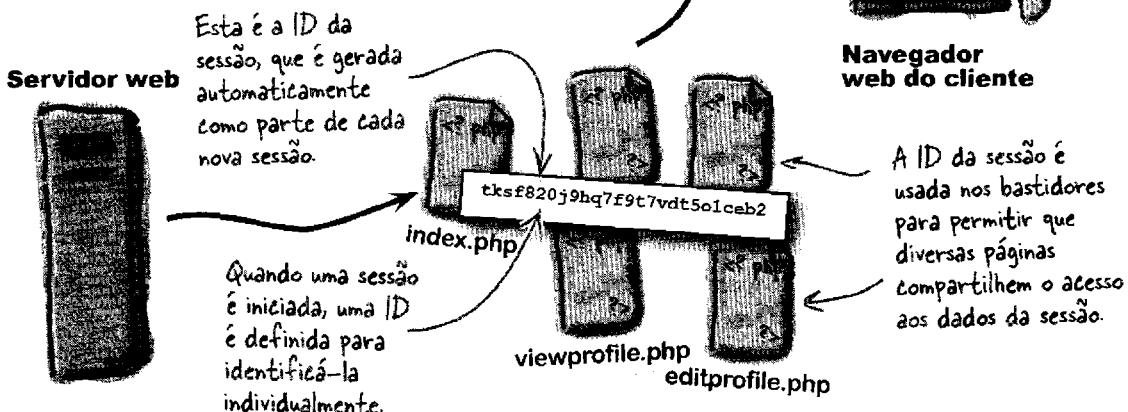
As sessões recebem esse nome por um motivo – elas têm um início e um fim muito claros. Os dados associados com uma sessão vivem e morrem de acordo com o tempo de vida da sessão, o qual você controla através de código PHP. A única situação em que você não tem controle do ciclo de vida da sessão é quando o usuário fecha o navegador, o que resulta no fim da sessão, quer você goste disso ou não.

Você precisa informar à sessão quando estiver pronto para iniciá-la, chamando a função PHP `session_start()`.

A função PHP `session_start()` inicia uma sessão e permite que você comece a armazenar dados em variáveis da sessão.

`session_start();` ← Esta função PHP inicia uma sessão.

Chamar a função `session_start()` não define quaisquer dados – a tarefa dela é apenas iniciar a sessão. A sessão é identificada internamente por uma ID individual, com a qual você geralmente não precisa se preocupar. Essa ID é usada pelo navegador para associar uma sessão com diversas páginas.



A ID não é destruída até que a sessão seja finalizada, o que acontece ou quando o navegador é fechado ou quando você chama a função `session_destroy()`.

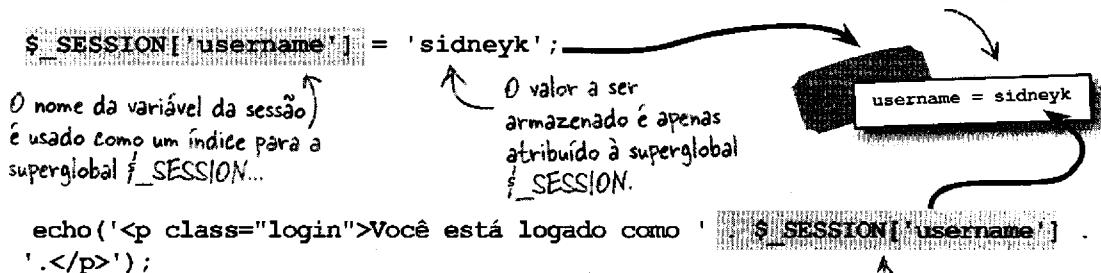
`session_destroy();` ← Esta função PHP finaliza uma sessão.

A função `session_destroy()` finaliza a sessão.

Quando você fecha uma sessão com esta função, ela não destrói automaticamente quaisquer variáveis de sessão que você tenha armazenado. Vamos dar uma olhada com mais detalhes no modo como as sessões armazenam dados, para descobrir por que isso acontece.

## Mantendo-se atualizado com os dados das sessões

O legal sobre as sessões é que elas são bastante semelhantes aos cookies, em termos de como você as usa. Uma vez iniciada uma sessão com uma chamada a `session_start()`, você pode começar a definir as variáveis, como por exemplo os dados de login do Mismatch, com a superglobal `$_SESSION`.



```
echo ('<p class="login">Você está logado como ' . $_SESSION['username'] . '</p>');
```

Ao contrário dos cookies, as variáveis de sessão não requerem nenhum tipo de função especial para defini-las – basta atribuir um valor à superglobal `$_SESSION`, certificando-se de usar o nome da variável como o índice do array.

E quanto a apagar as variáveis de sessão? Finalizar a sessão através de `session_destroy()` não destrói as suas variáveis, portanto, você precisa apagá-las manualmente se quiser que elas sejam eliminadas antes de o usuário fechar o navegador (`logout!`). Uma forma rápida e eficiente de destruir todas as variáveis de uma sessão é definir a superglobal `$_SESSION` como um array vazio.

`$_SESSION = array();`

Este código elimina todas as variáveis da sessão atual.

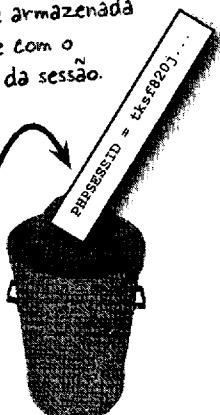
Mas isso ainda não é tudo. As sessões podem usar cookies nos bastidores. Se o navegador os tiver habilitados, a sessão pode definir um cookie para armazenar temporariamente a ID da sessão. Assim, para fechar completamente uma sessão via código PHP, você precisa também apagar qualquer cookie que possa ter sido criado automaticamente para armazenar essa ID no navegador. Como qualquer outro cookie, você o destrói definindo o seu prazo de validade como um tempo no passado. Tudo o que você precisa saber é o nome do cookie, que pode ser descoberto usando-se a função `session_name()`.

```
if (isset($_COOKIE[session_name()])) {
    setcookie(session_name(), '', time() - 3600);
}
```

Primeiramente, verifique se realmente existe um cookie da sessão.

Destrua o cookie da sessão definindo o seu prazo de validade para uma hora atrás.

Se a sessão estiver usando um cookie para ajudar a se lembrar da sua ID, então a ID é armazenada em um cookie com o mesmo nome da sessão.

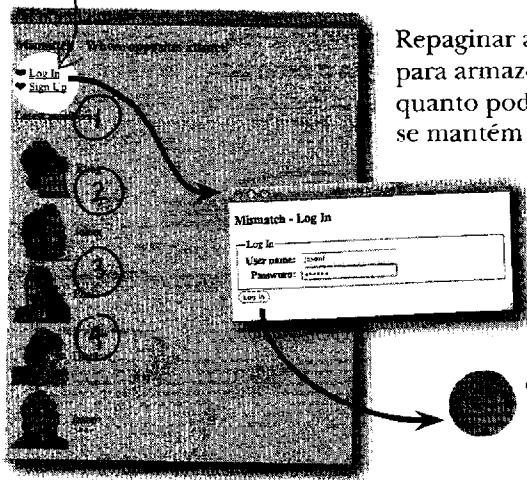


como o mismatch trabalha com sessões

Começa aqui!

## Renove o Mismatch com as sessões

Repaginar a aplicação Mismatch para que ele use sessões para armazenar dados de login não é tão complexo quanto pode parecer. Na verdade, o fluxo da aplicação se mantém quase o mesmo – você só precisa tomar um pouco mais de cuidado nas operações de iniciar e destruir a sessão e de limpar a sujeira uma vez terminada a sessão.



`session_start();`

A função `session_start()` abre a sessão.

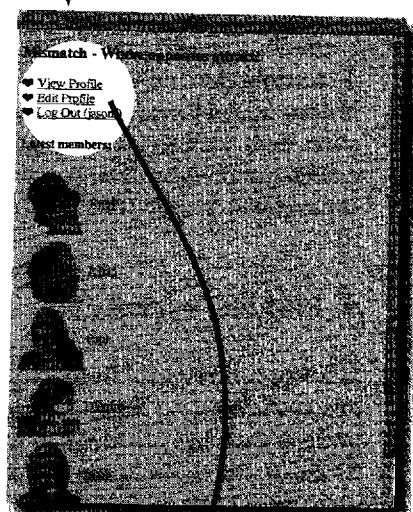
Se os cookies estiverem habilitados, o servidor cria um para armazenar a ID da sessão – caso contrário, a ID é passada juntamente com a URL de cada página.

`session_destroy();`

A função `session_destroy()` finaliza a sessão, impedindo que ela seja usada em outra página.

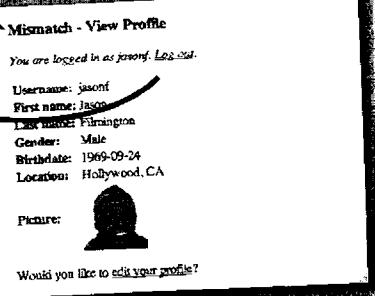
As variáveis da sessão são destruídas limpando-se o array `$_SESSION`.

As duas variáveis da sessão são criadas para armazenarem a ID e o nome do usuário, para o login.



Se um cookie tiver sido usado para armazenar a ID da sessão, ele é destruído.

Os dados do login agora são memorizados usando-se uma sessão, em vez de cookies.



# Faça Logout com sessões

Para se fazer o logout do usuário, é preciso um pouco mais de trabalho com as sessões do que em versões anteriores, em que usávamos apenas cookies. Estes passos precisam ser executados para se fazer logout com sucesso, nas sessões do Mismatch.

## 1 Apagar as variáveis da sessão.

Sem verificar, você não saberá com certeza se um cookie de sessão está sendo usado.

## 2 Verificar se existe um cookie de sessão e, se existir, apagá-lo.

Ok, esse aqui é um passo não essencial para o logout do usuário, mas é útil de qualquer forma.

## 3 Destruir a sessão.

## 4 Redirecionar o usuário à home page.



O script de logout do Mismatch está sendo reelaborado para usar sessões, em vez de apenas cookies, para a persistência dos logins. Escreva o código que está faltando para "sessionalizar" o script, e depois escreva a qual passo do processo de logout cada inserção corresponde.

```
<?php
    // Se o usuário estiver logado, apagar as vars de sessão para fazer o logout
    session_start();
    if (.....) {
        // Apaga as vars de sessão limpando o array $_SESSION.
        .....
        // Apaga o cookie de sessão, definindo o seu prazo de validade como uma hora atrás (3600)
        if (isset($_COOKIE[session_name()])) {
            .....
        }
        // Destroi a sessão
        .....
    }

    // Redireciona para a home page
    $home_url = 'http://'. $_SERVER['HTTP_HOST'] . dirname($_SERVER['PHP_SELF']) . '/index.php';
    header('Location: ' . $home_url);
?>
```

## Aponte seu lápis

### Solução

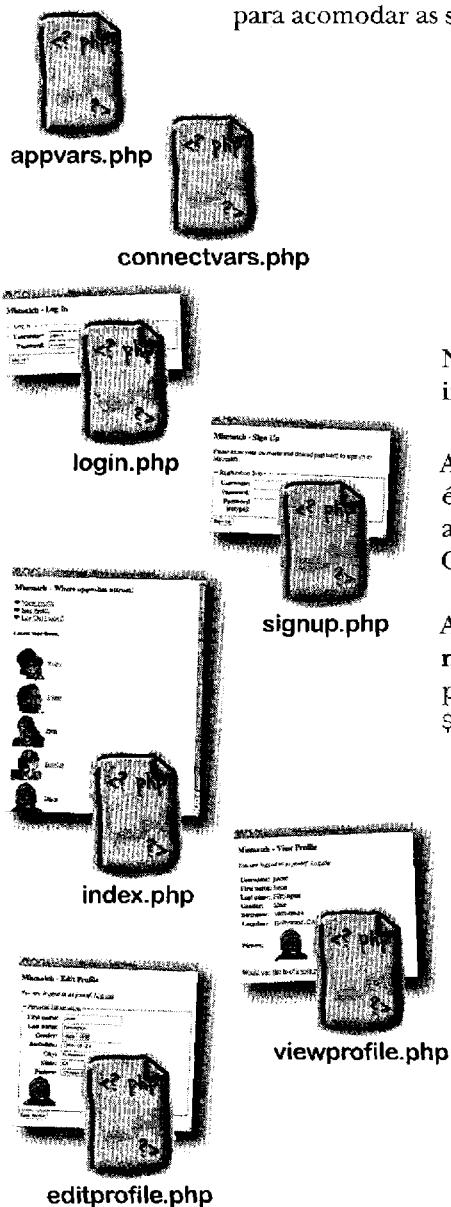
O script de logout do Mismatch está sendo reelaborado para usar sessões, em vez de apenas cookies, para a persistência dos logins. Escreva o código que está faltando para "sessionalizar" o script, e depois escreva a qual passo do processo de logout cada inserção corresponde.

- 1 Apagar as variáveis da sessão.
- 2 Verificar se existe um cookie de sessão e, se existir, apagá-lo.
- 3 Destruir a sessão
- 4 Redirecionar o usuário à home page.

Mesmo ao fazer o logout, você  
tem de iniciar a sessão para poder  
acessar as variáveis.

```
<?php
    // o usuário estiver logado, apagar as vars de sessão para fazer o logout
    session_start();
    if (isset($_SESSION['user_id'])) {
        // Apaga as vars de sessão limpando o array $_SESSION.
        $_SESSION = array(); (1) Agora, está sendo usada uma variável de sessão, em vez de um cookie, para verificar o status do login.
        // Para limpar as variáveis da sessão, atribua um array vazio à superglobal $_SESSION.
    }
    // Apaga o cookie de sessão, definindo o seu prazo de validade como uma hora atrás (3600)
    if (isset($_COOKIE[session_name()])) {
        setcookie(session_name(), "", time() - 3600); (2) Se existir um cookie da sessão, apague-o, definindo o seu prazo de validade como uma hora atrás.
    }
    // Destroi a sessão
    session_destroy(); (3) Destrua a sessão com uma chamada à função session_destroy().
}
// Redireciona para a home page
$home_url = 'http://'. $_SERVER['HTTP_HOST'] . dirname($_SERVER['PHP_SELF']) . '/index.php';
header('Location: ' . $home_url); (4)
?>
```

A mudança dos cookies para as sessões tem impacto sobre mais coisas do que apenas o script de logout. Ligue cada parte da aplicação Mismatch à modificação por que elas têm de passar para acomodar as sessões.



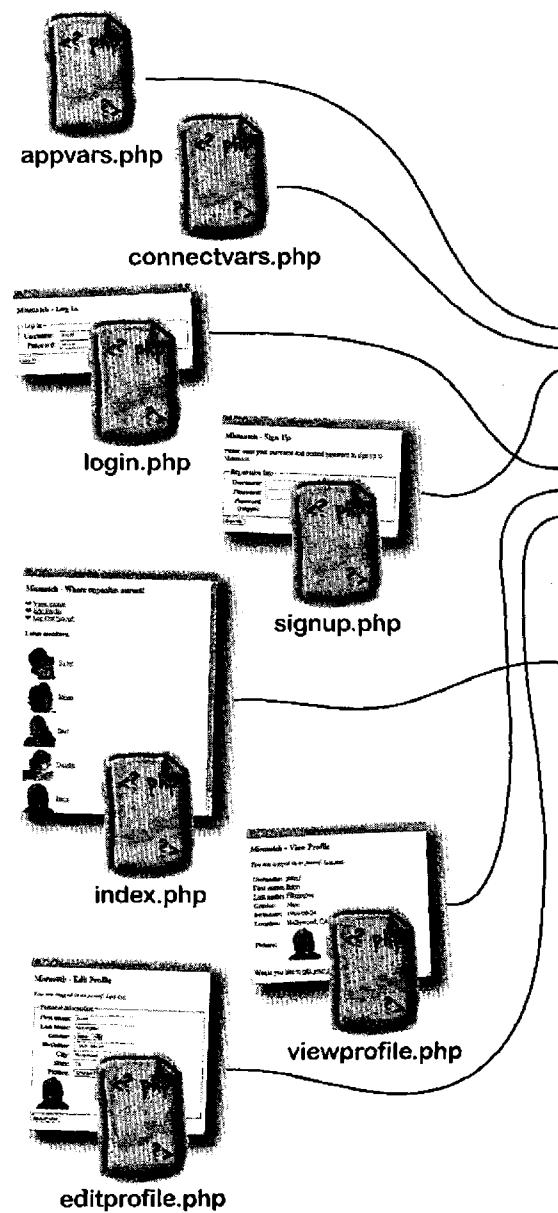
Nenhuma modificação, pois este script não tem influência direta na persistência dos logins.

As sessões precisam se lembrar de quem o usuário é. Chame a função `session_start()` para iniciar a sessão, e depois modifique as referências a `$_COOKIE` para `$_SESSION`.

As sessões precisam controlar o menu de navegação. Chame a função `session_start()` para iniciar a sessão, e depois mude as referências `$_COOKIE` por `$_SESSION`.

# COMO EU MUDO? SOLUÇÃO

A mudança dos cookies para as sessões tem impacto sobre mais coisas do que apenas o script de logout. Ligue cada parte da aplicação Mismatch à modificação por que elas têm de passar para acomodar as sessões.



Nenhuma modificação, pois este script não tem influência direta na persistência dos logins.

As sessões precisam se lembrar de quem o usuário é. Chame a função `session_start()` para iniciar a sessão, e depois modifique as referências a `$_COOKIE` para `$_SESSION`.

As sessões precisam controlar o menu de navegação. Chame a função `session_start()` para iniciar a sessão, e depois mude as referências `$_COOKIE` por `$_SESSION`.



## PONTOS DE BALA

- A autenticação HTTP é útil para se restringir o acesso a determinadas páginas, mas não oferece nenhuma forma de o usuário fazer "logout" após terminar de acessar a página.
- Os cookies lhe permitem armazenar no cliente (web browser) dados de pequeno porte, como por exemplo, os dados de login dos usuários.
- Todos os cookies têm um prazo de validade, que pode vencer num futuro muito distante ou então logo ao final da sessão do navegador.
- Para apagar um cookie, basta definir o seu prazo de validade para vencer em algum tempo no passado.
- As sessões oferecem recurso de armazenamento semelhante ao dos cookies, mas são armazenadas no servidor e, portanto, não ficam sujeitas às mesmas limitações do navegador – como por exemplo, a possibilidade de se desabilitar os cookies.
- As variáveis de sessão têm um tempo de vida limitado, e são sempre destruídas uma vez terminada a sessão (por exemplo, quando o navegador é fechado).

### *não existem* Perguntas Idiotas

**P:** A função `session_start()` é chamada em vários lugares diferentes, mesmo depois de a sessão ter sido iniciadas. Não estamos criando múltiplas sessões com cada uma dessas chamadas a `session_start()`?

**R:** Não. A função `session_start()` não serve apenas para iniciar uma nova sessão – ela também trabalha com sessões existentes. Assim, quando um script chama `session_start()`, a função primeiramente verifica se já existe uma sessão, o que é evidenciado pela presença de uma ID de sessão. Se não existir, a função gera nova ID e cria a nova sessão. Chamadas futuras a `session_start()` vindas de dentro da mesma aplicação irão reconhecer a sessão existente e irão usá-la, em vez de criar outra.

**P:** Como a ID da sessão é armazenada? É nesse ponto que as sessões às vezes usam cookies?

**R:** Sim. Embora os dados de sessões sejam armazenados no servidor e, portanto, tenham os benefícios de serem mais seguros e ficarem fora do controle do navegador, ainda precisa haver um mecanismo para o script saber da existência desses dados.

É para isso que serve a ID – identificar individualmente a sessão e os dados associados com ela. Essa ID precisa

persistir no cliente, de alguma forma, para que várias páginas possam fazer parte da mesma sessão. Uma forma pela qual essa persistência da ID é executada é através de um cookie, significando que a ID é armazenada em um cookie, que é então usado para associar um script a uma determinada sessão.

**P:** Se as sessões ainda dependem dos cookies desse jeito, qual é então a grande vantagem de usá-las em vez dos cookies?

**R:** As sessões não são inteiramente dependentes dos cookies. É importante entender que os cookies servem como uma otimização para se preservar a ID da sessão através de vários scripts, mas não como uma necessidade. Se os cookies estiverem desabilitados, a ID é passada de script para script através de uma URL, de forma semelhante a como já vimos dados serem transmitidos em uma requisição GET. Assim, as sessões podem funcionar perfeitamente sem cookies. Os detalhes sobre como as sessões reagem em resposta aos cookies estarem desabilitados são controlados no arquivo de configuração `php.ini` presente no servidor web, através das configurações de `session.use_cookies`, `session.use_only_cookies` e `session.use_trans_sid`.

**P:** Ainda parece estranho que as sessões possam usar cookies, quando a justificativa em usá-las é justamente que

elas supostamente são melhores que os cookies. Qual é a explicação?

**R:** Embora as sessões de fato ofereçam alguns benefícios claros em relação aos cookies, em determinados cenários, elas não necessariamente têm uma relação mutuamente exclusiva com os cookies. As sessões certamente têm o benefício de ficarem armazenadas no servidor em vez de no cliente, o que as torna mais seguras e confiáveis. Assim, caso você precise armazenar dados sensíveis de forma persistente, uma variável de sessão fornecerá mais segurança do que um cookie. As sessões também são capazes de armazenar quantidades maiores de dados do que os cookies. Portanto, existem vantagens claras em se usar sessões, independentemente de os cookies estarem disponíveis ou não.

Para os propósitos do Mismatch, as sessões oferecem uma conveniente solução para se armazenar dados de login no lado do servidor. Para os usuários que têm os cookies habilitados, as sessões fornecem melhor segurança e confiabilidade, ao mesmo tempo ainda usando os cookies como uma forma de otimização. E no caso dos usuários que não têm cookies habilitados, as sessões funcionam sem serem afetadas, ao passarem a ID da sessão através de uma URL, evitando os cookies completamente.

## Complete a sessão de transformações

Embora as diferentes partes do Mismatch afetadas pelas sessões as usem para realizar coisas diferentes, em última análise os scripts precisam de modificações semelhantes para se fazer a migração dos cookies para as sessões. Por exemplo, todos eles precisam chamar a função `session_start()` para iniciar as sessões. Além disso, todas as modificações envolvem a troca da superglobal `$_COOKIE` pela `$_SESSION`, que é responsável por armazenar as variáveis da sessão.

```
<?php  
    session_start();  
?>
```

Todos os scripts baseados em sessões começam com uma chamada a `session_start()`, para iniciar a sessão.

```
// Se o usuário não está logado, loga-o.  
if (!isset($_SESSION['user_id'])) {  
    if (isset($_POST['submit'])) {  
        // Conecta com o banco de dados  
        $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);  
  
        // Pega o dado do login digitado  
        $user_username = mysqli_real_escape_string($dbc, trim($_POST['username']));  
        $user_password = mysqli_real_escape_string($dbc, trim($_POST['password']));  
  
        if (!empty($user_username) && !empty($user_password)) {  
            // Olha o usuário e senha no banco  
            $query = "SELECT user_id, username FROM mismatch_user WHERE username = '$user_username' AND ".  
                    "password = SHA('$user_password')";  
            $data = mysqli_query($dbc, $query);  
  
            if (mysqli_num_rows($data) == 1) {  
                // O login é OK, defina a variável de sessão ID eo nome do usuário, e redireciona p/ home page.  
                $row = mysqli_fetch_array($data);  
                $_SESSION['user_id'] = $row['user_id'];  
                $_SESSION['username'] = $row['username'];  
                $home_url = 'http://'. $_SERVER['HTTP_HOST'] . dirname($_SERVER['PHP_SELF']) . '/index.php';  
                header('Location: ' . $home_url);  
            }  
            else {  
                // O usuário/senha estão incorretos, seta uma mensagem de erro  
                $error_msg = 'Desculpe, você deve digitar um usuário e senha válidos para fazer login.';  
            }  
        }  
    }  
}
```



login.php

O script de login usa sessões para se lembrar da ID e do nome do usuário, obtendo assim a persistência do login, e o faz usando a superglobal `$_SESSION` em vez da `$_COOKIE`.

```

// Gerar o menu de navegação
if (isset($_SESSION['username'])) {
    echo '<a href="viewprofile.php">View Profile</a><br />';
    echo '<a href="editprofile.php">Edit Profile</a><br />';
    echo '<a href="logout.php">Log Out (' . $_SESSION['username'] . ')</a>';
}
else {
    echo '<a href="login.php">Log In</a><br />';
    echo '<a href="signup.php">Sign Up</a>';
}

...
// Varre o array de dados de usuário formatado para HTML
echo '<h4>Latest members:</h4>';
echo '<table>';
while ($row = mysql_fetch_array($data)) {
    ...
    if (isset($_SESSION['user_id'])) {
        echo '<td><a href="viewprofile.php?user_id=' . $row['user_id'] . '">' . $row['first_name'] . '</a></td></tr>';
    }
    else {
        echo '<td>' . $row['first_name'] . '</td></tr>';
    }
}
echo '</table>';

```

A home page do Mismatch usa a superglobal `$_SESSION` em vez de `$_COOKIE` para acessar dados de login, ao mesmo tempo em que gera o menu e decide se deve ou não exibir um link para os perfis dos membros mais recentes.



Semelhante à página de login e à home page, o script Editar Perfil agora usa `$_SESSION` em vez de `$_COOKIE` para acessar os dados de login.

```

// Garantir que o usuário está logado.
if (!isset($_SESSION['user_id'])) {
    echo '<p class="login">Please <a href="login.php">log in</a> to access this page.</p>';
    exit();
}
else {
    echo '<p class="login">You are logged in as ' . $_SESSION['username'] .
        '<a href="logout.php">Log out</a>.</p>';
}

...
if (!empty($first_name) && !empty($last_name) && !empty($gender) && !empty($birthdate) &&
    !empty($city) && !empty($state)) {
    // Apenas seta a coluna picture se existir uma foto nova
    if (!$new_picture) {
        $query = "UPDATE mismatch_user SET first_name = '$first_name', last_name = '$last_name', " .
            "gender = '$gender', birthdate = '$birthdate', city = '$city', state = '$state' " .
            "picture = '$new_picture' WHERE user_id = '" . $_SESSION['user_id'] . "'";
    }
    else {
        $query = "UPDATE mismatch_user SET first_name = '$first_name', last_name = '$last_name', " .
            "gender = '$gender', birthdate = '$birthdate', city = '$city', state = '$state' " .
            "WHERE user_id = '" . $_SESSION['user_id'] . "'";
    }
    mysqli_query($dbc, $query);
}

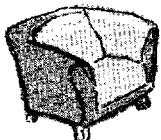
```



Embora não esteja sendo mostrado, o script Ver Perfil usa sessões praticamente da mesma forma que Editar Perfil.

*convera informal entre cookie e variavel de sessão*

## Conversa Informal



**Conversa de hoje: Cookie e variável de sessão discutem quem tem a melhor memória.**

**Cookie:**

Tem havido muita conversa aqui entre nós, cookies, sobre o que exatamente está acontecendo aí no servidor. Os boatos dão conta que você está tentando entrar no nosso território e roubar nossos empregos de armazenamento de dados. Qual é a história?

Não faz nenhum sentido para mim. O navegador é um lugar perfeitamente bom para se armazenar dados, e eu sou o cara perfeito para fazer esse trabalho.

Ahn, bem, essa é outra história. Se o usuário decidir me desabilitar, então claramente ele não tem nenhuma necessidade de armazenar dados.

Então eu suponho que a sua resposta é armazenar os dados no servidor? Que conveniente.

Ok, gênio. Já que você parece ter solução para tudo, por que é que você, às vezes, ainda me usa para armazenar a sua preciosa ID no navegador?



**Variável de sessão:**

Espera aí, roubar é uma palavra muito séria. A verdade é que às vezes simplesmente faz mais sentido armazenar os dados no servidor.

Mas e se o usuário te desabilitar?

Não é verdade. O usuário frequentemente nem sabe que a aplicação web está armazenando dados, porque, em muitos casos, são dados que só ficam nos bastidores, como por exemplo, os nomes de usuário. Assim, se você não estiver disponível, ele acaba sem nada.

Exatamente. E o legal é que o usuário não tem como desabilitar nada no servidor, assim você não precisa se preocupar se os dados poderão ou não realmente ser armazenados.

Er, bem, a maioria das pessoas não sabe disso, portanto não precisamos falar disso aqui. Podemos falar disso fora do microfone. O mais importante é que eu estou sempre alerta, pronto para armazenar dados no servidor.

## **Cookie:**

Não senhor, diga a todos o quanto você precisa de mim!

Ah, eu sei que pode, mas a verdade é que prefere depender de mim. E, no fundo, talvez até goste de mim.

Ah, então agora você vai me tirar só porque eu sou pequeno. Ok; posso não ser capaz de armazenar tanto quanto você, e admito que viver no cliente me torna um pouco menos seguro. Mas com certeza é mais emocionante! E eu tenho algo com que você só pode sonhar.

Bem, todo esse espaço de armazenamento e segurança de que você se orgulha tanto têm um custo... uma vida breve! Eu não queria ter de te dizer, mas toda a sua existência não ultrapassa a duração de uma sessão do navegador. Acho que é daí que vem o seu nome.

É simples. Eu não sou destruído junto com a sessão, apenas termino o meu prazo de validade. Assim, posso ser configurado para viver uma vida longa e próspera, muito mais longa do que o determinado por algum surfista feliz da web que acha bonito abrir e fechar o navegador a cada oportunidade que tem.

O problema é que esses mesmos programadores frequentemente definem o meu prazo de validade com um período tão curto que na verdade eu não chego a experimentar a vida longa que mereço. Quer dizer, eu...

## **Variável de sessão:**

Espera aí, roubar é uma palavra muito séria. A verdade é que às vezes simplesmente faz mais sentido armazenar os dados no servidor.

Olhe, eu não tenho nenhum problema com você. Só gostaria que fosse um pouco mais seguro. E você tem aquela limitação para o tamanho dos dados, que nem sempre são tão pequenos.

Ah, sim? E o que seria?

Quer dizer que você pode viver além de uma única sessão? Como isso é possível?!

Uau. A imortalidade deve ser sensacional. A minha única esperança é que algum programador preguiçoso acidentalmente se esqueça de me destruir ao fechar a sessão... mas o navegador ainda vai acabar comigo assim que for fechado.

Alô? Você está aí? Caramba, esse negócio de prazo de validade é sério mesmo.

**test drive the "sessionalizado" mismatch**



## TEST DRIVE

### Modifique o Mismatch para usar sessões em vez de cookies.

Modifique os scripts do Mismatch de modo que eles passem a usar sessões, em vez de cookies, para implementar a persistência do login (ou baixe os scripts do site da Alta Books, [www.altabooks.com.br](http://www.altabooks.com.br)). Isso envolverá modificações nos scripts index.php, login.php, logout.php, editprofile.php e viewprofile.php, principalmente para iniciar a sessão com uma chamada à função session\_start() e trocar as referências à superglobal \$\_COOKIE por referências a \$\_SESSION.

Envie os scripts para o seu servidor web e depois abra a página principal do Mismatch (index.php) em um navegador web. Tente fazer login e logout para se certificar de que tudo funciona como antes. A não ser que tivesse os cookies desabilitados antes, você não deverá perceber nenhuma diferença – isso é bom!

Muito legal. É bom poder fazer login mesmo sem ter os cookies habilitados.

Mismatch - Where opposites attract!

members:

- Ruby
- Johan
- Paul
- Dierdre
- Jason

Mismatch - Log In

User:  Password:

Mismatch - Edit Profile

You are logged in as jason. [Log out](#).

Personal Information

First name:	Jason
Last name:	Filmington
Gender:	<input checked="" type="radio"/> Male <input type="radio"/> Female
Birthdate:	1985-09-24
City:	Hollywood
State:	CA
Picture:	<input type="file" value="Choose file"/> no file selected

[Save Profile](#)

Gracias às sessões, os usuários que desabilitaram os cookies ainda podem fazer login e acessar os seus perfis.



**As sessões sem cookies poderão não funcionar se php.ini não estiver devidamente configurado no servidor.**

**Veja bem!**

Para as sessões poderem funcionar com os cookies desabilitados, precisa haver outro mecanismo para se enviar a ID da sessão de uma página para outra. Esse mecanismo envolve a anexação da ID à URL de cada página, o que ocorre automaticamente caso session.use\_trans\_id esteja definida como 1 (true) no arquivo php.ini do servidor. Se você não tiver a capacidade de alterar esse arquivo no seu servidor web, terá de anexar manualmente a ID à URL das páginas da sessão, caso os cookie estejam desabilitados, usando um código como este:

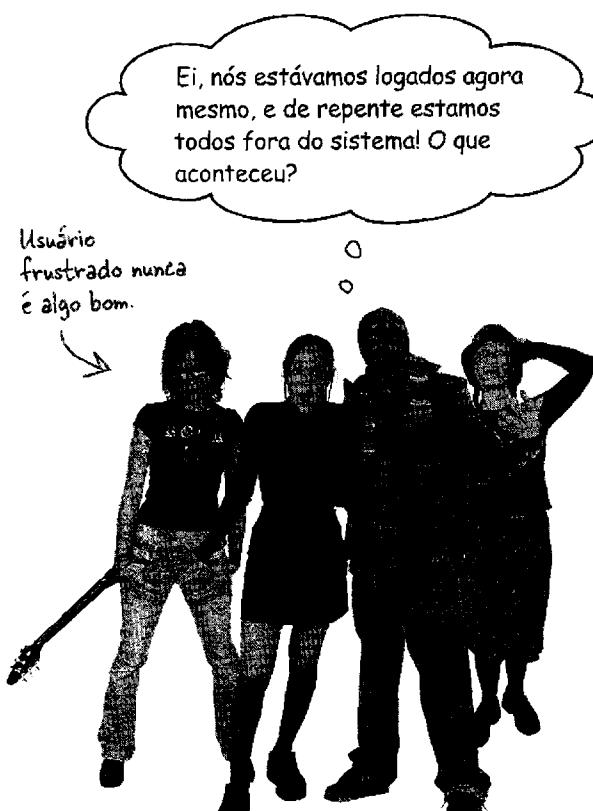
```
<a href="viewprofile.php?<?php echo $ID; ?>">view your profile</a>
```

A superglobal \$ID armazena a ID da sessão, a qual é enviada através da URL para que a página Editar Perfil saiba da existência da sessão.

*porque o logout automático?*

## Os usuários não estão se sentindo bem-vindos

Apesar de ser um bom aprimoramento em relação aos cookies, algo não está muito certo com a nova aplicação Mismatch baseada em sessões. Vários usuários relataram que estão sendo deslogados do aplicativo apesar de não terem clicado no link "Log Out". A aplicação não parece mais tão amistosa... isso é um grande problema.



Os usuários estão sendo deslogados do Mismatch mesmo sem ter clicado no link "Log Out".



Esta não é a mensagem que queremos que o Mismatch envie para os seus usuários.

A screenshot of the Mismatch application's home page. At the top, there is a header with the text "Mismatch - Where opposites attract!". Below the header, there are two buttons: "Log In" and "Sign Up". To the right, there is a section titled "Latest members:" featuring five user profiles with their names: Ruby, John@omo, Paul, Diane, and Jason. The text next to the profiles states: "A home page é apresentada aos usuários registrados. John@omo se eles fossem visitantes não cadastrados, sem que eles tenham feito logout." (The home page is presented to registered users. John@omo if they were visitors not registered, without them having to do a logout).



## PODER DO CÉREBRO

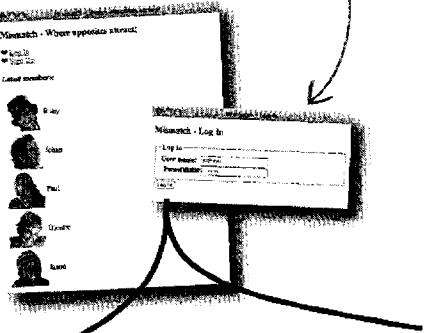
O que você acha que está fazendo os usuários serem deslogados automaticamente do Mismatch? Será algo que eles estão fazendo sem saber?

## *o tempo de vida dos cookies e sessões*

# As sessões têm vida curta...

O problema com os logouts automáticos no Mismatch tem a ver com o tempo de vida limitado das sessões. Como você deve se lembrar, as sessões só duram pelo tempo em que a instância atual do navegador permanece aberta, significando que todas as variáveis da sessão são eliminadas quando o usuário fecha o navegador. Em outras palavras, fechar o navegador resulta em o usuário ser deslogado, independentemente de ele gostar disso ou não. Isso não só é inconveniente, como também é um pouco confuso, uma vez que já temos um recurso de logout. Os usuários presumem que não sairão do sistema a não ser que cliquem deliberadamente no link Log Out.

Esteja você usando sessões ou cookies, o processo de login é o que inicia o modo persistente na aplicação.



O login com sessões resulta na criação de duas variáveis de sessão.

As variáveis de sessão são usadas para se memorizar a identidade do usuário.

O usuário fecha o navegador, mas pode não perceber que acabou de se deslogar do sistema.

As variáveis são destruídas junto com a sessão, quando o navegador é fechado.

Uma vez destruídas as variáveis da sessão, o usuário é deslogado... quer queira, quer não!

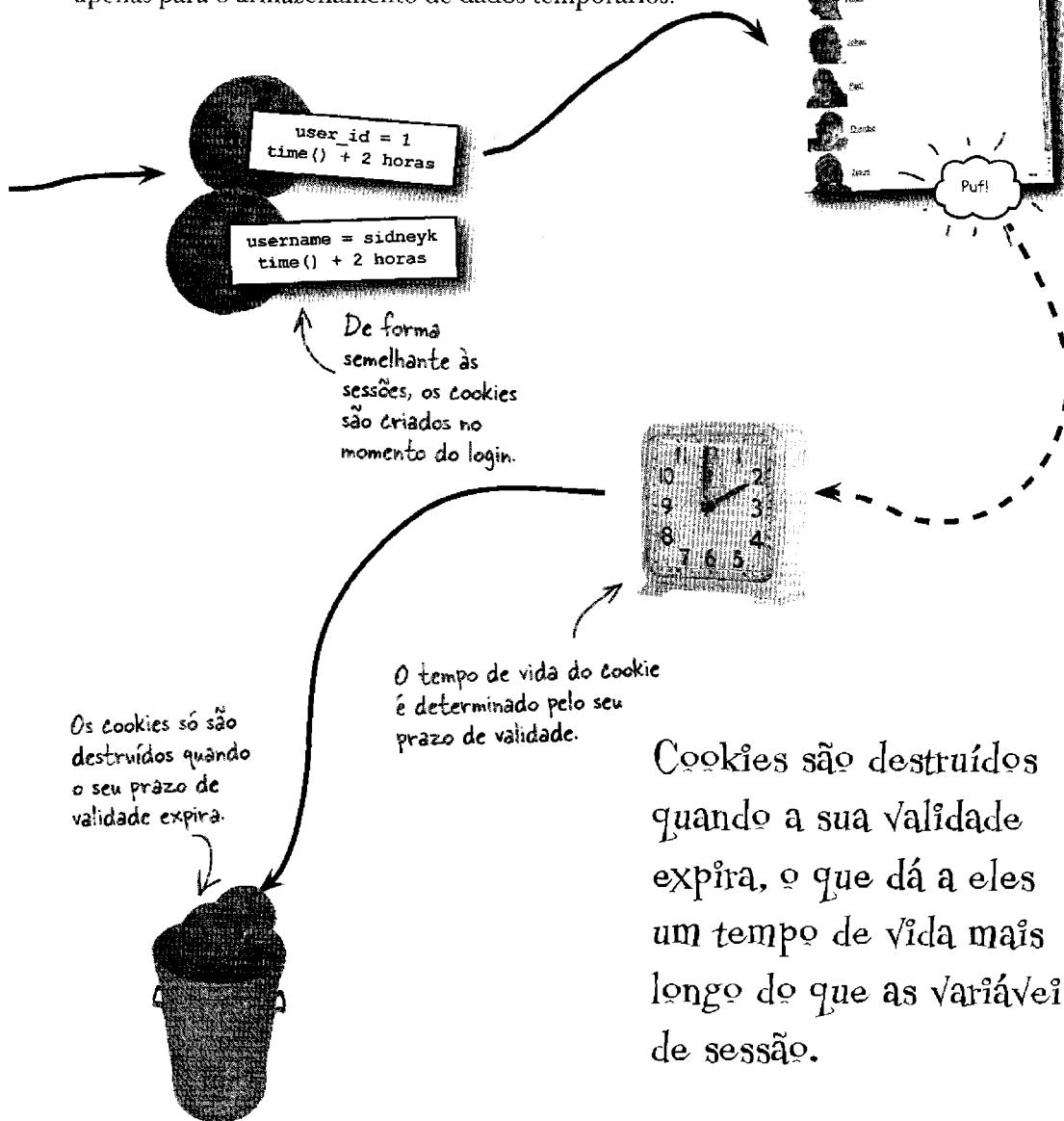
Embora você possa destruir uma sessão ao terminar de usá-la, não pode prolongar a sua existência para além da instância do navegador. Assim, as sessões são uma solução de armazenamento de prazo mais curto do que os cookies, uma vez que estes podem ter uma data de validade definida para daqui a horas, dias, meses ou anos. Isso significa que as sessões são inferiores aos cookies? Não, de jeito nenhum. Mas significa que as sessões apresentam um problema caso você precise memorizar informações para além de uma instância do navegador... como é o caso dos dados de login!

As variáveis de sessão são destruídas quando o usuário finaliza a sessão fechando o navegador.

## ...mas os cookies podem durar para sempre!

Ao contrário das variáveis de sessão, o tempo de vida de um cookie não fica limitado à instância do navegador; assim, os cookies podem continuar vivendo, pelo menos até vencer o seu prazo de validade. O problema é que os usuários têm a capacidade de destruir todos os cookies armazenados na sua máquina com uma simples configuração do navegador, portanto, não fique muito contente com a permanência dos cookies – de uma forma ou de outra, eles ainda se prestam apenas para o armazenamento de dados temporários.

Talvez não para sempre, mas tempo suficiente para durar mais do que a sessão.



## *usando cookies e sessões em paralelo*

Então faria sentido usar tanto sessões quanto cookies, sendo que esses ajudariam a manter os usuários logados por períodos de tempo mais longos? Isso funcionaria para os usuários que têm os cookies habilitados.

Desde que não esteja lidando com dados altamente sensíveis, em cujo caso a fraude de segurança dos cookies seria um argumento para se recomendar o uso apenas das sessões.

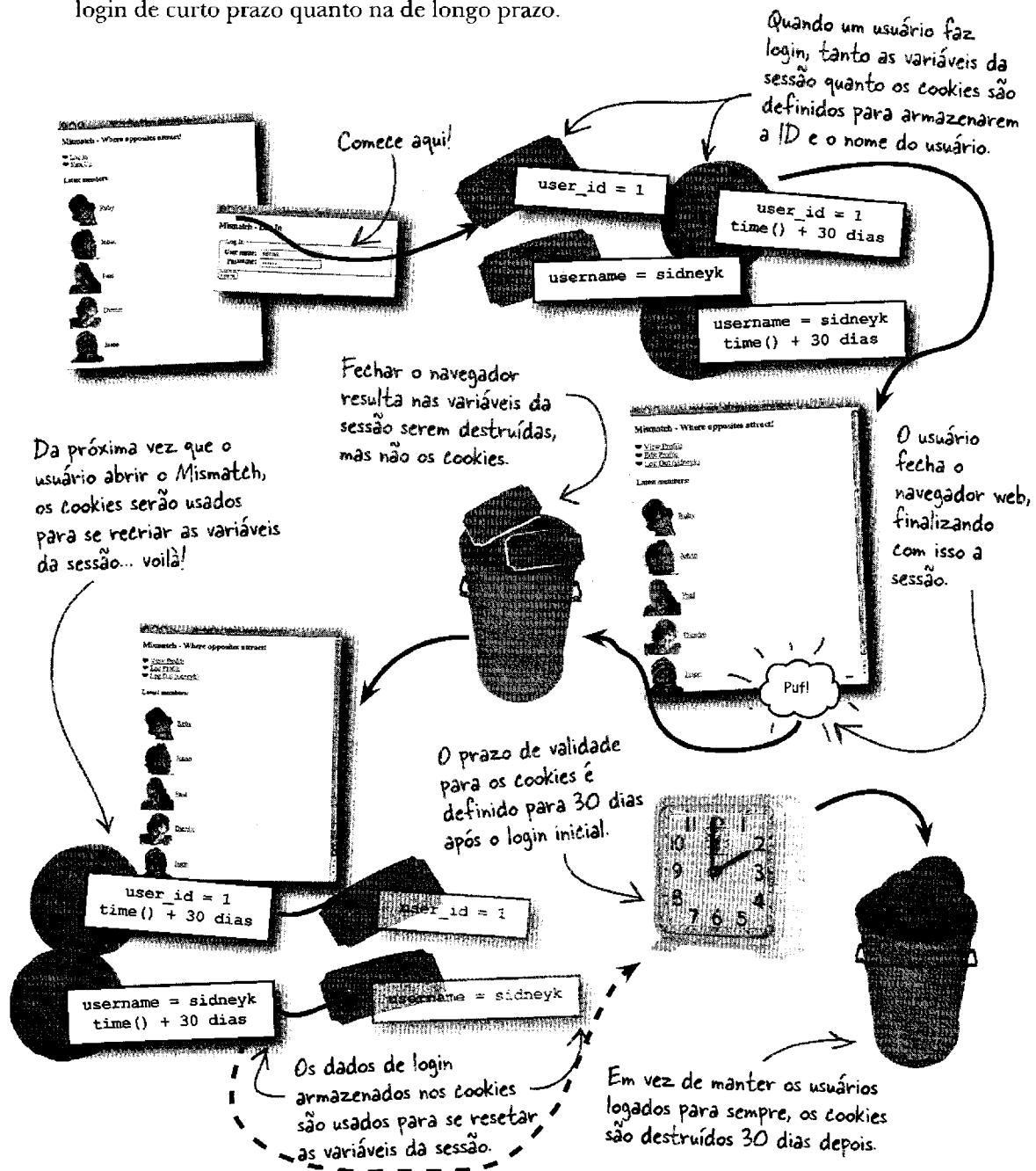
**Sim, não há nada errado em tirar proveito das vantagens das sessões e dos cookies para tornar os logins do Mismatch mais flexíveis.**

Na verdade, isso pode ser bastante útil. As sessões são melhores para a persistência de curto prazo, uma vez que elas são mais amplamente suportadas e não se limitam pelo navegador, enquanto que os cookies permitem que você mantenha os dados de login memorizados por um período de tempo mais longo. Certamente que nem todo mundo poderá se beneficiar do uso dos cookies, mas será benéfico para muitas pessoas, o que justifica esse uso. Sempre que puder aprimorar a experiência de usuário para uma parte significativa da sua base de usuários sem piorar para os outros, é bom fazê-lo.



# Sessões + Cookies = Persistência de login superior

Para a melhor persistência de login possível, você precisa ser criativo e combinar tudo o que aprendeu neste capítulo para tirar proveito dos benefícios tanto das sessões quanto dos cookies. Ao fazer isso, pode reestruturar a aplicação Mismatch de modo que ela tenha o melhor desempenho possível tanto na persistência de login de curto prazo quanto na de longo prazo.



## *não existem* Perguntas Idiotas

**P:** Então a razão para se optar entre sessões e cookies é a questão persistência de curto x de longo prazo?

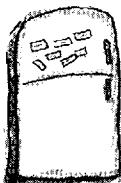
**R:** Não. Essa por acaso acabou sendo a estratégia que ajudou a guiar o design da aplicação Mismatch, mas cada aplicação é diferente, e existem outros aspectos das sessões e dos cookies que frequentemente precisam ser ponderados. Por exemplo, os dados armazenados em uma sessão são mais seguros do que aqueles armazenados em um cookie. Assim, mesmo que os cookies estejam habilitados e um deles esteja sendo usado unicamente para manter registro da ID da sessão, os dados armazenados na sessão propriamente dita são mais seguros do que se estivessem armazenados diretamente em um cookie. O motivo é que os dados das sessões são armazenados no servidor, o que torna bastante difícil serem acessados por usuários que não tenham os devidos privilégios. Assim, se você estiver lidando com dados que precisem ser seguros, as sessões são mais apropriadas que os cookies.

**P:** E quanto ao tamanho dos dados? Isso tem alguma influência?

**R:** Sim, o tamanho dos dados também faz diferença. As sessões são capazes de armazenar pedaços de dados maiores do que os cookies conseguem, portanto essa é outra razão para preferir as sessões caso você precise armazenar dados maiores do que algumas strings de texto simples. É claro que um banco de dados MySQL é ainda melhor para armazenar dados muito grandes, portanto cuidado para não se animar demais ao trabalhar com sessões.

**P:** Então por que eu deveria optar por uma sessão ou um cookie em vez de um banco de dados MySQL?

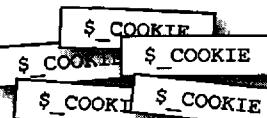
**R:** Conveniência. Armazenar dados em um banco demanda muito mais esforço, e não se esqueça de que os bancos são mais adequados para armazenar dados permanentes. Dados de login não são realmente permanentes, num contexto mais amplo. É aí que os cookies e as sessões entram em jogo – eles são melhores para dados que você precisa ter memorizados por um curto período de tempo, e depois jogados fora.



# Ímas de Geladeira do PHP

criando aplicações web personalizadas

A aplicação Mismatch foi reelaborada para usar sessões e cookies ao mesmo tempo, com o objetivo de conseguir a melhor persistência de login possível. O problema é que parte do código está faltando. Use os ímas de sessões e de cookies para completar o código.



```
.....
if (mysqli_num_rows($data) == 1) {
    // Login OK, definir as vars de sessão (e os cookies) de ID e nome do usuário,
    // e depois redirecionar para a home page
    $row = mysqli_fetch_array($data);

        ['user_id'] = $row['user_id'];
.....
        ['username'] = $row['username'];
.....

```



login.php

```
setcookie('user_id', $row['user_id'], time() + (60 * 60 * 24 * 30)); // expira em 30 dias
setcookie('username', $row['username'], time() + (60 * 60 * 24 * 30)); // expira em 30 dias
$home_url = 'http://'. $_SERVER['HTTP_HOST'] . dirname($_SERVER['PHP_SELF']) . '/index.php';
header('Location: ' . $home_url);
}
...

```

```
<?php
// Se o usuário estiver logado, apagar as vars de sessão para fazer o logout
session_start();
if (isset( ..... ['user_id'])) {

    // Apaga as vars de sessão limpando o array $_SESSION
    ..... = array();

    // Apaga o cookie de sessão, definindo o seu prazo de validade como uma hora atrás (3600)
    if (isset( ..... [session_name()])) {

        setcookie(session_name(), '', time() - 3600);
    }
    // Destroi a sessão
    session_destroy();
}
// Apaga os cookies de ID e de nome do usuário, definindo os seus prazos de validade como uma hora atrás (3600)
setcookie('user_id', '', time() - 3600);
setcookie('username', '', time() - 3600);
...

```

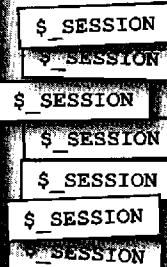


logout.php

```
<?php
session_start();

// Se as vars de sessão não estiverem definidas, tentar defini-las com um cookie
if (!isset( ..... ['user_id'])) {
    if (isset( ..... ['user_id']) && isset( ..... ['username'])) {
        ..... ['user_id'] = ..... ['user_id'];
        ..... ['username'] = ..... ['username'];
    }
}
?>
...

```



index.php

# Ímas de Geladeira do PHP - Solução



A aplicação Mismatch foi reelaborada para usar sessões e cookies ao mesmo tempo, com o objetivo de conseguir a melhor persistência de login possível. O problema é que parte do código está faltando. Use os ímas de sessões e de cookies para completar o código.

```
...  
if (mysqli_num_rows($data) == 1) {  
    // Login OK, definir as vars de sessão (e os cookies) de ID e nome do usuário,  
    // e depois redirecionar para a home page  
    $row = mysqli_fetch_array($data);  
  
    $_SESSION ['user_id'] = $row['user_id'];  
    $_SESSION ['username'] = $row['username'];  
  
    setcookie('user_id', $row['user_id'], time() + (60 * 60 * 24 * 30)); // expira em 30 dias  
    setcookie('username', $row['username'], time() + (60 * 60 * 24 * 30)); // expira em 30 dias  
    $home_url = "http://" . $_SERVER['HTTP_HOST'] . dirname($_SERVER['PHP_SELF']) . '/  
index.php';  
    header('Location: ' . $home_url);  
}  
...  
?
```

Os novos cookies são definidos, além das variáveis da sessão.

```
<?php  
// Se o usuário estiver logado, apagar as vars de sessão para fazer o logout  
session_start();  
if (isset($_SESSION ['user_id'])) {  
    // Apaga as vars de sessão limpando o array $_SESSION  
    ...  
    $_SESSION = array();  
  
    // Apaga o cookie de sessão, definindo o seu prazo de validade como uma hora atrás (3600)  
    if (isset($_COOKIE [session_name()])) {  
        setcookie(session_name(), '', time() - 3600);  
    }  
    // Destroi a sessão  
    session_destroy();  
}  
// Apaga os cookies de ID e de nome do usuário, definindo os seus prazos de validade como  
uma hora atrás (3600)  
setcookie('user_id', '', time() - 3600);  
setcookie('username', '', time() - 3600);  
...  
?
```

Agora, para se fazer o logout, é preciso apagar tanto o cookie da sessão quanto os novos cookies de login.

Caso o usuário não esteja logado através da sessão, verificar se os cookies estão definidos.

```
<?php  
session_start();  
  
// Se as vars de sessão não estiverem definidas, tentar defini-las com um cookie  
  
if (!isset($_SESSION ['user_id'])) {  
    if (isset($_COOKIE ['user_id']) & isset($_COOKIE ['username'])) {  
        $_SESSION ['user_id'] = $_COOKIE ['user_id'];  
        $_SESSION ['username'] = $_COOKIE ['username'];  
    }  
}
```

Defina as variáveis da sessão usando os cookies.

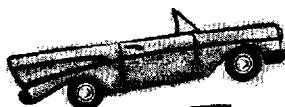
O mesmo código de index.php cookies/sessões deve ser colocado em editprofile.php e viewprofile.php.

login.php

logout.php

editprofile.php

viewprofile.php

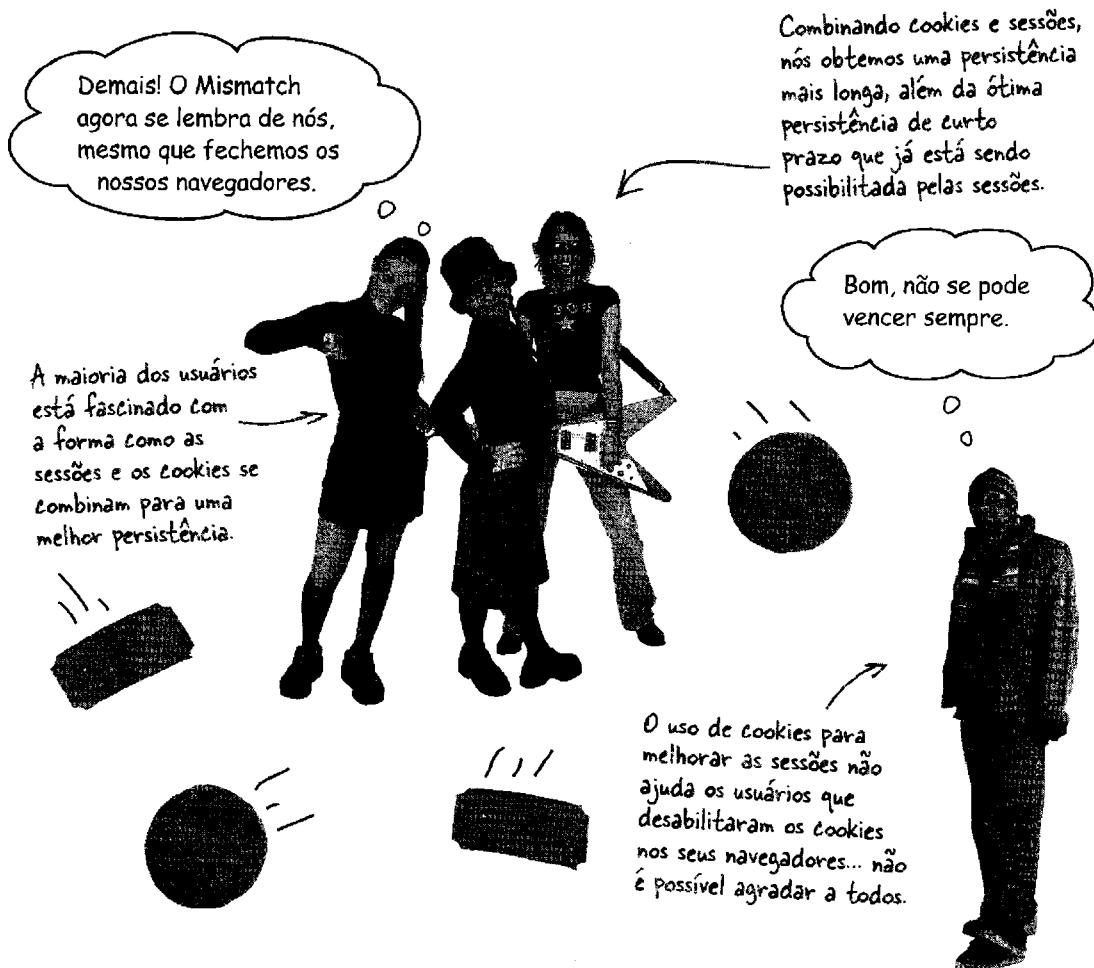


## TEST DRIVE

### Modifique o Mismatch para usar sessões e cookies.

Modifique os scripts do Mismatch de modo que eles usem tanto as sessões quanto os cookies para implementar a persistência do login (ou baixe os scripts no site da Alta Books, [www.altabooks.com.br](http://www.altabooks.com.br)). Será preciso modificar os scripts index.php, login.php, logout.php, editprofile.php e viewprofile.php.

Envie os scripts para o seu servidor web e depois abra a página principal do Mismatch (index.php) em um navegador. Tente fazer login e depois fechar o navegador, o que fará com que as variáveis da sessão sejam destruídas. Abra novamente a página principal e verifique se você ainda está logado – os cookies tornam isso possível, uma vez que eles persistem para além de uma sessão do navegador.





## Sua Caixa de Ferramentas do PHP & MySQL

Você aprendeu bastante sobre como criar um sistema de gerenciamento dos usuários para fazer parte da aplicação Mismatch. Vamos recapitular alguns dos pontos principais.

### `setcookie()`

Esta função interna do PHP é usada para se configurar um cookie no navegador, incluindo um prazo de validade opcional, após o qual o cookie é destruído. Se não for fornecido nenhum prazo, o cookie é apagado assim que o navegador se fechar.

### `$_COOKIE`

Esta superglobal interna do PHP é usada para se acessar dados dos cookies. Trata-se de um array, e cada cookie é armazenado como uma entrada no array. Assim, para se acessar o valor de um cookie, é preciso especificar o seu nome, na forma de índice do array.

### `session_start()`

Esta função interna do PHP inicia uma nova sessão ou reinicia uma já existente. Você precisa chamar esta função antes de acessar quaisquer variáveis da sessão.

### `SHA1($valor)`

Esta função do MySQL encriptografa um texto, resultando em uma string de 40 caracteres hexadecimais. Esta função fornece uma ótima maneira de se criptografar dados que precisem se manter irreconhecíveis mesmo dentro do banco de dados. É, porém, um processo sem volta, significando que não existe uma função de "descriptografar".

### `session_destroy()`

Esta função interna do PHP fecha uma sessão, e deve ser chamada sempre que você tiver terminado de usar uma determinada sessão. Esta função não destrói as variáveis da sessão, porém, é importante eliminá-las manualmente, esvaziando a superglobal `$_SESSION`.

### `$_SESSION`

Esta superglobal interna do PHP é usada para se acessar dados da sessão. Trata-se de um array, e cada variável da sessão é armazenada como uma entrada no array. Assim, para se acessar o valor de uma variável, é preciso especificar o seu nome, na forma de índice do array.



## QUEM FAZ O QUE?

Alguém bagunçou várias partes do código da aplicação Mismatch e não conseguimos nos lembrar do que elas fazem. Desenhe linhas conectando cada pedaço de código com o que ele faz.

### Código PHP/MySQL

```
empty($_COOKIE['user_id'])
```

```
setcookie(session_name(), '', time() - 3600);
```

```
SHA('$user_password')
```

```
session_destroy()
```

```
setcookie('user_id', $row['user_id'])
```

```
$_SESSION = array()
```

```
session_start()
```

```
isset($_SESSION['user_id'])
```

### Descrição

Usa uma variável de sessão para determinar se o usuário está logado ou não.

Use um cookie para determinar se o usuário está logado ou não.

Destrói um cookie de sessão, definido o seu prazo de validade como vencendo uma hora atrás.

Criptografa a senha do usuário, colocando-a em um formato irreconhecível.

Armazena a ID do usuário em um cookie.

Inicia uma sessão.

Fechá a sessão atual.

Destrói todas as variáveis de sessão.

## QUEM FAZ O QUE?

### Solução

Alguém bagunçou várias partes do código da aplicação Mismatch e não conseguimos nos lembrar do que elas fazem. Desenhe linhas conectando cada pedaço de código com o que ele faz.

<b>PHP/MySQL Code</b>	<b>Description</b>
<code>empty(\$_COOKIE['user_id'])</code>	Usa uma variável de sessão para determinar se o usuário está logado ou não.
<code>setcookie(session_name(), '', time() - 3600);</code>	Use um cookie para determinar se o usuário está logado ou não.
<code>SHA('\$user_password')</code>	Destroi um cookie de sessão, definido o seu prazo de validade como vencendo uma hora atrás.
<code>session_destroy()</code>	Criptografa a senha do usuário, colocando-a em um formato irreconhecível.
<code>setcookie('user_id', \$row['user_id'])</code>	Armazena a ID do usuário em um cookie.
<code>\$_SESSION = array()</code>	Inicia uma sessão.
<code>session_start()</code>	Fechá a sessão atual.
<code>isset(\$_SESSION['user_id'])</code>	Destroi todas as variáveis de sessão.

## 7 ½ Elimine Código Duplicado



### Compartilhar é Cuidar



**Guarda-chuvas não são a única coisa que pode ser compartilhada.** Em qualquer aplicação web, você poderá encontrar situações em que um mesmo código encontra-se duplicado em mais de um lugar. Isso não só é um desperdício de recursos, como também pode levar a problemas de manutenção, uma vez que inevitavelmente acabará fazendo modificações, e estas terão de ser efetuadas em mais de um lugar. A solução é eliminar o código duplicado, compartilhando-o. Em outras palavras, você mantém o código duplicado em apenas um lugar, e depois referencia esse código sempre que precisar dele. A eliminação de código duplicado resulta em aplicativos mais eficientes, de manutenção mais fácil e finalmente mais robustos.

*localizar o código duplicado*



A aplicação Mismatch evoluiu ainda mais desde a última vez que você a viu, com uma navegação aprimorada e uma aparência mais consistente. Mas essas melhorias acarretaram um custo... código duplicado. Apenas olhando as próprias páginas, veja se você consegue descobrir quais partes do Mismatch poderiam estar representando um problema de código duplicado. Circule e escreva comentários nessas partes da aplicação, e escreva também qualquer coisa não visível que você ache que também possa estar tendo problemas com código duplicado.

Mismatch - Where opposites attract!

[Home](#) • [View Profile](#) • [Edit Profile](#) • [Log Out \(Profile\)](#)

Latest members:

Member
Ruby
Johan
Paul
Dierdre
Jason

Copyright © 2009 Mismatch Enterprises, Inc.

index.php

**Mismatch - View Profile**

[Home](#) [View Profile](#) [Edit Profile](#) [Log Out \(jnetties\)](#)

Username: jnetties  
First name: Johan  
Last name: Netties  
Gender: Male  
Birthdate: 1981-11-03  
Location: Athens, GA

Picture: 

Would you like to [edit your profile?](#)

Copyright © 2009 Mismatch Enterprises, Inc.



**viewprofile.php**

**Mismatch - Edit Profile**

[Home](#) [View Profile](#) [Edit Profile](#) [Log Out \(jnetties\)](#)

Personal Information

First name:	Last name:
johan	Netties
name:	name:
Gender:	Male
Birthdate: 1981-11-03	
City: Athens	
State: GA	
Picture:	<input type="file"/> <input checked="" type="checkbox"/> johannes.jpg



Copyright © 2009 Mismatch Enterprises, Inc.



**editprofile.php**

## Exercício Solução

A aplicação Mismatch evoluiu ainda mais desde a última vez que você a viu, com uma navegação aprimorada e uma aparência mais consistente. Mas essas melhorias acarretaram um custo... código duplicado. Apenas olhando as próprias páginas, veja se você consegue descobrir quais partes do Mismatch poderiam estar representando um problema de código duplicado. Circule e escreva comentários nessas partes da aplicação, e escreva também qualquer coisa não visível que você acha que também possa estar tendo problemas com código duplicado.

**O título "Mismatch"**  
aparece em todas as páginas, com apenas o título detalhado variando de página para página.

**index.php**

**O menu de navegação é idêntico em todas as três páginas.**

**O rodapé da página, que contém as informações de copyright da aplicação, é sempre o mesmo.**

**Todas as páginas que dependem do login do usuário requerem exatamente o mesmo código de inicialização da sessão e de verificação do login.**

**viewprofile.php**

**editprofile.php**

```

<?php
session_start();
// If the session was not set yet, try to set them with a cookie
if (!isset($_SESSION['user_id'])) {
    if (isset($_COOKIE['user_id'])) {
        $_SESSION['user_id'] = $_COOKIE['user_id'];
        $_SESSION['username'] = $_COOKIE['username'];
    }
}

```

# O Mismatch está em pedaços

Portanto, a aplicação Mismatch tem alguns elementos comuns que estão repetidos nos principais scripts, no momento. Por que isso é tão ruim? Porque torna a manutenção da aplicação mais difícil. O que acontecerá se você decidir adicionar uma nova página, que requeira um novo item de menu? Você terá de modificar o código do menu em cada um dos scripts, para exibir o novo item. O mesmo se aplica ao aviso de copyright.

A solução para o problema é só armazenar cada informação apenas uma vez. Então, se algum dia o código precisar ser modificado, você só o modifica em um lugar. Com isso em mente, é possível repensar a organização do Mismatch em termos de componentes reutilizáveis.

## O cabeçalho das páginas



header.php

O script header.php contém o título da página, o qual referencia uma variável para apresentar um título diferente em cada página. O cabeçalho também inclui código HTML padrão e cuida de tarefas tais como vincular a folha de estilo CSS.

Este componente não resulta em código HTML visível, mas ele tem um papel vital no gerenciamento dos logins dos usuários em toda a aplicação Mismatch.

## O menu de navegação



navmenu.php

O script navmenu.php gera um menu de navegação para a aplicação, baseado em se o usuário está logado ou não. O menu de navegação apresenta links "Log In" e "Log Out" conforme necessário.

## O inicializador de sessão



startsession.php

O script startsession.php é responsável por iniciar a sessão e verificar se o usuário está logado.

## O rodapé das páginas



footer.php

O script footer.php exibe um aviso de copyright para a aplicação e fecha as tags HTML abertas no cabeçalho. Assim, o cabeçalho e o rodapé trabalham em conjunto, tendo sempre de ser usados juntos.

*mismatch precisa de um template*

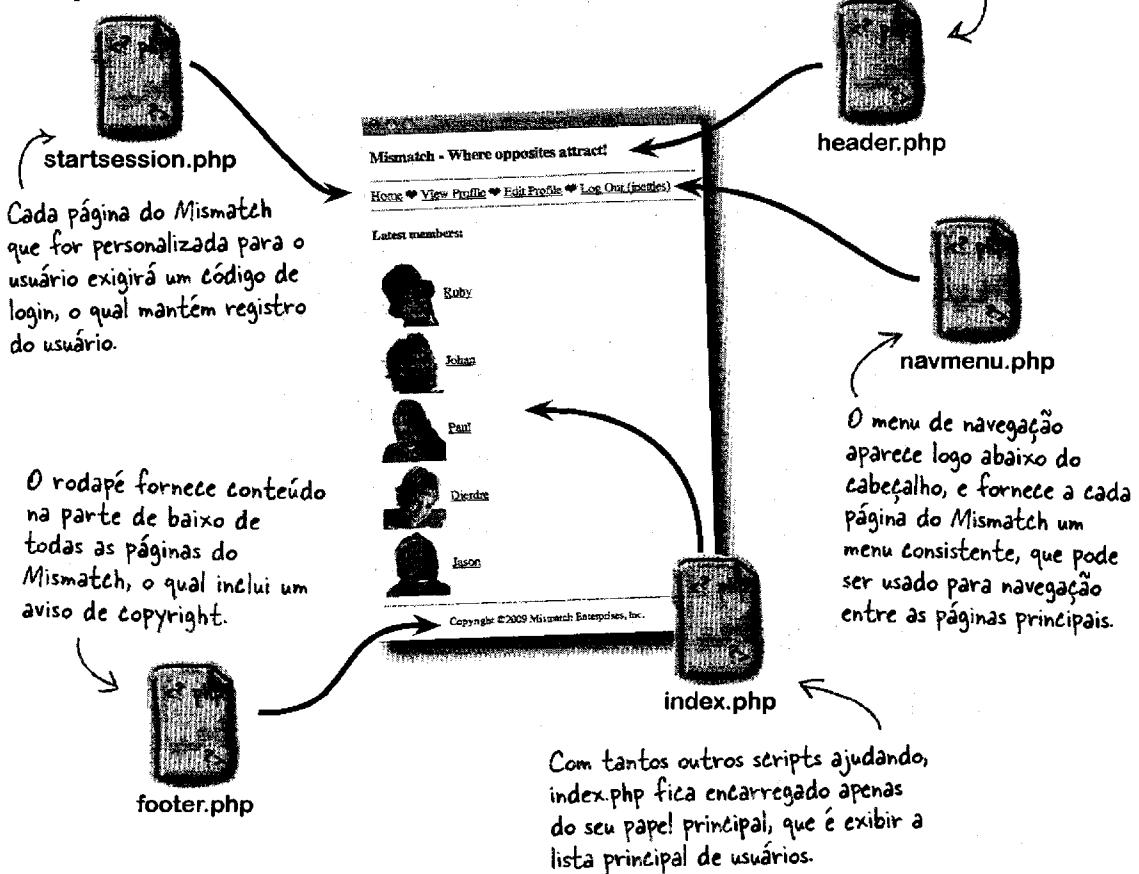
## Reconstruindo o Mismatch a partir de um template

Ok, então nós dividimos os Mismatch em vários scripts, mas como fazemos para eles funcionarem bem juntos? Você já está familiarizado com o conceito de arquivos incluídos, e estes fazem parte da solução. Mas você tem que pensar mais alto do que apenas incluir arquivos... você tem que pensar em termos de templates ("modelos"), os quais lhe permitem criar uma única página como a combinação de vários arquivos incluídos. O templates é como uma planta para as páginas da aplicação, onde todo o conteúdo, exceto aquele que é realmente único à página em questão, vem de arquivos incluídos.

A versão do Mismatch com templates requer que coloquemos os códigos comuns em scripts que desempenhem papéis bastante específicos: alguns responsáveis por gerar código HTML visual, outros por alguma outra função. A ideia é destilar o máximo possível de funcionalidades comuns em arquivos a serem incluídos e então só deixar em cada página o código que seja completamente único a ela.

Os templates  
permitem que uma  
aplicação PHP seja  
construída a partir  
de componentes de  
script reutilizáveis.

O cabeçalho aparece no alto  
de cada página do Mismatch,  
e exibe o título da aplicação  
juntamente com um título  
específico da página.



não existem  
**Perguntas Idiotas**

**P:** O que é exatamente um template? Ele não é apenas um monte de arquivos incluídos?

**R:** Sim. Um templates é um conjunto de arquivos incluídos, mas é um conjunto elaborado especificamente para dividir uma aplicação em componentes funcionais. O objetivo é reduzir cada página ao que é realmente específico a ela, e somente a ela. Assim cabeçalhos, rodapés, menus de navegação e quaisquer outras partes da aplicação que sejam as mesmas, ou semelhantes, em mais de uma página são candidatos ideais para inclusão em um templates. O resultado final é que você coloca o código dos templates em arquivos PHP que serão referenciados por outros scripts que precisem deles.

Você pode pensar no templates como um grupo de arquivos que vão além de apenas reduzir a quantidade de código duplicado – eles ajudam a organizar as funcionalidades de uma aplicação. O Mismatch é um exemplo relativamente simples de como usar os templates – aplicações PHP maiores e mais complexas, frequentemente empregam sistemas de templates bastante sofisticados.

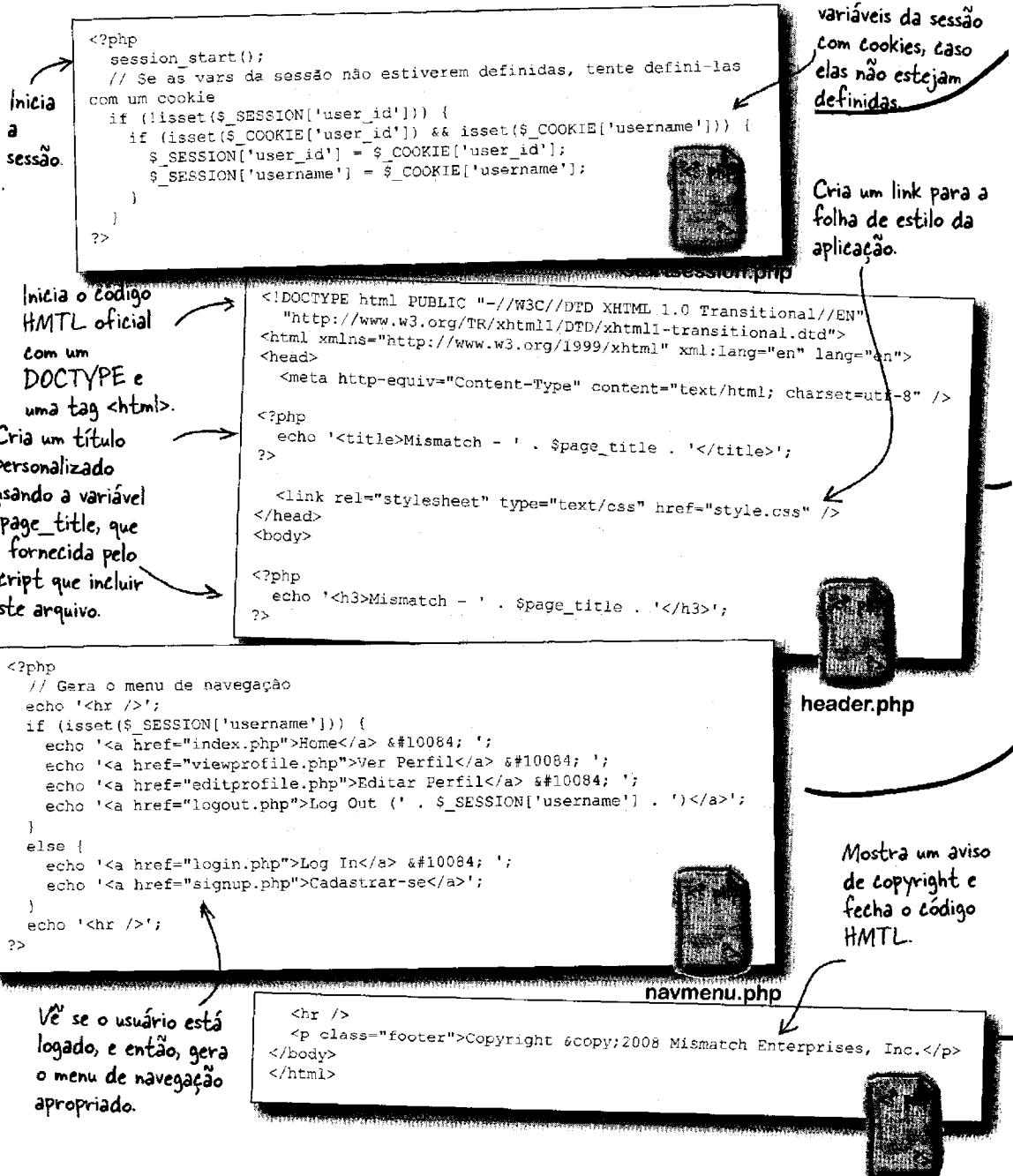
**P:** O código do template não tem de ser exatamente o mesmo para poder ser compartilhado entre vários scripts?

**R:** Não. É perfeitamente aceitável que o código do templates seja apenas semelhante, e não exatamente igual. O motivo é que você pode usar variáveis para permitir algum grau de personalização ao aplicar o script a diferentes páginas. O título das páginas no Mismatch é um exemplo perfeito disso. O modelo do cabeçalho é semelhante em cada página, no sentido que o título sempre começa com "Mismatch – ". Mas o título específico é diferente, e é por isso que precisamos de uma variável para fornecer um meio de mudar ligeiramente o título em cada página.

*mismatch—agora usando templates!*

## Reconstrua o Mismatch com templates

O trabalho de design envolvido em se dividir uma aplicação em templates geralmente é recompensador. No fim das contas, você terá um conjunto de scripts pequenos e bastante focados, bem como terá um código radicalmente simplificado nos scripts principais da aplicação, os quais agora dependem dos scripts templates.



O script `startsession.php` precisa ser o primeiro a ser incluído, para que a sessão seja iniciada e o restante do script tenha acesso aos seus dados.

A variável `$page_title` determina o título da página que é exibido dentro do cabeçalho.

```

<?php
    // Inicia a sessão
    require_once('startsession.php');

    // Insere o cabeçalho da página
    $page_title = 'Onde os opostos se atraem!';
    require_once('header.php');

    require_once('appvars.php');
    require_once('connectvars.php');

    // Mostra o menu de navegação
    require_once('navmenu.php');

    // Conecta-se ao banco de dados
    $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

    // Obtém os dados do usuário a partir do MySQL
    $query = "SELECT user_id, first_name, picture FROM mismatch_user WHERE first_name IS NOT NULL";
    "ORDER BY join_date DESC LIMIT 5";
    $data = mysqli_query($dbc, $query);

    // Faz loop através do array de dados do usuário, formatando-os como HTML
    echo '<h4>Membros mais novos:</h4>';
    echo '<table>';
    while ($row = mysqli_fetch_array($data)) {
        if (is_file(MM_UPLOADPATH . $row['picture']) && filesize(MM_UPLOADPATH . $row['picture']) > 0) {
            echo '<tr><td>';
        }
        else {
            echo '<tr><td>';
        }
        if (isset($_SESSION['user_id'])) {
            echo '<td><a href="viewprofile.php?user_id=' . $row['user_id'] . '">' . $row['first_name'] . '</a></td></tr>';
        }
        else {
            echo '<td>' . $row['first_name'] . '</td></tr>';
        }
    }
    echo '</table>';

    mysqli_close($dbc);
?>

<?php
    // Insere o rodapé da página
    require_once('footer.php');
?>

```

As variáveis da conexão e as da aplicação ainda são incluídas a partir de scripts separados, como antes.

O menu de navegação é gerado depois do cabeçalho, mas antes do corpo da página.

O código que não vem do modelo agora é realmente único a esta página, portanto há muito menos dele.

O rodapé finaliza a página, e precisa vir por último, uma vez que é ele que fecha as tags HTML.



uma aplicação php bem desenhada

## O Mismatch está novamente em pé... e muito melhor organizado

Embora o pensamento de destrinchar o Mismatch em pequenos pedaços possa nos ter deixado um pouco nervosos, o resultado final definitivamente valeu o esforço. A aplicação agora está dividida em vários novos arquivos templates (inclusos), os quais oferecem muito melhor organização e maximizam o compartilhamento de código. Se você precisar modificar um desses pedaços, basta editar um arquivo e o efeito se fará sentir em toda a aplicação... esse é o poder dos templates!

O código de inicialização da sessão é usado por qualquer página que requeira login do usuário.

O script `startsession.php` lida com as tarefas de login nos bastidores e não ocupa nenhum espaço virtual na página.

`startsession.php`

O menu de navegação fornece links úteis para as partes principais da aplicação.

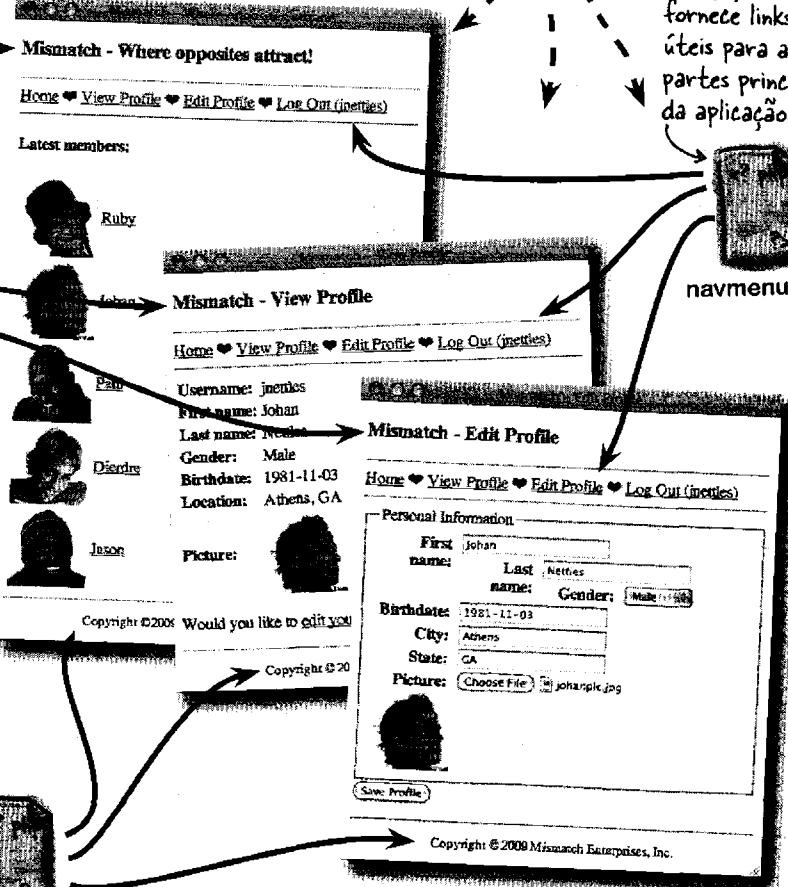
`navmenu.php`

O cabeçalho da página inclui código HTML e o título da página.

`header.php`

O rodapé da página contém informações de copyright para toda a aplicação... se você precisar mudar o aviso, basta modificá-lo em apenas um lugar!

`footer.php`



## 8 Controle Seu Mundo

# Colhendo Dados

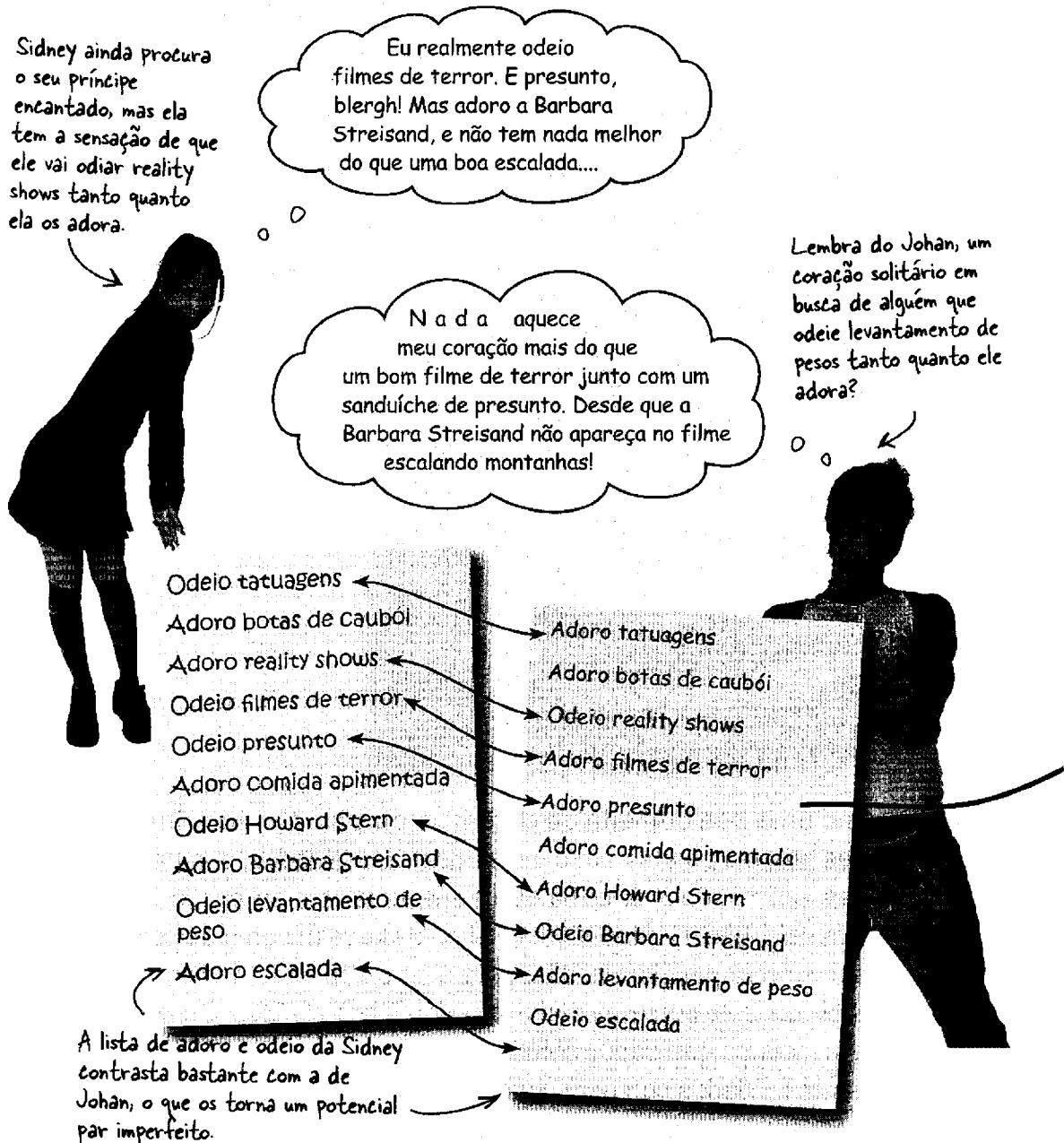


Do meu ponto de vista,  
é tudo uma questão de gerenciamento  
de dados. Primeiro eu organizo as ervilhas,  
depois seleciono algumas batatas, junto a  
elas um pouco de salsinha e algumas espigas  
de milho... e quando você vê, já temos um  
delicioso guisado!

**Nada como uma boa colheita de dados no outono.** Um sem-fim de informações prontas para serem examinadas, classificadas, comparadas, combinadas, enfim, qualquer coisa que a sua excelente aplicação web precisar que seja feito. Compensador? Sim. Mas assim como as colheitas na vida real, é preciso muito trabalho duro e uma boa dose de conhecimento para se obter controle sobre os dados em um banco MySQL. Os usuários da web exigem mais do que dados estáticos e enfadonhos. Eles querem dados enriquecedores... dados compensadores... dados relevantes. Então, o que você está esperando? Dê a partida no seu trator MySQL e mãos à obra!

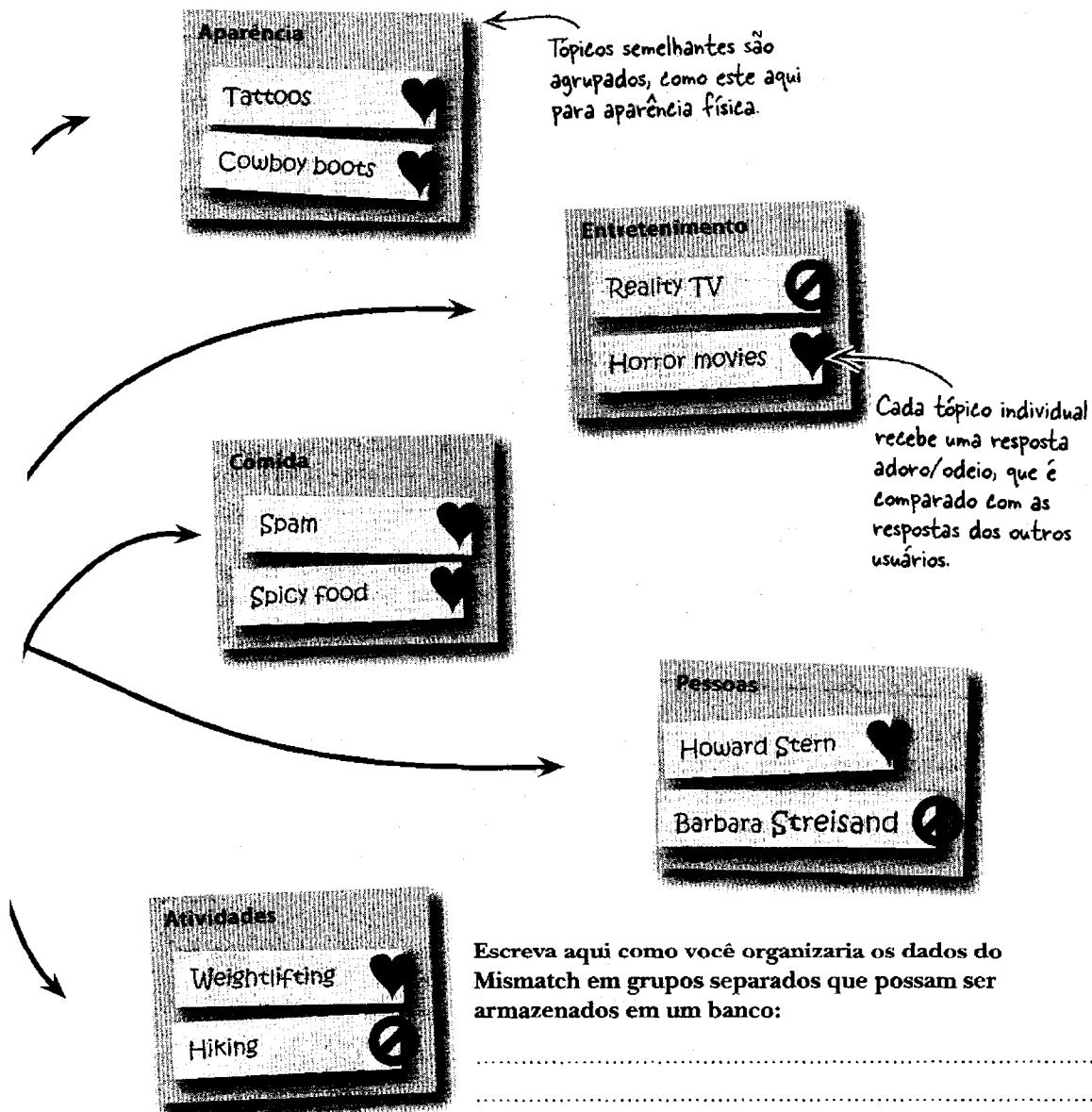
## Fazendo o desencontro perfeito

O aplicativo Mismatch tem um número cada vez maior de usuários registrados, e eles estão prontos para ver resultados. Precisamos permitir que os usuários achem os seus opositos ideais, comparando o que amam e o que odeiam com as preferências dos outros usuários, em busca de desencontros. Para cada "adoro" combinado com um "odeio", maior a probabilidade de um casal formar um par imperfeito.



## Os desencontros se referem aos dados

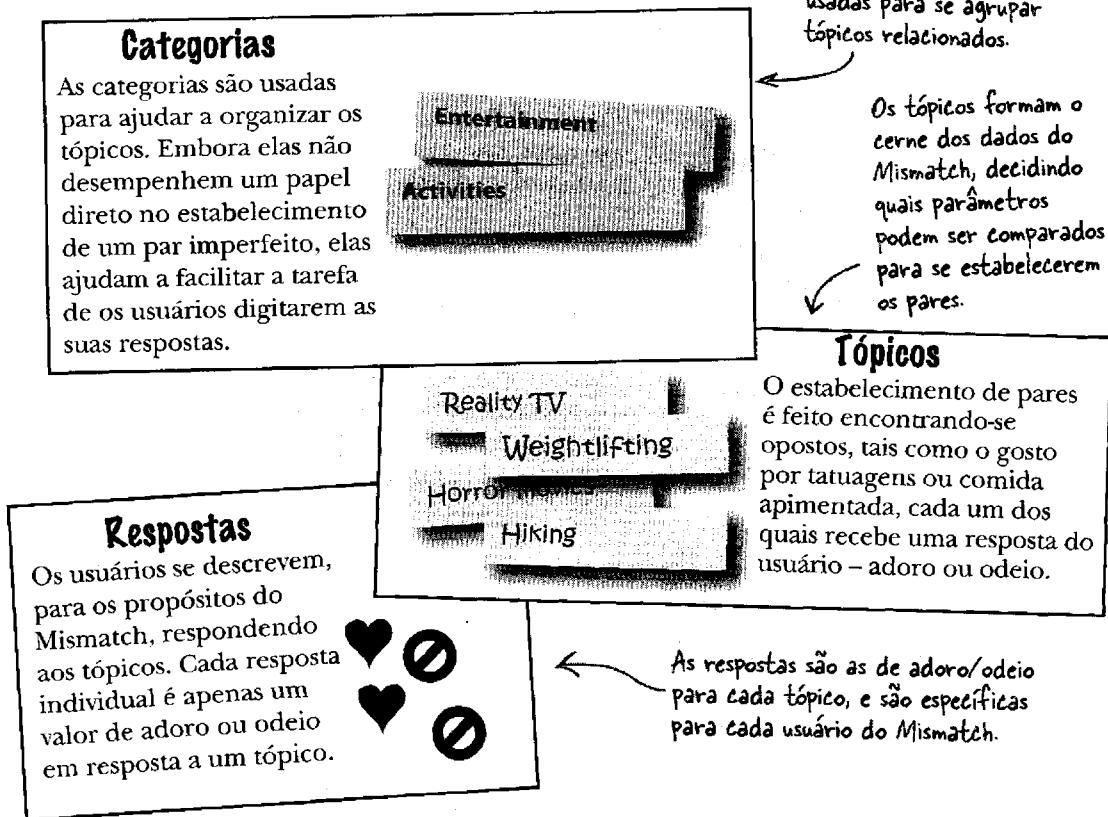
Para podermos estabelecer desencontros entre os usuários, precisamos primeiro determinar o melhor modo de organizar os dados que informam o que eles adoram e o que odeiam. Saber que as informações serão armazenadas em um banco de dados MySQL não é suficiente. Precisamos organizar esses tópicos de adoro/odeio de forma que sejam mais fáceis de gerenciar, permitindo que os usuários respondam a tópicos relacionados, indicando que eles adoram ou odeiam cada um.



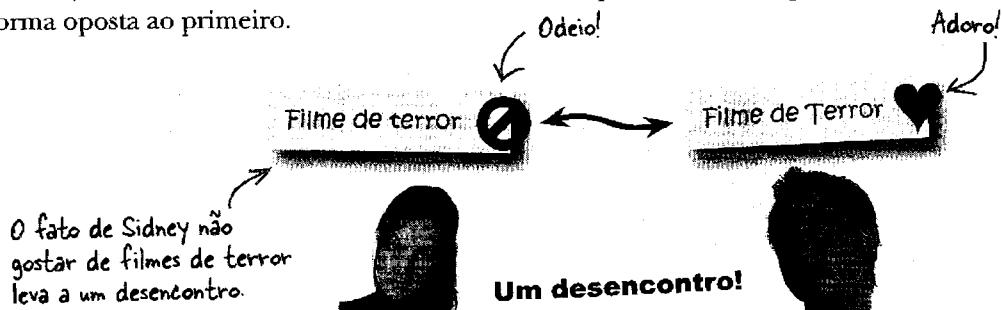
um modelo de dados para mismatch

## Divida os dados do Mismatch

Criar um modelo de dados para uma aplicação como o Mismatch é um passo extremamente importante, uma vez que ele tem um grande controle sobre o modo como a aplicação é construída. No caso do Mismatch, nós podemos dividir os seus dados em três tipos.



Como exatamente esses dados levam a um desencontro entre dois usuários? Nós comparamos as respostas que os usuários deram a cada tópico. Por exemplo, uma vez que Sidney e Johan têm respostas opostas para o tópico "Filmes de terror", achamos um desencontro nesse tópico particular. Para descobrir o melhor par imperfeito para um usuário, temos de encontrar o usuário com a maior quantidade de tópicos respondidos de forma oposta ao primeiro.



## Modele o banco de dados com um schema

Para podermos traduzir os requerimentos de dados da aplicação Mismatch em um projeto de banco de dados propriamente dito, precisamos de um schema. O schema é a representação de todas as estruturas do seu banco de dados, como por exemplo, tabelas e colunas, além do modo como eles se conectam uns aos outros. Criar uma representação visual do seu banco pode ajudá-lo a ver como as coisas se conectam quando você estiver escrevendo suas consultas, sem falar que ela esclarece o que cada coluna é responsável por fazer durante a conexão. Como um exemplo, vejamos o schema do banco de dados original do Mismatch, do capítulo anterior, que consistia apenas da tabela mismatch\_user.

O nome da tabela:

mismatch_user	
user_id	→
username	←
password	←
join_date	←
first_name	←
last_name	←
gender	←
birthdate	←
city	←
state	←
picture	←

Este símbolo indica que a coluna é uma chave primária para a tabela.

As outras colunas da tabela são listadas da forma como elas aparecem na estrutura do banco de dados.

Esta forma de se considerar a estrutura de uma tabela é um pouco diferente do que viu até agora. As tabelas normalmente têm sido representadas com os nomes das colunas na primeira linha, com os dados logo abaixo. Essa é uma ótima forma de se visualizar tabelas individuais e tabelas preenchidas com dados, mas não é muito prática quando queremos criar um diagrama estrutural de várias tabelas e do modo como elas se relacionam. E o Mismatch já está precisando de mais de uma tabela...

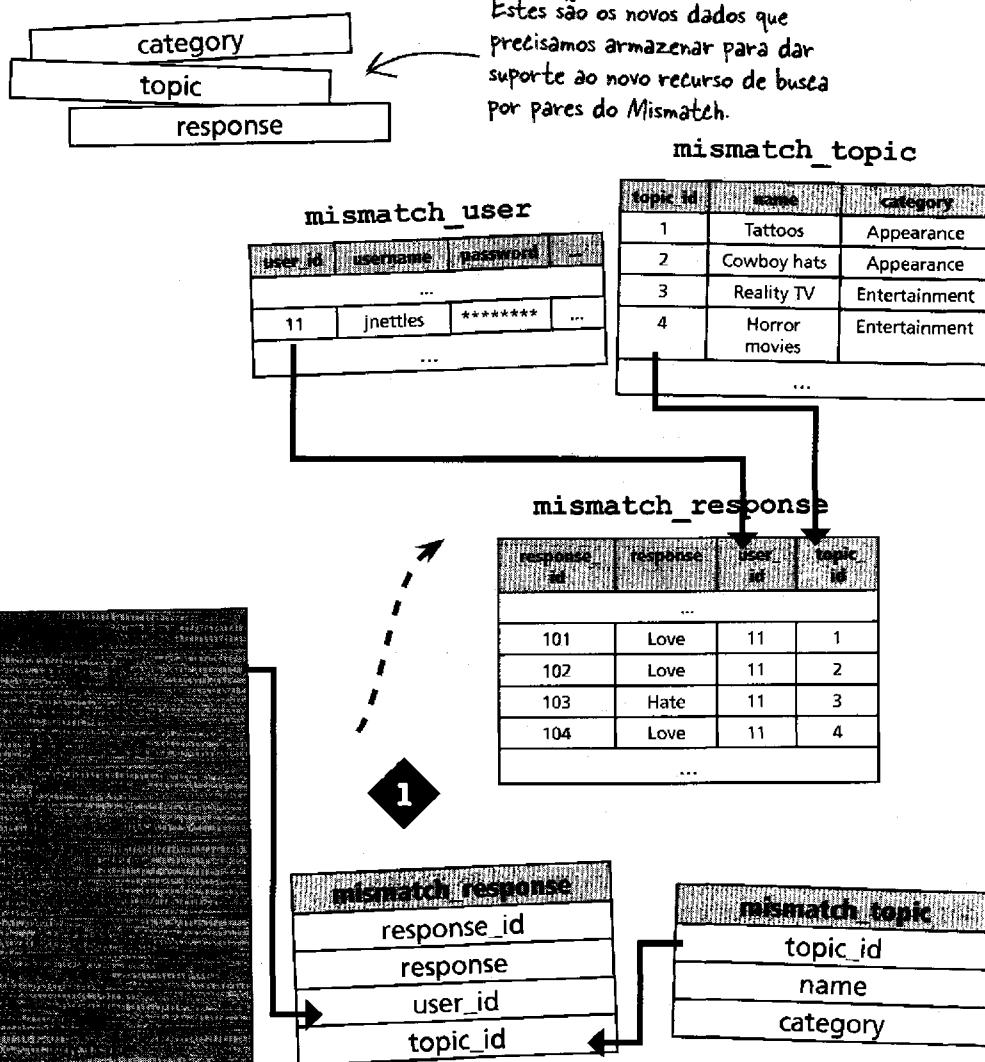
Uma descrição dos dados (as tabelas e as colunas) do seu banco de dados, junto com quaisquer outros objetos relacionados e a forma como eles se conectam, é um schema.

Criar um diagrama lhe permite manter o design da tabela separado dos dados contidos nela.

*escolha o melhor schema do mismatch*



O banco de dados do Mismatch está precisando armazenar as respostas dos usuários para tópicos do tipo adoro/odeio, bem como os nomes dos tópicos e as suas respectivas categorias. Apresentamos aqui três projetos de bancos de dados para incorporar as categorias, os tópicos e as respostas no banco do Mismatch. Circule o schema que você acha mais apropriado, e escreva por quê.



2

**mismatch\_user**

user_id	username	password	...
11	jnettles	*****	...
...			

**mismatch\_response**

response_id	category	topic	response	user_id
101	Appearance	Tattoos	Love	11
102	Appearance	Cowboy boots	Love	11
103	Entertainment	Reality TV	Hate	11
104	Entertainment	Horror movies	Love	11
...				

**mismatch\_response**

response_id	category	topic	response	user_id
-------------	----------	-------	----------	---------

**mismatch\_user**

user_id	username	password	topic	response
11	jnettles	*****	...	1 Love
11	jnettles	*****	...	2 Love
11	jnettles	*****	...	3 Hate
11	jnettles	*****	...	4 Love
...				

**mismatch\_topic**

topic_id	name	category
1	Tattoos	Appearance
2	Cowboy hats	Appearance
3	Reality TV	Entertainment
4	Horror movies	Entertainment
...		

topic\_id

response

topic\_id

name

category

3

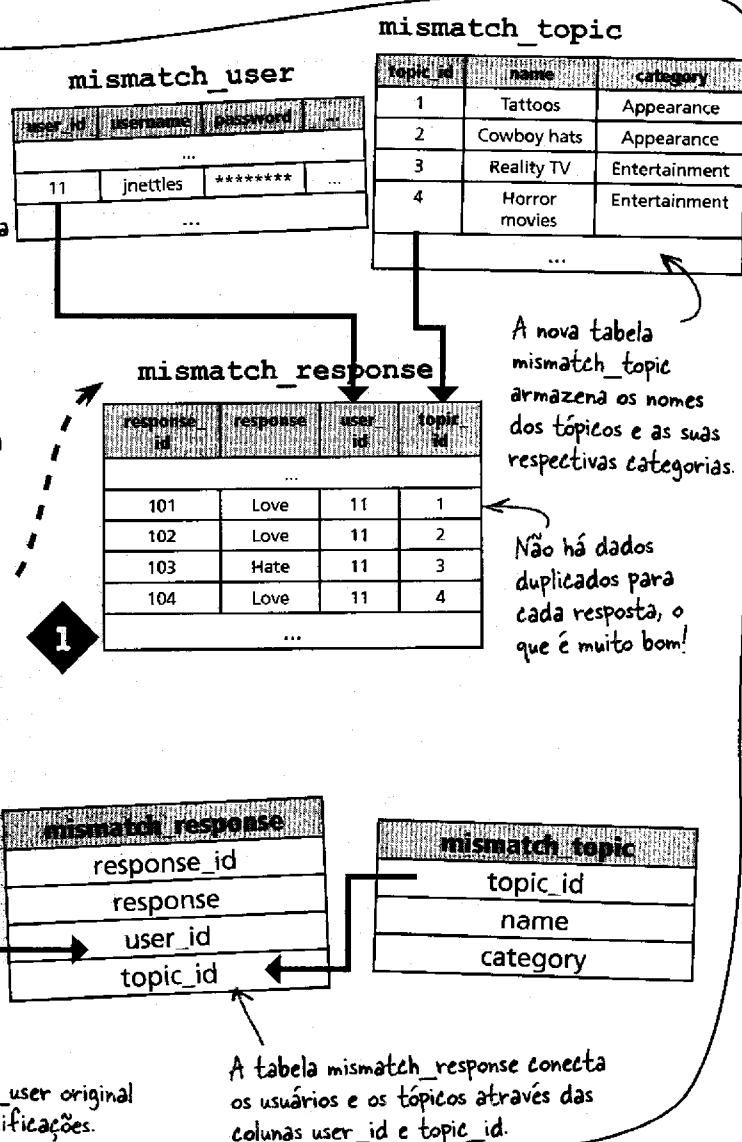


## Exercício Solução

O banco de dados do Mismatch está precisando armazenar as respostas dos usuários para tópicos do tipo adoro/odeio, bem como os nomes dos tópicos e as suas respectivas categorias. Apresentamos aqui três projetos de bancos de dados para incorporar as categorias, os tópicos e as respostas no banco do Mismatch. Circule o schema que você acha mais apropriado, e escreva por quê.

Em primeiro lugar, é importante ficar claro que os únicos dados novos no processo de se obter respostas adoro/odeio do usuário são as próprias respostas – tudo o mais no banco de dados permanece o mesmo, pelo menos do ponto de vista do usuário.

Quem disse que o mais simples é sempre o melhor? Este schema armazena as respostas em uma tabela própria para elas, separada dos outros dados que não sejam impactados diretamente por elas. Não há duplicação das respostas, porque os usuários, as categorias e os tópicos ficam todos fora da tabela mismatch\_response.



As respostas não ficam armazenadas dentro da tabela dos usuários, o que é ótimo. Mas existe uma quantidade excessiva de dados duplicados, porque as categorias e os tópicos são replicados para cada resposta.

mismatch\_user

user_id	username	password
...	...	...
11	jnettles	*****
...	...	...
...	...	...

mismatch\_response

response_id	category	topic	response	user_id
101	Appearance	Tattoos	Love	11
102	Appearance	Cowboy boots	Love	11
103	Entertainment	Reality TV	Hate	11
104	Entertainment	Horror movies	Love	11
...	...	...	...	...

mismatch\_response

response_id
category
topic
response
user_id

Os dados dos usuários são pessimamente replicados a cada resposta.

As categorias e os tópicos são duplicados para cada resposta, o que é um desperdício inaceitável de recursos.

Não é má ideia dividir as categorias e os tópicos em uma tabela para cada um, mas da forma como está sendo feito, temos de criar uma linha para cada resposta. Isso nos deixa com 4 Johans para 4 respostas. Nada bom!

mismatch\_user

user_id	username	password	topic_id	response
...	...	...	...	...
11	jnettles	*****	1	Love
11	jnettles	*****	2	Love
11	jnettles	*****	3	Hate
11	jnettles	*****	4	Love
...	...	...	...	...

mismatch\_topic

topic_id	name	category
1	Tattoos	Appearance
2	Cowboy hats	Appearance
3	Reality TV	Entertainment
4	Horror movies	Entertainment
...	...	...

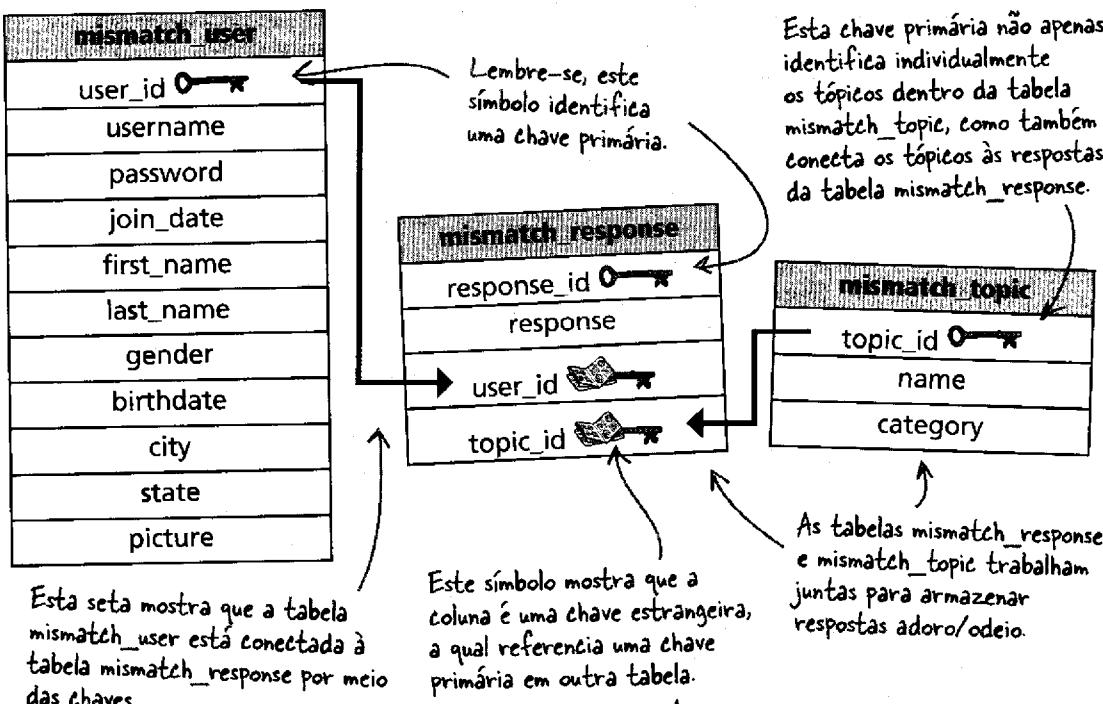
mismatch\_topic

topic_id
name
category

## Junte várias tabelas

Para conectarmos as tabelas de modo a formar um sistema de dados coeso, precisamos usar chaves. Nós já usamos chaves primárias para fornecer um identificador único para os dados de uma tabela, mas agora precisamos de chaves estrangeiras para vincular uma linha de uma tabela a uma linha de outra tabela. Uma chave estrangeira de uma tabela referencia a chave primária de outra, estabelecendo entre as duas uma conexão que pode ser usada em consultas.

O schema que elaboramos para o Mismatch no exercício anterior depende de um par de chaves estrangeiras na tabela mismatch\_response para conectar linhas de respostas a linhas de usuários e tópicos em outras tabelas.



Sem as chaves estrangeiras, seria muito difícil associar dados de uma tabela com dados de outra. E dividir os dados entre múltiplas tabelas é justamente o modo que encontramos para eliminar a duplicação de dados e conseguir maior eficiência para o banco. Assim, as chaves estrangeiras desempenham um papel importante em todo schema de bancos de dados, por mais simples que o banco seja.

Uma chave estrangeira (foreign key) é uma coluna, em uma tabela, que referencia a chave primária (primary key) de outra tabela.

As setas grandes mostram as chaves primárias conectando-se às chaves estrangeiras, para vincular as tabelas entre si.

## Chaves estrangeiras em ação

Muitas vezes, é útil visualizar os dados fluindo para as tabelas e conectando-as umas às outras através de chaves primárias e estrangeiras. Dar uma olhada com mais atenção nas tabelas do Mismatch, com alguns dados nelas, ajuda a revelar como as chaves primárias e estrangeiras se relacionam umas com as outras.

Sendo uma chave primária, user\_id precisa ser única dentro da tabela mismatch\_user. De fato, esse é o seu propósito — fornecer uma referência única a cada linha.

A chave estrangeira user\_id serve como uma referência às linhas dos usuários na tabela mismatch\_user, permitindo que você saiba qual usuário está associado com qual resposta.

A chave primária topic\_id serve como um índice único para as linhas da tabela mismatch\_topic.

Lembre-se, cada linha desta tabela corresponde a um usuário do Mismatch.

Cada linha desta tabela é uma única resposta adoro/odeio fornecida por um determinado usuário.

mismatch_user		
user_id	username	password
11	jnettles	*****
...	...	...

topic_id	name	category
1	Tattoos	Appearance
2	Cowboy hats	Appearance
3	Reality TV	Entertainment
4	Horror movies	Entertainment
...		

mismatch\_response

response_id	response	user_id	topic_id
101	Love	11	1
102	Love	11	2
103	Hate	11	3
104	Love	11	4
...			

Dentro da tabela mismatch\_response, você pode encontrar mais informações sobre o usuário que digitou uma determinada resposta procurando pela user\_id na tabela mismatch\_user. Da mesma forma, você pode descobrir o nome do tópico de uma resposta, bem como a sua categoria, olhando o topic\_id na tabela mismatch\_topic.

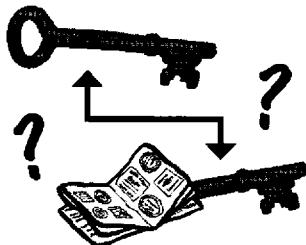
Vincular tabelas através de chaves primárias e chaves estrangeiras nos permite conectar os dados entre elas de uma forma consistente. Você pode até mesmo estruturar o seu banco de dados de forma a exigir uma correspondência entre as chaves primárias e as suas respectivas chaves estrangeiras. Isso é conhecido como integridade referencial, o que é uma forma chique de dizer que todas as referências das chaves precisam ser válidas.

Cada linha desta tabela é o nome/categoria de uma resposta adoro/odeio, mas não a resposta propriamente dita.

A chave estrangeira topic\_id referencia as linhas dos tópicos na tabela mismatch\_topic, e não é única, uma vez que muitos usuários diferentes terão as mesmas respostas para os mesmos tópicos.

A chave estrangeira user\_id vincula uma linha de resposta a uma linha de usuário na tabela mismatch\_user. Ela não é única, uma vez que um mesmo usuário pode ter diversas respostas adoro/odeio.

## tipos de relacionamento entre tabelas



Eu entendi que as chaves primárias e estrangeiras conectam várias tabelas entre si, mas a direção das setas nos diagramas tem algum significado?



### Sim, a direção das setas nos informa como as linhas de cada tabela se relacionam umas com as outras.

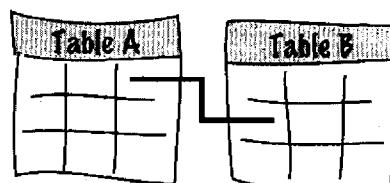
Falando mais especificamente, elas nos dizem quantas linhas em uma tabela podem ter linhas correspondentes em outra, e vice-versa. Este é um aspecto importantíssimo do projeto do schema, e envolve três padrões de dados possíveis: um-para-um, um-para-muitos e muitos-para-muitos.

### As tabelas podem corresponder linha por linha

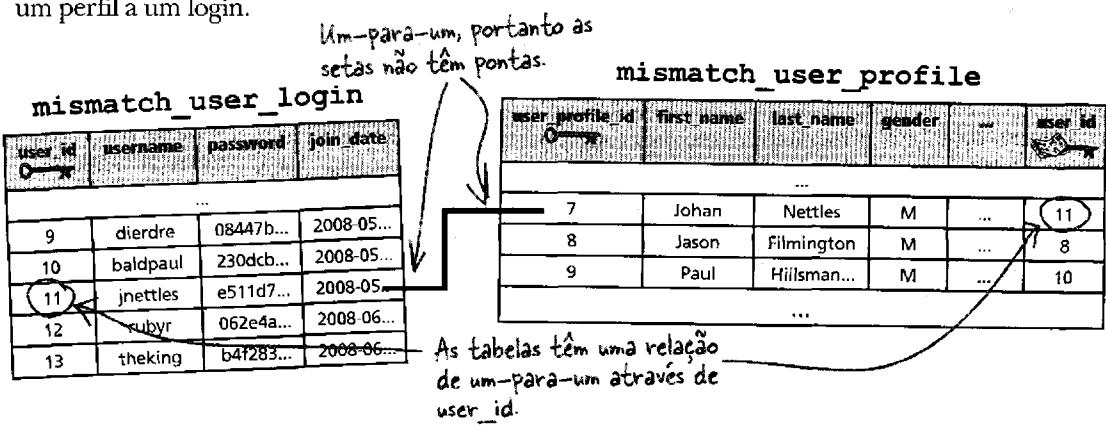
O primeiro padrão, um-para-um, determina que uma linha na Tabela A pode ter no máximo UMA linha correspondente na Tabela B, e vice-versa. Assim, só há uma correspondência em cada tabela para cada linha.

Como um exemplo, digamos que a tabela dos usuários do Mismatch seja separada em duas tabelas, uma apenas para as informações de login (Tabela A) e uma com os dados dos perfis (Tabela B). Ambas as tabelas contêm IDs para manter os usuários conectados aos seus perfis.

A coluna user\_id da tabela de login é uma chave primária que garante a individualidade de cada login. Na tabela dos perfis, user\_id é uma chave estrangeira, e desempenha um papel diferente, uma vez que a sua função é apenas conectar um perfil a um login.

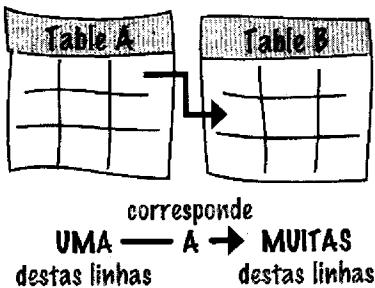


corresponde  
APENAS UMA — A — APENAS UMA  
destas linhas destas linhas



No que diz respeito às duas colunas user\_id, a tabela dos logins é considerada como tabela parent(pai), enquanto que a dos perfis é considerada como child(filho) – uma tabela com uma chave primária tem uma relação parent(pai)-child(filho) com a tabela que possui a chave externa correspondente.

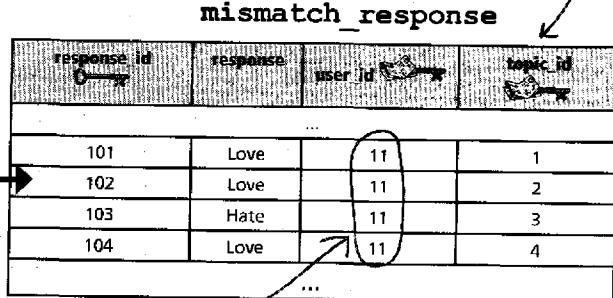
## **Uma linha leva a muitas outras**



### Chave primária.

mismatch_user				
user_id	username	password	...	...
9	dierdre	08447b...	...	...
10	baldpaul	230dcb...	...	...
11	jnettles	e511d7...	...	...
12	rubyr	062e4a...	...	...
13	theking	b4f283...	...	...

Há uma relação de um-para-muitos através de user\_id.



não existem

# Perguntas Ídiotas

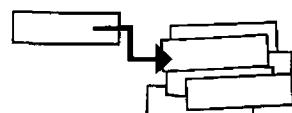
P: Como eu sei se as linhas de duas tabelas têm uma relação de um-para-um ou um-para-muitos?



Um-para-Um:  
exatamente  
uma linha  
da tabela  
parent(pai)  
se relaciona  
com uma  
linha da tabela  
child(filho).

**R:** Há uma tendência de se usar padrões de um-para-muitos com muito mais frequência do que um-para-um, e com bons motivos. É comum se ter uma tabela principal (parent) contendo dados primários, como por exemplo, os usuários do Mismatch, a qual se conecta a uma tabela secundária (child) em um arranjo de um-para-muitos. Isso acontece duas vezes no esquema do Mismatch, onde tanto os usuários quanto os tópicos têm relações de um-para-muitos com as respostas.

Em muitos casos, as linhas de uma relação de um-para-um em duas tabelas podem ser combinadas em uma só tabela. Porém, certamente existem situações em que faz sentido optar pelo padrão de um-para-um, como no exemplo hipotético dos perfis dos usuários na página anterior, onde há uma motivação de segurança para se colocar uma parte dos dados em uma tabela separada.

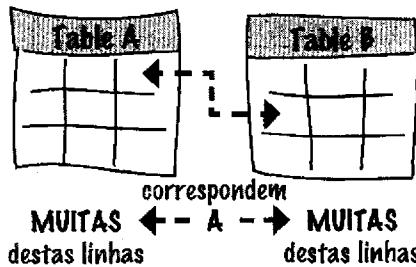


Um para Muitos:  
exatamente uma  
linha da tabela  
parent(pai)  
se relaciona  
com várias  
linhas da tabela  
child(filho).

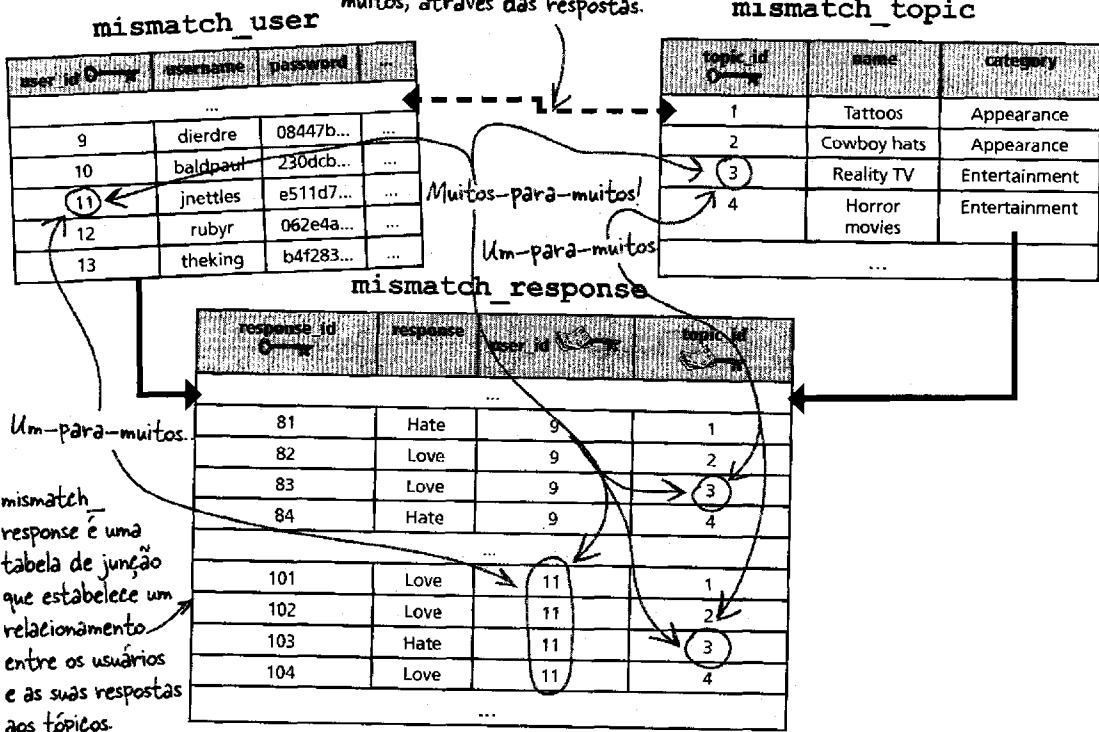
*o relacionamento muitos para muitos*

## Checando linhas de muitas-para-muitas

O terceiro e último padrão de relacionamento de dados é o de muitos-para-muitos, que tem muitas linhas de dados na Tabela A correspondendo a muitas linhas na Tabela B... parece um caso de sobrecarga de dados! Mas na realidade não é. Existem várias situações em que um padrão muitos-para-muitos é o mais apropriado. Seria o caso do Mismatch? Vamos dar uma olhada.

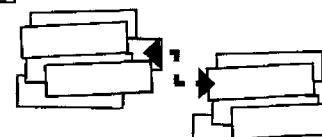


Os usuários e os tópicos têm uma relação de muitos-para-muitos, através das respostas.



O padrão muitos-para-muitos no Mismatch é indireto, significando que ele ocorre através da tabela mismatch\_response. Mas o padrão ainda existe. Basta ver quantas das mesmas user\_ids e topic\_ids aparecem em mismatch\_response.

Além de armazenar os dados das respostas, a tabela mismatch\_response está agindo como o que é conhecido como tabela de junção, ao servir como um conveniente intermediário para os usuários e os tópicos. Sem a tabela de junção, nós teríamos muitos dados duplicados, o que é ruim. Caso você não esteja convencido disso, volte para o exercício dos esquemas no início deste capítulo e dê uma olhada com atenção no Projeto 2. Naquele projeto, a tabela mismatch\_topic era repetida na tabela mismatch\_response, resultando em muita duplicação de dados.



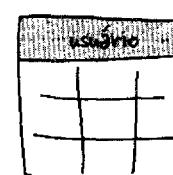
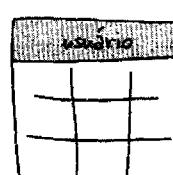
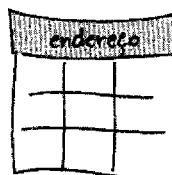
Muitos-para-Muitos:  
Várias linhas de uma tabela parent(pai) se relacionam com várias linhas de uma tabela child(filho).

# Qual é a Relação?

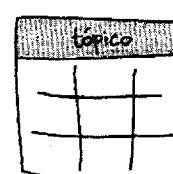
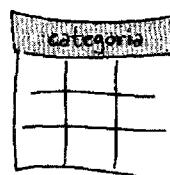
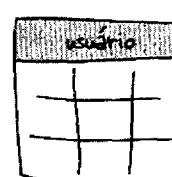
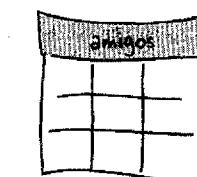
Em cada uma das tabelas abaixo, existem colunas circuladas que poderiam ser movidas para as suas próprias tabelas. Escreva se cada uma dessas colunas seria melhor representada por uma relação de um-para-um, um-para-muitos ou muitos-para-muitos com a sua tabela original, e depois desenhe a relação na forma de uma linha conectando as duas tabelas, com as setas apontando para os lados apropriados.

## Relação

mismatch_user	
id_do_usuario	key
...	
endereço	
empresa	
amigos	



mismatch_topic	
id_do_tópico	key
nome	
categoria	



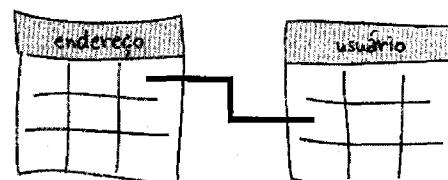
# Qual é a Relação? - Solução

Em cada uma das tabelas abaixo, existem colunas circuladas que poderiam ser movidas para as suas próprias tabelas. Escreva se cada uma dessas colunas seria melhor representada por uma relação de um-para-um, um-para-muitos ou muitos-para-muitos com a sua tabela original, e depois desenhe a relação na forma de uma linha conectando as duas tabelas, com as setas apontando para os lados apropriados.

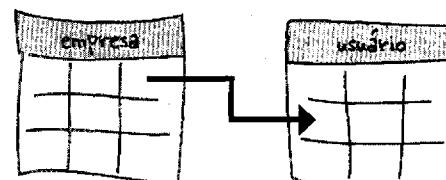
Só há um endereço para cada usuário, o que significa que a linha de endereço tem uma relação de um-para-um com cada linha de usuário.

mismatch user	
	id do usuário
...	
endereço	
empresa	
amigos	

## RELATIONSHIP

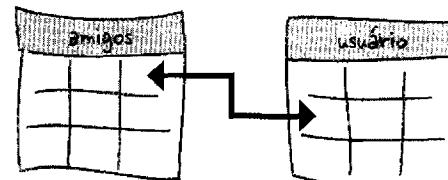


um-para-um



um-para-muitos

Vários usuários podem trabalhar para a mesma empresa, resultando em uma relação de um-para-muitos entre uma linha de empresa e várias linhas de usuários.



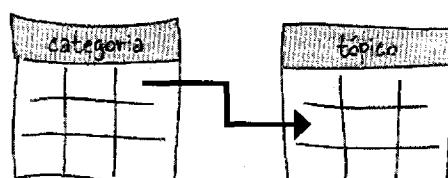
muitos-para-muitos

Vários usuários podem ter vários amigos, significando que as linhas de amigos têm uma relação de muitos-para-muitos com as linhas de usuários.

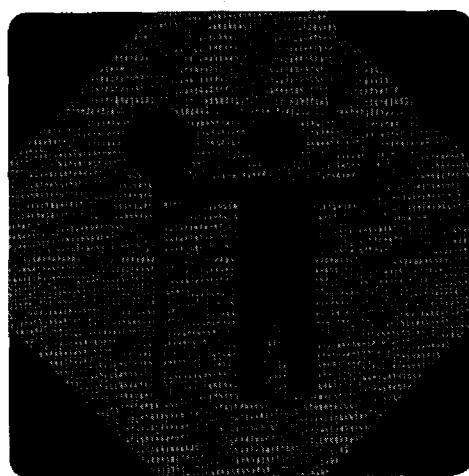
Vários tópicos podem pertencer à mesma categoria, resultando em uma relação de um-para-muitos entre uma linha de categoria e várias linhas de tópicos. Mas cada tópico não pode pertencer a mais de uma categoria.

mismatch topic	
	id do tópico
...	
nome	
categoria	

Tabelas que contêm uma relação de muitos-para-muitos geralmente são conectados através de uma tabela de junção, que não é mostrada aqui.



um-para-muitos



### **Espere um momento! Tire um segundo para organizar o banco de dados do Mismatch, para que possamos encontrar os pares imperfeitos.**

Baixe os arquivos .sql da aplicação Mismatch no site da Alta Books, em [www.altabooks.com.br](http://www.altabooks.com.br). Esses arquivos contêm as instruções SQL que criam as tabelas necessárias: mismatch\_user, mismatch\_topic e mismatch\_response. Certifique-se de rodar a instrução de cada arquivo .sql em uma ferramenta de MySQL, para ter todas as tabelas de que precisa para iniciar o Mismatch.

Quando estiver tudo pronto, execute uma instrução DESCRIBE em cada uma das novas tabelas (mismatch\_topic e mismatch\_response) para checar as suas estruturas. Essas tabelas são importantíssimas para os scripts PHP que estamos prestes a escrever para o Mismatch.

```

File Edit Window Help Levels
mysql> DESCRIBE mismatch_topic;
+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| topic_id | int(11) | NO | PRI | NULL    | auto_increment |
| name | varchar(48) | NO | UNI | NULL    |                |
| category | varchar(48) | NO |     | NULL    |                |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

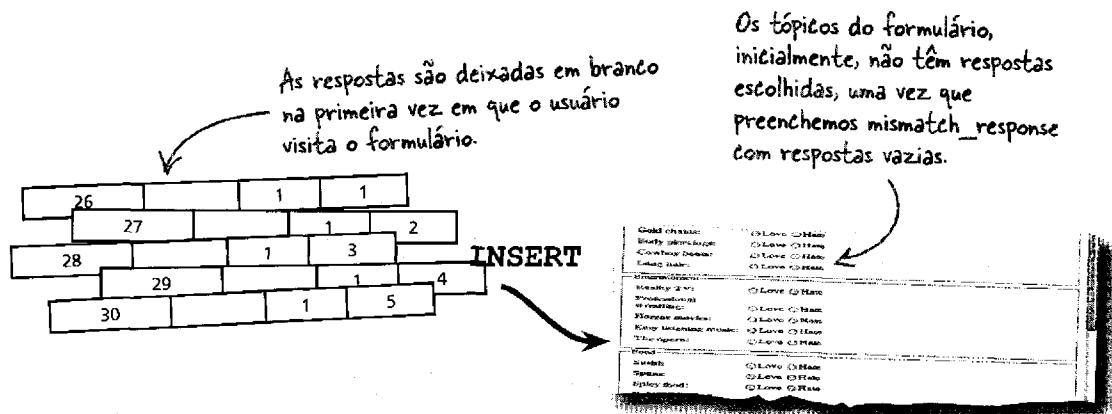
File Edit Window Help Help
mysql> DESCRIBE mismatch_response;
+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| response_id | int(11) | NO | PRI | NULL    | auto_increment |
| user_id | int(11) | NO |     | NULL    |                |
| topic_id | int(11) | NO |     | NULL    |                |
| response | tinyint(4) | NO |     | 0        |                |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

A chave estrangeira
topic_id vincula-se
à chave primária da
tabela mismatch_
topic.
  
```

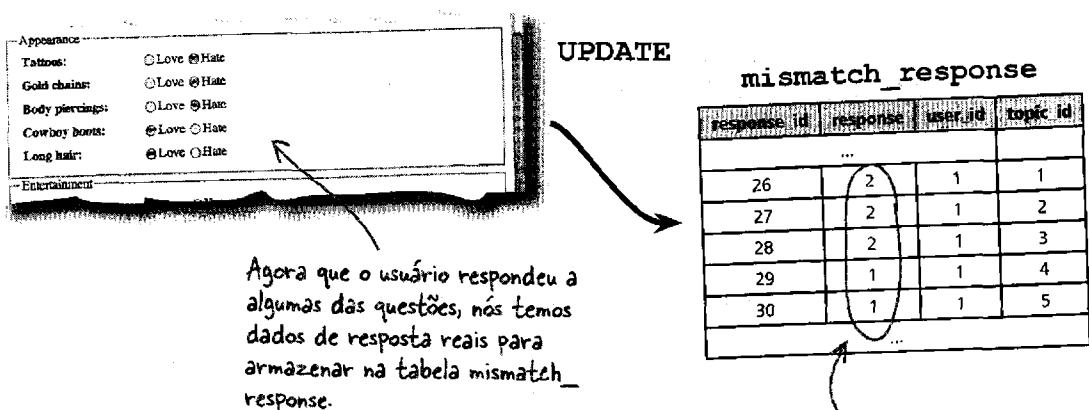
*colocando resposta em mismatch\_response*

## Coloque as respostas no banco de dados

Embora possa parecer que nós devamos começar gerando o formulário, este depende da existência de dados de resposta na tabela mismatch\_response. Portanto, vamos começar pelo começo: temos de "preencher" a tabela mismatch\_response com linhas de respostas vazias, na primeira vez que o usuário acessar o questionário. Isso nos permitirá gerar o formulário a partir da tabela mismatch\_response sem termos de nos preocupar se o usuário realmente já forneceu quaisquer respostas.

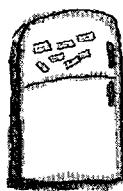


Assim, do ponto de vista do formulário, sempre existe uma linha de dados, na tabela mismatch\_response, para cada questão. Isso significa que, quando o usuário submete o formulário, nós apenas atualizamos as linhas de dados para cada resposta fornecida.



Embora o processo de se armazenar respostas no banco do Mismatch tenha, em última análise, dois passos, o primeiro deles (INSERT) só ocorre uma vez para cada usuário. Uma vez que as respostas vazias tenham sido adicionadas, todas as modificações futuras são realizadas no segundo passo, por meio de UPDATEs SQL.

As respostas são atualizadas no banco de dados, para registrar as informações fornecidas pelo usuário.



## Ímas de Geladeira do PHP & MySQL

O código seguinte cuida de inserir respostas vazias na tabela mismatch\_response, na primeira vez em que o usuário visita o formulário. Ele também atualiza as respostas quando o usuário faz modificações e submete o formulário. Infelizmente, parte do código se perdeu e precisa ser reestabelecida. Use os ímas para completar o código que está faltando.

```

...
// Se este usuário jamais respondeu ao questionário, inserir respostas vazias no
// banco de dados
$query = "SELECT * FROM mismatch_response WHERE user_id = '" . $_SESSION['user_
id'] . "'";
$data = mysqli_query($dbc, $query);

if ( .....($data) == 0) {

    // Primeiramente, obtém a lista de IDs dos tópicos a partir da respectiva tabela
    $query = "SELECT.....FROM mismatch_topic ORDER BY category_id, topic_id";

    $data = mysqli_query($dbc, $query);
    $topicIDs = array();
    while ($row = mysqli_fetch_array($data)) {
        array_push($topicIDs, $row['topic_id']);
    }
    // Insere linhas de respostas na tabela respectiva, uma para cada tópico
    foreach ($topicIDs as $topic_id) {

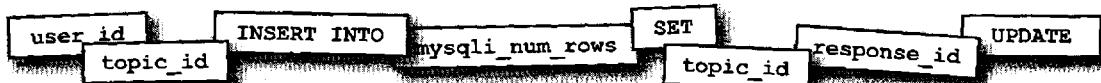
        $query = ".....mismatch_response " .
            "( ..... , ..... ) VALUES ('" . $_SESSION['user_id'] . "',
'$topic_id')";

        mysqli_query($dbc, $query);
    }
}
// Se o formulário tiver sido submetido, escreve as respostas no banco
if (isset($_POST['submit'])) {
    // Escreve as linhas de respostas na respectiva tabela
    foreach ($_POST as $response_id => $response) {

        $query = "..... mismatch_response ..... response = '$response'
"
        "WHERE ..... = '$response_id'";
        mysqli_query($dbc, $query);
    }
    echo '<p>As suas respostas foram registradas.</p>';
}

...

```





## Ímas de Geladeira do PHP & MySQL - Solução

O código seguinte cuida de inserir respostas vazias na tabela mismatch\_response, na primeira vez em que o usuário visita o formulário. Ele também atualiza as respostas quando o usuário faz modificações e submete o formulário. Infelizmente, parte do código se perdeu e precisa ser reestabelecida. Use os ímas para completar o código que está faltando.

```
...  
// Se este usuário jamais respondeu ao questionário, inserir respostas vazias  
no banco de dados  
$query = "SELECT * FROM mismatch_response WHERE user_id = '' . $_  
SESSION['user_id'] . "";  
$data = mysqli_query($dbc, $query);  
  
if (mysqli_num_rows($data) == 0) { ←  
    // Primeiramente, obtém a lista de IDs dos tópicos a partir da respectiva tabela  
  
    $query = "SELECT ... topic_id ... FROM mismatch_topic ORDER BY category_id,  
topic_id";  
  
    $data = mysqli_query($dbc, $query);  
    $topicIDs = array();  
    while ($row = mysqli_fetch_array($data)) {  
        array_push($topicIDs, $row['topic_id']);  
    }  
  
    // Insere linhas de respostas na tabela respectiva, uma para cada tópico  
    foreach ($topicIDs as $topic_id) {  
        $query = " ... INSERT INTO mismatch_response ... ".  
        "( ... user_id , ... topic_id ) VALUES ('" . $_SESSION['user_id'] . "",  
'$topic_id')";  
  
        mysqli_query($dbc, $query);  
    }  
  
    // Se o formulário tiver sido submetido, escreve as respostas no banco  
    if (isset($_POST['submit'])) {  
        // Escreve as linhas de respostas na respectiva tabela  
        foreach ($_POST as $response_id => $response) {  
  
            $query = " ... UPDATE mismatch_response ... SET response = '$response'  
            "WHERE response_id = '$response_id'";  
  
            mysqli_query($dbc, $query);  
        }  
        echo '<p>As suas respostas foram registradas.</p>';  
    }  
...  
"
```

Verifica se a consulta  
retornou 0 linhas de  
dados... não há dados!

Para gerar um array vazio de  
respostas, primeiramente nós  
temos de obter todos os tópicos,  
na respectiva tabela.

A linha da resposta se encontra  
"não respondida" neste ponto,  
uma vez que o usuário ainda não  
escolheu "adoro" ou "odeio" no  
formulário.

Só o que muda quando  
o usuário submete o  
formulário é a coluna  
resposta da respectiva  
tabela, portanto é só isso  
que nós atualizamos.

não existem  
Perguntas Idiotas

P: Qual é a da função array\_push()? Acho que ainda não usamos essa função..

R: Realmente, ainda não a usamos, porque ainda não precisamos criar um array dinamicamente, um elemento de cada vez. A função array\_push() anexa um novo elemento ao final do array, fazendo-o crescer em um elemento. No código da página anterior, nós estamos usando array\_push() para criar um array de IDs de tópicos a partir da tabela mismatch\_topic. Esse array é então usado para se inserir respostas em branco na tabela mismatch\_response... uma para cada tópico..

1 Use INSERT para adicionar linhas vazias de respostas ao banco de dados, na primeira vez que o usuário acessar o formulário.

**FEITO**

Bam! Acabamos de acertar dois coelhos virtuais com uma cajadada só, e agora já temos metade do script pronto.

2 Use UPDATE para modificar as linhas de respostas com base nas informações fornecidas pelo usuário.

**FEITO**

3 Use SELECT para obter os dados necessários para se gerar o formulário.

4 Gere o formulário HTML com o questionário a partir dos dados de resposta.

Mas ainda faltam dois passos para podermos começar a fazer pares no Mismatch...

## É possível comandar um formulário com dados

Não é novidade que formulários web são usados para se obter dados dos usuários através de campos de texto, listas de seleção, botões de rádio, etc., mas pode não ser muito óbvio o fato de que você pode gerar formulários HTML a partir de dados do banco, usando PHP. A ideia, no caso do Mismatch, é gerar dinamicamente um formulário HTML contendo o questionário criado a partir de dados de resposta. O script do formulário assume que já existem dados de resposta, o que o permite gerar o formulário a partir da tabela mismatch\_response. Nós sabemos que é seguro assumir isso, porque acabamos de escrever o código que adiciona respostas vazias na primeira vez que o usuário visita o formulário.

**Os formulários comandados por dados (data-drive) dependem de dados presentes em um banco MySQL para gerar campos de formulários HTML.**

The diagram illustrates the data-drive process for generating an HTML form from a MySQL database table. It shows a table named 'mismatch\_response' with columns: response\_id, response, user\_id, and topic\_id. Arrows point from each column to specific parts of the generated HTML code and the resulting browser form. Annotations explain how the primary key 'response\_id' identifies individual fields, how the table data generates the form, and how the 'checked' attribute controls radio button selection.

**mismatch\_response**

response_id	response	user_id	topic_id
0			

A chave primária `response_id` é usada para se identificar individualmente os campos do formulário HTML e se associar cada campo com uma linha do banco.

O código do formulário HTML é gerado a partir de dados na tabela `mismatch_response`.

O atributo `checked` controla a seleção dos botões de rádio.

```

<form method="post" action="">
  <p>How do you feel about each topic?</p>
  <fieldset>
    <legend>Appearance</legend>
    <label for="76">Tattoos:</label><input type="radio" id="76" name="76" value="1" checked="checked" />Love<br />
    <label for="77">Gold chains:</label><input type="radio" id="77" name="77" value="1" checked="checked" />Love
    <input type="radio" id="77" name="77" value="2" />Hate<br />
    >Love
    <input type="radio" id="78" name="77" value="2" />Hate<br />
    <label for="78">Body piercings:</label><input type="radio" id="78" name="78" value="1" checked="checked" />Love
    <input type="radio" id="78" name="78" value="2" />Hate<br />
    >Love
    <input type="radio" id="79" name="78" value="2" />Hate<br />
    <label for="79">Cowboy boots:</label><input type="radio" id="79" name="79" value="1" checked="checked" />Love
    <input type="radio" id="79" name="79" value="2" />Hate<br />
    >Love
    <input type="radio" id="80" name="79" value="2" />Hate<br />
    <label for="80">Long hair:</label><input type="radio" id="80" name="80" value="1" checked="checked" />Love
    <input type="radio" id="80" name="80" value="2" />Hate<br />
  </fieldset>
  <fieldset>
    <legend>Entertainment</legend>
    ...
  </fieldset>
</form>

```

How do you feel about each topic?

Appearance

Tattoos:	<input checked="checked" type="radio"/> Love <input type="radio"/> Hate
Gold chains:	<input checked="checked" type="radio"/> Love <input type="radio"/> Hate
Body piercings:	<input checked="checked" type="radio"/> Love <input type="radio"/> Hate
Cowboy boots:	<input checked="checked" type="radio"/> Love <input type="radio"/> Hate
Long hair:	<input checked="checked" type="radio"/> Love <input type="radio"/> Hate

Entertainment

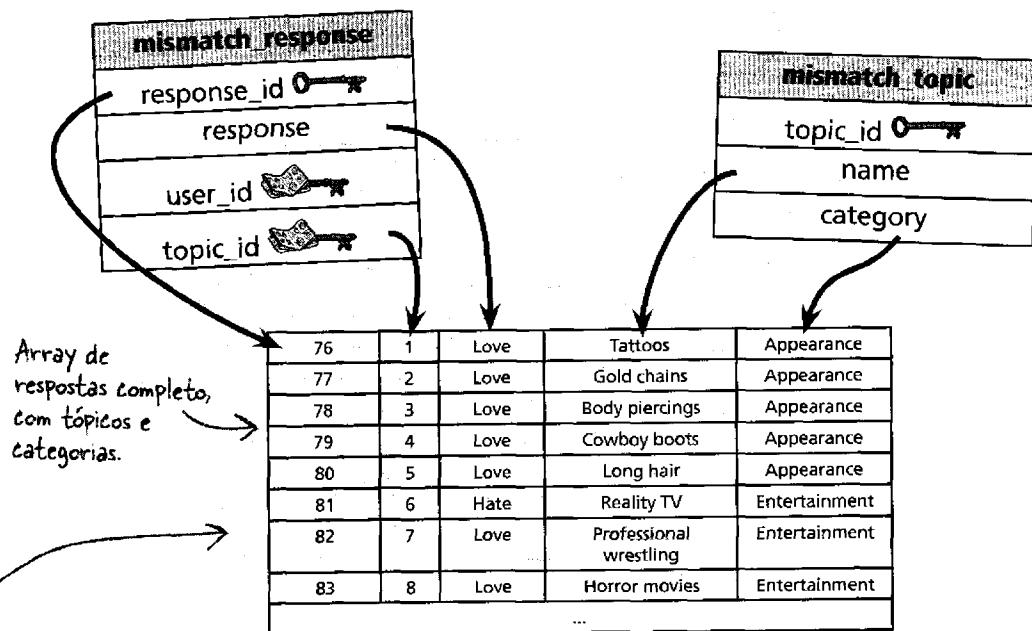
Reality TV:	<input checked="checked" type="radio"/> Love <input type="radio"/> Hate
Professional wrestling:	<input checked="checked" type="radio"/> Love <input type="radio"/> Hate
Horror movies:	<input checked="checked" type="radio"/> Love <input type="radio"/> Hate

**O formulário reflete as respostas do usuário para cada tópico.**



## Exercício

O questionário do Mismatch é gerado a partir de respostas do usuário que são armazenadas na tabela mismatch\_response. Para gerarmos o código do formulário HTML, é necessário ler essas respostas, certificando-nos de verificar o nome do tópico e o da categoria para cada resposta, na tabela mismatch\_topic. O código a seguir cria um array de respostas com tópicos e categorias, realizando duas consultas: a primeira obtém as respostas do usuário, enquanto que a segunda verifica o tópico e a categoria para cada resposta. O problema é que parte do código está faltando... preencha as lacunas para fazê-lo funcionar!



```
// Obtém os dados de resposta do banco, para gerar o formulário
$query = "SELECT response_id, topic_id, response FROM mismatch_
response "
    "WHERE user_id = '" . $_SESSION['user_id'] . "'";
$data = mysqli_query($dbc, $query);
$responses = array();
while ($row = mysqli_fetch_array($data)) {
    // Verifica o nome do tópico correspondente à resposta, na
    // tabela dos tópicos
    $query2 = ".....".
        "WHERE topic_id = '" . $row['topic_id'] . "'";
    $data2 = mysqli_query($dbc, .....);
    if (mysqli_num_rows(.....) == 1) {
        $row2 = mysqli_fetch_array($data2);
        $row['topic_name'] = .....
        $row['category_name'] = .....
        array_push($responses, $row);
    }
}
```

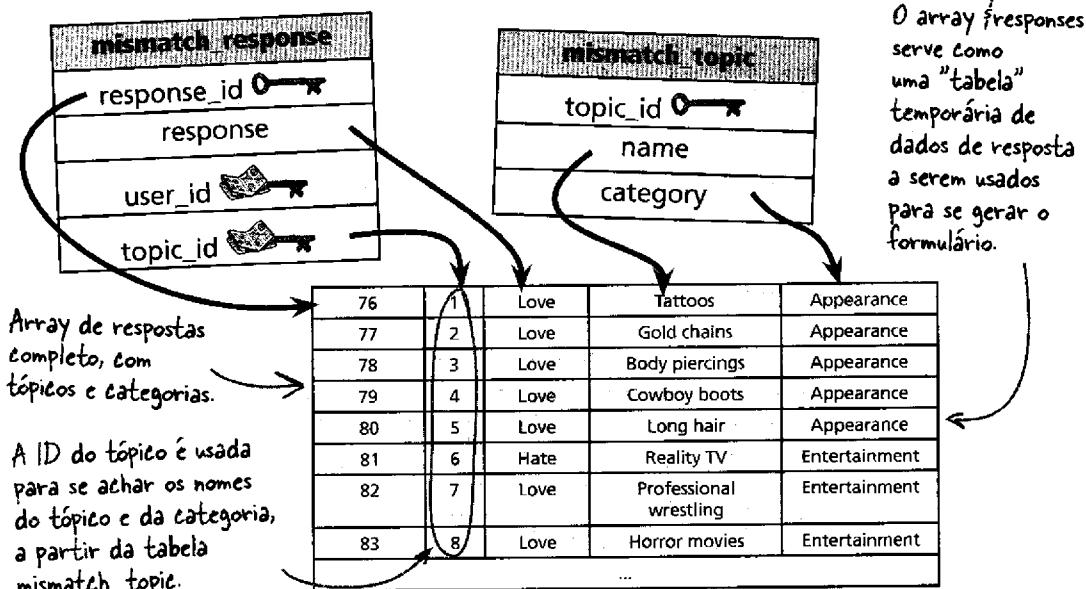
Esta função PHP lhe informa quantas linhas de dados foram retornadas como resultados da consulta.

## exercício solução



### Exercício Solução

O questionário do Mismatch é gerado a partir de respostas do usuário que são armazenadas na tabela mismatch\_response. Para gerarmos o código do formulário HTML, é necessário ler essas respostas, certificando-nos de verificar o nome do tópico e o da categoria para cada resposta, na tabela mismatch\_topic. O código a seguir cria um array de respostas com tópicos e categorias, realizando duas consultas: a primeira obtém as respostas do usuário, enquanto que a segunda verifica o tópico e a categoria para cada resposta. O problema é que parte do código está faltando... preencha as lacunas para fazê-lo funcionar!



```
// Obtém os dados de resposta do banco, para gerar o formulário
$query = "SELECT response_id, topic_id, response FROM mismatch_
response " .
```

É muito importante usar novas variáveis para se realizar uma segunda consulta (interna), de modo que a consulta original não seja afetada.

```
WHERE user_id = '" . $_SESSION['user_id'] . "'";
```

```
$data = mysqli_query($dbc, $query);
```

```
$responses = array();
```

```
while ($row = mysqli_fetch_array($data)) {
```

```
// Verifica o nome do tópico correspondente à resposta, na
tabela dos tópicos
```

```
$query2 = "...SELECT name, category FROM mismatch_topic....."
```

```
"WHERE topic_id = '" . $row['topic_id'] . "'";
```

```
$data2 = mysqli_query($dbc, $query2);
```

```
if (mysqli_num_rows($data2) == 1) {
```

```
$row2 = mysqli_fetch_array($data2);
```

```
$row['topic_name'] = ...$row2['name'];
```

```
$row['category_name'] = ...$row2['category'];
```

```
array_push($responses, $row);
```

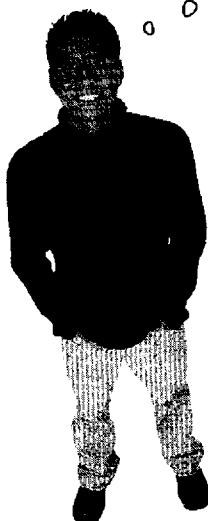
Certifique-se de haver dados de resposta aqui.

Os nomes do tópico e da categoria são adicionados ao array de resposta, atribuindo-se dados originários da segunda consulta.

A função ARRAY-PUSH() adiciona um item para o final de um array.

3

**FEITO**  
Use SELECT para obter os dados necessários para se gerar o



A resposta do usuário é armazenada como texto no banco de dados, como "Adoro" ou "Odeio" literalmente? Se sim, isso não seria pouco eficiente?

### **Não e Sim, e é por isso que é importante usar o tipo de dados mais eficiente possível para armazenar dados em um banco MySQL.**

Pensando bem, uma resposta no Mismatch é na verdade uma resposta do tipo true/false (verdadeiro/falso), porque ela é sempre um (adorei) ou o outro valor (odeio). Na verdade, um terceiro valor (desconhecido) pode ser útil para informar à aplicação que o usuário ainda não respondeu um determinado tópico. Então, nós precisamos manter registro de três valores possíveis para cada resposta. Esse tipo de problema de armazenamento é ideal para um número, como por exemplo, TINYINT. Com esse tipo de dados, basta você usar diferentes valores numéricos para representar cada resposta possível.



**Desconhecido = 0      Adoro = 1      Odeio = 2**

Minimizar os requerimentos de armazenamento de dados é uma parte importante do projeto de bancos de dados, e neste caso é uma parte útil, porém, importante da aplicação Mismatch. Essas respostas numéricas desempenham um papel direto na geração de campos do formulário do Mismatch.



### **Aponte seu lápis**

*Não se preocupe com os botões de rádio para "Odeio" por enquanto – eles são gerados exatamente da mesma forma.*

O seguinte código faz um loop através do array de respostas que você acabou de criar, gerando um campo do formulário HTML para cada botão de rádio "Adoro". Escreva o código que está faltando para que o campo fique inicialmente marcado, caso a resposta seja definida como "adoro" (1). Além disso, certifique-se de que o valor da tag <input> seja definido corretamente.

```

foreach ($responses as $response) {
    ...
    if (.....) {
        echo '<input type="radio" name="' . $response['response_id'] .
            '" value=..... checked=..... />Love ';
    }
    else {
        echo '<input type="radio" name="' . $response['response_id'] .
            '" value=..... />Adoro ';
    }
}

```

aponte seu lápis solução



## Aponte seu lápis Solução

O seguinte código faz um loop através do array de respostas que você acabou de criar, gerando um campo do formulário HTML para cada botão de rádio "Adoro". Escreva o código que está faltando para que o campo fique inicialmente marcado, caso a resposta seja definida como "adoro" (1). Além disso, certifique-se de que o valor da tag <input> seja definido corretamente..

O botão de rádio para "Adoro" é marcado com base no valor da resposta (1 representa "adoro", no banco).

```
foreach ($responses as $response) {  
    ...  
    if ($response['response'] == 1) {  
        echo '<input type="radio" name="' . $response['response_id'] .  
              '" value="1" checked="checked" />Love ';  
    }  
    else {  
        echo '<input type="radio" name="' . $response['response_id'] .  
              '" value="1" />Love ';  
    }  
}
```

O valor da tag <input> é definido como "1", para que fique mais fácil de armazenar a resposta no banco, quando o formulário for submetido.

Se esta resposta for definida como "adoro" (1), marca o botão de seleção, definindo o seu atributo checked com o valor "checked".

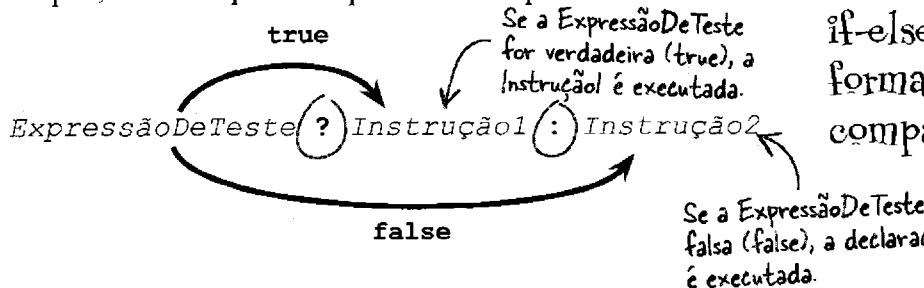
Deixar checked="checked" resulta no botão de seleção ficar desmarcado, se a resposta não estiver definida como "adoro" (1).

```
foreach ($responses as $response) {  
    ...  
    if ($response['response'] == 2) {  
        echo '<input type="radio" name="' . $response['response_id'] .  
              '" value="2" checked="checked" />Hate ';  
    }  
    else {  
        echo '<input type="radio" name="' . $response['response_id'] .  
              '" value="2" />Hate ';  
    }  
}
```

Caso esteja curioso, o código para gerar os botões "Odeio" funciona exatamente da mesma forma – ele apenas procura por uma resposta ligeiramente diferente... mas, na verdade, existe uma maneira mais elegante de gerar tanto os botões "Adoro" quanto os "Odeio" com menos código...

## Falando em eficiência...

A eficiência do banco de dados não é o único tipo de eficiência que vale a pena considerar. Há também a eficiência do código, que pode vir em muitas formas. Uma forma é tirando-se proveito da linguagem PHP para simplificar declarações if-else. O operador ternário é uma forma útil de se programar declarações if-else simples, de modo que elas fiquem mais compactas.



O operador ternário, na verdade, é apenas um atalho para se escrever uma declaração if-else. Ele pode ser útil para simplificar essas declarações, especialmente quando você está fazendo uma atribuição de variável ou gerando código HTML em resposta à condição if. Eis aqui o mesmo código para o botão de seleção "Adoro", reescrito para usar o operador ternário:

```
echo '<input type="radio" name="' . $response['response_id'] . '" value="1" ' .
($response['response_id'] == 1 ? ('checked="checked"' : '') . ' />Adoro ';
```

Este teste true/false controla o resultado do operador ternário.

Se o valor de resposta armazenado em \$response ['response'] for igual a 1, então o atributo checked será gerado como parte da tag <input>, resultando no seguinte botão de seleção "Adoro", marcado:

```
<input type="radio" name="279" value="1" checked="checked" />Love
```

Esta parte do código da tag <input> é controlada pelo operador ternário.

O atributo checked da tag <input> agora é gerado usando-se o operador ternário, em vez de uma declaração if-else.



Por outro lado, um valor de resposta diferente de 1 impedirá o atributo checked de ser gerado, resultando em uma tag <input> não marcada para o botão "Adoro".

O operador ternário ? : pode ser usado para se programar declarações if-else de uma forma mais compacta.

o script inteiro `questionnaire.php`

## Gere o formulário do questionário Mismatch

Agora nós já temos pedaços suficientes do questionário do Mismatch para usarmos o array de resposta (`$responses`) que criamos anteriormente para gerar o formulário HTML inteiro. Lembre-se, esse array foi construído obtendo-se as respostas atuais do usuário presentes na tabela `mismatch_response`. Vamos ver o código de geração do questionário no contexto do script `questionnaire.php` inteiro.

```
<?php
    // Inicia a sessão
    require_once('startsession.php');

    // Insere o cabeçalho da página
    $page_title = 'Questionnaire';
    require_once('header.php');

    require_once('appvars.php');
    require_once('connectvars.php');

    // Assegura que o usuário está logado antes de seguir adiante
    if (!isset($_SESSION['user_id'])) {
        echo '<p class="login">Por favor <a href="login.php">faça login para acessar esta página.</p>';
        exit();
    }

    // Mostra o menu de navegação
    require_once('navmenu.php');

    // Conecta-se ao banco de dados
    $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

    // Se este usuário jamais respondeu ao questionário, inserir respostas vazias no banco
    $query = "SELECT * FROM mismatch_response WHERE user_id = '" . $_SESSION['user_id'] . "'";
    $data = mysqli_query($dbc, $query);
    if (mysqli_num_rows($data) == 0) {
        // Primeiramente, obtém a lista de IDs dos tópicos a partir da respectiva tabela
        $query = "SELECT topic_id FROM mismatch_topic ORDER BY category_id, topic_id";
        $data = mysqli_query($dbc, $query);
        $topicIDs = array();
        while ($row = mysqli_fetch_array($data)) {
            array_push($topicIDs, $row['topic_id']);
        }

        // Insere linhas de respostas na tabela respectiva, uma para cada tópico
        foreach ($topicIDs as $topic_id) {
            $query = "INSERT INTO mismatch_response (user_id, topic_id) VALUES ('" . $_SESSION['user_id'] .
                "', '$topic_id')";
            mysqli_query($dbc, $query);
        }
    }

    // Se o questionário tiver sido submetido, escreve as respostas do formulário no banco
    if (isset($_POST['submit'])) {
        // Escreve as linhas de respostas do questionário na tabela de respostas
        foreach ($_POST as $response_id => $response) {
            $query = "UPDATE mismatch_response SET response = '$response' ";
        }
    }
}
```

Inclui os arquivos template que iniciam a sessão e exibem o cabeçalho da página.

Restringe a página aos usuários logados.

questionnaire.php

```

        "WHERE response_id = '$response_id'";
        mysqli_query($dbc, $query);
    }
    echo '<p>As suas respostas foram registradas..</p>';
}

// Obtém os dados de resposta do banco, para gerar o formulário
$query = "SELECT response_id, topic_id, response FROM mismatch_response WHERE user_id = ''";
$_SESSION['user_id'] . "";
$data = mysqli_query($dbc, $query);
$responses = array();
while ($row = mysqli_fetch_array($data)) {
    // Verifica o nome do tópico correspondente à resposta, na tabela dos tópicos
    $query2 = "SELECT name, category FROM mismatch_topic WHERE topic_id = '' .
    $row['topic_id'] .
    '';
    $data2 = mysqli_query($dbc, $query2);
    if (mysqli_num_rows($data2) == 1) {
        $row2 = mysqli_fetch_array($data2);
        $row['topic_name'] = $row2['name'];
        $row['category_name'] = $row2['category'];
        array_push($responses, $row);
    }
}

```

**3**

Obtém a categoria da primeira resposta, para iniciar o processo antes de entrar no loop.

Cada categoria é criada como um conjunto de campos, para ajudar a organizar os tópicos.

```
mysqli_close($dbc);
```

```

// Gera o formulário do questionário, fazendo uso através do array de respostas
echo "<form method='post' action=''. SERVER['PHP_SELF'] .'><br>";
echo "<p>Qual é sua opinião sobre cada tópico?</p> ";
$category = $responses[0]['category_name'];
echo "<fieldset><legend>". $responses[0]['category_name'] . "</legend>";
```

**4**

Aqui, o operador ternário é usado para se modificar o rótulo para tópicos não respondidos.

```

foreach ($responses as $response) {
    // Se inicia um novo conjunto de campos se a categoria tiver se modificado
    if ($category != $response['category_name']) {
        $category = $response['category_name'];
        echo "</fieldset><fieldset><legend>". $response['category_name'] . "</legend>";
```

```

    // Exibe o campo do tópico no formulário
    echo "<label>". ($response['response'] == NULL ? "class='error'" : '') . " for='".
        $response['response_id'] . "'>". $response['topic_name'] . "</label>";
```

```

    echo "<input type='radio' id='".$response['response_id']."' name='".
        $response['response_id']."' value='1'". ($response['response'] == 1 ? ' checked="checked"' : '') . ">Love";
    echo "<input type='radio' id='".$response['response_id']."' name='".
        $response['response_id']."' value='2'". ($response['response'] == 2 ? ' checked="checked"' : '') . ">Hate<br>";
```

```

    echo "</fieldset>";
    echo "<input type='submit' value='Salvar o Questionário' name='submit' />";
    echo "</form>";
```

```

// Insere a página
require_once('footer.php');
```

?>

Cada uma destas instruções echo gera um botão de seleção - um para "Adoro" e outro para "Odeio".

Lembre-se, 1 = adoro e 2 = odeio.

Cada tópico é criado como um rótulo seguido de botões de seleção "Adoro" e "Odeio".

**FEITO**  
Gere o formulário HTML com o questionário a partir dos dados de resposta.



## TEST DRIVE

### Teste o novo questionário do Mismatch.

Modifique o Mismatch para que ele use o novo script do questionário. (ou baixe a aplicação no site da Alta Books, em [www.altabooks.com.br](http://www.altabooks.com.br)). Para isso, você terá de criar um novo script *questionnaire.php*, bem como adicionar um item de menu “Questionário” ao script *navmenu.php*, para que os usuários possam acessá-lo.

Envie os scripts para o seu servidor web e depois abra a página principal do Mismatch (*index.php*) em um navegador. Certifique-se de fazer login, e então clique no item de menu “Questionário” para acessá-lo. Repare que nenhum dos tópicos tem repostas, uma vez que esta é a sua primeira visita ao questionário. Responda aos itens e submeta o formulário. Retorne à página principal, e depois volte ao questionário para confirmar que as suas respostas foram corretamente carregadas a partir do banco de dados.

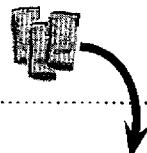
How do you feel about each topic?

Appearance	
Tattoos:	<input checked="" type="radio"/> Love <input type="radio"/> Hate
Gold chains:	<input checked="" type="radio"/> Love <input type="radio"/> Hate
Body piercings:	<input checked="" type="radio"/> Love <input type="radio"/> Hate
Cowboy boots:	<input checked="" type="radio"/> Love <input type="radio"/> Hate
Long hair:	<input checked="" type="radio"/> Love <input type="radio"/> Hate
Entertainment	
Reality TV:	<input checked="" type="radio"/> Love <input type="radio"/> Hate
Professional wrestling:	<input checked="" type="radio"/> Love <input type="radio"/> Hate
Horror movies:	<input checked="" type="radio"/> Love <input type="radio"/> Hate

O script do questionário permite que os usuários respondam a itens adoro/odeio e armazenem os resultados no banco de dados.

As questões no formulário são geradas dinamicamente a partir do banco de dados  
– se você adicionar novos tópicos, o formulário se modificará.

Faça o Download!



O código-fonte completo da aplicação Mismatch está disponível para download no site da Alta Books:

[www.altabooks.com.br](http://www.altabooks.com.br)

não existem  
Perguntas Idiotas

**P:** Como o botão de seleção "Adoro" sabe que o resultado do operador ternário é uma string?

**R:** O operador ternário sempre resolve em uma das duas instruções separadas pelos dois-pontos, com base no valor (true ou false) da expressão de teste. Se essas instruções forem strings, então o resultado do operador será uma string. É isso que torna o operador tão útil – você pode inseri-lo no meio de uma atribuição ou concatenação.

**P:** O operador ternário faz o meu script rodar mais rápido?

**R:** Não, provavelmente não. O operador ternário serve para adicionar eficiência estilística ao seu código, e não exatamente para melhorar a eficiência do desempenho, o que significa dizer que a sua vantagem é exigir menos linhas de código. Em alguns casos, é mais conciso usar o operador ternário do que uma declaração if-else completa, embora os dois sejam logicamente equivalentes. Mesmo assim, não fique muito entusiasmado com o operador ternário, porque ele pode tornar o código mais difícil de entender, caso você esteja tentando substituir uma declaração if-else complexa. A ideia é usar o operador ternário em lugares onde a eliminação do if-else pode realmente ajudar a simplificar o código, em vez de torná-lo mais complicado. Isso geralmente envolve usar o operador para controlar seletivamente um valor sendo atribuído a uma variável ou inserido em uma expressão. No caso dos botões de seleção do Mismatch, esta última abordagem foi usada para se controlar seletivamente a inserção de um atributo HTML (checked).

**P:** Como é possível gerar o questionário do Mismatch a partir da tabela mismatch\_response, quando o usuário ainda não respondeu a nenhum tópico?

**R:** Excelente pergunta. O formulário do questionário tem de lidar com dois cenários possíveis: o usuário está respondendo ao questionário pela primeira vez, ou o usuário já o respondeu e está revendo as respostas. No primeiro cenário, não há nenhuma resposta ainda, portanto a tabela mismatch\_response ainda não tem dados para este usuário. Mas nós ainda temos de gerar o formulário dinamicamente. Poderíamos usar a tabela mismatch\_topic para isso. Porém, não funcionaria para o segundo cenário, porque desta vez o formulário precisa ser gerado com base nas respostas específicas do usuário; lembre-se, os botões de seleção para "Adoro" e "Odeio" são gerados como parte do formulário. Assim, nós temos um problema, no sentido que o código para gerar o formulário é completamente diferente, dependendo de se os usuários já responderam ou não o questionário. Não apenas isso, e se o usuário tiver respondido apenas a algumas questões? Pode ficar muito complicado bem rápido. A solução adotada pelo Mismatch é preencher antecipadamente a tabela mismatch\_response com respostas em branco, na primeira vez que o usuário acessa o questionário. Isso nos permite sempre gerar o formulário a partir da tabela mismatch\_response, e não precisamos nos preocupar com a complicaçāo de gerar o formulário diferentemente dependendo de se o usuário já respondeu ou não ou de quais tópicos ele tenha respondido. É claro que o código para geraçāo do formulário ainda não é exatamente trivial, mas é mais simples do que se não tivéssemos usado esta abordagem.

Faça esta  
Experiencia!

Para simplificar o código, o Mismatch não se ajusta automaticamente a novos tópicos, pelo menos não no que se refere a usuários que já tenham respondido o questionário. Assim, você terá de esvaziar a tabela mismatch\_response após adicionar um novo tópico.

- 1 Adicione um novo tópico à sua própria tabela mismatch\_topic, com esta instrução SQL:

```
INSERT INTO mismatch_topic  
    (name, category) VALUES  
    ('Virtual guitars', 'Activities')
```

- 2 Esvazie todos os dados da tabela mismatch\_response, com esta instrução SQL:

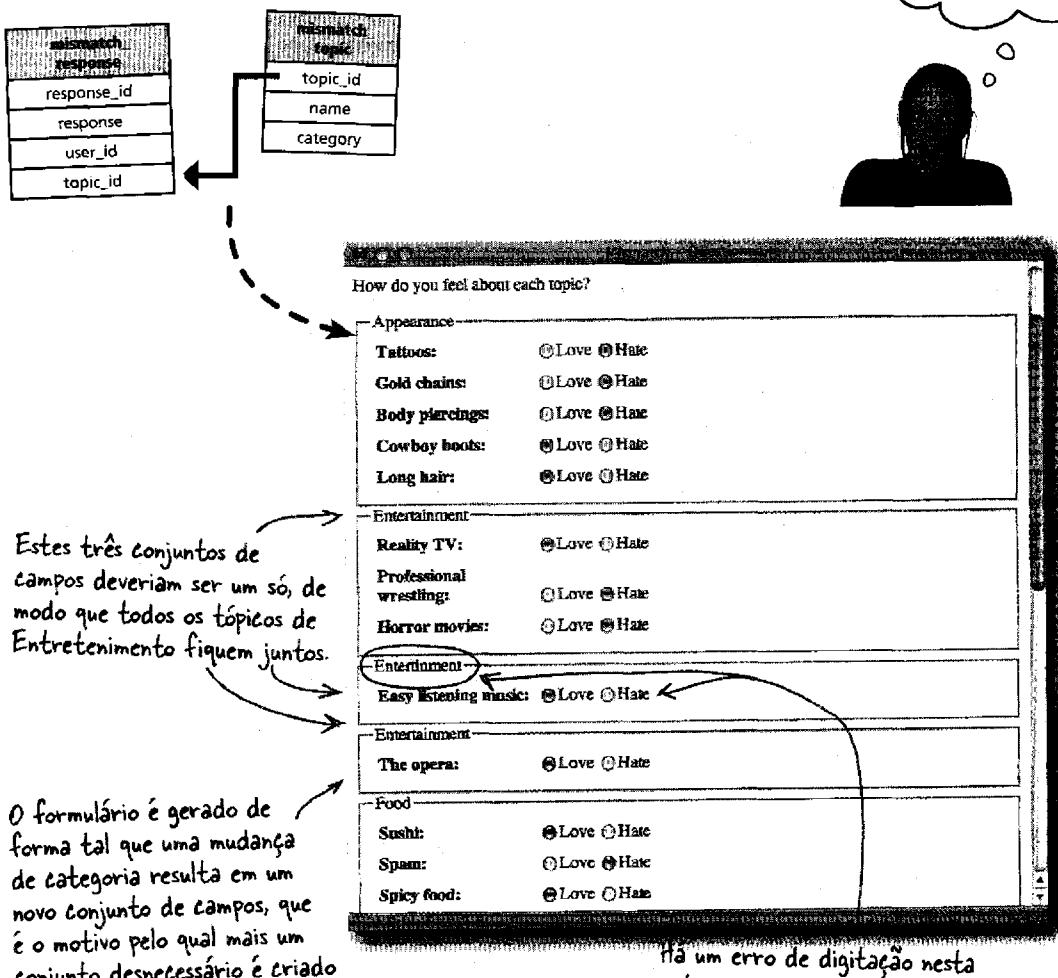
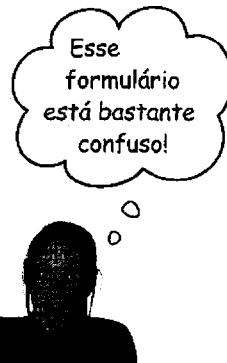
```
DELETE FROM mismatch_response
```

- 3 Visualize o questionário na aplicação Mismatch para ver o novo tópico.

- 4 Responda ao novo tópico, submeta o formulário e verifique a sua resposta gravada.

## Agora os dados estão comandando o formulário

Foi preciso algum trabalho, mas agora a aplicação Mismatch está gerando dinamicamente o questionário a partir de respostas armazenadas no banco de dados. Isso significa que quaisquer modificações feitas ao banco serão automaticamente refletidas no formulário – essa é a vantagem de se gerar a interface de usuário de uma aplicação web a partir de um banco de dados. Mas o que acontece quando nós temos dados inválidos?



Os dados estão determinando o formulário como queríamos, mas algo ainda está errado. Parece que uma das categorias foi escrita no banco de dados com um erro de digitação, fazendo o código PHP gerar um conjunto de campos separado para ela. Isso é um grande problema, porque arruina o efeito de se usar conjuntos para ajudar a organizar o formulário e facilitar a resposta aos tópicos.



Nós já sabemos que esta categoria com erro de digitação está causando problemas....

...e aqui temos mais uma causando o mesmo problema no formulário.

### mismatch\_topic

topic_id	name	category
8	Horror movies	Entertainment
9	Easy listening music	Entertainment
10	The opera	Entertainment
11	Sushi	Food
12	Spam	Food
13	Spicy food	Food
14	Peanut butter & banana sandwiches	Food
15	Martini	Food
16	Howard Stern	People
17	Bill Gates	People
18	Barbara Streisand	People

People
Howard Stern: <input checked="" type="radio"/> Love <input type="radio"/> Hate
Bill Gates: <input type="radio"/> Love <input checked="" type="radio"/> Hate
Barbara Streisand: <input type="radio"/> Love <input checked="" type="radio"/> Hate

**Frank:** É fácil. É só corrigir a grafia da categoria na tabela mismatch\_topic.

**Joe:** Mas tem mais de uma categoria escrita errada. E agora que pensei nisso, não entendo por que os nomes das categorias têm de ser armazenados mais de uma vez.

**Jill:** Concordo. Nós nos demos ao trabalho de eliminar dados duplicados ao projetarmos o esquema do banco de dados, e aqui estamos nós com um monte de nomes de categorias duplicados. E não apenas isso, mas temos até alguns nomes com erros de digitação.

**Frank:** Ok, e se nos livrássemos dos nomes das categorias, e talvez fizéssemos a referência a elas por números? Assim não haveria o risco de termos erros de digitação.

**Joe:** É verdade, mas ainda precisamos dos nomes das categorias para usá-los como títulos no questionário.

**Jill:** Talvez possamos nos referir às categorias por número, sem jogar fora os nomes. É mais ou menos o que já estamos fazendo com os tópicos na tabela mismatch\_topic, não é?

**Joe:** Certo! Nós não queríamos armazenar um monte de nomes de tópicos duplicados na tabela mismatch\_response, e portanto, colocamos esses nomes na tabela mismatch\_topic, e vinculamos os tópicos às respostas usando chaves numéricas.

**Frank:** Vocês estão dizendo que poderíamos resolver o problema dos nomes de categorias duplicados criando uma nova tabela para as categorias?

**Jill:** É exatamente isso. Nós podemos criar uma nova tabela mismatch\_category, onde cada nome de categoria é armazenado exatamente uma vez. E depois, conectamos as categorias aos tópicos usando chaves primárias e estrangeiras entre mismatch\_topic e mismatch\_category. Brilhante!

## Procure por um pouco de normalidade

O processo de reelaborar o banco de dados do Mismatch de modo a eliminar dados duplicados, e a dividir e conectar tabelas de uma forma lógica e consistente é conhecido como normalização. A normalização é um assunto relativamente aprofundado no que diz respeito ao projeto de bancos de dados, e pode ser um tanto intimidante. Mas não precisa ser. Existem várias técnicas simples de projeto de bancos de dados, das quais nós podemos nos aproveitar para tornar os nossos bancos MySQL muito melhores do que se simplesmente tentássemos adivinhar como é que os dados deveriam ser dispostos.

Eis alguns passos gerais que você pode usar para iniciar o processo de design do banco de dados, os quais levarão naturalmente a um banco de dados mais "normal":

**1. Escolha uma coisa, apenas uma coisa que você queira que a tabela descreva.**

Qual é a principal coisa que você quer que seja o assunto da sua tabela?

**2. Faça uma lista das informações que você precisa saber sobre essa coisa, ao usar a tabela.**

Como você irá usar a tabela?

**3. Usando a lista, divida as informações sobre essa coisa em pedaços que você possa usar para organizar a tabela.**

Qual é o modo mais fácil de consultar essa tabela?

Um conceito fundamental na normalização é a ideia dos dados atômicos, que são dados divididos até a sua menor forma possível que faça sentido, considerando-se o uso do banco de dados. Por exemplo, as colunas `first_name` e `last_name` do banco de dados do Mismatch são atômicas no sentido de que elas dividem o nome do usuário em mais pedaços (primeiro nome e sobrenome) do que uma única coluna para o nome inteiro o faria. Isso é necessário no Mismatch porque queremos poder nos referir ao usuário apenas pelo seu primeiro nome.

Talvez não seja sempre necessário que a aplicação divida o nome em colunas separadas para primeiro nome e sobrenome, porém, em cujo caso uma tabela "name" sozinha seria atômica o suficiente. Assim, quando estiver dividindo a "coisa" da tabela em partes, pense em como os dados serão usados, e não apenas no que eles representam.

Normalizar significa elaborar um banco de dados de modo a reduzir a duplicação de dados e a melhorar as relações entre os dados.

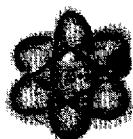


Dados atômicos são dados que foram divididos até a menor forma necessária para um determinado banco de dados.

## Ao normalizar, pense em termos de átomos

Para ajudar a transformar as suas ideias sobre o projeto do banco em ações, é útil fazer perguntas específicas aos seus dados. Isso ajudará a determinar como os dados se encaixam na tabela, e permitirá saber se eles realmente foram divididos até a sua representação atômica apropriada. Ninguém disse que dividir o átomo seria fácil, mas esta lista de perguntas poderá lhe ajudar.

Tornar os seus dados atômicos é o primeiro passo para se criar uma tabela normalizada.



### 1. Qual é a **principal coisa** que a sua tabela descreve?

A sua tabela descreve avistamentos de OVNIs, assinaturas de uma lista de emails, pontuações máximas de um videogame, nomes de românticos incuráveis?



### 2. Como você **usará** a tabela para chegar até essa **principal coisa**?

Elabore sua tabela de modo que ela seja fácil de consultar!



### 3. As suas **colunas** contêm **dados atômicos**, para tornar suas consultas curtas e diretas?

Certifique-se de que os dados sejam apenas tão pequenos quanto necessário.

não existem

### Perguntas Ídiotas

**P:** Eu devo tentar dividir os meus dados até os menores pedaços possíveis?

**R:** Não necessariamente. Tornar os seus dados atômicos significa dividi-los até os menores pedaços de que você precisa para criar uma tabela eficiente, e não até os menores pedaços possíveis.

Não dividia os seus dados mais do que realmente precisar. Se não precisar de colunas extras, não as adicione à tua.

**P:** Como os dados atômicos me ajudam?

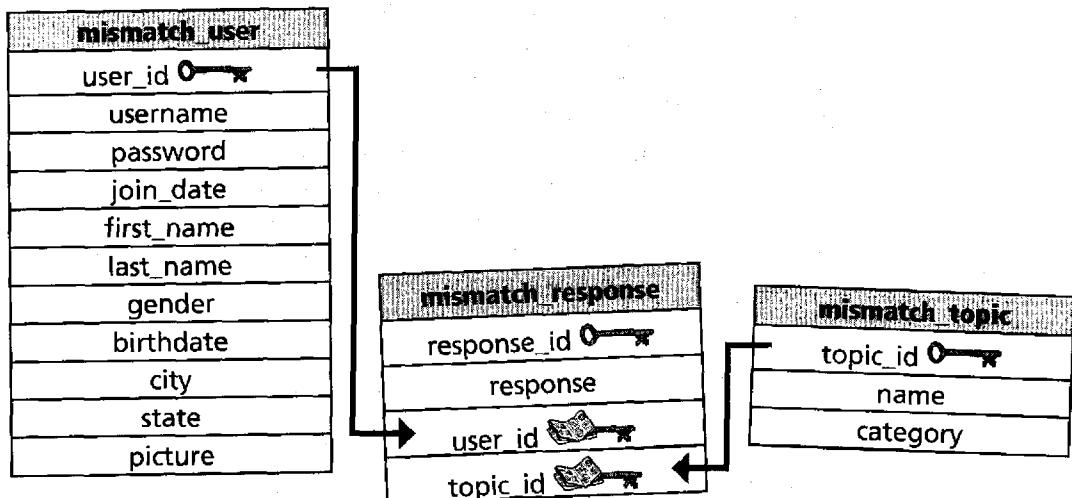
**R:** Eles lhe ajudam a garantir que os dados da sua tabela sejam precisos. Por exemplo, se você tiver uma coluna para o endereço de um avistamento de ETs, poderá querer dividir o endereço em duas colunas: uma para a rua e uma para o número. Então, você poderá certificar-se de registrar apenas números na coluna referente ao número.

Os dados atômicos também lhe permitem executar consultas de modo mais eficiente, porque as consultas ficam mais fáceis de escrever e rodam em menos tempo, o que é eficiente quando você tem uma quantidade muito grande de dados armazenados.

*normalize a banco de dados do mismatch*



O banco de dados do Mismatch está precisando ser normalizado, para resolver o problema de nomes de categorias duplicados. Dada a estrutura existente do banco, esboce um projeto modificado que resolva esse problema, eliminando o risco de erros na inserção dos dados. Escreva comentários no projeto, explicando como ele funciona.



não existem  
**Perguntas Ídiotas**

**P:** Como eu faço para aplicar o terceiro passo de normalização ao Mismatch, para consertar o problema hipotético com a cidade/estado/CEP?

**R:** A solução é colocar os dados de localização do usuário em uma tabela própria, e então conectar a tabela mismatch\_user à nova tabela por meio de uma chave estrangeira. Assim, você poderia criar uma tabela chamada mismatch\_location, com uma chave primária location\_id, junto com colunas para armazenar a cidade e o estado do usuário, por exemplo. Então, as colunas city ("cidade") e state ("estado") são removidas de mismatch\_user e substituídas por uma chave estrangeira location\_id. Problema resolvido! O que faz esse design funcionar é que a coluna location\_id na verdade usa o CEP como chave primária, eliminando o problema de dependência de colunas que não sejam chaves.

**P:** Caramba, isso parece um monte de trabalho só para atender a um requerimento de projeto exigente. Isso é realmente necessário?

**R:** Sim e não. Os dois primeiros passos da normalização são obrigatórios, porque os dados atômicos e as chaves primárias são essenciais para qualquer bom projeto de bancos de dados. É no terceiro passo que você passa a ter que pesar a atratividade de um projeto perfeito, por um lado; contra as realidades práticas do que a aplicação realmente precisa, por outro. No caso do problema com cidade/estado/CEP do Mismatch, provavelmente vale a pena aplicar esse requerimento, pela simplicidade que ele proporciona. Essa não é uma decisão a ser tomada sem pensar bem, e muitos puristas defendem que você deve aderir estritamente a todos três passos. A boa notícia é que a coluna com o CEP é puramente hipotética, e não faz parte realmente da tabela mismatch\_user, portanto não precisamos realmente nos preocupar com ela.

**P:** Mesmo sem uma coluna para o CEP, não seria necessário mover city ("cidade") e state ("estado") para as suas próprias tabelas, para atender ao terceiro passo?

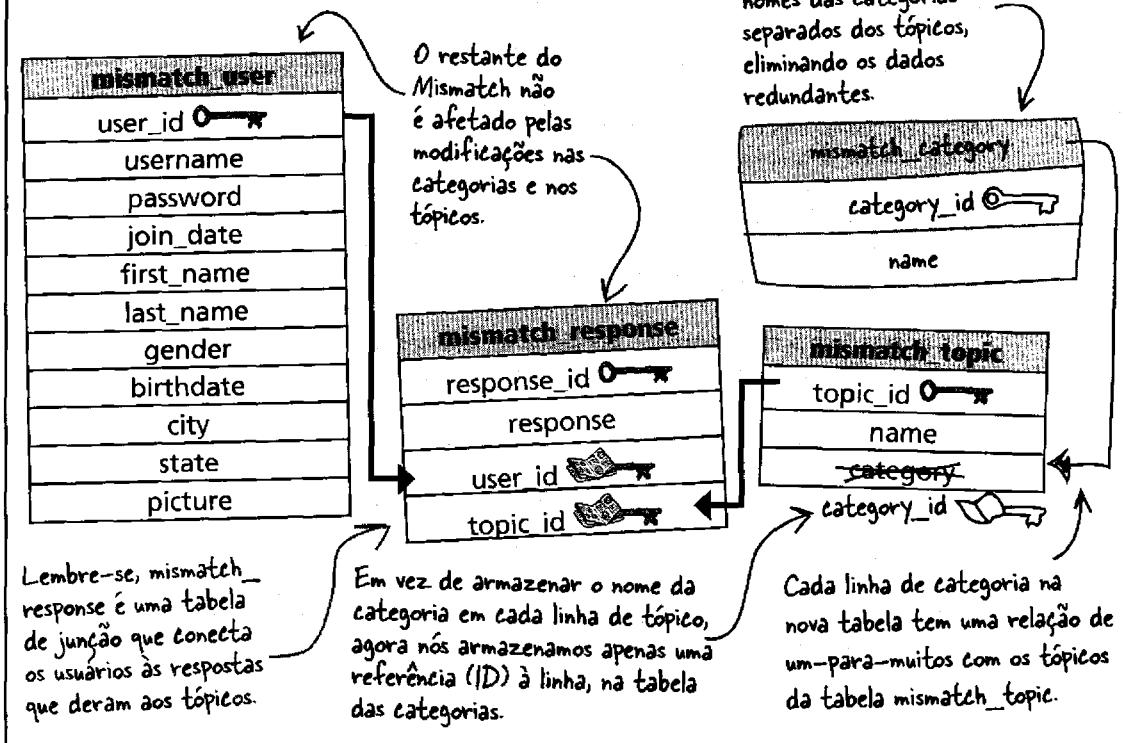
**R:** Possivelmente. Sem dúvida você acabará tendo dados duplicados, referentes a cidades e estados, na tabela mismatch\_user. O problema em separar cidade e estado numa tabela própria, sem uso do CEP, é que você teria de preencher essas tabelas com todas as cidades e estados existentes. Caso contrário, os usuários sem dúvida escreveriam alguns nomes de cidades com erros ortográficos, e ainda acabaria tendo dados problemáticos. Este é um bom exemplo de caso em que você tem que pesar seriamente o benefício da normalização estrita contra a realidade de uma aplicação real. Uma interessante solução possível para todos os problemas é usar o CEP na tabela mismatch\_user, em vez de cidade e estado, e então procurar a cidade e o estado em uma tabela estática ou em algum serviço web, conforme necessário. Isso é mais complexidade do que precisamos no momento, portanto vamos ficar com as colunas city (cidade) e state (estado).

## a base de dados mismatch—agora normalizada!



### Exercício Solução

O banco de dados do Mismatch está precisando ser normalizado, para resolver o problema de nomes de categorias duplicados. Dada a estrutura existente do banco, esboce um projeto modificado que resolva esse problema, eliminando o risco de erros na inserção dos dados. Escreva comentários no projeto, explicando como ele funciona.



## não existem Perguntas Idiotas

**P:** Como exatamente a nova tabela `mismatch_category` resolve o problema de dados duplicados?

**R:** A nova tabela separa os nomes das categorias da tabela `mismatch_topic`, permitindo que eles sejam armazenados isoladamente. Com as categorias armazenadas em tabela própria, não é mais necessário duplicar os seus nomes – você tem uma linha para cada categoria, e essas linhas são então referenciadas por linhas da tabela `mismatch_topic`. Isso significa que as linhas de categorias da tabela `mismatch_category` têm uma relação de um-para-muitos com as linhas de tópicos da tabela `mismatch_topic`.

**P:** Então isso significa que `mismatch_category` tem apenas cinco linhas, uma para cada categoria?

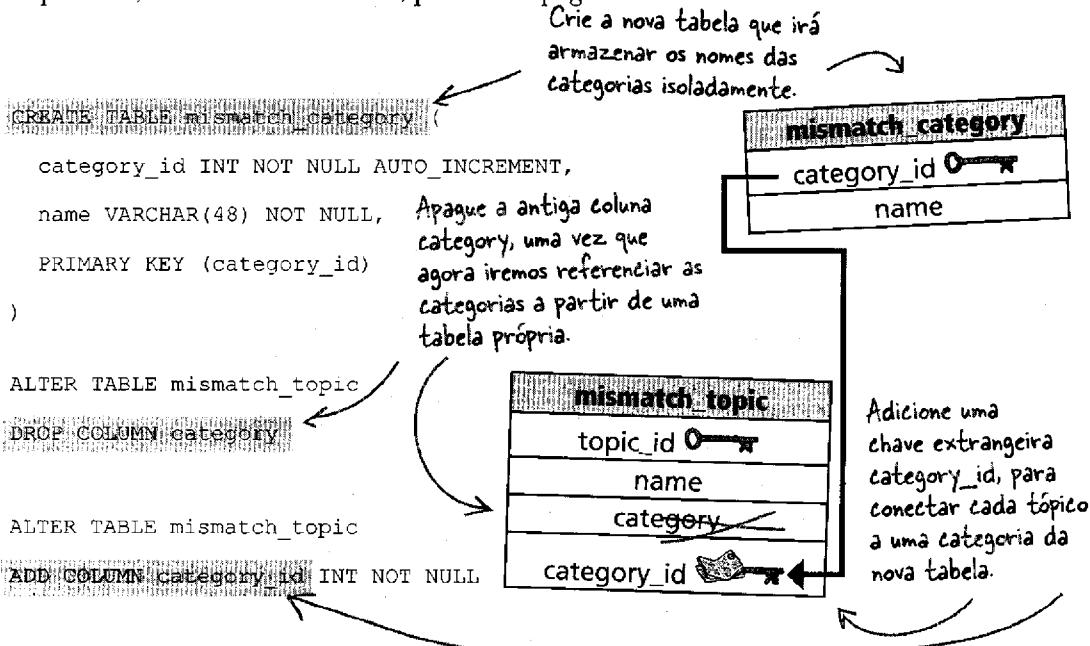
**R:** Sim, isso mesmo: `mismatch_category`

category_id	name
1	Appearance
2	Entertainment
3	Food
4	People
5	Activities

O nome de cada categoria só é armazenado uma vez!

# Alterando o banco de dados do Mismatch

Para tirar proveito do novo schema, o banco de dados do Mismatch requer algumas modificações estruturais. Mais especificamente, nós precisamos criar uma tabela mismatch\_category, e depois conectá-la a uma nova chave estrangeira na tabela mismatch\_topic. E uma vez que a velha coluna category da tabela mismatch\_topic, contendo os dados duplicados, não é mais necessária, podemos apagá-la.



A nova tabela mismatch\_category precisa ser preenchida com os dados referentes às categorias, o que é feito através de algumas instruções INSERT.

```

INSERT INTO mismatch_category (name) VALUES ('Appearance')
INSERT INTO mismatch_category (name) VALUES ('Entertainment')
INSERT INTO mismatch_category (name) VALUES ('Food')
INSERT INTO mismatch_category (name) VALUES ('People')
INSERT INTO mismatch_category (name) VALUES ('Activities')
  
```

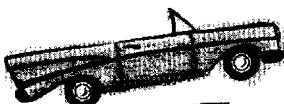
A nova coluna category\_id precisa, então, ser preenchida com dados para ligar corretamente cada tópico à sua categoria apropriada da tabela mismatch\_category.

```

UPDATE mismatch_topic SET category_id = 3
WHERE name = 'Martinis'
  
```

Esta ID deve ser a mesma ID autoincrementada que a categoria tem na tabela mismatch\_category.

**test drive as tabelas normalizadas mismatch**



## TEST DRIVE

### Crie e preencha a nova tabela mismatch\_category.

Usando alguma ferramenta MySQL, execute o comando SQL CREATE TABLE da página anterior para adicionar uma nova tabela, chamada mismatch\_category, ao banco de dados do Mismatch. Em seguida, emita instruções INSERT para preencher a tabela com dados referentes às categorias. Agora, execute as duas instruções ALTER para modificar a tabela mismatch\_topic para que ela tenha uma coluna category\_id. Finalmente, use UPDATE em cada linha da tabela mismatch\_topic, para que as suas colunas category\_id apontem para a categoria correta na tabela mismatch\_category.

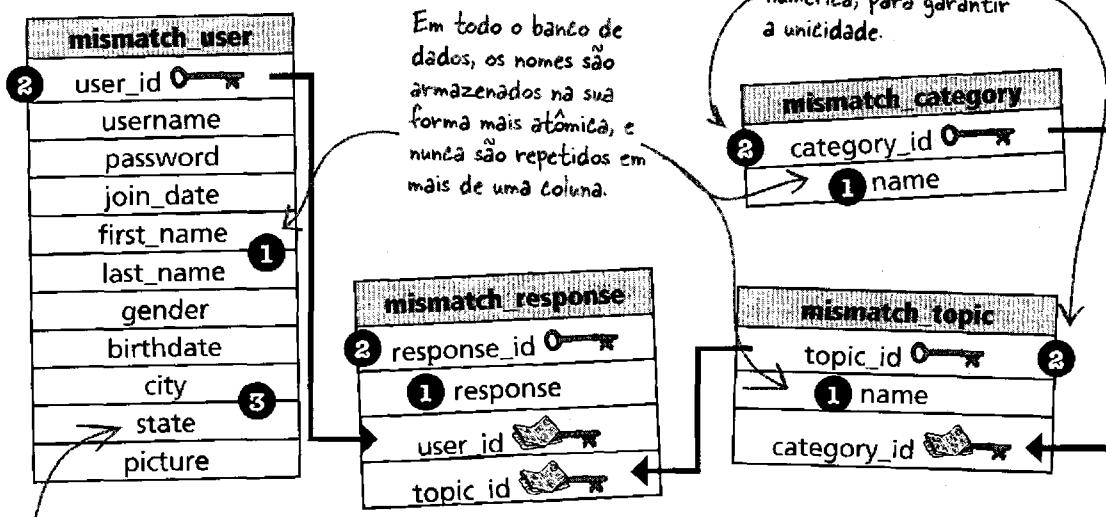
Agora execute um SELECT em cada uma das tabelas, só para certificar-se de que está tudo certo.

## Então, o Mismatch realmente é normal?

Sim. Se você aplicar as três regras principais da normalidade a cada uma das tabelas do Mismatch, verá que ele passa no teste com louvor. Mas mesmo se não passasse, nem tudo estaria perdido. Assim como nas pessoas, existem graus de normalidade no que se refere aos bancos de dados. O importante é tentar elaborar bancos que sejam completamente normais, só aceitando menos que isso quando houver um motivo muito bom para burlar as regras.

- ① Certifique-se de que as suas colunas sejam atômicas.
- ② Dê a cada tabela a sua própria chave primária.
- ③ Certifique-se de as colunas que não são chaves não sejam dependentesumas das outras.

Todas as tabelas têm uma chave primária numérica, para garantir a unicidade.



Sem a hipotética dependência do CEP, as colunas que se referem à localização do usuário não têm mais problemas de dependências.

O novo projeto das tabelas do Mismatch não afeta as consultas que estão sendo feitas no código do script do questionário?

mismatch_category	
category_id	name

```
...
// Obtém os dados de resposta do banco para gerar o formulário
$query = "SELECT response_id, topic_id, response FROM mismatch_response " .
    "WHERE user_id = '" . $_SESSION['user_id'] . "'";
$data = mysqli_query($dbc, $query);
$responses = array();
while ($row = mysqli_fetch_array($data)) {
    // Verifica o nome do tópico correspondente à resposta na tabela dos tópicos
    $query2 = "SELECT name, category FROM mismatch_topic " .
        "WHERE topic_id = '" . $row['topic_id'] . "'";
    $data2 = mysqli_query($dbc, $query2);
    if (mysqli_num_rows($data2) == 1) {
        $row2 = mysqli_fetch_array($data2);
        $row['topic_name'] = $row2['name'];
        $row['category_name'] = $row2['category'];
        array_push($responses, $row);
    }
}
...

```

questionnaire.php

**Sim. De fato, a maioria das modificações estruturais em bancos de dados exige que nós ajustemos quaisquer consultas que envolvam as tabelas afetadas.**

Neste caso, modificar o projeto do banco para adicionar a tabela mismatch\_category afeta qualquer consulta envolvendo a tabela mismatch\_topic. Isso ocorre porque o projeto anterior tinha as categorias armazenadas diretamente na tabela mismatch\_topic. Após colocarmos as categorias na sua própria tabela, o que nós agora sabemos é uma ótima ideia graças à normalização, torna-se necessário rever as consultas e programá-las para trabalhar com uma nova tabela (mismatch\_category).

## Ligue com pontos

Uma vez que os joins envolvem mais de uma tabela, é importante ser claro sobre cada coluna referenciada em um join. Mais especificamente, você deve identificar a tabela de cada coluna, para não haver nenhuma confusão – diferentes tabelas frequentemente têm colunas com os mesmos nomes, principalmente no que se refere às chaves. Basta prefixar o nome da coluna com o nome da tabela e um ponto. Por exemplo, eis aqui a consulta INNER JOIN anterior, que cria um conjunto de resultados de IDs de tópicos e nomes de categorias:

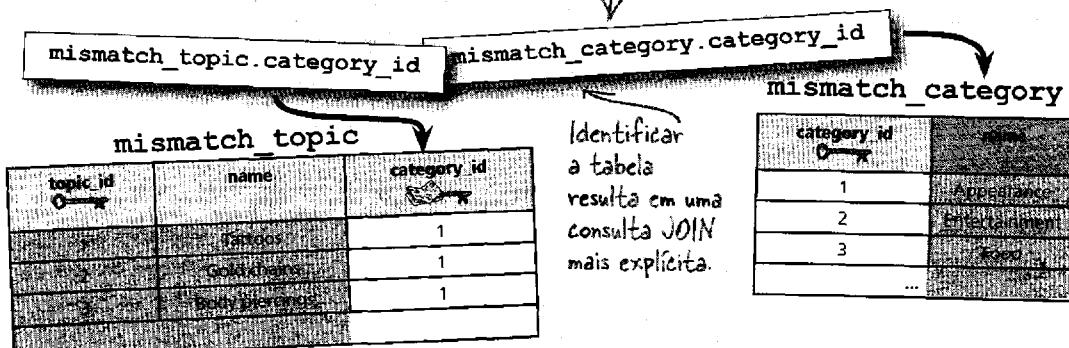
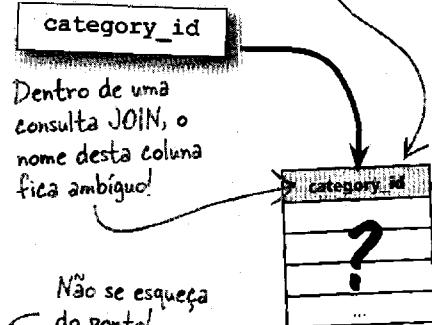
```
Este é o nome da tabela.          O ponto!
SELECT mismatch_topic.topic_id, mismatch_category.name
  FROM mismatch_topic
    INNER JOIN mismatch_category
      ON mismatch_topic.category_id = mismatch_category.category_id
```

Este é o nome da coluna dentro da tabela, separado do nome da tabela por um ponto.

Eis outra referência a tabela/coluna usando a notação com ponto.

Sem a capacidade de especificar as tabelas associadas com as colunas, nesta consulta, nós teríamos uma ambiguidade problemática. Na verdade, seria impossível entender a parte ON da consulta, porque ela estaria verificando se a coluna category\_id é igual a ela mesma, presumivelmente dentro da tabela mismatch\_topic. Por esse motivo, é sempre uma boa ideia ser bastante explícito na hora de identificar as tabelas associadas com colunas, ao criar consultas JOIN.

Sozinho, o nome da coluna não nos diz nada sobre a qual tabela ele pertence.



## Certamente podemos fazer mais com os inner joins

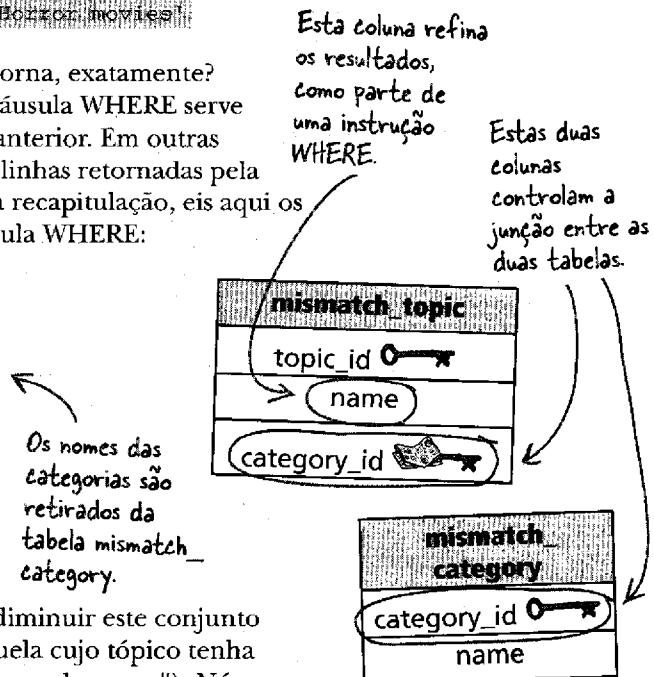
Os inner joins não servem apenas para combinar dados de duas tabelas. Uma vez que um inner join é, em última instância, apenas uma consulta, você ainda pode usar construtos normais de consultas para controlar ainda mais os resultados. Por exemplo, caso queira obter uma linha específica do conjunto de resultados unidos, você pode colocar uma instrução WHERE na consulta INNER JOIN para isolar apenas essa linha.

```
SELECT mismatch_topic.topic_id, mismatch_category.name
  FROM mismatch_topic
    INNER JOIN mismatch_category
      ON (mismatch_topic.category_id = mismatch_category.category_id)
     WHERE mismatch_topic.name = 'Horror movies'
```

Então, o que é que esta consulta retorna, exatamente? Lembre-se, primeiramente, que a cláusula WHERE serve como um refinamento da consulta anterior. Em outras palavras, ela restringe o número de linhas retornadas pela consulta INNER JOIN original. Para recapitulação, eis aqui os resultados do inner join sem a cláusula WHERE:

1	Appearance
2	Appearance
3	Appearance
4	Appearance
5	Appearance
6	Entertainment
7	Entertainment
8	Entertainment
...	

As IDs dos tópicos são extraídas da tabela mismatch\_topic.



A cláusula WHERE tem o efeito de diminuir este conjunto de resultados a uma única linha, aquela cujo tópico tenha o nome igual a 'Horror movies' ("Filmes de terror"). Nós temos de olhar novamente na tabela mismatch\_topic para ver qual é essa linha.

topic_id	name	category_id
...		
7	Professional wrestling	2
8	Horror movies	2
9	Easy listening music	2
...		

Esta linha bate com a cláusula WHERE.

O conjunto de resultados da INNER JOIN original é reduzido a esta linha solitária, devido à cláusula WHERE.

A cláusula WHERE restringe os resultados da junção a apenas uma linha.

## **Joins, ao trabalho!**

Assim, os joins possibilitam envolver mais de uma tabela em uma consulta, na prática retirando dados de mais de um lugar e colocando-se em uma única tabela de resultados. A consulta do Mismatch que cria um array de respostas é um candidato perfeito para joins, uma vez que ela contém nada menos do que três consultas aninhadas para lidar com várias tabelas. Vamos começar com o código original:

```
// Obtém os dados de resposta do banco, para gerar o formulário
$query = "SELECT response_id, topic_id, response FROM mismatch_response "
        "WHERE user_id = '" . $_SESSION['user_id'] . "' ";
$data = mysqli_query($dbc, $query);
$responses = array();
while ($row = mysqli_fetch_array($data)) {
    // Verifica o nome do tópico correspondente à resposta, na tabela dos
    tópicos
    $query2 = "SELECT name, category_id FROM mismatch_topic "
              "WHERE topic_id = '" . $row['topic_id'] . "' ";
    $data2 = mysqli_query($dbc, $query2);
    if (mysqli_num_rows($data2) == 1) {
        $row2 = mysqli_fetch_array($data2);
        $row['topic_name'] = $row2['name'];
    }
    // Verifica o nome da categoria correspondente ao tópico, na
    tabela das categorias
    $query3 = "SELECT name FROM mismatch_category "
              "WHERE category_id = '" . $row2['category_id'] . "' ";
    $data3 = mysqli_query($dbc, $query3);
    if (mysqli_num_rows($data3) == 1) {
        $row3 = mysqli_fetch_array($data3);
        $row['category_name'] = $row3['name'];
        array_push($responses, $row);
    }
}

Com um join,
tanto o nome
da categoria
```

E aqui vai a nova versão do código, usando uma junção:

```
// Obtém os dados de resposta do banco, para gerar o formulário
$query = "SELECT response_id, topic_id, response FROM mismatch_response "
        . "WHERE user_id = '" . $_SESSION['user_id'] . "'";
$data = mysqli_query($dbc, $query);
$responses = array();
while ($row = mysqli_fetch_array($data)) {
    // Verifica os nomes do tópico e da categoria correspondentes à resposta,
    // respectivas tabelas
    $query2 = "SELECT mc.name AS topic_name, mcu.name AS category_name "
        . "FROM mismatch_topic AS mt "
        . "INNER JOIN mismatch_category AS mc USING (category_id) "
        . "WHERE mt.topic_id = '" . $row['topic_id'] . "'";
    $data2 = mysqli_query($dbc, $query2);
    if (mysqli_num_rows($data2) == 1) {
        $row2 = mysqli_fetch_array($data2);
        $row['topic_name'] = $row2['topic_name'];
        $row['category_name'] = $row2['category_name'];
        array_push($responses, $row);
    }
}
```

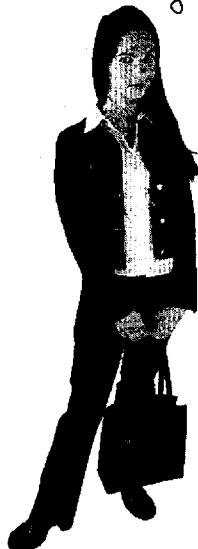
Os joins são  
mais eficientes  
e requerem  
menos código  
do que consultas  
aninhadas.

As duas últimas consultas do código são responsáveis por obter os nomes do tópico e da categoria das suas respectivas tabelas – uma consulta por tabela.

Com um join, é possível obter tanto o nome do tópico quanto o da categoria, em uma só consulta.

São usados aliases para ajudar a simplificar o código.

- A ID do tópico é usado como base para a consulta principal, mas a ID da categoria controla o próprio join.



Não entendo, ainda temos uma consulta extra que procura o nome da categoria. Se os joins continuam tão bons, por que ainda precisamos de duas consultas?

### Nós não precisamos de duas consultas, pelo menos não se usarmos todo o potencial dos joins.

É possível unir mais de duas tabelas, que é o que realmente precisamos fazer no código do array de respostas do Mismatch. Precisamos de uma única consulta que realize três coisas: obter todas as respostas para o usuário, obter o nome do tópico para cada resposta, e finalmente obter o nome da categoria para cada resposta. O novo e melhorado código da página anterior realiza as duas últimas em uma só consulta, que envolve um join entre as tabelas mismatch\_topic e mismatch\_category. O ideal é que tivéssemos uma única consulta, com dois joins, para acertar os três coelhos com apenas uma cajadada em forma de join.

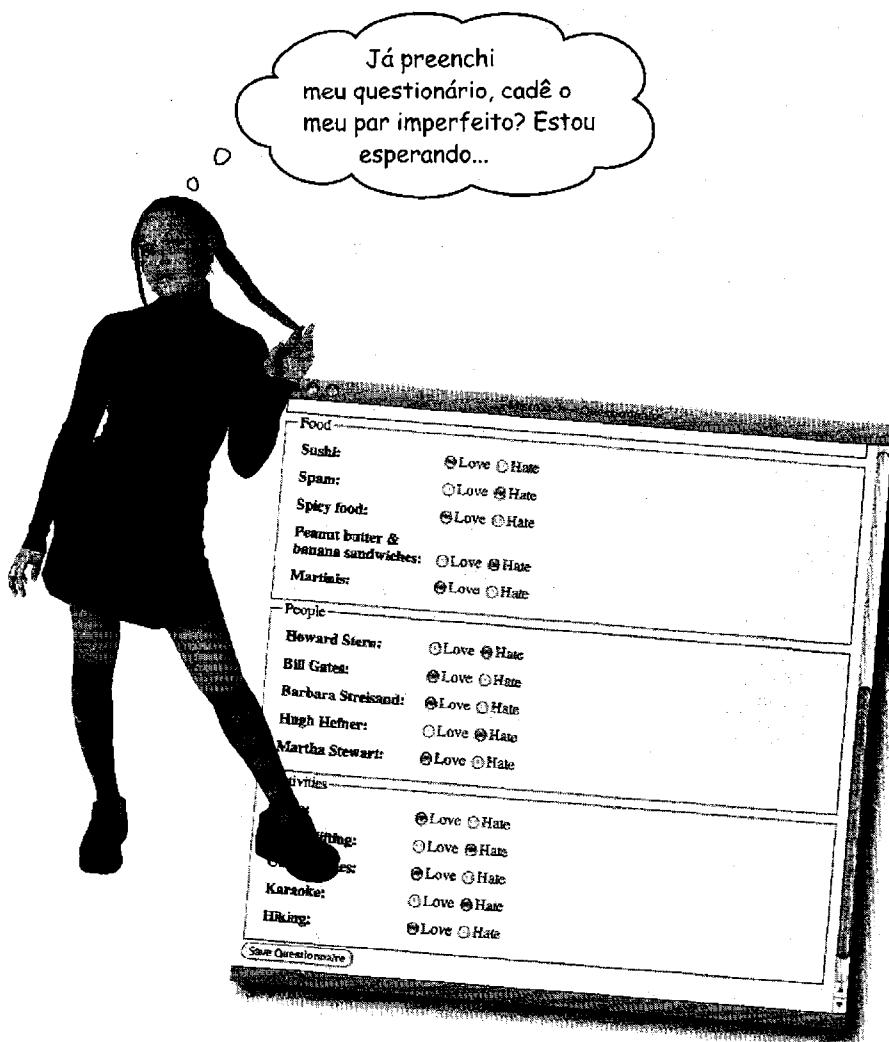


### Exercício

Segue algum código capaz de obter dados de resposta do banco com apenas uma consulta, graças ao uso inteligente dos joins. Seja inteligente e escreva a consulta SQL que fará a junção entre as tabelas mismatch\_response, mismatch\_topic e mismatch\_category.

```
// Obtém os dados de resposta do banco, para gerar o formulário
$query = .....  
.....  
.....  
.....  
.....  
.....  
  
$data = mysqli_query($dbc, $query);
$responses = array();
while ($row = mysqli_fetch_array($data)) {
    array_push($responses, $row);
}
```

## hora de brincar mismatch



O questionário de Sidney está preenchido, armazenado e pronto para ser usado no Mismatch.

**O Mismatch agora se lembra das respostas dos usuários, mas ainda não está fazendo o que gostaríamos com elas... como por exemplo formar pares!**

O conjunto dos dados dos usuários só nos leva até metade do caminho para formarmos pares imperfeitos. A aplicação Mismatch ainda precisa de um mecanismo para disparar a flecha de Cupido no banco de dados para encontrar conexões amorosas. Para isso, precisamos examinar de alguma forma as respostas de todos os usuários no banco, para ver quais formam pares imperfeitos ideais.



Encontrar um par imperfeito ideal parece bem complicado, com todas aquelas categorias, tópicos e respostas. Tem certeza que é possível fazer isso?

**Certamente; nós só precisamos de uma forma consistente de calcular quantos tópicos dois usuários quaisquer têm com respostas opostas.**

Se elaborarmos uma forma razoavelmente simples de calcular quantos tópicos respondidos de forma oposta dois usuários quaisquer têm, torna-se possível fazer um loop através do banco de dados, comparando usuários. A pessoa com o maior número de opiniões contrárias a um determinado usuário é o par imperfeito ideal para esse usuário!

 **Par imperfeito!**

**Escreva abaixo como você faria para calcular a "desencontrabilidade" de dois usuários, usando dados armazenados no banco do Mismatch:**

.....  
.....  
.....

## a lógica do mismatch

# O amor é um jogo de números

Como você deverá se lembrar, as respostas do Mismatch são armazenadas na tabela mismatch\_response na forma de números, com 0, 1 e 2 tendo significados especiais em relação a uma determinada resposta.

Desconhecido = 0      Adoro = 1      Odeio = 2

Esses são os dados usados para se calcular a desencontrabilidade entre dois usuários, e o que estamos procurando especificamente é uma combinação de adoro com odeio ou de odeio com adoro. Em outras palavras, estamos procurando por linhas de resposta em que response é um 1 em oposição a um 2 ou vice-versa.

	Body piercing	Caveira tatuagem	Long hair	Entertainment	Reality TV
28	2	0	1	3	
29	1	1	1	4	
30	1	1	1	5	
31	1	1	1	5	

	Body piercing	Caveira tatuagem	Long hair	Entertainment	Reality TV
278	1	1	11	3	
279	1	1	11	4	
280	1	1	11	5	
281	2	0	11	5	

Uma resposta adoro (1) combinada com uma odeio (2) para o mesmo tópico (reality shows) resulta em uma combinação positiva.

response_id	response	user_id	topic_id

A segunda coluna destas linhas é a resposta, que pode ser 0 (não respondido), 1 (adoro) ou 2 (odeio).

Ainda nos falta uma forma de determinar, através de código PHP, os momentos em que ocorre uma combinação entre duas respostas. Certamente poderíamos escrever algumas declarações if-else para verificar se há um 1 e um 2 ou então um 2 e um 1, mas a solução pode ser mais elegante do que isso. Em qualquer um dos cenários, adicionar os valores das duas respostas resulta no valor 3. Assim, podemos usar uma simples equação para detectar a combinação desejada entre duas respostas quaisquer:

**Se RespostaA + RespostaB = 3, temos uma combinação!**

Assim, achar uma conexão amorosa acaba sendo mesmo uma simples questão de matemática. Isso basta no que se refere a comparar as combinações individuais, mas não atende ao problema maior de como realmente criar o script Meu Par Imperfeito.

## Cinco passos para um desencontro com sucesso

Encontrar o par imperfeito ideal para alguém não é apenas uma questão de comparar linhas de respostas. O script Meu Par Imperfeito tem de seguir uma série de passos cuidadosamente orquestrados para formarmos pares com sucesso. Estes passos são a chaves para finalmente satisfazermos os usuários e usarmos de forma significativa as respostas que eles deram ao questionário.

**1 Obter as respostas do usuário na tabela mismatch\_response, certificando-se de juntar os nomes dos tópicos aos resultados.**

**2 Inicializar os resultados de busca do Mismatch, incluindo as variáveis que registram o "melhor par imperfeito".**

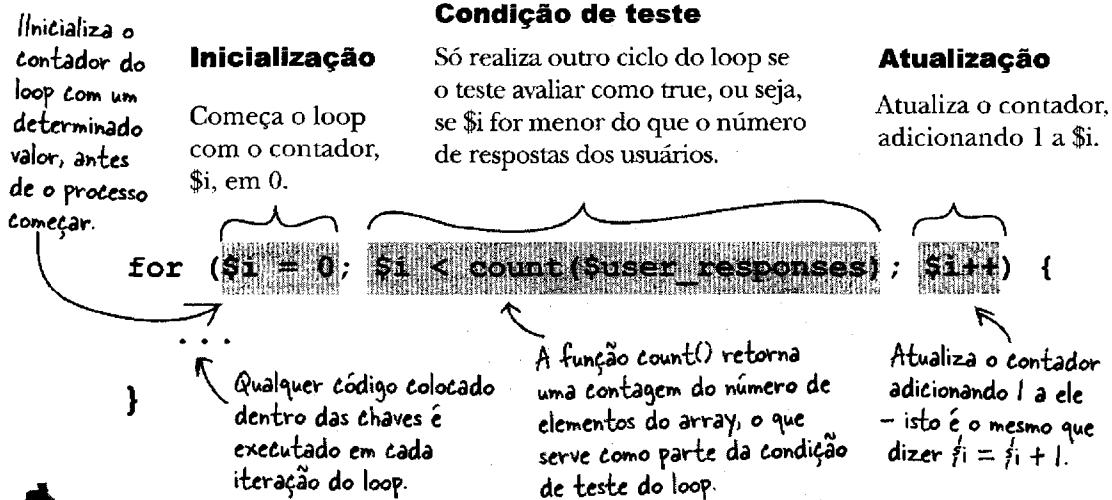
**3 Fazer um loop através da tabela dos usuários, comparando as respostas dos demais às do usuário em questão. Para isso, comparar as respostas de cada pessoa no banco de dados às respostas correspondentes do usuário em questão. Um "placar" mantém registro de quantas respostas opostas o usuário tem em relação a cada pessoa.**

**4 Após cada ciclo do loop, analisar se o par imperfeito atual é melhor do que o melhor par encontrado até agora. Em caso afirmativo, armazená-lo como o novo "melhor par imperfeito", certificando-se de armazenar também os tópicos em oposição.**

**5 Certificar-se de que um "melhor par imperfeito" foi encontrado, e então fazer consulta para obter mais informações sobre o usuário considerado como o melhor par, e mostrar os resultados.**

## Tudo o que precisamos é um loop FOR

O PHP oferece outro tipo de loop que tem exatamente a funcionalidade de que precisamos para as comparações de respostas no Mismatch. Chama-se loop for, e é ótimo para repetir algo uma determinada quantidade de vezes. Por exemplo, os loops for são ótimos para tarefas de contagem, como contar regressivamente até zero ou contar progressivamente até algum valor. Eis a estrutura de um loop for, que revela o modo como o loop pode ser estruturado para atravessar um array usando uma variável para contar (\$i).



### Exercício

O Passo 3 do script Meu Par Imperfeito faz a comparação de dois usuários através de um loop através de cada uma das suas respostas, calculando uma "pontuação" com base em quantas respostas são opostas. Com os pedaços de dados a seguir, termine de escrever o loop for que calculará essa pontuação.

<b>\$mismatch_responses</b>	<b>\$score</b>
<b>\$user_responses</b>	É o array com as respostas do usuário que é o par potencial sendo considerado.
É o array com as respostas do usuário.	A pontuação a ser calculada dentro do loop.
<pre> for (\$i = 0; \$i &lt; count(\$user_responses); \$i++) {     if (...) + ..... == ..... {         .....         array_push(\$topics, \$user_responses[\$i]['topic_name']);     } } </pre>	

*não existem*  
**Perguntas Idiotas**

**P:** Por que não simplesmente usamos um loop foreach para calcular a pontuação, em vez do loop for?

**R:** Embora um loop foreach funcionasse perfeitamente para fazermos o loop através de todas as respostas, ele não nos forneceria um índice (`i$`) em qualquer iteração do loop. Esse índice é importante porque o código está usando tanto para acessar o array das respostas do usuário A quanto o das respostas do usuário B. Um loop foreach eliminaria a necessidade de um índice para um dos dois arrays, mas não ambos. Assim, precisamos de um loop for regular, com um índice que possa ser usado para se acessar elementos semelhantes de cada array.

**P:** Qual é o propósito de se armazenar as respostas do usuário B em um array próprio?

**R:** O array com as respostas do usuário B é importante para que os usuários saibam exatamente como eles se compararam com os seus pares ideais. Não é suficiente simplesmente dizermos a identidade da pessoa que é o par imperfeito ideal – melhor do que isso é dizer em que tópicos específicos o usuário tem respostas diferentes das dessa pessoa. Isso ajuda a dar um pouco mais de contexto ao resultado, e permite que os usuários saibam um pouco mais sobre por que esta pessoa em particular seria um par imperfeito tão bom.

**P:** No Passo 5 do script, como seria possível não haver um melhor par para o usuário em questão?

**R:** Embora improvável, você tem de considerar o cenário em que haja apenas um usuário no sistema inteiro, em cujo caso não haveria ninguém com quem comparar esse usuário.

## exercício solução

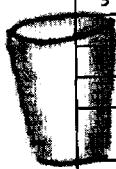


### Exercício

### Solução

\$user\_responses

1	1	Hate	Tattoos
2	2	Hate	Gold chains
3	3	Hate	Body piercings
4	Love	Cowboy boots	
5	Love	Long hair	
6	Love	Reality TV	
7	Hate	Professional wrestling	
8	Hate	Horror movies	
...			



\$mismatch\_responses

76	1	Love	Tattoos
77	2	Love	Gold chains
	3	Love	Body piercings
	4	Love	Cowboy boots
	5	Love	Long hair
	6	Hate	Reality TV
	7	Love	Professional wrestling
83	8	Love	Horror movies
...			



\$score

A variável \$score acaba na faixa de 0 (nenhum par encontrado) até o número total de tópicos (desencontro perfeito).

A pontuação é incrementada a cada resposta oposta encontrada.

Lembre-se, o número de respostas do usuário é o mesmo que o número total de tópicos, uma vez que as respostas vêm direto do questionário.

```
for ($i = 0; $i < count($user_responses); $i++) {
    if ($user_responses[$i]['response'] + $mismatch_responses[$i]['response'] == 3) {
        $score += 1;
    }
    array_push($topics, $user_responses[$i]['topic_name']);
}
```

O contador do loop é usado para se passar através de cada resposta do usuário.

Cada tópico com respostas opostas é adicionado a um array, para que possa ser exibido ao usuário quando o sistema lhe apresenta o par imperfeito encontrado.

Uma combinação positiva consiste de um adoro (1) combinado a um odeio (2), de modo que adicionar os dois sempre resulta em um valor 3, caso haja a combinação procurada.

**FEITO**

- 3 Fazer um loop através da tabela dos usuários, comparando as respostas dos demais às do usuário em questão.

# Finalizando o desencontro

O loop que acabamos de criar para calcular a pontuação de desencontro faz parte de um script maior (`mymismatch.php`), o qual cuida de encontrar o par imperfeito ideal para o usuário, e depois exibir as informações.

Este script encontra o par imperfeito para o usuário!



`mymismatch.php`

...

```
// Só procura pelo par se o usuário tiver respostas armazenadas

$query = "SELECT * FROM mismatch_response WHERE user_id = '" . $_SESSION['user_id'] . "'";

$data = mysqli_query($dbc, $query);
if (mysqli_num_rows($data) != 0) {
    // Primeiro, obtém as respostas do usuário a partir da respectiva tabela (JOIN
    // para obter o nome do tópico)

    $query = "SELECT mr.response_id, mr.topic_id, mr.response, mt.name AS topic_name "
        . "FROM mismatch_response AS mr "
        . "INNER JOIN mismatch_topic AS mt "
        . "USING (topic_id) "
        . "WHERE mr.user_id = '" . $_SESSION['user_id'] . "'";

    $data = mysqli_query($dbc, $query);

    $user_responses = array();
    while ($row = mysqli_fetch_array($data)) {
        array_push($user_responses, $row);
    }
}

// Inicializa os resultados da busca
```

Só é possível achar um par para o usuário que tiver respondido o questionário.

Um familiar JOIN é usado para se obter o nome do tópico, ao se selecionar as respostas do usuário.

O array `$user_responses` armazena todas as respostas do usuário.

Estas variáveis mantêm registro da busca do Mismatch, à medida que o processo ocorre.

Aguente firme, ainda tem muito mais - vire a página!

...

**faça um par imperfeito!**



## TEST DRIVE

### Encontre o seu par imperfeito ideal!

Modifique o Mismatch para que ele use o novo script Meu Par Imperfeito (ou baixe a aplicação no site da Alta Books, em [www.altabooks.com.br](http://www.altabooks.com.br)). Será preciso criar um novo script mymismatch.php, bem como adicionar um item de menu "Meu Par Imperfeito" ao script navmenu.php, para que os usuários possam acessá-lo.

Envie os scripts ao seu servidor web e depois abra a página principal do Mismatch (index.php) em um navegador web. Certifique-se de que está logado e que já respondeu o questionário, e então clique no item de menu "Meu Par Imperfeito" para encontrar o seu par.



# Ímas de Geladeira do Schema do Banco de Dados

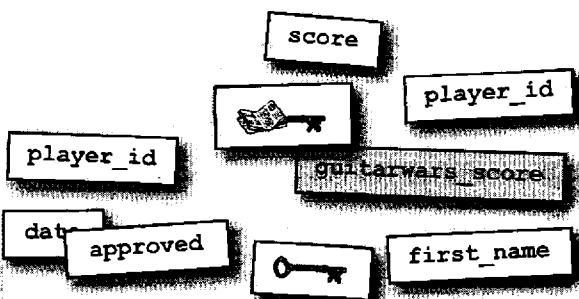
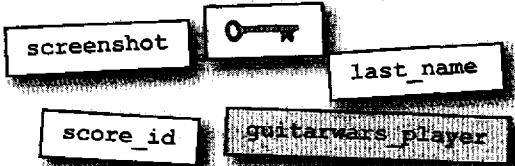
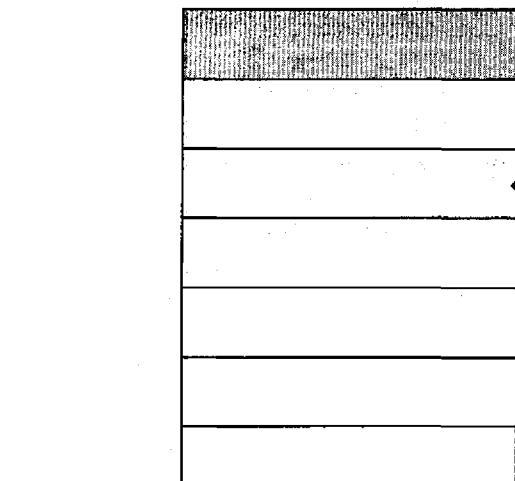
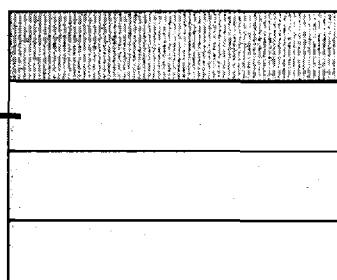
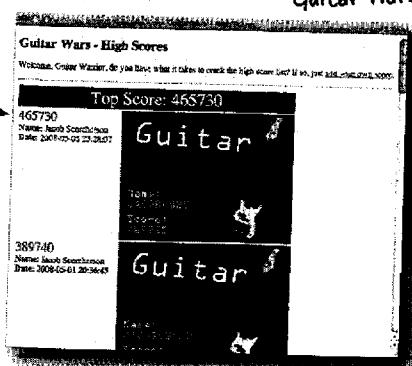
Lembra-se da aplicação Guitar Wars, de séculos atrás? A sua tarefa é estudar o banco de dados do Guitar Wars, o qual poderia se aproveitar de um pouco de normalização, e criar um schema melhor. Use todos os ímas abaixo para completar os nomes das tabelas e colunas, e identifique também as chaves primária e estrangeira.

O banco de dados determina a exibição das pontuações na página principal do Guitar Wars.

Eis o banco de dados original do Guitar Wars, o qual armazena pontuações submetidas pelos usuários.

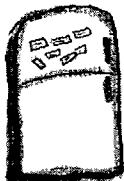
guitar_wars	date	name	score	screenshot	approved

Eis o novo e melhorado schema que você precisa criar com os ímas... boa sorte!



Ímas do schema do banco de dados solução

## Ímas de Geladeira do schema do Banco de Dados - Solução



A tabela precisa de uma chave primária, que é uma parte importante de qualquer banco de dados normalizado.

guitarwars
date
name
score
screenshot
approved

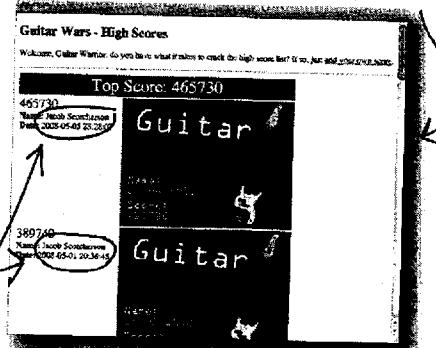
Uma vez que o mesmo usuário pode postar diversas pontuações, a coluna name resulta em dados redundantes... nada bom!

A nova coluna score\_id serve como uma chave primária muito necessária para a tabela score.

A tabela score referencia os jogadores através de uma nova chave estrangeira.

guitarwars_score
2 score_ 0
player_id
date
score
screenshot
approved

As tabelas agora têm novos nomes, uma vez que servem a propósitos mais específicos.



O nome de cada jogador agora é dividido em primeiro nome e sobrenome, para ser mais atômico, e só é armazenado uma vez, não importando quantas pontuações o jogador envie..

guitarwars_player
player_ 0
first_name
last_name

Uma relação de um-para-um entre jogadores e pontuações!

A redundância nos dados dos nomes dos usuários é resolvida criando-se uma nova tabela para armazenar os nomes dos jogadores, a qual se conecta à tabela score através de uma chave.

O banco de dados do Guitar Wars não tinha nenhum problema de dependência nas colunas.

Eis aqui as regras mais uma vez, apenas para assegurarmos que você não as esqueceu!

1 Certifique-se de que as suas colunas sejam atômicas.

2 Dé a cada tabela a sua própria chave primária.

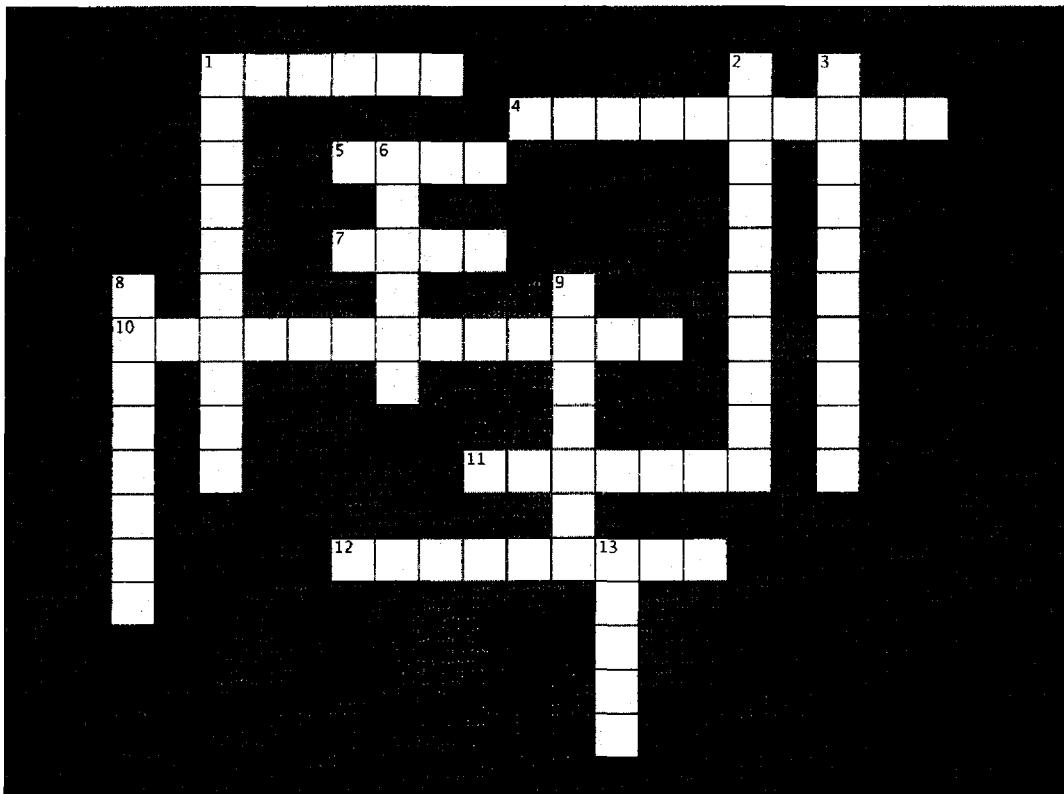
3 Certifique-se de as colunas que não são chaves não sejam dependentes umas das outras.

O banco de dados determina a exibição das pontuações na página principal do Guitar Wars.



## Palavras-Cruzadas PHP & MySQL

Está preocupado sobre se o seu par imperfeito ideal ainda está por aí esperando para ser encontrado? Tire esse pensamento da cabeça completando estas palavras cruzadas.



### Horizontais

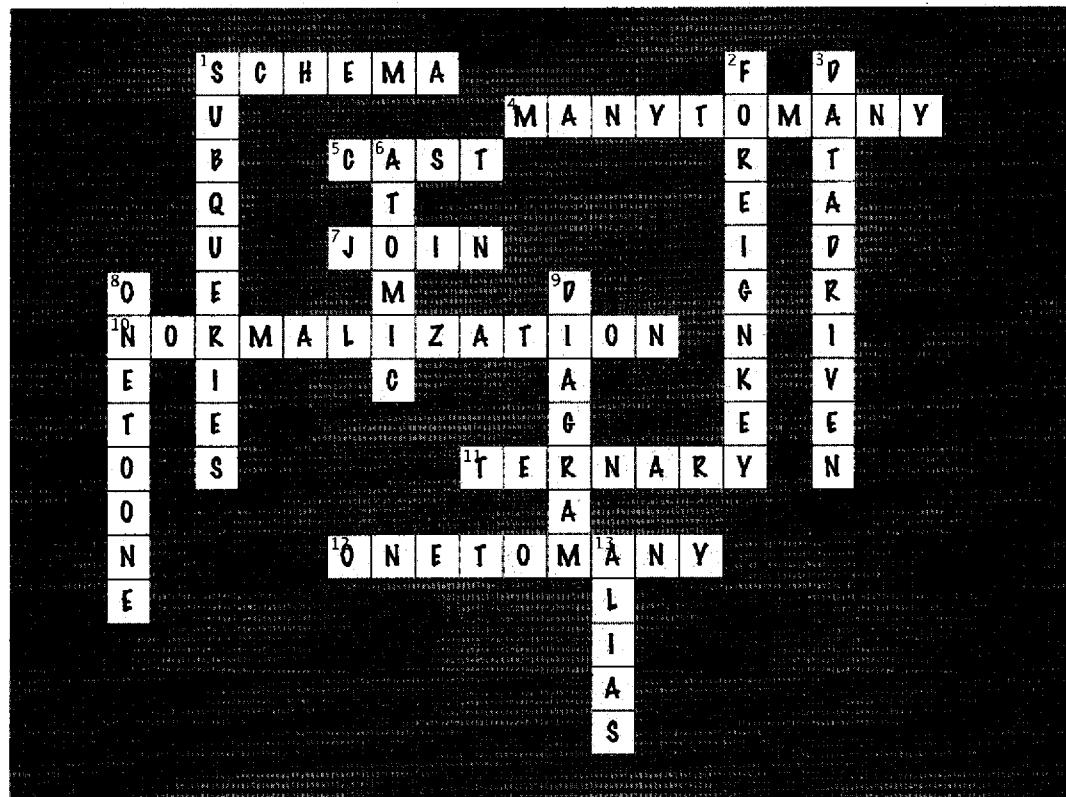
1. Uma representação de todas as estruturas, como tabelas e colunas, do seu banco de dados, junto com o modo como elas se conectam.
4. Isto acontece quando múltiplas linhas de uma tabela relacionam-se com múltiplas linhas de outra.
5. Isto lhe permite converter entre diferentes tipos de dados PHP.
7. Use isto para combinar resultados de uma tabela com os resultados de outra, em uma consulta.
10. O processo de se eliminar redundâncias e outros problemas de design em um banco de dados.
11. Você pode abreviar algumas instruções if-else com este útil operador.
12. Quando uma linha de uma tabela relaciona-se com múltiplas linhas de outra

### Verticais

1. Uma junção possibilita nos livrarmos de joins.
2. Uma coluna, em uma tabela, que referencia a chave primária de outra (em inglês).
3. Quando um formulário é gerado a partir de um banco de dados, ele é considerado ..... .. .. ..
6. Não é nuclear, apenas dados no menor tamanho apropriado para um determinado banco de dados.
8. Linhas de duas tabelas têm esta relação quando há exatamente uma linha em uma tabela para cada linha da outra.
9. Um destes pode ajudar enormemente a se entender o projeto de uma tabela.
13. Um nome temporário usado para se referenciar uma determinada informação, em uma consulta.



# Palavras-Cruzadas PHP & MySQL - Solução





## Sua Caixa de Ferramentas do PHP & MySQL

Um bom número de novas técnicas para bancos de dados MySQL foi apresentado neste capítulo, sem mencionar alguns novos truques com o PHP. Vamos fazer uma rápida recapitulação!

### Schemas e Diagramas

Um esquema é uma representação de todas as estruturas (tabelas, colunas, etc.) do seu banco de dados, juntamente com o modo como elas se conectam. Um diagrama é uma representação visual do seu banco, incluindo detalhes sobre as colunas específicas responsáveis por conectar tabelas.

?

O operador ternário é um construtor PHP que funciona como uma decisão if-else bastante compacta. É útil para realizar cálculos simples com base em uma expressão true/false.

### INNER JOIN

Este tipo de join combina dados de duas tabelas que têm linhas em comum. Ao contrário de uma consulta normal o join permite que você obtenha dados de mais de uma tabela, o que é extremamente útil quando um banco de dados consiste de múltiplas tabelas.

### Normalização

A normalização é o processo de se alterar o projeto de um banco de dados de modo a reduzir a duplicação de dados e melhorar a colocação das relações entre os dados. O objetivo é produzir um design robusto que se adapte bem ao crescimento dos dados.

### FOR (...)

Um loop que é adequado a operações com um número específico de iterações. Você cria um loop fazendo inicializando um contador, estabelecendo uma condição de teste e especificando como o contador deve ser atualizada após cada iteração.

### AS nome

Esta instrução SQL estabelece um alias, que é um nome usado para se identificar um determinado dado em uma consulta. Os aliases são frequentemente usados para se simplificar consultas abreviando nomes de tabelas e colunas muito longas. Podem ser usados também para renomear dados de resultados quando a coluna original da tabela não é específica o suficiente.

riskyjobs precisa de uma ferramenta de busca em seu site

## Um bom trabalho arriscado é difícil de encontrar

O novo site RiskyJobs.biz tem como missão ajudar as empresas a encontrar as pessoas certas para os trabalhos mais arriscados oferecidos por elas. O modelo de negócios é simples: para cada candidato que conseguirmos achar para um emprego arriscado recebemos uma comissão. Quanto mais combinações de candidatos e empregos conseguirmos, maior o nosso lucro.

O RiskyJobs precisa de ajuda para melhorar a sua funcionalidade de busca por vagas. Do jeito como está agora, há um banco de dados cheio de empregos arriscados só esperando para serem descobertos pelas pessoas certas. Vamos dar uma olhada no formulário de busca do RiskyJobs e no banco de dados subjacente, que contém as vagas disponíveis.

Este formulário de busca simples chama um script que faz a busca na tabela riskyjobs.

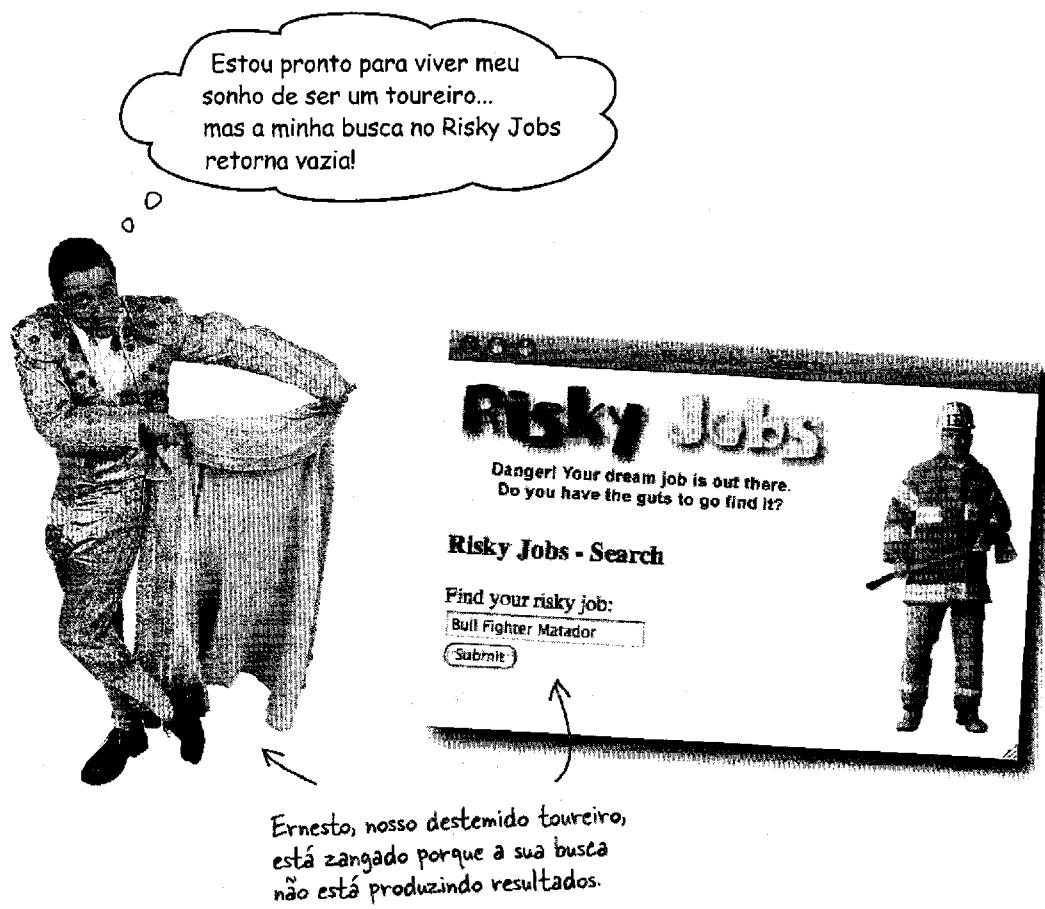
O formulário de busca açãoa uma consulta na tabela riskyjobs, a qual procura por empregos que batam com o critério de busca.

A tabela riskyjobs contém títulos e descrições de empregos, junto com informações sobre a localização e a data do anúncio.

job_id	title	description	city	state	zip	company	ad_date
1	Matador	Bustling dairy farm...	Rutland	VT	05701	Mad About Milk Dairies	2008-03-11 10:51:24
2	Paparazzo	Top celebrity...	Beverly Hills	CA	90210	Diva Pursuit, LLC	2008-03-24 10:51:24
3	Shark Trainer	Training sharks to do...	Orlando	FL	32801	SharkBait, Inc.	2008-04-28 03:12:45
4	Firefighter	The City of Dataville...	Dataville	OH	45490	City of Dataville	2008-05-22 12:34:17
5	Voltage Checker	You'll be out in the...	Durham	NC	27701	Shock Systems, LLC	2008-06-28 11:16:30
6	Crocodile Dentist	Do you love animals...	Everglades City	FL	34139	Ravenous Reptiles	2008-07-14 10:51:24
7	Custard Walker	We need people...	Albuquerque	NM	87101	Pie Technologies	2008-07-24 10:54:05
8	Electric Bull Repairer	Hank's Honky Tonk...	Hoboken	NJ	07030	Hank's Honky Tonk	2008-07-27 11:22:28

Cada anúncio é identificado individualmente através da chave primária job\_id.

Mostra os resultados da busca!

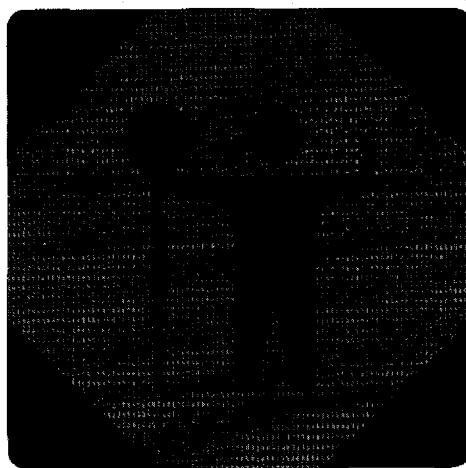


## Aponte seu lápis

Quando o formulário do Risky Jobs é submetido, a string de busca é armazenada na variável \$user\_search, que é colocada na consulta SQL abaixo para ser feita a busca propriamente dita. Escreva aqui quantas linhas do banco de dados riskyjobs da página anterior serão encontradas como resultado da busca de Ernesto.

```
$search_query = "SELECT job_id, title, state, description FROM riskyjobs
"
"WHERE title = '$user_search'";
$result = mysqli_query($dbc, $search_query);
```

Escreva sua resposta aqui! → .....



**Tempo! Tire um momento para se familiarizar com o banco de dados do Risky Jobs... e experimente fazer algumas buscas.**

Baixe o arquivo riskyjobs.sql, da aplicação Risky Jobs, no site da Alta Books em [www.altabooks.com.br](http://www.altabooks.com.br). Esse arquivo contém as instruções SQSQL para criar e preencher a tabela riskyjobs com dados de exemplo.

Após ter executado as instruções de riskyjobs.sql em alguma ferramenta MySQL, experimente fazer algumas consultas para simular as buscas dos usuários. Eis alguns exemplos:

```
SELECT * FROM riskyjobs
```

← Esta consulta seleciona todas as colunas de todas as vagas presentes na tabela riskyjobs.

```
SELECT job_id, title, description FROM riskyjobs  
WHERE title = 'Bull Fighter Matador'
```

← Esta consulta obtém a ID, o título e a descrição de vagas cujo título seja exatamente "Bull Fighter Matador".

```
SELECT job_id, title, description FROM riskyjobs  
WHERE description LIKE '%animals%'
```

↑  
Esta consulta usa LIKE para encontrar vagas com a palavra "animals" em qualquer lugar da descrição.

**Faça o Download!**



O código-fonte completo da aplicação Risky Jobs está disponível para download no site da Alta Books: [www.altabooks.com.br](http://www.altabooks.com.br)



## Ímas de Geladeira da Cláusula LIKE

Temos um monte de cláusulas LIKE bagunçadas pela mesa. Você é capaz de ligar as cláusulas com os seus resultados? Quais ímas não serão encontrados por nenhuma destas cláusulas LIKE?

Algunas poderão ter mais de uma resposta.

LIKE '%er'

LIKE '% T%'

LIKE 'c%'

LIKE '%test %'

LIKE '%Tipper Cow%'

LIKE '%do\_'

LIKE '%ma%'

Team Mascot  
(Mascote)

Cliff Diver  
(Mergulhador de Penhascos)

Rodeo Clown  
(Palhaço de Rodeio)

Crash Test Dummy  
(Boneco de Testes de Impacto)

Human Cannonball  
(Bala Humana)

Cat Herder  
(Pastor de Gatos)

Snake Charmer  
(Encantador de Serpentes)

(Pet Food Tester)  
Experimentador de Rações para Animais

Cow Tipper  
(Virador de Vacas)

Matador  
(Toureiro)

Politician  
(Político)

Shark Finder  
(Caçador de Tubarões)



## TEST DRIVE

### Leve o formulário de busca do Risky Jobs para dar uma volta.

Baixe a aplicação Risky Jobs no site da Alta Books, em [www.altabooks.com.br](http://www.altabooks.com.br). O script search.php contém o código para a geração das consultas, no qual você acabou de trabalhar, e é usado para processar os dados de busca digitados no formulário da página search.html.

Envie o script e os outros arquivos do Risky Jobs para o seu servidor web, e depois abra o formulário de busca (search.html) em um navegador. Faça algumas busca para ver como o seu código gerador de consultas se comporta. Não se esqueça de testar a busca "Bull Fighter Matador" de Ernesto, que é um bom teste para o novo código com implode().

**Risky Jobs**

Danger! Your dream job is out there.  
Do you have the guts to go find it?

**Risky Jobs - Search Results**

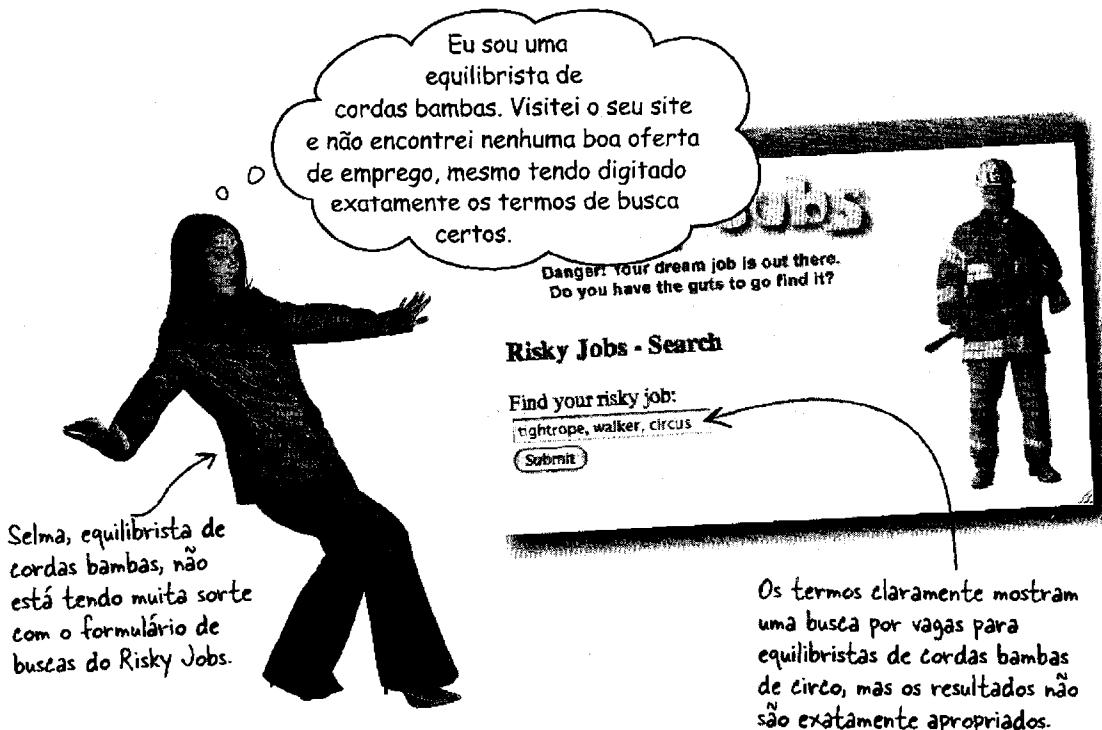
Job Title	Description	ID	Date
Prize Fighter	Up and coming super fly goat weight boxer needs opponent to help build his winning record. Sloppy hands, and a glass jaw are preferred. No experience required. This is not a full-time salaried position. We'll meet you in the alley behind the ring to share the purse. Let Ron King make you a championship fighter, or at least help you lose to one!		2008-11-14 21:49:31
Toreador	Lovely bovines waiting for your superior non-violent cape waving skills. Must pass basic bull fighting aptitude test.		2008-11-14 11:22:28
Electric Bull Repairer	Hank's Honky Tonk needs an experienced electric bull repairer. Free rides (after you fix it) and half off hot wings are some of the benefits.		

Tantos empregos para se escolher! Não acho que sirvo para luta livre ou consertar touros mecânicos, mas acho que poderei encontrar o meu trabalho dos sonhos aqui... algum dia..

Ernesto não encontrou imediatamente o seu trabalho arriscado perfeito, mas ele definitivamente está fazendo progresso, agora que o script de busca procura por cada termo individualmente.

515

*podemos melhorar a busca?*



**Risky Jobs**

Danger! Your dream job is out there.  
Do you have the guts to go find it?

**Risky Jobs - Search Results**

Job Title	Description	State	Date Posted
Master Cat Juggler	Are you a practitioner of the lost art of cat juggling? Banned in forty countries, only the Jim Ruiz Circus has refined cat juggling for the sophisticated tastes of the modern audience. Ply your trade with premiere cat jugglers at our circus, the only place on earth to master synchronized cat juggling. It's true, juggling them is even harder than herding them. We are an equal opportunity employer, and look forward to adding you to our team. Please be prepared to undergo a thorough battery of tests to prove your deft handling of felines. Only the cream of the crop will be accepted into our Master Cat Juggler program.	AZ	2008-11-14 21:13:35
Tightrope Tester	If the thought of dangling for hours on end from great heights is your idea of a good time, then this job just may be for you. Every one of our tightropes goes through rigorous a 43 point test, culminating in a real live human hanging for a prolonged period of time. That could be you! We do provide safety nets but you'll need to bring your own helmet and gloves. Here at our manufacturing and testing facility in Big Top, Montana, we offer an incredible employment package with benefits ranging from Bring Your Pet to Work Week and Formal Fridays. We will need three references, including your verified maximum hang time and number of past falls. We're the circus behind the circus!	MT	2008-11-14 21:17:16

Será que o problema são os termos de busca, ou realmente não há vagas para equilibristas?



## Aponte seu lápis

Escreva abaixo a consulta SQL gerada quando Selma digita "tightrope, walker, circus" ("corda bamba", "caminhante", "circo") como a sua busca, e depois escreva quaisquer problemas que você acha que poderão ocorrer.

.....  
.....  
.....

préprocessando a string de busca

## Aponte seu lápis Solução

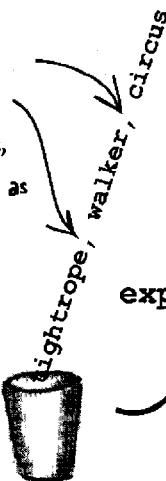
Escreva abaixo a consulta SQL gerada quando Selma digita "tightrope, walker, circus" como a sua busca, e depois escreva quaisquer problemas que você acha que poderão ocorrer.

SELECT \* FROM riskyjobs

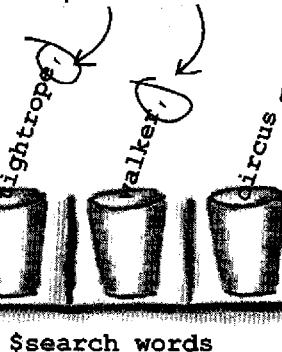
WHERE description LIKE '%tightrope,%' OR description LIKE '%walker,%' OR

description LIKE '%circus%'

A função  
explode() usa  
espaços como  
delimitadores,  
mas não pega as  
vírgulas.

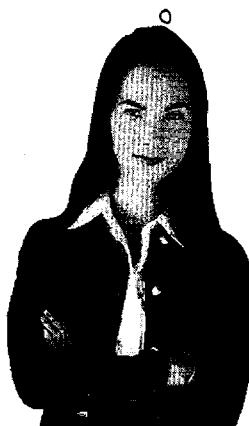


As vírgulas são consideradas como  
parte dos termos de busca, e não  
como separadores entre os termos.



O único termo de busca  
que está realmente  
sendo usado é "circus",  
porque ele não tem uma  
vírgula acidentalmente  
ligada a ele.

Não vejo qual é o problema. Basta chamar a  
função explode() duas vezes - uma para se livrar  
dos espaços e a outra para se livrar das vírgulas.

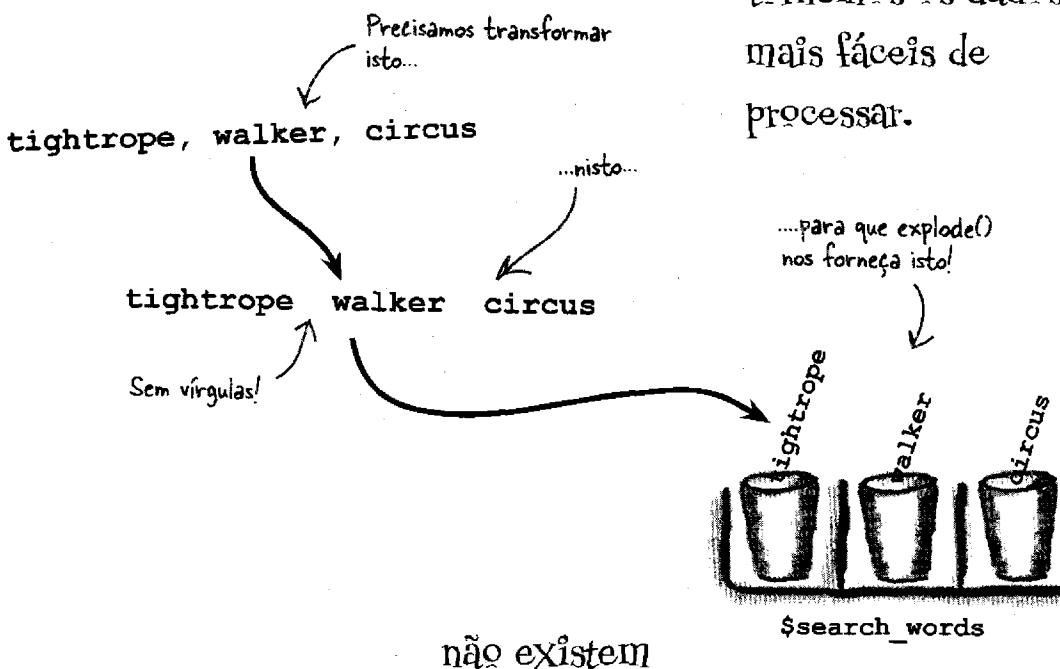


**A função explode() lhe permite dividir uma string em substrings, mas neste caso nós já temos substrings.**

A primeira chamada à função explode() nos deixa com várias strings armazenadas em um array, portanto não existe mais uma única string a ser explodida. E tentar explodir cada uma das strings do array provavelmente só criaria mais problemas. Em vez de tentar resolver o problema do delimitador com várias chamadas a explode(), o que nós precisamos é pré-processar a string de busca, para reduzi-la a um único delimitador antes de sequer fazermos a chamada a explode(). Então, essa função poderá fazer o que faz melhor – dividir a string usando um delimitador.

## Pré-processse a string de busca

Nós queremos passar à função explode() uma string que ela possa dividir sem problemas, de uma só vez. Como fazemos isso? Certificando-nos de que explode() só precise se preocupar com um delimitador, por exemplo um caractere espaço. Isso significa que temos de pré-processar a string de busca, de modo que cada termo seja separado por um espaço, mesmo que o usuário tenha digitado vírgulas.




---

## Perguntas Ídiotas

**Q:** Podemos usar mais de um caractere como delimitador, ao explodirmos a string de busca?

**A:** Sim, você pode especificar qualquer número de caracteres para servir como o seu delimitador, mas isso não é o mesmo que especificar diferentes delimitadores, e não resolverá o nosso problema aqui. Se usássemos explode(' , ', \$user\_search) para dividir a nossa string, isso usaria uma vírgula e um espaço combinados como o delimitador e funcionaria se o usuário digitasse "tightrope, walker, circus". Mas falharia se ele digitasse "tightrope walker circus". Nesse caso, acabariamos tendo apenas uma longa string – nada bom.

**Q:** Podemos simplesmente apagar as vírgulas, em vez de transformá-las em espaços?

**A:** Isso só funcionará se os usuários separarem os seus termos de busca com uma vírgula e um espaço, e não podemos confiar que será sempre assim. Se apagássemos as vírgulas, correríamos o risco de transformar "tightrope.walker" em "tightropewalker", o que provavelmente não encontraria nada no banco de dados do Risky Jobs.

*php função str\_replace()*

## Substitua caracteres de busca indesejados

Se você pensar bem, o pré-processamento da string de busca no Risky Jobs funciona de modo bastante parecido com o recurso de localizar e substituir de um processador de texto. No nosso caso, queremos encontrar vírgulas e substituí-las por espaços. A função str\_replace() do PHP lhe permite fazer exatamente isso, usando três parâmetros: o texto que você quer encontrar, o texto que deseja colocar no lugar do primeiro e a string na qual você quer realizar a operação de localizar e substituir. Eis um exemplo de str\_replace() em ação:

Esta é a substring que você deseja substituir..

...e esta é a string a ser colocada no lugar da primeira.

```
$clean_search = str_replace('milhares', 'centenas',  
    'Ganhe milhares de reais logo no primeiro mês.  
Candidate-se agora!');
```

O terceiro parâmetro é a string que será modificada. Aqui, estamos adicionando um pouco de verdade ao anúncio, substituindo "milhares" por "centenas".

Mas, e quanto às vírgulas na string de busca? A função str\_replace() trabalha igualmente bem na substituição de caracteres individuais:

Lembre-se, esta é a substring que você está substituindo..

...e esta é a string substituta.

```
$clean_search = str_replace(',', ' ', 'tightrope,  
walker, circus');
```

Sempre que aparecer uma vírgula nesta string, ela será substituída por um espaço.

Depois que este código for executado, a variável \$clean\_string conterá a string "tightrope walker circus".



Você vê algo suspeito nos resultados da função str\_replace()? Você acha que a substituição das vírgulas por espaços irá funcionar da maneira que queremos?



Dado o código PHP abaixo, mostre qual seria o output do array `$search_words` para cada uma das seguintes strings de busca. Certifique-se de escrever dados para os elementos apropriados do array e risque os elementos que sobrarem caso o array `$search_words` acabe contendo menos elementos.

```
$clean_search = str_replace(',', ' ', $user_search);
$search_words = explode(' ', $clean_search);
```

**bull,matador cape**



`$search_words`

**3 espaços!**  
bull matador cape



`$search_words`

**bull , matador cape**



`$search_words`

**2 espaços!**  
bull,matador, cape



`$search_words`

## exercício solução



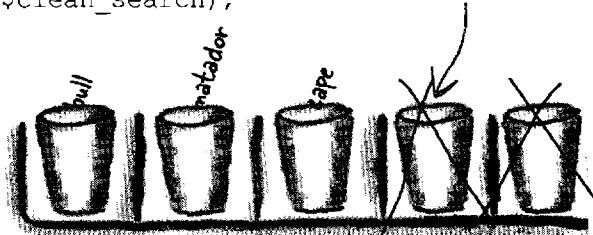
### Exercício Solução

Dado o código PHP abaixo, mostre qual seria o output do array `$search_words` para cada uma das seguintes strings de busca. Certifique-se de escrever dados para os elementos apropriados do array, e risque os elementos que sobrarem caso o array `$search_words` acabe contendo menos elementos..

```
$clean_search = str_replace(',', ' ', $user_search);  
$search_words = explode(' ', $clean_search);
```

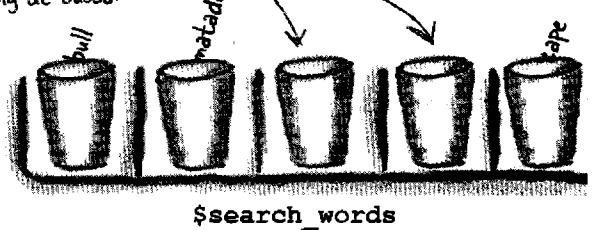
Este array tem  
apenas três elementos.

**bull,matador cape**



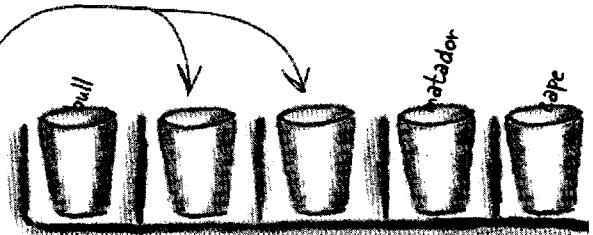
Estes dois elementos do array na verdade estão vazios, por causa daqueles dois espaços extras entre matador e cape na string de busca.

**bull matador cape**



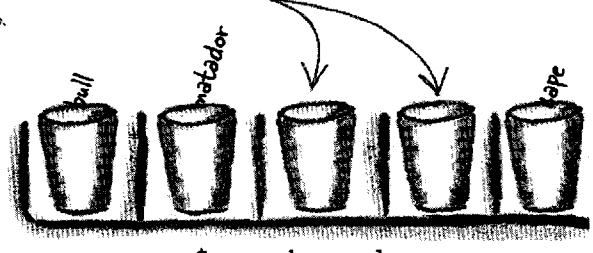
3 espaços!

**bull , matador cape**

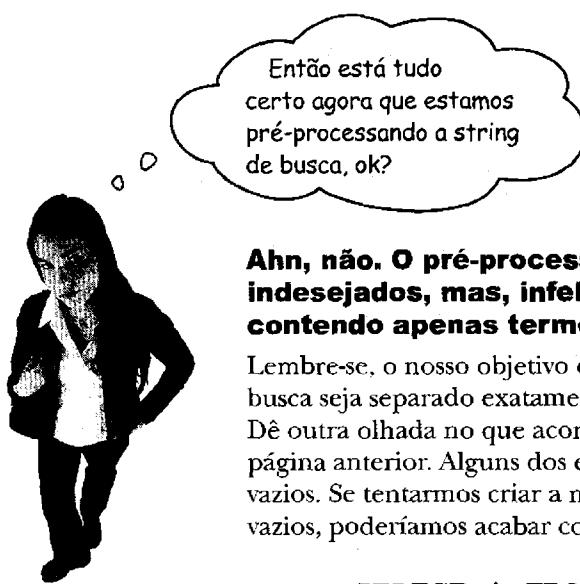


Novamente, dois elementos vazios, porque a vírgula é substituída por um espaço.

**bull,matador, cape**



2 espaços!



**Ahn, não. O pré-processamento nos livra de caracteres indesejados, mas, infelizmente, não resulta em um array contendo apenas termos de busca úteis.**

Lembre-se, o nosso objetivo é ter uma string em que cada termo de busca seja separado exatamente pelo mesmo delimitador, um espaço. Dê outra olhada no que aconteceu nos três últimos exemplos da página anterior. Alguns dos elementos do array \$search\_words estão vazios. Se tentarmos criar a nossa cláusula WHERE com os elementos vazios, poderíamos acabar com algo parecido com isto:

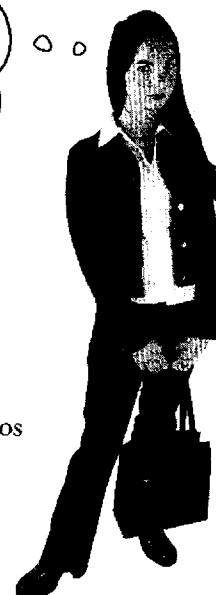
```
SELECT * FROM riskyjobs
WHERE description LIKE '%bull%' OR
description LIKE '%matador%' OR
description LIKE '% %' OR
description LIKE '% %' OR
description LIKE '%cape%'
```

Estes espaços irão encontrar cada espaço que esteja presente em cada descrição de emprego. Eles são um problema e tanto.

Mas esses espaços não vão encontrar nada, certo?

### Errado! Eles vão encontrar tudo.

Se houver um único espaço, em qualquer lugar da descrição de um emprego (e é certo que haverá pelo menos um), esta consulta o encontrará e o retornará como resultado. Assim, todos os anúncios no banco de dados do Risky Jobs serão encontrados como resultado por esta consulta. Precisamos nos livrar desses elementos vazios no array antes de podermos construir a consulta SQL para tornar o script de busca útil novamente.

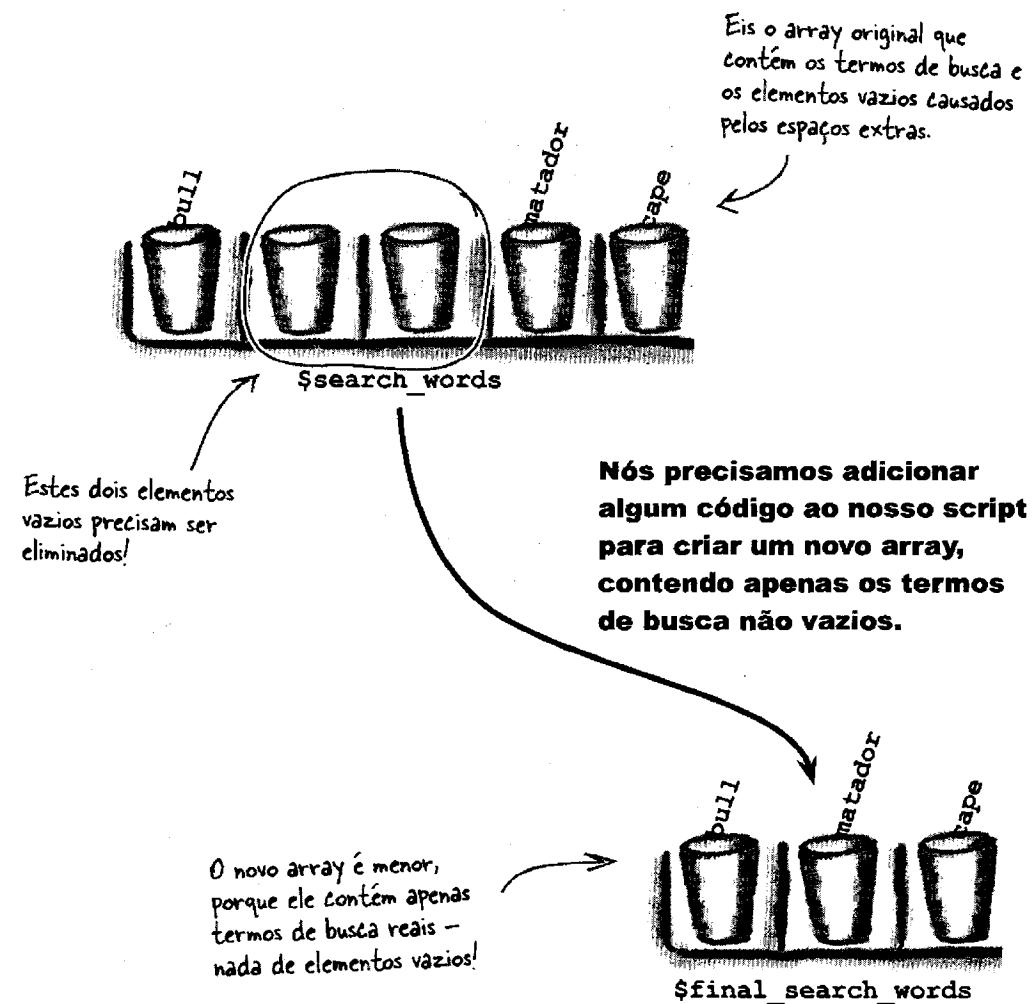


*tirando itens vazios da busca*

## A consulta precisa de termos de busca legítimos

A boa notícia é que não é muito difícil consertar os nossos termos de busca antes de usá-los em uma consulta. Teremos de criar um novo array que contenha apenas os termos de busca reais. Para isso, copiaremos todos os elementos não vazios do nosso array original para um segundo array, e então usaremos este último para construir a consulta SELECT.

Para construir o novo array, podemos usar um loop foreach para realizar ciclos através de cada elemento do array original, e depois usar uma declaração if para encontrar os elementos não vazios. Sempre que encontrarmos um elemento não vazio, basta adicioná-lo ao novo array. O processo se dá mais ou menos desta forma:



# Copie elementos não vazios para um novo array

Agora, vamos dar uma olhada no código que copia os elementos não vazios do nosso array `$search_words` para o novo array `$final_search_words`.

```
$search_query = "SELECT * FROM riskyjobs";

// Extrai as palavras-chaves de busca e as coloca em um array
$clean_search = str_replace(',', ' ', $user_search);
$search_words = explode(' ', $clean_search);
$final_search_words = array();
if (count($search_words) > 0) {
    foreach ($search_words as $word) {
        if (!empty($word)) {
            $final_search_words[] = $word;
        }
    }
}
```

Nada de novo aqui – substitui vírgulas por espaços usando `str_replace()`.

Faz um loop através de cada elemento do array `$search_word`. Se o elemento não estiver vazio, coloca-o no array chamado `$final_search_words`.

Após verificar que há pelo menos um termo de busca no array `$search_words`, o loop `foreach` percorre o array, procurando por elementos não vazios. Quando encontra um, usa o operador `[]` para adicionar o elemento ao final do array `$final_search_words`. É assim que o novo array é criado.

E depois? Bem, nós geramos a consulta SELECT da mesma forma que antes, exceto pelo fato de que agora usamos o array `$final_search_words` em vez de `$search_words`:

```
// Gera uma cláusula WHERE usando todas as palavras-chaves
de busca
$where_list = array();
if (count($final_search_words) > 0) {
    foreach($final_search_words as $word) {
        $where_list[] = "description LIKE '%$word%'";
    }
}
$where_clause = implode(' OR ', $where_list);

// Adiciona a cláusula WHERE à consulta de busca.
if (!empty($where_clause)) {
    $search_query .= " WHERE $where_clause";
}
```

Este é o mesmo código que já usamos para criar a cláusula WHERE da consulta de busca, mas desta vez ele usa o novo array `$final_search_words`, que não contém nenhum elemento vazio.

Esse código nos fornece uma consulta de busca que não possui mais nenhum elemento vazio. Eis a nova consulta para a busca “bull, matador, cape” (“touro, toureiro, capa”):

```
SELECT * FROM riskyjobs
WHERE description LIKE '%bull%' OR
description LIKE '%matador%' OR
description LIKE '%cape%'
```



Será que esta busca dará aos nossos usuários os resultados que eles querem?



## TEST DRIVE

### Atualize o script Search para que ele pré-processse a string de busca do usuário.

Atualize o script search.php de modo que ele use as funções explode() e implode() para pré-processar a string de busca do usuário e gerar uma consulta SELECT mais robusta. Depois, envie o script para o seu servidor web e experimente fazer algumas buscas.

**Risky Jobs**

Danger! Your dream job is out there.  
Do you have the guts to go find it?

**Risky Jobs - Search Results**

Job Title	Description	State	Date Posted
Tightrope Walker	Fledgling big top looking for three-ring professional with 1-3 years of experience to perform tightrope acrobatics with pudgy elephant. Willingness to sweep excrement a big plus. Excellent benefits including medical and dental plans, 401 (k), stock ownership and discount purchase plan, prescription coverage, merchandise discount, short and long term disability insurance, life and business travel insurance, vision discount plan, auto and home insurance discounts, medical care and dependent care reimbursement, educational assistance, paid vacation and holidays, and adoption assistance. Flexible starting salaries based on skills and abilities, experience and geographic market. Promotion opportunities based on performance. The only thing stopping you from the highest wire in the big tent is your desire and work ethic...and your balance! Other duties include planning & organizing wires, handling minor elephant administration, processing comment cards from children. Leading by example (don't fall!), showing initiative and a sense of urgency and being results-driven help acrobatic professionals become successful. If you want to be challenged and your talent needs mentoring and opportunity, Bingling Brothers can offer you a fast track to success!	TX	2008-11-14 21:16:19
Master Cat Juggler	Are you a practitioner of the lost art of cat juggling? Banned in forty countries, only the Jim Ruiz Circus has refined cat juggling for the sophisticated tastes of the modern audience. Ply your trade with premiere cat jugglers at our circus, the only place on earth to master synchronized cat juggling. It's true, juggling them is even harder than herding them. We are an equal opportunity employer, and look forward to adding you to our team. Please be prepared to undergo a thorough battery of tests to prove your deft handling of felines. Only the cream of the crop will be accepted into our Master Cat Juggler program.	AZ	2008-11-14 21:13:35
Tightrope Tester	If the thought of dangling for hours on end from great heights is your idea of good time, then this job just may be for you. Every one of our tightropes goes through rigorous a 43 point test, culminating in a real	MT	2008-11-14 21:17:16

A busca de Selma por "tightrope, walker, circus" agora parece estar encontrando anúncios mais relevantes.



Estou obtendo anúncios de emprego, mas estou recebendo descrições enormes para cada anúncio. Não preciso de todas essas informações. Talvez seja melhor eu tentar em hazardpays.com, onde eles mostram apenas parte da descrição, e eu posso ver mais anúncios por página.

**Embora o Risky Jobs esteja fazendo um trabalho muito melhor para achar anúncios, as enormes descrições deles são um exagero.**

O que realmente está irritando Selma é que para ver mais anúncios, em seu navegador, ela tem de passar por várias telas. Não é necessário mostrar a descrição inteira de cada emprego nos resultados de busca. O ideal é mostrarmos apenas parte da descrição de cada anúncio, talvez apenas as primeiras frases.

**Escreva abaixo como nós poderíamos diminuir as descrições dos anúncios, de modo que eles não fiquem tão grandes nos resultados da busca:**

.....  
.....  
.....

a função php substr()

## Às vezes você só precisa de parte de uma string

Uma vez que a extensão das descrições de empregos no banco de dados do Risky Jobs varia bastante, algumas delas sendo muito longas, nós poderíamos melhorar os resultados de busca reduzindo as descrições a um tamanho melhor. E para não confundirmos os usuários, podemos colocar reticências (...) ao final de cada uma, para deixar claro que eles estão vendo apenas parte de cada descrição

Você passa à função a string original e dois números inteiros. O primeiro é o índice de onde você deseja começar a substring, e o segundo é a sua extensão, em caracteres. A sintaxe é a seguinte:

A função PHP substr() é perfeita para extraímos parte de uma string.

**substr(string, inicio, extensão)**

The diagram shows the substr function signature: substr(string, inicio, extensão). Three arrows point from explanatory text below to each parameter. The first arrow points to 'string' with the text 'Esta é a string original, da qual queremos extrair uma substring.'. The second arrow points to 'inicio' with the text 'Isto especifica onde começar a substring...'. The third arrow points to 'extensão' with the text '...e isto é o número de caracteres a serem retornados.'

No que diz respeito à função substr(), você pode pensar na string como sendo um array em que cada caracter é um elemento diferente. Considere a seguinte string:

```
$job_desc = 'Are you a practioner of the lost art of cat juggling? ';
```

De forma semelhante a elementos de um array, cada caracter desta string possui um índice, começando e 0 e indo até o final da string.

**Are you a practioner of the lost art of cat juggling?**

The string 'Are you a practioner of the lost art of cat juggling?' is shown with vertical lines above each character indicating its index. The indices are labeled 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ..., 50, 51, 52. Ellipses are used between indices 9 and 50, and between 50 and 52.

Podemos usar esses índices com a função substr() para obter partes da string:

Comece no 4, retorno 3 caracteres → substr(\$job\_desc, 4, 3) → you

Comece no 49 e, uma vez que não especificamos o segundo argumento, significa que a função deve seguir até o final da string. → substr(\$job\_desc, 49) → ing?  
→ substr(\$job\_desc, 0, 3) → Are  
→ substr(\$job\_desc, 0, 9) → Are you a

## Extraia substrings do início ou do fim

A função substr() não se limita a obter substrings a partir do início de uma string. Você pode também extraer caracteres começando do final da string. A extração ainda correrá da esquerda para a direita; você apenas usa um índice negativo para identificar o início da substring.

Are you a practitioner of the lost art of cat juggling?

-53 -52 -51 -50 ...

... -3 -2 -1

Eis alguns exemplos:

Comece em -53, retorna 7 caracteres. → substr(\$job\_desc, -53, 7) → Are you

Comece em -9 e retorne tudo a partir daí. → substr(\$job\_desc, -9) → juggling?



### Aponte seu lápis

Segue abaixo algum código PHP para gerar uma tabela HTML para os resultados de busca do Risky Jobs. Termine o código, cuja tarefa é limitar a descrição dos anúncios a 100 caracteres, e também reduza o texto referente à data do anúncio de modo que ele mostre apenas o mês, o dia e o ano.

```
echo '<table border="0" cellpadding="2">';
echo '<td>Job Title</td><td>Description</td><td>State</td><td>Date
Posted</td>';
while ($row = mysqli_fetch_array($result)) {
    echo '<tr class="results">';
    echo '<td valign="top" width="20%">' . $row['title'] . '</td>';
    echo '<td valign="top" width="50%">' . ..... .
'...</td>';
    echo '<td valign="top" width="10%">' . $row['state'] . '</td>';
    echo '<td valign="top" width="20%">' . ..... .
'</td>';
    echo '</tr>';
}
echo '</table>';
```

**aponte o seu lápis solução**

## Aponte seu lápis Solução

Segue abaixo algum código PHP para gerar uma tabela HTML para os resultados de busca do Risky Jobs. Termine o código, cuja tarefa é limitar a descrição dos anúncios a 100 caracteres, e também reduza o texto referente à data do anúncio de modo que ele mostre apenas o mês, o dia e o ano.

```

echo '<table border="0" cellpadding="2">';
echo '<td>Título</td><td>Descrição</td><td>Estado</td><td>Data do Anúncio</td>';
while ($row = mysqli_fetch_array($result)) {
    echo '<tr class="results">';
    echo '<td valign="top" width="20%">' . $row['title'] . '</td>';
    echo '<td valign="top" width="50%">' . substr($row['description'], 0, 100) . '...</td>';
    echo '<td valign="top" width="10%">' . $row['state'] . '</td>';
    echo '<td valign="top" width="20%">' . substr($row['date_posted'], 0, 10) . '</td>';
    echo '</tr>';
}
echo '</table>';

```

*Coloque reticências ao final, para indicar que isto é apenas parte da descrição*

*Todos os dados de date posted começam com MM-DD-AAAA, o que ocupa exatamente 10 caracteres.*



### Nota Geek

É possível dispensar a função PHP substr() e limitar os dados da descrição na própria consulta SQL. Para isso, usa-se uma função MySQL muito semelhante, chamada SUBSTRING(), a qual aceita os mesmos argumentos que substr(). A única diferença é que o índice começa em 1, em vez de 0. Assim, para obter os 100 primeiros caracteres da descrição, usamos isto:

```
SELECT SUBSTRING(job_description, 1, 100)
  FROM riskyjobs;
```

A vantagem de usarmos a função PHP é que teremos tanto a descrição parcial quanto a descrição inteira disponíveis para nosso uso. Se usarmos o MySQL, só teremos a descrição parcial, e teríamos de criar outra consulta para obter a descrição completa.

*não existem  
Perguntas Idiotas*

**P:** Substr() trabalha com valores numéricos?

**R:** Não, ela funciona apenas com strings. Porém, se você tiver um número armazenado como CHAR, VARCHAR ou TEXT, ao obtê-lo através de SQL, esse número é tratado pelo PHP como uma string, e não um número, de modo que você pode usar a função substr() nele.

**P:** E se o valor referente à extensão for maior do que a string? Será retornada uma string com espaços depois dela, para atender ao valor da extensão?

**R:** Será retornada a string inteira. Mas a função não insere espaços depois da string para mudar-lhe a extensão. Por exemplo, o seguinte código retorna a string 'dog': substr('dog', 0, 10).



## TEST DRIVE

### Ajuste o script Search de modo a limitar o texto exibido para descrições e datas dos anúncios.

Modifique o script search.php de modo que ele passe a usar a função PHP substr() para reduzir os textos de descrição e a data dos anúncios retornados pela busca. Depois, envie o script para o seu servidor web e teste-o com algumas buscas.

Selma agora está contente,  
porque ela pode ver os resultados  
da sua busca sem ter de passar  
pelas enormes descrições dos  
anúncios.



### Risky Jobs

Danger! Your dream job is out there.  
Do you have the guts to go find it?

#### Risky Jobs - Search Results

Job Title	Description	State	Date Posted
Tightrope Walker	Fledgling big top looking for three-ring professional with 1-3 years of experience to perform tightr...	TX	2008-11-14
Master Cat Juggler	Are you a practitioner of the lost art of cat juggling? Banned in forty countries, only the Jim Ruiz...	AZ	2008-11-14
Tightrope Tester	If the thought of dangling for hours on end from great heights... then thi...	MT	2008-11-14



Eu realmente gostaria de  
ver os resultados classificados pela  
data dos anúncios ou talvez por estado.  
Eu realmente gostaria de achar um  
emprego de toureiro em Vermont.

A data do anúncio também  
ficou um pouco mais fácil de ler,  
agora que mostra apenas a data  
e não a hora de postagem.

### PODER DO CÉREBRO

Como nós poderíamos mudar o layout da página  
e a consulta para podermos classificar os  
resultados pr data, estado ou título?



adicionando classificação à busca dos resultados

## Podemos classificar nossos resultados com múltiplas consultas

Para permitirmos que os visitantes classifiquem os resultados das suas buscas, é preciso que eles possam identificar como querem os resultados ordenados. Talvez um formulário... ou um botão? Na verdade, é bem mais simples que isso. Podemos usar HTML para transformar cada cabeçalho de coluna, na tabela de resultados, em links. Os usuários poderão então clicar no link para indicar por qual coluna eles querem classificar os resultados.

**Risky Jobs**  
Danger! Your dream job is out there.  
Do you have the guts to go find it?

**Risky Jobs - Search Results**

Job Title	Description	State	Date Posted
Prize Fighter	Up and coming super fly gnat weight boxer needs an opponent to help build his winning record. Slow f...	MO	2008-11-14
Toreador	Lovely hovines waiting for your superior non-violent cape waving skills. Must pass basic bull fighti...	ID	2008-11-14
Electric Bull Repairer	Hank's Honky Tonk needs an experienced electric bull repairer. Free rides (after you fix it) and hal...	NJ	2008-11-14

Nós podemos usar esses links para recarregar o mesmo script Search, porém com uma consulta que classifique os resultados de acordo com o link clicado. Já sabemos como usar ORDER BY para estruturar uma consulta com resultados classificados. Se criarmos diferentes consultas SQL para classificar através de cada coluna individual, permitiremos que o usuário ordene os resultados alfabeticamente por título, descrição ou estado, ou cronologicamente por data do anúncio.

Eis a consulta SQL para classificar os resultados alfabeticamente, pela descrição:

```
SELECT * FROM riskyjobs
WHERE description LIKE '%Bull%' OR description LIKE '%Fighter%' OR
description LIKE '%Matador%'
ORDER BY description
```

Este comando classifica os resultados da consulta pelas descrições dos empregos, em ordem alfabética crescente.



## Aponte seu lápis

Escreva três consultas diferentes que classifiquem os resultados do Risky Jobs de acordo com o título do anúncio, o estado e a data. Assuma que o usuário tenha digitado a string de busca "window, washer, skyscraper" ("janela, lavador, arranha-céus").

Como poderíamos reescrever essas consultas se você quisesse ver os títulos e os estados em ordem inversa? E se quisesse ter os anúncios mais novos aparecendo primeiro?

**ORDER BY** ordena o resultado da consulta



## Aponte seu lápis Solução

Escreva três consultas diferentes que classifiquem os resultados do Risky Jobs de acordo com o título do anúncio, o estado e a data. Assuma que o usuário tenha digitado a string de busca "window, washer, skyscraper" ("janela, lavador, arranha-céus").

`SELECT * FROM riskyjobs`

`WHERE description LIKE '%window%' OR description LIKE '%washer%' OR  
description LIKE '%skyscraper%'`

`ORDER BY job_title`

O padrão de  
**ORDER BY** é

**ASC**endente

(crescente), que é

o mesmo que dizer

**ORDER job\_title**

**ASC.**

`SELECT * FROM riskyjobs`

`WHERE description LIKE '%window%' OR description LIKE '%washer%' OR  
description LIKE '%skyscraper%'`

`ORDER BY state`

`ASC.`

`SELECT * FROM riskyjobs`

`WHERE description LIKE '%window%' OR description LIKE '%washer%' OR  
description LIKE '%skyscraper%'`

`ORDER BY date_posted`

Como poderíamos reescrever essas consultas se você quisesse ver os títulos e os estados em ordem inversa? E se quisesse ter os anúncios mais novos aparecendo primeiro?

`SELECT * FROM riskyjobs`

`WHERE description LIKE '%window%' OR description LIKE '%washer%' OR  
description LIKE '%skyscraper%'`

`ORDER BY job_title DESC`

Poderíamos  
usar estes, caso  
já tenhamos  
classificado por  
uma das colunas,  
e então o  
usuário clica  
nela novamente  
para inverter a  
ordem.

`SELECT * FROM riskyjobs`

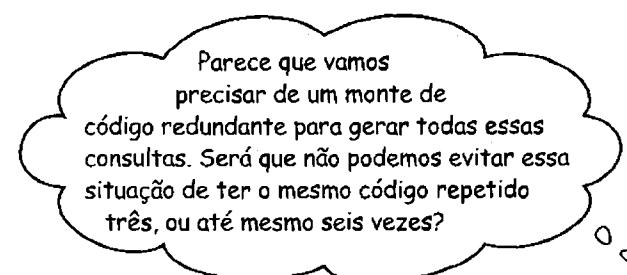
`WHERE description LIKE '%window%' OR description LIKE '%washer%' OR  
description LIKE '%skyscraper%'`

`ORDER BY state DESC`

`SELECT * FROM riskyjobs`

`WHERE description LIKE '%window%' OR description LIKE '%washer%' OR  
description LIKE '%skyscraper%'`

`ORDER BY date_posted DESC`



**Sim. Embora seja verdade que precisaremos executar uma consulta diferente sempre que o usuário clicar em um link diferente, é possível construir uma única consulta baseada no link clicado.**

Na primeira vez que os resultados são exibidos, nenhum link foi clicado ainda, portanto não temos de nos preocupar com a classificação. Podemos simplesmente pegar as palavras-chaves submetidas no nosso formulário e construir uma consulta sem ORDER BY. Os resultados são exibidos com cabeçalhos clicáveis, cada um dos quais com um link de volta para o script, mas com uma ordem de classificação diferente. Assim, cada link consiste de uma URL com as palavras-chaves originais e um parâmetro chamado sort, que indica a ordem na qual os resultados devem aparecer.



O que realmente nos ajudaria aqui seria criar a nossa própria função personalizada, a qual possa pegar informações sobre o modo de classificar os dados e retornar uma string com a cláusula WHERE e ORDER BY nos seus devidos lugares. A nossa nova função personalizada consulta o parâmetro sort para descobrir como classificar os resultados da busca. Eis aqui os passos que a função tem de seguir:

- 1 Pré-processar as palavras-chaves da busca, e armazená-las em um array.
- 2 Opcionalmente, usar um parâmetro sort que diz à função qual coluna usar para a classificação.
- 3 Livrar-se de quaisquer palavras-chaves vazias.
- 4 Criar uma cláusula WHERE contendo todas as palavras-chaves de busca.
- 5 Verificar se o parâmetro sort tem um valor. Se tiver, anexar uma cláusula ORDER BY.
- 6 Retornar a consulta recém-formada.

Isso pode parecer muito trabalho, mas já temos a maior parte do código escrito. Só precisamos transformá-lo em uma função. Mas antes de fazermos isso, vamos dar uma olhada no modo de escrever funções personalizadas...

## As funções lhe permitem reutilizar código

Uma função é um bloco de código separado do resto do seu código, o qual você pode executar em qualquer ponto do seu código que quiser. Até agora, tem usado funções internas que o PHP já criou para você. `explode()`, `substr()` e `mysqli_query()` são funções pré-definidas pelo PHP, e podem ser usadas em qualquer script.

Mas você pode também escrever as suas próprias funções personalizadas, para implementar recursos não fornecidos pela linguagem. Criando uma função personalizada, você pode usar o seu próprio código várias vezes, sem ter de repeti-lo no seu script. Em vez disso, basta chamar a função pelo nome, quando quiser rodar o seu código.

Segue um exemplo de função personalizada chamada `replace_commas()`, a qual substitui vírgulas por espaços, em uma string:

Para criar uma função personalizada, você a inicia com a palavra "function".

Este é o nome da sua função e você tem liberdade para escolher o nome que quiser - faça-o o mais descritivo possível.

Um conjunto de parênteses vem depois do nome da função. Você pode enviar um ou mais valores para a sua função na forma de argumentos, cada um separado por vírgula - neste caso, só há um argumento.

```
function replace_commas($str) {  
    $new_str = str_replace(',', ' ', $str);  
    return $new_str;
```

As chaves indicam onde estará o código da função, da mesma forma como em um loop ou uma instrução.

}

Uma função é capaz de retornar um valor ao código que a chamou - neste caso, nós retornamos a string alterada.

Quando estiver pronto para usar uma função personalizada, basta chamar-lá pelo nome e digitar, entre parênteses, quaisquer valores que ela espere receber. Se a função tiver sido elaborada para retornar um valor, você pode atribuí-la a uma nova variável, desta forma:

Nós passamos uma string, "tightrope, walker, circus".

```
$clean_search = replace_commas('tightrope, walker, circus');
```

A função retorna uma nova string, com as vírgulas substituídas por espaços.

# Construa uma consulta com uma função personalizada

Nós já escrevemos a maior parte do código de que precisamos para criar a função personalizada que irá gerar as consultas de busca no Risky Jobs.

Só o que falta é colocar o código dentro de um esqueleto de função

PHP. Eis a função personalizada build\_query():

```
function build_query($user_search) {
    $search_query = "SELECT * FROM riskyjobs";

    // Extrai as palavras-chaves de busca e as coloca em um array
    $clean_search = str_replace(',', ' ', $user_search);
    $search_words = explode(' ', $clean_search);
    $final_search_words = array();
    if (count($search_words) > 0) {
        foreach ($search_words as $word) {
            if (!empty($word)) {
                $final_search_words[] = $word;
            }
        }
    }

    // Gera uma cláusula WHERE usando todas as palavras-chaves
    // de busca
    $where_list = array();
    if (count($final_search_words) > 0) {
        foreach ($final_search_words as $word) {
            $where_list[] = "description LIKE '%$word%'";
        }
    }
    $where_clause = implode(' OR ', $where_list);

    // Adiciona a cláusula WHERE à consulta de busca
    if (!empty($where_clause)) {
        $search_query .= " WHERE $where_clause";
    }
}

return $search_query;
```

Estamos passando para a função o array \$user\_search que criamos a partir dos dados digitados no formulário.

Nada de novo dentro da função!

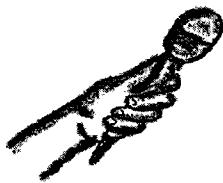
Na verdade, isto aqui é novo. É aqui que retornamos a nova consulta, para que o código que chamou a função possa utilizá-la

A função build\_query() retorna uma consulta SQL completa, baseada na string de busca passada a ela através do argumento de \$user\_search. Para usar a função, nós simplesmente repassamos a ela os dados de busca digitados pelo usuário, e então armazenamos o resultado em uma nova string, que chamaremos de \$search\_query:

```
$search_query = build_query($user_search);
```

Isto nos permite capturar o valor que a nossa função retorna, neste caso, a nossa nova consulta.

Este é o valor submetido pelo usuário através do formulário.



## Funções Personalizadas Expostas

**Na entrevista desta semana:**  
**Funções personalizadas: o quanto personalizadas elas realmente são?**

**Use a Cabeça:** Olhe, nós estamos todos curiosos sobre uma coisa: o que há de tão errado com código redundante? Quer dizer, ele é fácil de criar, basta copiar e colar e pronto.

**Função Personalizada:** Ah, nem me fale de código redundante. Ele é simplesmente feio e torna o seu código mais difícil de ler. Isso já é ruim o suficiente. Mas existe um motivo ainda muito mais importante para se evitar código redundante.

**Use a Cabeça:** Que é...?

**Função Personalizada:** Bem, e se algo se modificar no seu código? Isso acontece com bastante frequência.

**Use a Cabeça:** E daí? As coisas mudam o tempo todo. Você simplesmente vai lá e conserta.

**Função Personalizada:** Mas e se a coisa que se modificou estava no seu código redundante? E ela estava em cinco, ou talvez dez, lugares diferentes espalhados pela sua aplicação?

**Use a Cabeça:** Não vejo qual é o problema. Basta achar todas as ocorrências, consertá-las e pronto.

**Função Personalizada:** Ok, tudo bem. Mas e se você se esquecer de consertar uma das ocorrências? Vocês, programadores, são humanos. Se cometer um erro, poderá ser muito difícil achar a fonte do problema.

**Use a Cabeça:** Sim, é claro que isso pode acontecer. Mas como é que você ajuda?

**Função Personalizada:** Ah, aí está a beleza de ser eu. Se esse código estivesse em uma função, você só precisaria modificá-lo uma vez. Uma vez, e pronto.

**Use a Cabeça:** Eu tenho que admitir que isso é bem interessante. Mas ainda não vejo por que eu deveria me dar ao trabalho de usar você. Quer dizer, você é bem limitado, não é? Você só consegue usar strings.

**Função Personalizada:** Ei, espere um minuto aí, amigo! Eu posso usar qualquer tipo de dados que você me enviar. Desde que o meu código lide com eles da forma correta, eu posso usar quaisquer dados. Caramba, eu usei até um array no último exemplo. Eu diria que isso é bem sofisticado.

**Use a Cabeça:** Mas você retornou uma string.

**Função Personalizada:** Eu posso retornar qualquer coisa que você quiser. A questão é aproveitar ao máximo o que eu posso realizar, e me utilizar corretamente.

**Use a Cabeça:** Essa é outra questão. Você é muito exigente. Você sempre quer receber dados.

**Função Personalizada:** De onde você está tirando essas ideias malucas? Você pode me chamar sem nenhuma variável, se quiser, e se eu estiver configurada dessa maneira. Se você não quiser me enviar dados, é só não escrever nenhuma variável nos parênteses juntos ao meu nome, quando me criar. Embora eu não consiga pensar em muitos motivos pelos quais você não queira me enviar dados, nem recebê-los de volta com uma instrução de retorno.

**Use a Cabeça:** Acabou o nosso tempo. Muito obrigado pela entrevista.

**Função Personalizada:** Não há de quê. Eu vivo para servir. Ou será que sirvo para viver? Ou vivo e sirvo? Sei lá, alguma dessas coisas.



## TEST DRIVE

### Modifique o script Search para que ele use a função build\_query().

Crie a nova função build\_query() no script search.php, certificando-se de substituir o código original por uma chamada à nova função. Envie o script para o seu servidor web e experimente fazer uma busca no navegador, para certificar-se de que está tudo funcionando bem.

Essa nova função personalizada build\_query() é legal, mas ela ainda não está classificando os resultados da busca. Será que nós poderíamos adicionar outro parâmetro para fazer isso?



### Sem problemas. Nós podemos passar dois parâmetros à função build\_query(), em vez de um só.

Nós já estamos passando à função o argumento \$user\_search, que contém os termos de busca do usuário. Agora nós precisamos de outro argumento, \$sort, que indica como classificar os dados. O novo argumento \$sort precisa controlar a ordem dos dados retornados pela consulta nas seis maneiras que estabelecemos anteriormente: classificando pelas colunas job\_title, state e date\_posted da tabela riskyjobs, tanto em ordem crescente quanto em decrescente.

Nós poderíamos armazenar as strings de ORDER BY em \$sort para indicar a ordem de classificação. Ou poderíamos usar os números de 1 a 6 para representar cada ordem, desta forma:

```
$sort == 1 ➔ ORDER BY job_title
$sort == 2 ➔ ORDER BY job_title DESC
$sort == 3 ➔ ORDER BY state
$sort == 4 ➔ ORDER BY state DESC
$sort == 5 ➔ ORDER BY date_posted
$sort == 6 ➔ ORDER BY date_posted DESC
```

Nós escolhemos arbitrariamente esses números e os seus significados.

Não existem regras especiais sobre isso, fora usar as suas escolhas de forma consistente.

Não faz muito sentido classificar pela descrição dos empregos, uma vez que a ordem alfabética não significa muita coisa na descrição.

Mas os números não são ainda mais cripticos quando alguém está lendo o seu código? Sem comentários para ajudar, sim, mas há um motivo mais importante para usarmos números aqui. Se usássemos strings ORDER BY, os nossos dados apareceriam na URL do script como parte do link de cada cabeçalho. Isso inadvertidamente revelaria os nomes das nossas colunas, o que é algo que você não quer tornar público por motivo de segurança.

## Permitindo ao usuário escolher a ordenação

Ok, eu entendi como o novo argumento \$sort funciona, mas como saber qual valor de \$sort nós devemos passar para a nossa função?  
Não precisamos obter essa informação com o usuário?



**Sim, o usuário precisa especificar o modo de classificar os resultados, da mesma forma como ele especifica os próprios termos de busca.**

A boa notícia é que já sabemos como queremos implementar essa funcionalidade: nós vamos transformar os cabeçalhos das colunas na nossa página de resultados em links. Quando o usuário clica em um determinado cabeçalho, como por exemplo, "Estado", nós repassaremos à função build\_query() o número referente à classificação por estado.

Mas ainda precisamos fazer com que a ordem de classificação vá do link até o script. Podemos fazer isso quando estivermos gerando os links para os cabeçalhos, anexando um parâmetro sort à URL:

Os resultados da busca são gerados como parte de uma tabela HTML, e é por isso que temos uma tag <td> aqui.

```
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '
```

```
'?usersearch=' . $user_search . '&sort=3">State</a></td>';
```

A nossa função build\_query() precisa das palavras-chaves de busca do usuário para exibir resultados, portanto, fornecemos essas informações através da URL.

Queremos recarregar esta página quando os usuários clicarem no cabeçalho de uma coluna para classificar os resultados, portanto, temos de criar este formulário como autorreferente.

Nós repassamos dados para indicar a classificação desejada para a busca. Uma vez que este é o link para a classificação por estado, "sort" é igual a 3.

Quando a página dos resultados é gerada, cada link (exceto "Descrição") tem a sua própria URL personalizada, incluindo um valor sort para definir como os resultados devem ser classificados.

```
<a href="search.php?usersearch=bull fighter matador&sort=1">
```

The screenshot shows a web page titled "Risky Jobs". The main heading says "Danger! Your dream job is out there. Do you have the guts to go find it?". Below it, the sub-heading "Risky Jobs - Search Results" is displayed. The page lists several job entries with columns for "Job Title", "Description", "State", and "Date Posted". A cursor is hovering over the "State" column header, which is underlined, indicating it is the active sorting column. Handwritten annotations explain the code logic: "lookin" is written near the "Description" column, and "Classificar pela descrição não seria muito informativo, portanto, não há motivo para transformar esse cabeçalho em um link." is written next to the "State" column header. Two additional annotations show URLs: "<a href="search.php?usersearch=bull fighter matador&sort=3">" pointing to the "State" column header, and "<a href="search.php?usersearch=bull fighter matador&sort=5">" pointing to the "Date Posted" column header.



Hm. Entendi como esses links funcionam nas três primeiras consultas, mas e quanto às outras três ORDER BYs que classificam por ordem decrescente? Onde elas ficam?

**Joe:** Normalmente, o mesmo cabeçalho permitiria ao usuário classificar em ordem crescente ou decrescente.

**Jill:** Correto. Cada vez que o usuário clica no cabeçalho, este apenas inverte a ordem.

**Frank:** Isso não significa que nós agora precisamos de um jeito de manter registro do estado dos cabeçalhos a cada vez que o usuário clica neles, uma vez que o cabeçalho precisará apontar para um link diferente do link para o qual ele aponta atualmente?

**Joe:** Não entendi o que você quis dizer.

**Frank:** Bem, os cabeçalhos não fazem sempre a mesma classificação. Por exemplo, se você clicar em "Título" e ele classificar o resultado pelos títulos dos anúncios, em ordem crescente, então o link precisará se modificar para classificar pelos títulos, mas em ordem decrescente, na próxima vez que for clicado.

**Jill:** É isso mesmo. Mas tenha em mente que cada tipo de classificação tem um número na URL do link, para que o script saiba qual tipo de ordenamento usar. E uma vez que nós estamos gerando esses links, podemos controlar exatamente quais números de sort colocamos em cada link.

**Joe:** Entendo. Então o nosso desafio é estruturar de alguma forma o nosso código, de modo que ele possa gerar o link correto baseado no estado atual da classificação, certo?

**Frank:** Ah, já sei! Não podemos resolver isso com algumas declarações if? Quer dizer, é para esse tipo de tomada de decisão que essas declarações servem, certo?

**Joe:** Sim, isso funcionaria, mas estamos falando de várias decisões envolvendo exatamente os mesmos dados – o tipo de classificação. Seria muito bom se conseguíssemos bolar uma forma melhor de tomar essas decisões, que não seja com um monte de declarações if-else aninhadas.

**Jill:** Muito bem observado, e esta é uma oportunidade perfeita para testarmos uma nova declaração que eu descobri. A declaração switch nos permite tomar múltiplas decisões, muito mais do que duas, com base em apenas um valor.

**Frank:** Parece bom. Vamos tentar.

**Joe:** Concordo. Qualquer coisa para evitar declarações if-else complicadas. Elas me dão dor de cabeça!

**Jill:** Sim, a mim também. Acho que a declaração switch pode ser exatamente o que precisamos...

# SWITCH toma muito mais decisões do que IF

A declaração switch oferece uma forma eficiente de verificar um valor e executar um dentre vários blocos de código, dependendo desse valor. Isso é algo que iria exigir um pequeno exército de declarações if-else, principalmente em situações envolvendo mesmo um número pequeno de opções.

Em vez de escrever declarações if-else aninhadas para verificar cada valor possível, em vez disso, você escreve uma declaração switch com um rótulo case correspondente a cada valor possível. No final de cada rótulo case, você coloca a declaração break;, que instrui o PHP a sair da declaração switch inteira e não considerar quaisquer outros casos. Isso garante que o PHP irá executar o código em um e somente um caso.

Vamos dar uma olhada em um exemplo que usa switch:

```
switch ($benefit_code) {  
    case 1:  
        $benefits = 'Plano de saúde, 10 dias de licença médica';  
        break; // A declaração break diz ao PHP para sair da instrução switch.  
    case 2:  
        $benefits = 'Apenas em caso de morte e esquartejamento, um mês de licença remunerada';  
        break;  
    case 3:  
    case 4:  
        $benefits = 'Boa sorte!';  
        break;  
    default:  
        $benefits = 'Nenhum.';  
}  
  
echo 'Nós oferecemos quatro pacotes de benefícios';  
echo 'O plano que você selecionou: ' . $benefits;
```

Este é o valor que a declaração switch está verificando - ele controla toda a declaração.

Este código só é executado quando \$benefit\_code é igual a 1.

Se você precisar fazer a mesma coisa para dois ou mais valores, basta omitir a declaração break até chegar ao último valor..

Quaisquer valores armazenados em \$benefit\_code que não sejam 1, 2, 3 ou 4 farão com que o código padrão seja executado.

Não exatamente. Só existem três pacotes, uma vez que 3 e 4 são a mesma coisa, devido ao fato de 3 não ter um break.

Uma declaração SWITCH contém uma série de rótulos CASE que executam diferentes blocos de código, dependendo do valor de uma variável.



O Risky Jobs tem uma nova função chamada generate\_sort\_links() que permite aos usuários classificar os resultados de busca clicando nos cabeçalhos dos resultados. Infelizmente, estão faltando partes importantes do código. Complete o código da função. E não se esqueça dos números para cada tipo de busca:

1 = por título, crescente; 2 = por título, decrescente; 3 = por estado, crescente; 4 = por estado, decrescente; 5 = por data de postagem, crescente; e 6 = por data de postagem, decrescente.

```
.....generate_sort_links($user_search, $sort) {
$sort_links = '';

.....($sort) {
case 1:
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">Título</a></td><td>Description</td>';
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">Estado</a></td>';
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">Data</a></td>';

.....
case 3:
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">Job Title</a></td><td>Description</td>';
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">State</a></td>';
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">Date Posted</a></td>';

.....
case 5:
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">Job Title</a></td><td>Description</td>';
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">State</a></td>';
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">Date Posted</a></td>';

.....
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">Job Title</a></td><td>Description</td>';
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">State</a></td>';
$sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
'&sort=.....">Date Posted</a></td>';

}
return .....;
```



Este é o conjunto padrão de cabeçalhos que deverá aparecer caso o usuário ainda não tenha escolhido um método de classificação.

a função completa generate\_sort\_links()



## Exercício Solução

O Risky Jobs tem uma nova função chamada generate\_sort\_links() que permite aos usuários classificar os resultados de busca clicando nos cabeçalhos dos resultados. Infelizmente, estão faltando partes importantes do código. Complete o código da função. E não se esqueça dos números para cada tipo de busca: 1 = por título, crescente; 2 = por título, decrescente; 3 = por estado, crescente; 4 = por estado, decrescente; 5 = por data de postagem, crescente; e 6 = por data de postagem, decrescente.

```
function generate_sort_links($user_search, $sort) {
    $sort_links = '';
    switch ($sort) {
        case 1:
            $sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
                '&sort=1....>Título</a></td><td>Description</td>';
            break;
        case 3:
            $sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
                '&sort=3....>Estado</a></td>';
            break;
        case 5:
            $sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
                '&sort=5....>Data</a></td>';
            break;
        case 2:
            $sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
                '&sort=2....>Job Title</a></td><td>Description</td>';
            break;
        case 4:
            $sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
                '&sort=4....>State</a></td>';
            break;
        case 6:
            $sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
                '&sort=6....>Date Posted</a></td>';
            break;
        default:
            $sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
                '&sort=1....>Job Title</a></td><td>Description</td>';
            $sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
                '&sort=3....>State</a></td>';
            $sort_links .= '<td><a href = "' . $_SERVER['PHP_SELF'] . '?usersearch=' . $user_search .
                '&sort=5....>Date Posted</a></td>';
    }
    return $sort_links;
}
```

Se \$sort for igual a 1, significa que já classificamos pelo título, portanto, agora precisamos reclassificar em ordem decrescente.

Se \$sort ainda não tiver sido definido ou se for igual a 2, 4 ou 6, nós temos de exibir os links originais que classificam os dados em ordem crescente.

## Dê a build\_query() a capacidade de classificar

Agora nós temos duas funções para lidar com as buscas no Risky Jobs. build\_query() constrói uma consulta SQL com base em termos de busca digitados pelo usuário, e generate\_sort\_links() gera hyperlinks para os cabeçalhos do resultado da busca, os quais permitem ao usuário classificar os resultados. Mas build\_query() ainda não está totalmente pronta, uma vez que a consulta gerada por essa função ainda não faz a classificação. A função tem de adicionar uma cláusula ORDER BY à consulta. Mas precisa ser a ORDER BY correta, conforme determinado por um novo argumento \$sort:

```

function build_query($user_search, $sort) {
    $search_query = "SELECT * FROM riskyjobs";
    ...
    // Adiciona a cláusula WHERE à consulta de busca
    if (!empty($where_clause)) {
        $search_query .= " WHERE $where_clause";
    }

    // Classifica a consulta de buscas usando a
    // configuração de classificação
    switch ($sort) {
        // Crescente - por título
        case 1:
            $search_query .= " ORDER BY title";
            break;
        // Decrescente - por título
        case 2:
            $search_query .= " ORDER BY title DESC";
            break;
        // Crescente - por estado
        case 3:
            $search_query .= " ORDER BY state";
            break;
        // Decrescente - por estado
        case 4:
            $search_query .= " ORDER BY state DESC";
            break;
        // Crescente - por data (anúncios mais antigos)
        case 5:
            $search_query .= " ORDER BY date_posted";
            break;
        // Decrescente - por data (anúncios mais novos)
        case 6:
            $search_query .= " ORDER BY date_posted DESC";
            break;
        default:
            // Sem ordenação fornecida, não ordena consulta
    }
}

return $search_query;

```

*Agora nós estamos passando o argumento \$sort para a nossa função, além de \$user\_search.*

*Eis aqui as adições ao código de build\_query(). Esta declaração switch verifica o valor de \$sort e adiciona a instrução ORDER BY correspondente ao final da consulta de busca.*

*Quando os usuários carregam a página de resultados sem clicar em nenhum cabeçalho, \$sort estará vazio, portanto, nós não classificamos os resultados por padrão.*

*Nós retornamos \$search\_query como antes, mas desta vez com uma cláusula ORDER BY ao final.*

teste o script revisado search.php



## TEST DRIVE

**Reelabore o script Search para que ele use as duas novas funções personalizadas.**

Crie a nova função generate\_sort\_links() no script search.php e depois adicione o novo código à função build\_query(), para que ele gere uma consulta com resultados classificados. Não se esqueça de fazer a chamada a generate\_sort\_links() no script, no lugar do código que ecoa os cabeçalhos dos resultados.

Envie o script ao seu servidor web, abra a página search.html em um navegador e experimente fazer uma busca. Agora clique nos cabeçalhos das colunas da página de resultados para classificar os anúncios com base nos diferentes tipos de informações. Não se esqueça de clicar no mesmo cabeçalho mais de uma vez, para alternar entre a ordem crescente e a decrescente

Risky Jobs

Danger! Your dream job is out there.  
Do you have the guts to go find it?

Risky Jobs - Search

Find your risky job!

Bull Fighter Matador

Submit

A função build\_query() pega os termos de busca que o usuário digitou, explode a consulta em um array, remove quaisquer strings vazias do array, e cria uma consulta SQL com os termos e uma ORDER BY correspondente ao valor de classificação, se houver.

Risky Jobs

Danger! Your dream job is out there.  
Do you have the guts to go find it?

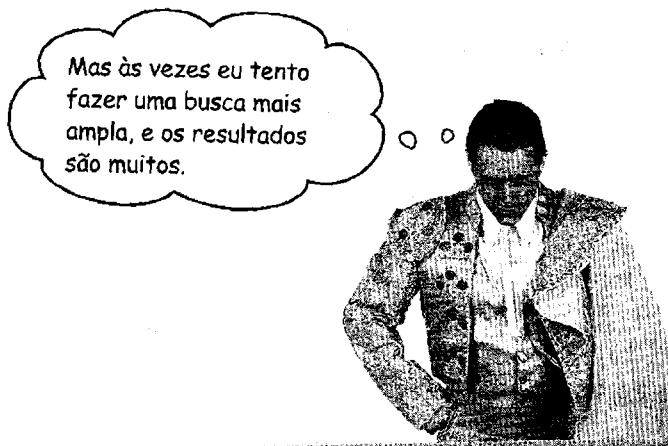
Risky Jobs - Search Results

Job Title	Description	State	Date Posted
Matador	Bustling dairy farm looking for part-time matador to entertain spirited bull with mild case of ADD... ...	VT	2008-03-11
Firefighter	The City of Dataville is hiring firefighters. No experience required - you will be trained. Non-smo...	OH	2008-05-22
Electric Bull Repairer	Ham's Howly Tock needs an experienced electric bull repairer. Free rides (after you fix it) and hal...	NJ	2008-11-14

A função  
generate\_sort\_  
links() gera  
os cabeçalhos  
cliqueáveis, incluindo  
o empacotamento  
das opções de  
classificação na  
URL de cada link.

Agora  
eu consigo ver os anúncios  
mais antigos, aqueles em que os  
empregadores estão ficando realmente  
desesperados para achar um toureiro  
em Vermont.





## Risky Jobs

Danger! Your dream job is out there.  
Do you have the guts to go find it?

### Risky Jobs - Search Results

Job Title	Description	State	Date Posted
Custard Walker	We need people willing to test the theory that you can walk on custard. We're going to fill a swi...	NM	2008-07-24
Shark Trainer	Training sharks to do cute tricks for the audiences at our new water theme park. You'll spend tim...	FL	2008-04-28
Voltage Checker	You'll be out in the field checking a.c. and d.c. voltages in the range of 3 to 250 or more volts. Y...	NC	2008-06-28
Antenna Installer	You'll be installing antennas and other metallic broadcast receiving equipment on the roofs of Miami...	FL	2008-09-04
Elephant Proctologist	Needed: experienced proctologist willing to work with large animals. Elephants at our zoo (in San Fr...	CA	2008-07-29
Airplane Engine Cleaner	Jet airplanes needing engines cleaned. In need of clean-minded individuals willing to handle rust an...	TX	2008-08-17
Matador	Bustling dairy farm looking for part-time matador to entertain spirited bull with mild case of ADD....	VT	2008-03-11
Paparazzo	Top celebrity photography firm looking for seasoned paparazzo to stalk temperamental lip-syncing pop...	CA	2008-03-24
Tightrope Walker	Fledgling big top looking for three-ring professional with 1-3 years of experience to perform tightr...	TX	2008-11-14
Crocodile Dentist	Do you love animals and hate plaque? Well, then this might be the job for you! Our crocodile farm ...	FL	2008-07-14
Mime	We need some fresh new faces. Full health insurance and shin pads provided. Must love kids....	NY	2008-11-02
Pet Food Tester	We pride ourselves on how good our pet food tastes. Now you can help make our products even better...	MO	2008-11-09
Toreador	Lovely bovines waiting for your superior non-violent cape waving skills. Must pass basic bull fighti...	ID	2008-11-14
Electric Bull Repairer	Hank's Honky Tonk needs an experienced electric bull repairer. Free rides (after you fix it) and hal...	NJ	2008-07-27
	City of Dataville is hiring firefighters. No experience required - you will be trained. Non-smo...	OH	2008-05-22
	Bozo needs an acrobat. Slow f...	MO	2008-11-14
	Ice juggling? In Rurz...	AZ	2008-11-14
	Lead from great men thi...	MT	2008-11-14



Isto é um  
bocado de  
anúncios  
para uma só  
pessoa ler de  
uma só vez.

## PODER DO CÉREBRO

Como será que os outros sites evitam ter muitos resultados em uma mesma página?

**use paginação para mostrar os resultados**

## Podemos paginar os nossos resultados

Atualmente, nós estamos exibindo todos os nossos resultados em uma só página, o que é um problema quando uma busca encontra muitos anúncios. Em vez de forçar os usuários a irem de cima a baixo em uma página enorme para ver todos os resultados, podemos usar uma técnica chamada paginação para exibi-los. Quando você pagina os resultados, divide o conjunto de resultados em grupos, e então, exibe cada grupo em uma página web separada, desta forma:

Cada página mostra cinco resultados, junto com links para se acessar as outras páginas de resultados. Os usuários podem facilmente clicar para acessar cada página, sem precisar ficar subindo e descendo com o mouse.

The screenshot shows a search results page for 'Risky Jobs'. At the top, there's a banner with the text 'Danger! Your dream job is out there. Do you have the guts to go find it?' and a small image of a person in a hard hat. Below the banner, the title 'Risky Jobs - Search Results' is displayed. The results are presented in a grid format with two columns. The first column contains job titles and descriptions, while the second column contains state and date posted information. Navigation links for pages 1, 2, 3, 4, and 5 are shown at the bottom of each column. A large speech bubble on the right side contains the text: 'Isso é ótimo, mas como nós dividimos os nossos resultados em grupos? A nossa consulta SQL retorna todos os resultados encontrados pela string de busca.'

Job Title	Description	State	Date Posted
Custard Walker	We need peo walk on cust	NY	2008-11-02
Shark Trainer	Training sha our new wa	MO	2008-11-09
Voltage Checker	You'll be ou in the range	ID	2008-11-14
Anicona Installer	You'll be in broadcas	NJ	2008-11-14
Elephant Proctologist Needed: ex large anima	Paparazzo		2008-05-22
< 1 2 3 4 >	Tightrope Walker		
	Crocodile Dentist		
	< 1 2 3 4 >		

Estes links permitem que os usuários naveguem através de várias páginas.

A página atual não é um link – esta é a segunda página de resultados.

A paginação divide os resultados em grupos, e exibe cada grupo em uma página web própria.

**Nós precisamos de uma consulta que retorne apenas um subconjunto dos resultados, e não todos eles.**

Afortunadamente, o SQL já nos oferece uma forma de fazer isso: a cláusula LIMIT. Vamos rever LIMIT e aprender como podemos usá-la para dividir os nossos resultados em grupos de cinco...



# Obtenha apenas as linhas de que você precisa com LIMIT

A chave para controlar quais linhas nós iremos exibir em uma dada página é adicionar mais uma cláusula à nossa consulta de busca, a cláusula LIMIT. Para obter um máximo de cinco linhas, nós adicionamos LIMIT 5 ao final da nossa consulta, desta forma:

```
SELECT * FROM riskyjobs
ORDER BY job_title
LIMIT 5
```

Só retorna os cinco primeiros resultados, não importando quantos tenham sido realmente encontrados.

Sem uma cláusula WHERE, esta consulta retorna todos os anúncios do banco de dados, o que é equivalente a procurar sem nenhum termo de busca.

**LIMIT**  
controla quais e quantas linhas são retornadas por uma consulta SQL.

Se você se lembrar, nós usamos a função personalizada build\_query() para criar a nossa consulta no Risky Jobs. Para forçá-la a exibir apenas os cinco primeiros resultados, nós simplesmente adicionamos LIMIT 5 ao final da string da consulta, após esta ter sido criada:

```
$query = build_query($user_search, $sort);
```

```
$query .= " $query . " LIMIT 5";
```

Adicionar uma cláusula LIMIT ao final de uma consulta limita o número de linhas retornadas por ela, neste caso a cinco.

Isso funciona bem para se obter as cinco primeiras linhas dos resultados, mas e quanto às cinco linhas seguintes, e às cinco linhas depois dessas? Para obter linhas posteriores no conjunto de resultados, temos de modificar um pouco o nosso LIMIT. Mas como? LIMIT 10 obteria as 10 primeiras linhas, portanto isso não funcionaria. Nós precisamos obter as linhas 6 a 10, e para isso nós usamos LIMIT com uma sintaxe diferente. Quando você fornece dois argumentos a LIMIT, o primeiro controla o número de linhas que quer pular, e o segundo controla o número de linhas que você recebe. Por exemplo, eis o modo de obter as linhas de 11 a 25, o que seria a terceira páginas dos resultados:

```
$query = build_query($user_search, $sort);
```

```
$query = $query . " LIMIT 10 - 5";
```

O primeiro argumento diz a LIMIT quantas linhas pular – as dez primeiras.

O segundo argumento controla quantas linhas são retornadas – cinco, o mesmo que antes.

**use LIMIT para ajudar nos resultados de paginação**

## Controle os links das páginas com LIMIT

Uma tarefa importante da paginação é fornecer links que permitam ao usuário navegar entre as diferentes páginas de resultados. Nós podemos usar a cláusula LIMIT para definir os links de navegação na parte de baixo de cada página. Por exemplo, os links "próximo" e "anterior" podem ter, cada um, o seu LIMIT. O mesmo se aplica aos links numéricos que permitem ao usuário pular diretamente para uma página específica dos resultados.

Eis as cláusulas LIMIT para as três primeiras páginas dos resultados, junto com LIMITs para alguns dos links:

The figure displays three separate instances of a web page titled "Risky Jobs - Search Results". Each instance shows a list of job titles and descriptions, with navigation links at the bottom. The first instance has a LIMIT of 0, 5. The second has a LIMIT of 5, 5. The third has a LIMIT of 10, 5. In all cases, the LIMIT values are explicitly labeled below the navigation arrows.

Job Title	Description
Circus Walker	We need people willing to test the walk on circus. We're going to fill a position.
Shark Trainer	Training sharks to do cute tricks for our new water theme park. You'll spray.
Vintage Checker	You'll be out in the field checking items in the range of 3 to 250 or more years.
Antenna Installer	You'll be installing antennas and other broadcast receiving equipment on buildings.
Elephant Proctologist	Needed: experienced proctologists for large animals. Elephants at our zoo fit.

**Risky Jobs - Search Results**

Danger! Your dream job is out there.  
Do you have the guts to go find it?

Job Title Description

Circus Walker We need people willing to test the walk on circus. We're going to fill a position.

Shark Trainer Training sharks to do cute tricks for our new water theme park. You'll spray.

Vintage Checker You'll be out in the field checking items in the range of 3 to 250 or more years.

Antenna Installer You'll be installing antennas and other broadcast receiving equipment on buildings.

Elephant Proctologist Needed: experienced proctologists for large animals. Elephants at our zoo fit.

<- 1 2 3 4 ->

LIMIT 0, 5

**Risky Jobs - Search Results**

Danger! Your dream job is out there.  
Do you have the guts to go find it?

Job Title Description

Airplane Engine Cleaner Jet airplanes needing engines cleaned. Individuals willing to handle.

Matador Battling dairy farm looking for part-time entertain spangled bull with mild case of.

Paparazzo Top celebrity photography firm looking paparazzo to stalk temperamental lip.

Tightrope Walker Needing big top looking for three with 1-3 years of experience to perform.

Crocodile Dentist Do you love animals and hate plaque? This might be the job for you! Our except.

<- 1 2 3 4 ->

LIMIT 5, 5

**Risky Jobs - Search Results**

Danger! Your dream job is out there.  
Do you have the guts to go find it?

Job Title Description

Mime We need some fresh new faces. Full and skin pads provided. Must love.

Pet Food Tester We pride ourselves on how good our food is. Now you can help make our products.

Torador Lovely business waiting for your expert sword waving skills. Must pass basic.

Electric Bull Repairer Hank's Honky Tonk needs an experienced repairer. Free rides (after you fix it).

Breifighter The City of Dataville is hiring firef

experienced required - you will be

<- 1 2 3 4 ->

LIMIT 10, 5



Sem problemas. Nós só precisamos escrever um monte de consultas, com um LIMIT diferente para cada uma, certo?

**Mais ou menos isso. Nós precisamos de um LIMIT diferente dependendo da página e do link, mas podemos gerá-lo em vez de escrever várias consultas.**

Tudo o que precisamos fazer é modificar um pouco mais a nossa função build\_query(), para adicionar o LIMIT correto ao final da consulta que ela constrói.

## Mantenha registro dos dados da paginação

Para adicionar a nova funcionalidade de paginação a `build_query()`, nós precisamos configurar e manter registro de algumas variáveis que determinem quais resultados devemos consultar e mostrar em uma dada página. Essas variáveis são importantes também para determinar como os links de navegação na parte de baixo da página serão gerados.

### **\$cur\_page**

Obtém a página atual, `$cur_page`, a partir da URL do script via `$_GET`. Se não houver nenhuma página atual para ser passada através da URL, define `$cur_page` como a primeira página (1).

### **\$results\_per\_page**

Este é o número de resultados por página, que você pode escolher para melhor se adequar à aparência da página, colocando a quantidade de resultados que caibam bem no layout da sua página. É daqui que vem o segundo argumento para a cláusula `LIMIT`.

### **\$skip**

Calcula o número de linhas a serem puladas antes de começar a exibir as linhas da página atual. Esta variável é o que controla onde cada página começa, em termos de resultados, fornecendo o primeiro argumento para a cláusula `LIMIT`.

### **\$total**

Executa uma consulta que obtém todas as linhas, sem nenhum `LIMIT`, e depois conta os resultados e armazena o resultado em `$total`. Em outras palavras, este é o número total de resultados da busca.

### **\$num\_pages**

Calcula o número de páginas, `$num_pages`, usando `$total` dividido por `$results_per_page`. Assim, para cada busca, existe uma quantidade `$total` de linhas de resultados, mas estas são exibidas uma página por vez, com cada página contendo um número `$results_per_page` de combinações. Existem `$num_pages` páginas, e a página atual é identificada por `$cur_page`.

## Defina as variáveis da paginação

A maioria das variáveis de paginação pode ser definida puramente através de informações fornecidas pela URL, que fica acessível por meio da superglobal `$_GET`. Por exemplo, as variáveis `$sort`, `$user_search` e `$cur_page` surgem todas diretamente a partir de dados GET. Nós podemos então usar essas variáveis para calcular quantas linhas devem ser puladas para chegarmos à primeira linha de dados, `$skip`. A variável `$results_per_page` é um pouco diferente, no sentido de que nós simplesmente a definimos com o número de resultados que queremos ter em cada página, o que é mais uma questão de gosto pessoal, levando-se em consideração o layout da página de resultados.

```
Obtém a ordem de  
classificação, que é  
um número inteiro  
na faixa de 1 a 6.  
  
Obtém a página atual,  
partir da URL  
via GET. Se $sort = $_GET['sort'];  
não houver uma página atual,  
define $cur_page como 1.  
// Obtém a configuração de classificação e as palavras-chaves de busca a partir da URL, usando GET  
$user_search = $_GET['usersearch'];  
  
Obtém a string de busca que o usuário digitou no formulário.  
  
Se o número da página não estiver definido, o padrão é a primeira página.  
  
Define o número de resultados por página.  
Calcula o número da primeira linha da página, $skip.  
$results_per_page = 5; // número de resultados por página  
$skip = ($cur_page - 1) * $results_per_page;  
  
Este cálculo resulta em 0 para a página 1, 5 para a página 2, 10 para a página 3, etc.
```

Ainda faltam duas importantes variáveis: `$total` e `$num_pages`. Essas variáveis só podem ser definidas após realizarmos uma consulta inicial para descobrir quantos resultados foram encontrados no banco. Uma vez que saibamos quantos resultados nós temos, é possível definir essas variáveis e então usar `LIMIT` nos resultados...

# Revise a consulta para ter resultados paginados

Agora que nós temos nossas variáveis definidas, precisamos revisar o script Search para que ele, em vez de fazer consultas para todos os resultados, faça-as apenas para o subconjunto de resultados de que precisamos para a página que o usuário está visualizando no momento. Para isso, primeiramente fazemos uma consulta para que a variável \$total possa ser definida, e a variável \$num\_pages possa ser calculada. Em seguida, fazemos uma segunda consulta que usa \$skip e \$results\_per\_page para gerar uma cláusula LIMIT, que adicionamos ao final da consulta. Eis a seção do script search.php revisada, com essas adições salientadas:

```

    mysqli_num_rows() retorna uma
    contagem de quantas linhas foram
    retornadas pela consulta.

    Esta consulta obtém
    todas as linhas, sem
    nenhum LIMIT.

// Consulta para obter o total de resultados
$query = build_query($user_search, $sort);
$result = mysqli_query($dbc, $query);
$total = mysqli_num_rows($result);

$num_pages = ceil($total / $results_per_page);

    A função mysqli_num_
    rows() retorna o número
    total de linhas

    Calcula o número de páginas,
    dividindo o número total
    de linhas pelo número de
    resultados por página, e depois
    arredondando o resultado.

    A função ceil() arredonda
    um número até o inteiro
    mais próximo - o "teto".

// Consulta novamente, para obter somente o subconjunto dos resultados

$query = $query . "LIMIT $skip, $results_per_page";
$result = mysqli_query($dbc, $query);
while ($row = mysqli_fetch_array($result)) {
    echo '<tr class="results">';
    echo '<td valign="top" width="20%">' . $row['title'] . '</td>';
    echo '<td valign="top" width="50%">' . substr($row['description'], 0,
100) . '...</td>';
    echo '<td valign="top" width="10%">' . $row['state'] . '</td>';
    echo '<td valign="top" width="20%">' . substr($row['date_posted'], 0,
10) . '</td>';
    echo '</tr>';
}
echo '</table>';

    Pula este
    número de
    linhas...
    ...e retorna este
    número de linhas.

    Emite uma segunda consulta, mas
    desta vez limitando os resultados
    para a página atual.

```

## Gere os links de navegação da página

Então, nós definimos algumas variáveis e construímos uma nova consulta SQL que retorna um subconjunto de resultados para a página. Só o que ainda precisamos fazer é gerar os links de navegação da parte de baixo da página de resultados: o link "previous" ("página anterior"), links numéricos para cada página de resultados, e o link "next" ("página seguinte"). Nós já temos todas as informações de que precisamos para criar os links. Vamos revê-las para nos certificarmos de que está claro o modo como elas serão usadas.

### \$user\_search

Cada link ainda precisa saber o que o usuário está realmente procurando, portanto nós temos que passar os termos de busca na URL de cada link.

### \$num\_pages

Nós precisamos saber quantas páginas existem, para podermos gerar links para cada uma delas.

### \$cur\_page

Os links de navegação dependem inteiramente da página atual, de forma que é muito importante que essa informação seja empacotada na URL de cada link.

### \$sort

A ordem de classificação também influí sobre os links de paginação, porque a ordem precisa ser mantida ou então a paginação inteira não faria muito sentido.

Ok, nós sabemos quais são as informações de que precisamos para gerar os links de navegação, portanto, estamos prontos para escrever o código PHP necessário. Esse código poderia simplesmente ser colocado no script search.php, mas que tal se o colocássemos na sua própria função personalizada? Dessa forma, o código principal do script que gera os resultados da busca pode permanecer muito mais simples, exigindo apenas uma linha de código para gerar os links das páginas – uma chamada a esta nova função, que chamaremos de `generate_page_links()`.

O único senão é que nós não queremos que esta função seja chamada se houver apenas uma página de resultados. Assim, precisamos verificar o número de páginas antes de chamar a nova função `generate_page_links()`. Eis o modo como podemos realizar a verificação e chamar a função, sem nos esquecermos de enviar as informações necessárias na forma de argumentos para a função:

```
if ($num_pages > 1) {  
    echo generate_page_links($user_search, $sort, $cur_page, $num_pages);
```

Primeiramente, verifique se existe mais de uma página de resultados; caso contrário, não gere os links.

Envie a string de busca; a ordem de classificação; a página atual e o número total de páginas a serem usadas na criação dos links.



## Ímas de Geladeira do PHP & MySQL

A função generate\_page\_links() está quase pronta, mas estão faltando alguns pedaços de código. Use os ímas para preencher o código que está faltando e dar ao Risky Jobs a capacidade de gerar links de navegação para as páginas.

```

function generate_page_links($user_search, $sort, $cur_page,
$num_pages) {
    $page_links = '';

    // Se esta página não for a primeira, gera o link "previous"
    if (.....) {

        $page_links .= '<a href="'. $_SERVER['PHP_SELF'] .
            '?usersearch=' . $user_search .
            '&sort=' . $sort .

            '&page=' . (.....) . '"><-</a> ';

    }
    else {
        $page_links .= '<- ';
        ↗ O link "previous" aparece como uma
        ↗ seta para a esquerda, "<-".
    }

    // Faz um loop através das páginas, gerando os links com os
    números das páginas
    for($i = 1; $i <= $num_pages; $i++) {

        if (.....) {

            $page_links .= ' ' . $i;
        }
        else {
            $page_links .= ' <a href="'. $_SERVER['PHP_SELF'] .
                '?usersearch=' . $user_search .
                '&sort=' . $sort .
                '&page=' . $i . '">' . $i . '</a>';
        }
    }

    // Se esta página não for a última, gera o link "next"
    if (.....) {

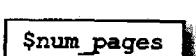
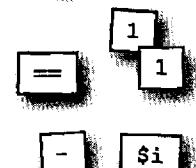
        $page_links .= ' <a href="'. $_SERVER['PHP_SELF'] .
            '?usersearch=' . $user_search .
            '&sort=' . $sort .

            '&page=' . ($cur_page + 1) . '">-</a>';

    }
    else {
        $page_links .= ' ->';
        ↗ O link "next" aparece como uma
        ↗ seta para a direita, "->".
    }

    return $page_links;
}

```



a função completa `generate_page_links()`



## Ímas de Geladeira do PHP & MySQL - Solução

A função `generate_page_links()` está quase pronta, mas estão faltando alguns pedaços de código. Use os ímas para preencher o código que está faltando e dar ao Risky Jobs a capacidade de gerar links de navegação para as páginas.

```
function generate_page_links($user_search, $sort, $cur_page, $num_pages) {
    $page_links = '';

    // Se esta página não for a primeira, gera o link "previous"
    if ($cur_page > 1) {
        $page_links .= '<a href="' . $_SERVER['PHP_SELF'] .
            '?usersearch=' . $user_search .
            '&sort=' . $sort . ' <->' .
            '&page=' . ($cur_page - 1) . '></a>';
    } else {
        $page_links .= '<->';
    }

    // Faz um loop através das páginas, gerando os links com os números das
    // páginas
    for ($i = 1; $i <= $num_pages; $i++) {
        if ($cur_page == $i) {
            $page_links .= ' ' . $i;
        } else {
            $page_links .= ' <a href="' . $_SERVER['PHP_SELF'] .
                '?usersearch=' . $user_search .
                '&sort=' . $sort .
                '&page=' . $i . '>' . $i . '</a>';
        }
    }

    // Se esta página não for a última, gera o link "next"
    if ($cur_page < $num_pages) {
        $page_links .= '<a href="' . $_SERVER['PHP_SELF'] .
            '?usersearch=' . $user_search .
            '&sort=' . $sort .
            '&page=' . ($cur_page + 1) . '></a>';
    } else {
        $page_links .= '<->';
    }

    return $page_links;
}
```

Nós ainda precisamos enviar os dados da busca do usuário, bem como a ordem de classificação, na URL de cada link.

O link "previous" aparece como uma seta para a esquerda, "<->".

Certifique-se de que cada link aponte de volta para o mesmo script - nós estamos apenas enviando um número de página diferente com cada link.

O link para uma página específica é simplesmente o número da página.

# Montando o script Search completo

E finalmente nós chegamos ao script Search completo do Risky Jobs, que exibe os resultados apropriados com base nos termos de busca do usuário, gera cabeçalhos clicáveis para a classificação dos resultados, faz a paginação desses resultados e gera links de navegação na parte de baixo de cada página.

```
<?php
    // Esta função cria uma consulta de busca a partir das palavras-chaves
    // de busca e da configuração de classificação
    function build_query($user_search, $sort) {
        ...
        return $search_query;
    }

    // Esta função cria links com base na configuração de classificação
    // especificada
    function generate_sort_links($user_search, $sort) {
        ...
        return $sort_links;
    }

    // Esta função cria links de navegação, com base na página atual e no
    // número de páginas

    function generate_page_links($user_search, $sort, $cur_page, $num_pages)
    {
        ...
        return $page_links;
    }

    // Obtém a configuração de classificação e as palavras-chaves de busca a
    // partir da URL, usando GET
    $sort = $_GET['sort'];
    $user_search = $_GET['usersearch'];

    // Calcula as informações de paginação.
    $cur_page = isset($_GET['page']) ? $_GET['page'] : 1;
    $results_per_page = 5; // número de resultados por página
    $skip = ($cur_page - 1) * $results_per_page;

    // Começa a gerar a tabela de resultados
    echo '<table border="0" cellpadding="2">';

    // Gera os cabeçalhos dos resultados
    echo '<tr class="heading">';
    echo generate_sort_links($user_search, $sort);
    echo '</tr>';

    // Chama a função generate_sort_links() para criar os links para os
    // cabeçalhos dos resultados, e depois os exibe..
    Aguente firme,
    ainda tem mais!

```



*o final search.php*

## O script Search completo, continua...

```
// Conecta-se ao banco de dados
require_once('connectvars.php');
$dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

// Consulta para obter o total de resultados
$query = build_query($user_search, $sort);
$result = mysqli_query($dbc, $query);
$total = mysqli_num_rows($result);
$num_pages = ceil($total / $results_per_page);

// Consulta novamente, para obter somente o subconjunto dos resultados
$query = $query . " LIMIT $skip, $results_per_page";
$result = mysqli_query($dbc, $query);
while ($row = mysqli_fetch_array($result)) {
    echo '<tr class="results">';
    echo '<td valign="top" width="20%">' . $row['title'] . '</td>';
    echo '<td valign="top" width="50%">' . substr($row['description'], 0,
100) . '...</td>';
    echo '<td valign="top" width="10%">' . $row['state'] . '</td>';
    echo '<td valign="top" width="20%">' . substr($row['date_posted'], 0,
10) . '</td>';
    echo '</tr>';
}
echo '</table>';

// Gera links de navegação, se tivermos mais de uma página
if($num_pages > 1) {
    echo generate_page_links($user_search, $sort, $cur_page, $num_pages);
}

mysqli_close($dbc);
?>
```

*Chama build\_query() para criar a consulta SQL para a busca.*

*aqui a cláusula LIMIT busca apenas os subconjuntos do resultado*

*E aqui vai o código que escrevemos para reduzir a descrição e a data de postagem dos anúncios, usando a função substr().*

*Mantém a casa arrumada, fechando a conexão com o banco de dados.*

*não existem*

## Perguntas Ídiotas

**P:** Nós realmente precisamos enviar as informações de busca, classificação e paginação para generate\_page\_links()?

**R:** Sim. E o motivo para isso tem a ver com o fato de as funções, quando bem elaboradas, não devem manipular dados fora do seu próprio código. Assim, uma função só deve acessar dados enviados a ela em um argumento, e só fazer modificações em dados que ela retornar.

**I:** Ok, e quanto a ecoar dados? Por que generate\_page\_links() simplesmente não faz echo dos links?

**R:** O mesmo problema. Ao ecoar dados para navegador, a função estaria, na prática, dando um passo maior que as pernas, ao fazer uma modificação em algum outro lugar. É muito mais difícil realizar debug e manutenção quando não está claro quais dados a função está modificando. A solução é fazer sempre a função retornar os dados que está afetando, e aí então você pode fazer o que quiser com eles, mas *fora da função*.



## TEST DRIVE

### Finalize o script Search do Risky Jobs.

Adicione a nova função generate\_page\_links() ao script search.php, certificando-se de adicionar também o código que chama essa função após verificar se existe mais de uma página de resultados. Além disso, crie e inicialize as variáveis usadas como argumentos para a função. E não se esqueça de atualizar o código da consulta para que ela use LIMIT para obter o subconjunto correto de resultados para cada página.

Quando estiver tudo pronto, envie o novo script search.php para o seu servidor web, e depois abra a página search.html em um navegador. Experimente fazer algumas buscas, e não se esqueça de procurar por alguns termos que você saiba que vão gerar muitos resultados, para ver as novas funcionalidades de paginação em ação. Para obter o máximo de páginas, faça uma busca sem digitar nenhum termo no formulário.

## Risky Jobs

Danger! Your dream job is out there.  
Do you have the guts to go find it?

**Risky Jobs - Search Results**

Title	Description	State	Date Posted
Matador	Bustling dairy farm looking for part-time matador to entertain spirited bull with mild case of ADD...	VT	2008-03-11
Paparazzo	Top celebrity photography firm looking for seasoned paparazzo to stalk temperamental lip-syncing pop...	CA	2008-03-24
Shark Trainer	Training sharks to do cute tricks for the audiences at our new water theme park. You'll spend tim...	FL	2008-04-28
Firefighter	The City of Dataville is hiring firefighters. No experience required - you will be trained. Non-smo...	OH	2008-05-22
Voltage Checker	You'll be out in the field checking a.c. and d.c. voltages. NC in the range of 3 to 250 or more volts. Y...	NC	2008-06-28

< 1 2 3 4 >

Finalmente  
encontrei meu  
emprego dos sonhos!  
Vermont, aí vou eu!

**Faça o Download!**

Não se esqueça: o código-fonte completo da aplicação Risky Jobs está disponível para download no site da Alta Books:

[www.altabooks.com.br](http://www.altabooks.com.br)

Ernesto encontrou um  
emprego com a quantidade  
perfeita de risco!

você está aqui ▶

559

caixa de ferramentas do php &amp; mysql



## Sua Caixa de Ferramentas do PHP & MySQL

O script Search do Risky Jobs exigiu uma boa quantidade de novas técnicas de PHP e MySQL. Vamos recapitular algumas das mais importantes.

### LIKE

Use LIKE para procurar por dados dentro de uma consulta SQL com exata necessidade de uma combinação exata. Coloque um % na frente e/ou depois de um termo para dizer a LIKE que o termo pode ter outros caracteres em torno dele.

### explode(), implode()

A função explode() do PHP divide uma string em um array de substrings separadas por um delimitador comum. Por exemplo, espaço ou vírgula. implode() faz o oposto — cria uma única string a partir de um array de strings inserindo um delimitador entre elas.

### substr()

Esta função do PHP extrae uma parte de uma string com base em argumentos que você fornece. Você pode obter o inicio da string, o final dela ou qualquer parte do meio.

### str\_replace()

Chame esta função do PHP para realizar sua operação de localizar e substituir em uma string de texto, substituindo um caractere ou sequência de caracteres por outro.

### Função personalizada

Um pedaço de código PHP organizado em um pacote nomeado e reutilizável. A ideia é isolar o código que realiza uma determinada tarefa de modo que ele possa ser reutilizado com o mínimo de esforço e de duplicação de código.

### switch-case

Um constructo bônus de decisões do PHP, que lhe permite executar um conjunto variado de códigos baseado em um dado valor. O comando if é substituído por switch/case, dividido em casos individuais, zero ou mais. Pode integrar seu código com uma estrutura de decisão na forma de uma grande switch.

### LIMIT

A cláusula LIMIT permite-lhe controlar o número de linhas que são retornadas por uma consulta SQL. É isso comente isso. LIMIT pode também limitar o resultado de conjuntos de resultados permitindo que você forme um subset dentro dos resultados.



## **Regras Para Substituição**



A professora trocou o  
hamster da sala por outro!  
Será que ela achou que não  
iríamos perceber?

**As funções string são adoráveis. Mas, ao mesmo tempo, são limitadas.** Certamente elas podem realizar tarefas como lhe dizer a extensão da sua string, truncar a string, trocar certos caracteres por outros. Mas às vezes você precisa de liberdade para lidar com manipulações de texto mais complexas. É aqui que as expressões regulares podem ajudar. Elas podem modificar strings de forma precisa, com base em um conjunto de regras, em vez de um só critério.

## Risky Jobs permite que os usuários submetam currículos

O site riskyjobs.biz cresceu. A empresa agora permite que os usuários digitem os seus currículos e informações de contato em um formulário web, para que os empregadores que usam o Risky Jobs possam encontrá-los mais facilmente. O formulário se parece com o seguinte:

Além das informações de contato normais, o usuário deve digitar o seu emprego desejado, bem como o seu currículo.

Danger! Your dream job is out there!  
Do you have the guts to go find it?

Risky Jobs - Registration

Register with Risky Jobs, and post your resume.

First Name: \_\_\_\_\_  
Last Name: \_\_\_\_\_  
Email: \_\_\_\_\_  
Phone: \_\_\_\_\_  
Desired Job: \_\_\_\_\_

Paste your resume here:

Submit

O novo formulário de registro do Risky Jobs permite que os usuários digitem informações pessoais, de modo que os potenciais empregadores possam encontrá-los.

As informações dos nossos usuários que estão procurando emprego ficam armazenadas em uma tabela que pode ser consultada por empregadores, recrutadores e headhunters para identificarem potenciais novos contratados. Mas há um problema... os dados digitados no formulário aparentemente não são confiáveis!

Nome: Four Fingers  
Sobrenome: McGraw  
Email: four@gregs-listnet  
Telefone: 555-098  
Emprego Desejado: Malabarista de facas

Nome: Jimmy  
Sobrenome: Swift  
Email: JS@sim-u-duck.com  
Telefone: 636 4652  
Emprego Desejado: Ninja

Primeiro eu queria contratar um ninja e não consegui, porque o número de telefone dele estava faltando. Agora, o meu email para este malabarista voltou. Estou cheio desses dados inválidos do banco de currículos da Risky Jobs.

Os empregadores podem fazer buscas no banco de dados do Risky Jobs e então entrar em contato com as pessoas para, possivelmente, contratá-las... desde que as informações suficientes de contato suficientes tenham sido digitadas!





Segue abaixo parte do código do script registration.php, que exibe e processa os dados digitados pelo usuário no formulário para se cadastrar. Escreva o que você acha que há de errado com o código, e como ele poderia ser modificado para resolver o problema dos dados inválidos.

```
<?php
if (isset($_POST['submit'])) {
    $first_name = $_POST['firstname'];
    $last_name = $_POST['lastname'];
    $email = $_POST['email'];
    $phone = $_POST['phone'];
    $job = $_POST['job'];
    $resume = $_POST['resume'];
    $output_form = 'no';

    if (empty($first_name)) {
        // $first_name está em branco
        echo '<p class="error">Você se esqueceu de digitar seu nome.</p>';
        $output_form = 'yes';
    }

    if (empty($last_name)) {
        // $last_name está em branco
        echo '<p class="error">Você se esqueceu de digitar seu sobrenome.</p>';
        $output_form = 'yes';
    }

    if (empty($email)) {
        // $email está em branco
        echo '<p class="error">Você se esqueceu de digitar seu endereço de email.</p>';
        $output_form = 'yes';
    }

    if (empty($phone)) {
        // $phone está em branco
        echo '<p class="error">Você se esqueceu de digitar seu número de telefone.</p>';
        $output_form = 'yes';
    }
}

Continua a validar os campos não vazios referentes ao emprego
desejado e ao currículo.

else {
    $output_form = 'yes';
}

if ($output_form == 'yes') {
?>
Exibe o formulário.
```

## exercício solução



### Exercício Solução

```
<?php
if (isset($_POST['submit'])) {
    $first_name = $_POST['firstname'];
    $last_name = $_POST['lastname'];
    $email = $_POST['email'];
    $phone = $_POST['phone'];
    $job = $_POST['job'];
    $resume = $_POST['resume'];
    $output_form = 'no';

    if (empty($first_name)) {
        // $first_name está em branco
        echo '<p class="error">Você se esqueceu de digitar seu nome.</p>';
        $output_form = 'yes';
    }

    if (empty($last_name)) {
        // $last_name está em branco
        echo '<p class="error">Você se esqueceu de digitar seu sobrenome</p>';
        $output_form = 'yes';
    }

    if (empty($email)) {
        // $email está em branco
        echo '<p class="error">Você se esqueceu de digitar seu endereço de
email.</p>';
        $output_form = 'yes';
    }

    if (empty($phone)) {
        // $phone está em branco
        echo '<p class="error">Você se esqueceu de digitar seu número de
telefone.</p>';
        $output_form = 'yes';
    }
}

else {
    $output_form = 'yes';
}

if ($output_form == 'yes') {
?>
    Exibe o formulário
...
}
```

Segue abaixo parte do código do script registration.php, que exibe e processa os dados digitados pelo usuário no formulário para se cadastrar. Escreva o que você acha que há de errado com o código e como ele poderia ser modificado para resolver o problema dos dados inválidos.

O script verifica se há campos em branco no formulário, o que é bom, mas alguns dos campos têm dados mais especializados que precisam aderir a um determinado formato.

Não há muito mais que possamos verificar no que diz respeito a nome e sobrenome, portanto este código está ok.

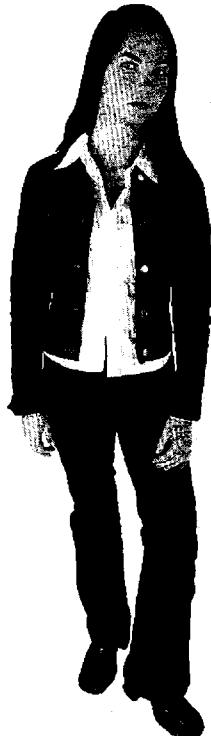
Um endereço de email tem um formato bastante específico, que nós devemos verificar antes de aceitar os dados fornecidos pelo usuário.

Four Finger McGraw se esqueceu de colocar um ponto perto do final do seu endereço de email - o formulário precisa capturar esse tipo de erro!

A mesma coisa com o número de telefone - o formulário não deve ser aceito a não ser que tenhamos certeza de que o telefone está no formato correto.

Jimmy Swift não forneceu o DDD com o seu número de telefone, o que o formulário deveria ter exigido.

O que nós realmente precisamos é de uma forma de verificar endereços de email e números telefônicos, os dois campos do formulário que têm um formato específico. Para os demais campos, basta verificar se eles não estão vazios.



Por que nós não usamos algumas funções string para consertar os dados inválidos? Não poderíamos ter usado str\_replace() para adicionar os dados que estão faltando?

**Você pode consertar alguns dados com funções string, mas elas não ajudam muito quando os dados precisam atender a um padrão bastante específico.**

As funções strings são adequadas para operações de localizar-e-substituir simples. Por exemplo, se os usuários estiverem submetendo os seus números telefônicos usando pontos para separar os blocos de dígitos, em vez de hífens, nós poderíamos facilmente escrever algum código com str\_replace() para substituir os pontos pelos hífens.

Porém, para qualquer coisa que não podemos prever, como por exemplo, o DDD do número de Jimmy Swift, precisamos pedir à pessoa que está submetendo o formulário para ser mais clara. E a única maneira que temos de saber que está faltando o DDD é entendendo o padrão exato de um número telefônico. O que nós realmente precisamos é de uma validação mais avançada, para garantir que coisas como números telefônicos e endereços de email sejam digitados da forma correta.

Ok, mas ainda assim, não podemos usar funções string para fazer essa validação?



**As funções string não são realmente úteis para validação de dados, a não ser que a validação seja muito primitiva.**

Pense na forma como tentaria validar um endereço de email usando funções string. O PHP tem uma função chamada strlen(), a qual lhe diz quantos caracteres existem em uma string. Mas não existe uma quantidade de caracteres pré-definida para coisas como endereços de email. É claro que essa função poderia ajudar no caso dos números telefônicos, porque eles frequentemente contêm uma quantidade consistente de números, mas você ainda poderá ter de lidar com pontos, hífens e parênteses.

Voltando aos endereços de email, o formato deles é simplesmente complexo demais para que as funções strings sejam úteis. O que nós estamos realmente procurando aqui é por padrões específicos de dados, o que requer uma estratégia de validação que possa checar os dados do usuário em relação a um padrão, para ver se os dados são válidos. Modelar padrões para os seus dados é o cerne desse tipo de validação.

*decida como deverá ser a aparência dos seus dados*

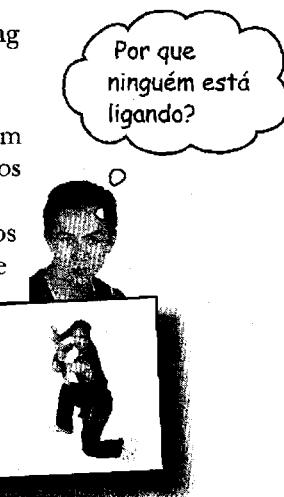
## Decida como deverá ser a aparência dos seus dados

O nosso desafio é especificar, clara e exatamente, qual deve ser a aparência de uma determinada informação, nos mínimos detalhes. Considere o número telefônico de Jimmy. É bem óbvio, para um observador humano, que está faltando o DDD nesse número. Mas a validação de formulários não é feita por humanos; ela é executada por código PHP. Isso significa que nós temos de "ensinar" o nosso código como procurar por uma string de dados digitados pelo usuário, e determinar se os dados atendem ao padrão estabelecido para os números telefônicos.

Elaborar um padrão desses pode ser um desafio, que exige pensar bem sobre a gama de possibilidades para o tipo de informação. Os números telefônicos são relativamente simples, uma vez que eles possuem, nos Estados Unidos, 10 dígitos com delimitadores opcionais. Os endereços de email são outra história, mas iremos nos ocupar deles mais adiante neste capítulo.

Para um ser humano, é fácil ver que Jimmy se esqueceu de colocar o seu DDD, mas não é tão simples assim fazer essa mesma observação a partir de código PHP.

First Name: Jimmy  
Last Name: Swift  
Email: JS@sim-u-duck.com  
Phone: 636 4652  
Desired Job: Ninja



não existem

### Perguntas Idiotas

**P:** Ainda não sei se entendi por que não posso simplesmente usar `isset()` e `empty()` para a validação do nosso formulário.

**R:** Essas duas funções lhe dizem se o usuário que submeteu o formulário colocou ou não dados em um campo de texto, mas elas não lhe dizem nada sobre os dados propriamente ditos. No que diz respeito à função `empty()`, não existe absolutamente nenhuma diferença entre o usuário digitar \*(707) 827-700\* ou \*4FG8SXY12\* no campo de telefone do nosso formulário. Isso seria um grande problema para sites como o Risky Jobs, que depende de dados confiáveis para entrar em contato com os candidatos a empregos.

**P:** Se `isset()` e `empty()` não

funcionam, não podemos simplesmente ter alguém para verificar os dados depois de eles serem colocados no banco?

**R:** Você pode, mas a essa altura frequentemente já é tarde demais para consertar os dados inválidos. Se estiver faltando o DDD em um número telefônico, nós teremos de pedir ao usuário que esclareça as coisas, submetendo novamente os dados desse campo. Se esperar para verificar os dados só depois que eles já estiverem no banco, talvez você não tenha como contatar o usuário para lhe dizer que alguns dos seus dados estavam inválidos. E uma vez que o usuário provavelmente não vai perceber que cometeu um engano, é possível que ele jamais fique sabendo que há algo de errado. Assim, o melhor plano de ação é validar os dados dos usuários

imediatamente, quando eles submeterem o formulário. Dessa maneira, você poderá exibir uma mensagem de erro e pedir que eles preencham o formulário novamente.

**P:** Mas então como decidimos se os dados digitados são válidos ou não?

**R:** Isso depende do tipo dos dados. Diferentes tipos de informações têm diferentes regras que precisam seguir: qual tipo de caracteres elas contêm, quantos caracteres, em qual ordem esses caracteres devem estar. Assim, você precisa comunicar essas regras ao seu código PHP. Vamos dar uma olhada com mais detalhes nas regras que governam os números telefônicos...



## Aponte seu lápis

Escreva abaixo todas as formas de representar um número telefônico que você conseguir se lembrar.

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

Quais são algumas regras que é razoável esperar que os seus usuários sigam ao preencher o formulário? Por exemplo, números telefônicos não devem conter letras.

Eis uma regra,  
como exemplo.



Nós poderíamos exigir regras como: somente dígitos podem ser usados neste campo.

.....  
.....  
.....  
.....  
.....



## Aponte seu lápis Solução

Escreva abaixo todas as formas de representar um número telefônico que você conseguir se lembrar.

Espaços, traços,  
parênteses, e às vezes  
pontos podem aparecer em  
números telefônicos

É possível até mesmo incluir letras  
em um número telefônico, mas isso  
já estaria perto de ultrapassar  
os limites do que nós devemos  
considerar como um número válido.

555 636-4652  
(555) 636-4652  
(555)636-4652  
(555) 6364-652  
555636-4652  
555 636-4652  
555,636,4-652  
5556364-652  
555-636-4652  
555 ME NINJA

Eis uma  
regra,  
como  
exemplo.

Quais são algumas regras que é razoável esperar que os seus usuários sigam ao preencher o formulário? Por exemplo, números telefônicos não devem conter letras.

Nós poderíamos exigir regras como: somente dígitos podem ser usados neste campo.

Nós poderíamos dividir o número em três campos do formulário, um para o DDD, um para os três primeiros dígitos, e um para os quatro últimos dígitos.

Ou nós poderíamos informar às pessoas que o número de telefone delas deve seguir o formato (555)636-4652. Nós fazemos as regras.

Existem tantos padrões possíveis. Como poderemos criar regras para dar conta de todos eles?



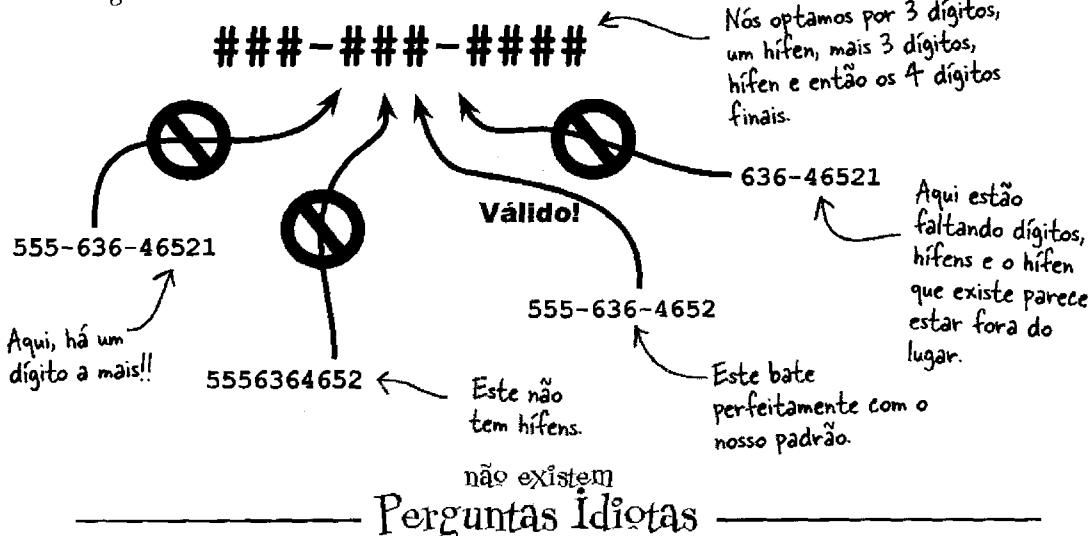
**Existem algumas coisas que sabemos com certeza sobre números telefônicos, e podemos usar essas coisas para criar as regras.**

Primeiramente, eles não podem começar com 1 (código de interurbano) nem 0 (telefonista). Em segundo lugar, deve haver 10 dígitos. E mesmo que algumas pessoas possam ter maneiras inteligentes de representar os seus números com letras, os números telefônicos são essencialmente números – 10 dígitos quando incluímos o DDD.

## Formule um padrão para números telefônicos

Para ir além da validação básica, como a de `empty()` e `isset()`, nós precisamos adotar um padrão para os nossos dados. No caso do número telefônico, isso significa que temos de optar por apenas um formato que esperamos receber do campo telefone de nosso formulário. Uma vez que decidimos por um padrão/formato para o telefone, podemos validar esses dados.

Segue o que provavelmente é o formato de telefone mais comum em uso hoje em dia, pelo menos para números dos EUA. Usar este formato significa que, se os dados que os usuários digitarem não se conformarem a ele, o script PHP irá rejeitar o formulário e exibir uma mensagem de erro.



### Perguntas Idiotas

**P:** Eu tenho que usar exatamente esse padrão para os números telefônicos?

**R:** Esse é o que estamos usando para o Risky Jobs porque ele é basicamente o padrão, mas ao elaborar os seus próprios formulários você deve escolher um que faça sentido para você. Apenas tenha em mente que, quanto mais comum o padrão for, maior a probabilidade de os usuários o seguirem.

**P:** Nós não poderíamos simplesmente dizer aos usuários para digitar um padrão como `#####` e depois usar as funções `string` do PHP para nos certificarmos de que os dados contenham 10 caracteres numéricos?

**R:** Você poderia fazer isso e seria suficiente se esse fosse o padrão esperado pelos seus usuários. Infelizmente, na realidade esse não é um padrão muito bom, porque a maioria das pessoas não escreve os seus números telefônicos com os números todos juntos desse jeito. É um pouco fora do padrão, o que significa que os usuários não estarão acostumados com ele, e será menos provável que irão segui-lo.

**P:** E dai? O padrão é meu, eu posso fazer o que quiser; certo?

**R:** Claro, mas ao mesmo tempo você quer que os seus usuários tenham uma boa experiência. Caso contrário, eles não voltarão ao seu site.

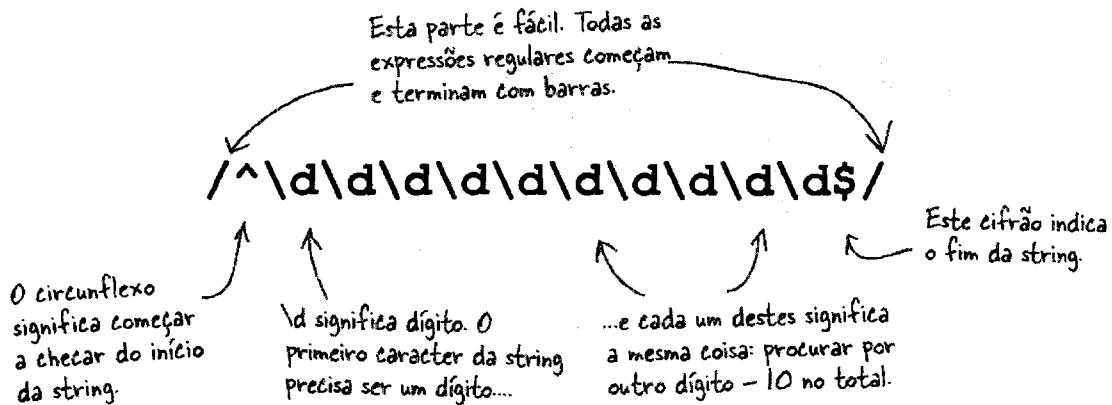
**P:** Ok, eu não poderia então usar três campos de texto para o número: um para o DDD, depois três dígitos no segundo, e finalmente os últimos quatro dígitos no terceiro. Aí então eu poderia usar as funções `string` do PHP.

**R:** Sim, você poderia, e alguns sites fazem isso. Mas usar os padrões lhe dá mais flexibilidade. E os padrões são úteis para muito mais coisas do que apenas verificar se o usuário digitou o número de telefone corretamente, como você verá mais adiante neste capítulo..

## Confira padrões com relação a expressões regulares

O PHP oferece uma maneira poderosa de criar e verificar padrões em um texto. Você pode criar regras que lhe permitem procurar por padrões em strings de texto. Essas regras são conhecidas como expressões regulares ou simplesmente regex. Uma expressão regular representa um padrão de caracteres a encontrar. Com a ajuda das expressões regulares, você pode descrever no seu código as regras que deseja impor as suas strings.

Como exemplo, eis uma expressão regular que procura por 10 dígitos seguidos. Este padrão só irá coincidir com a uma string que consista de um número com 10 dígitos. Se a string for maior ou menor do que isso, não irá checar. Se a string contiver qualquer coisa que não seja um número, não irá coincidir. Vamos aos detalhes,

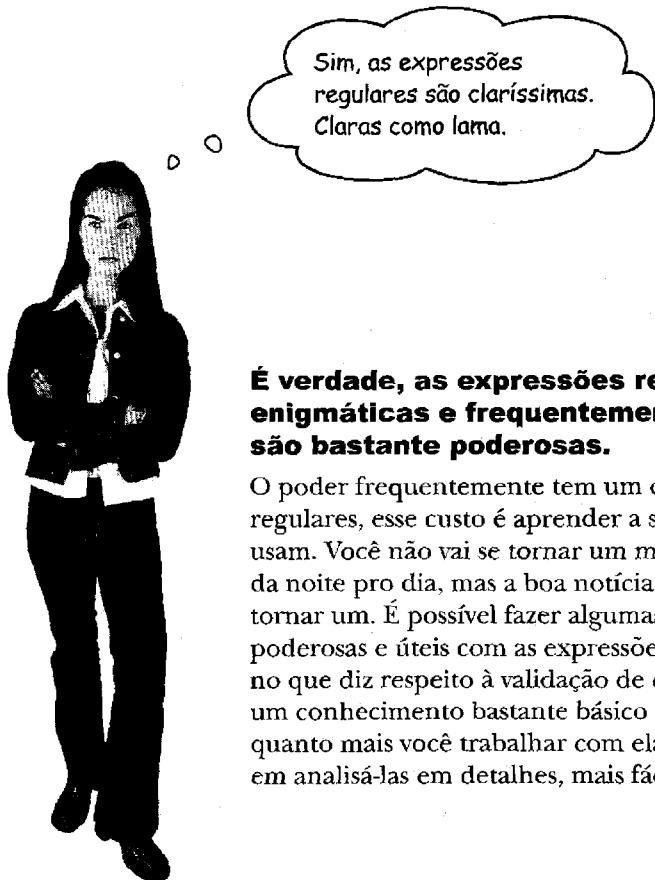


Existe também uma forma mais concisa de escrever esta mesma expressão regular, utilizando chaves. As chaves são usadas para indicar repetição:

/^\d{10}\\$/

Isto significa o mesmo que o padrão acima. {10} é uma forma abreviada de dizer "10 dígitos".

As expressões regulares são regras usadas para coincidir padrões em uma ou mais strings.



**É verdade, as expressões regulares são enigmáticas e frequentemente difíceis de ler... mas são bastante poderosas.**

O poder frequentemente tem um custo; no caso das expressões regulares, esse custo é aprender a sintaxe enigmáticas que elas usam. Você não vai se tornar um mestre das expressões regulares da noite pro dia, mas a boa notícia é que você não precisa se tornar um. É possível fazer algumas coisas impressionantemente poderosas e úteis com as expressões regulares, principalmente no que diz respeito à validação de campos de formulário, com um conhecimento bastante básico dessas expressões. Além disso, quanto mais você trabalhar com elas e quanto mais prática tiver em analisá-las em detalhes, mais fáceis de entender elas serão.

## Crie padrões usando metacaracteres

Poder checar dígitos em uma string de texto usando \d é bem legal, mas se isso fosse tudo o que as expressões regulares possibilitassem, o seu uso seria tremendamente limitado. Apenas checar dígitos não é suficiente nem para a validação de números telefônicos no Risky Jobs, uma vez que precisaremos checar caracteres como espaços, hifens e até mesmo letras.

Felizmente, a funcionalidade regex do PHP lhe permite usar expressões mais especiais, como \d, para checar essas coisas. Essas expressões são chamadas de metacaracteres. Vamos dar uma olhada em alguns dos metacaracteres regex usados mais frequentemente.

Os metacaracteres nos permitem descrever padrões de texto dentro de uma expressão regular.

### \d

Como você viu na página anterior, este metacaracter procura por um dígito. Qualquer número de 0 à 9 – em outras palavras, se você quisesse encontrar um número com dois dígitos, teria de usar \d\d ou \d{2}.

### \s

Procura por espaço em branco. Isso não significa apenas o caracter espaço que você obtém na tela ao digitar a barra de espaço; \s também encontra caracteres tab, ou um newline ou carriage return. Novamente, lembre-se que \s só encontra um desses caracteres por vez. Se você quiser encontrar exatamente dois caracteres espaços seguidos, precisa usar \\$\\$ ou \s{2}.

O metacaracter ponto final encontra qualquer outro caracter, exceto um newline. Encontra letras ou dígitos, assim como \w, bem como espaços ou tabs, como \s.

### \w

Procura por qualquer caracter alfanumérico – em outras palavras, ou uma letra ou um número. Encontra um dos seguintes caracteres: a – z e A – z (tanto maiúsculas quanto minúsculas), bem como 0-9 (assim como \d) ..

Nós também vimos o metacaracter circunflexo na página anterior. Ele procura pelo começo da string, de modo que você pode usá-lo para indicar que precisa haver uma combinação no início de uma string de texto, e não em qualquer lugar da string. Por exemplo, a regex /^d{3}/ encontrará a string "300 cadastros", mas não a string "Nós recebemos 300 cadastros".

### \$

Procura pelo final de uma string. Você pode usar este metacaracter com ^ para marcar o início e o fim da sua string, especificando exatamente onde ela começa e onde termina. Por exemplo /^w{5}\\$|d{3}\\$/ irá encontrar "Nanny 411", mas não "Nanny 411 é ótima" nem "Ligue para Nanny 411".

Esses metacaracteres são legais, mas se você quisesse ter um caracter específico na sua regex? Basta usar esse caracter na expressão. Por exemplo, se quisesse encontrar o número de telefone "707-827-7000" exato, você usaria a regex /707-827-7000/.

CINEMA FAZ O QUE?

Ligue cada expressão regular ao número telefônico a que ela corresponde.

Regex

String que ela encontra

`/^\d{3}\s\d{7}$/`**5556364652**`/^\d{3}\s\d{3}\s\d{4}$/`**555 636 4652**`/^\d{3}\d{3}-\d{4}$/`**555636-4652**`/^\d{3}-\d{3}-\d{4}$/`**555 ME NINJA**`/^\d{3}\s\w\w\w\s\w{5}$/`**555 6364652**`/^\d{10}$/`**555-636-4652**

## QUEM FAZ O QUE SOLUÇÃO

Ligue cada expressão regular ao número telefônico a que ela corresponde.

`/^\d{3}\s\d{7}$/` ————— **5556364652**

`/^\d{3}\s\d{3}\s\d{4}$/` ————— **555 636 4652**

`/^\d{3}\d{3}-\d{4}$/` ————— **555636-4652**

Este é o padrão que o Risky Jobs usa para encontrar números de telefones no formato **###-###-####**.

`/^\d{3}-\d{3}-\d{4}$/` ————— **555 ME NINJA**

`/^\d{3}\s\w\w\s\w{5}$/` ————— **555 6364652**

Os metacaracteres `\w` neste padrão encontram letras.

`/^\d{10}$/` ————— **555-636-4652**

Este padrão é composto totalmente por dígitos, de modo que ele só consegue encontrar um número telefônico que não tenha espaços nem hifens.



## SINTA-SE como uma Expressão Regular

Sua tarefa é interpretar o papel da expressão regular, e aceitar ou rejeitar números telefônicos dos usuários do Risky Jobs. Marque os números que você considerar válidos, e deixe os demais em branco. Escreva por que os números inválidos são inválidos.

Esta é a expressão regular para o número telefônico – seja ela!

`/^\d{3}-\d{3}-\d{4}$/`



(555) 935-2659



(555)672-0953



555-343-8263

Surf pode ser uma atividade bem arriscada, principalmente quando você trabalha de isca profissional para tubarões!



555-441-9005



555.903.6386



555-612-8527-8724

seja expressão regular solução

## SINTA-SE como uma Expressão Regular - Solução



Sua tarefa é interpretar o papel da expressão regular, e aceitar ou rejeitar números telefônicos dos usuários

do Risky Jobs. Marque os números que você considerar válidos e deixe os demais em branco. Escreva por que os números inválidos são inválidos.

Esta é a expressão regular para o número telefônico – seja ela!

`/^\d{3}-\d{3}-\d{4}$/`



Parenteses não são permitidos e espaços também não..

(555) 935-2659



(555)672-0953



555-343-8263

Sem parenteses, por favor.



As nossas expressões regulares requerem hífens e não pontos.

555-441-9005



555.903.6386



Ei, tem quatro números a mais aqui. Isso é uma extensão em um escritório?

555-612-8527-8724



Às vezes as pessoas colocam dígitos a mais nos seus números telefônicos, como por exemplo, uma extensão de quatro dígitos no final. Existe alguma maneira de encontrarmos esses padrões também?

**Sim, mas a chave é especificar esse tipo de padrão como opcional, na sua expressão regular.**

Se modificássemos a nossa regex para `^\d{3}\d{3}\d{4}-\d{4}$`, estariamos exigindo que a nossa string tivesse uma extensão de quatro dígitos no final, e não conseguiríamos mais encontrar números como "555-636-4652". Mas podemos usar expressões regulares para indicar que partes da string são opcionais. As regexes têm suporte a um recurso chamado quantificadores, que lhe permitem especificar quantas vezes os caracteres ou metacaracteres devem aparecer em um padrão. Na verdade, você já viu os quantificadores em ação, em regexes como esta:

`/^\d{10}$/`

Isto quer dizer "o dígito deve aparecer 10 vezes seguidas".

Aqui, as chaves agem como quantificador, para dizer quantas vezes o dígito precedente deve aparecer. Vamos dar uma olhada em alguns outros quantificadores frequentemente usados.

### {min, max}

Quando há dois números dentro das chaves, separados por vírgula, isso indica uma faixa de possíveis vezes que o caractere ou metacaractere deve ser repetido. Aqui, estamos dizendo que ele deve aparecer 2, 3 ou 4 vezes seguidas.

### +

O caractere ou metacaractere precedente deve aparecer uma ou mais vezes.

### \*

O caractere ou metacaractere pode aparecer uma ou mais vezes... ou não aparecer nenhuma vez.

### ?

O caractere ou metacaractere precedente deve aparecer uma vez ou então não aparecer nenhuma vez.

O quantificador especifica

Assim, se quiséssemos encontrar aqueles dígitos opcionais no final do número telefônico, poderíamos usar o seguinte padrão:

quantas vezes um metacaractere deve aparecer.

`/^\d{3}-\d{3}-\d{4}(-\d{4})?$/`

Coloque entre parênteses a seção à qual o quantificador se aplica.

O ponto de interrogação faz com que o hífen e os quatro últimos dígitos sejam opcionais.



Você se  
esqueceu de uma coisa. Os  
números telefônicos nos EUA não  
podem começar com 0 nem 1.

**Você tem toda razão. 0 é o número para chamar uma telefonista e 1 é o código para interurbanos nos EUA.**

Nós queremos simplesmente o código DDD e o número.  
Precisamos nos certificar que o primeiro dígito não seja 1 nem 0.  
E para isso, precisamos de uma classe de caracteres.

As classes de caracteres lhe permitem encontrar caracteres a partir de um conjunto específico de valores. Com uma classe de caracteres, você pode procurar por uma faixa de dígitos, e pode também procurar por um conjunto de valores. E você pode adicionar um circunflexo para procurar por tudo que não esteja no conjunto.

Para indicar que alguns caracteres ou metacaracteres pertencem a uma classe de caracteres, tudo o que você precisa fazer é colocá-los entre colchetes, [ ]. Vejamos alguns exemplos de classes de caracteres em ação:

## [0-2]

Isto encontrará uma faixa de números, especificamente 0, 1 ou 2.

## [A-D]

Isto encontrará A, B, C ou D.

Em uma classe de caracteres, o ^ significa "não corresponde".

## [^b-f]

Este circunflexo tem um significado diferente quando usado dentro de uma classe de caracteres. Em vez de dizer "as strings devem começar com...", o circunflexo significa "encontrar tudo exceto..."

Isto encontrará tudo exceto b, c, d, e ou f.

Uma classe de caracteres é um conjunto de regras para se encontrar um caractere.

Escreva aqui uma expressão regular que encontre números telefônicos internacionais:

.....  
.....  
.....

# Ajuste os padrões com classes de caracteres

Com a ajuda das classes de caracteres, podemos refinar a nossa expressão regular para números de telefone, de modo tal que ela não encontre combinações inválidas de dígitos. Dessa forma, se alguém accidentalmente digitar um DDD que comece com 0 ou 1, podemos exibir uma mensagem de erro. Eis como seria a nossa nova e melhorada regex:

`/^ [2-9] \d{2} - \d{3} - \d{4} $/`

A classe de caracteres diz que o nosso primeiro caractere deve ser qualquer dígito de 2 a 9, inclusive.

...e nós estamos procurando por mais dois dígitos que podem ser qualquer valor entre 0 e 9.

...seguidos de mais três dígitos...

...e um hífen e mais os 4 últimos dígitos.

`^` e `$` especificam que a nossa regex deve cobrir toda a string de texto que estamos comparando. Em outras palavras, a string não pode conter nenhum outro caractere que não pertença ao número telefônico.

não existem

## Perguntas Idiotas

**P:** Então as classes de caracteres permitem que você especifique uma faixa de caracteres para comparar com a string de texto?

**R:** Sim, a classe de caracteres lhe permite especificar, na sua expressão regular, que qualquer membro de um conjunto de caracteres especificado poderá ser usado para se encontrar a string de texto, em vez de apenas um caractere. Por exemplo, a classe `[aeiou]` encontrará uma instância de qualquer vogal minúscula, e a classe `[m-zM-Z]` encontrará uma instância de qualquer letra da segunda metade do alfabeto, seja minúscula ou maiúscula.

E a classe `[0-9]` é equivalente ao metacaracter `\d`, que na verdade é apenas uma forma abreviada de dizer a mesma coisa.

**P:** Não é preciso colocar espaços ou vírgulas entre os caracteres ou faixas que eu especificar nas classes de caracteres?

**R:** Não. Se você fizer isso, esses caracteres extras serão interpretados como parte do conjunto de caracteres a serem comparados com a string de texto. Por exemplo, a classe

`[m-z, M-Z]`

encontraria não apenas as letras minúsculas e maiúsculas de m até z, mas também uma vírgula ou um espaço, o que provavelmente não é o que você quer.

uma vez, com uma classe de caracteres? Por exemplo, uma ou mais vogais consecutivas?

**R:** Basta adicionar um quantificador depois da classe de caracteres. A expressão `/(aeiouAEIOU)+/` encontrará uma ou mais vogais seguidas.

**P:** Eu pensei que os quantificadores só se aplicassem ao caractere imediatamente antes deles.

**R:** Geralmente é assim, mas se o quantificador vier logo depois de uma classe de caracteres, ele se aplica à classe inteira. E se você quiser fazer com que o quantificador se aplique a toda uma série de caracteres que não estejam em uma classe, poderá colocar esses caracteres entre parênteses, para indicar que eles devem ficar agrupados. Como exemplo, a expressão regular `/(hello)+/` encontrará uma ou mais ocorrências consecutivas da palavra "hello" em uma string de texto.

**P:** E se eu quisesse encontrar uma palavra escrita de duas formas diferentes, como "ketchup" ou "catsup"?

**R:** Você pode usar a barra vertical (`|`) nas suas expressões regulares para indicar um conjunto de opções. Assim, a expressão regular `/(ketchup|catsup|catchup)/` encontrará qualquer uma das três grafias mais comuns para essa palavra.

**P:** E se eu quisesse encontrar um caractere mais de



o o

E quanto a colocar caracteres como pontos finais ou de interrogação em uma expressão regular?  
Se eu colocar essas coisas, o PHP não vai achar que elas são metacaracteres ou quantificadores, e com isso, atrapalhar o processamento da minha regex?

**Se quiser usar caracteres reservados na sua expressão regular, você precisa fazer escape deles.**

Na sintaxe das expressões regulares, há um pequeno conjunto de caracteres que recebem significados especiais, porque são usados para indicar coisas como metacaracteres, quantificadores e classes de caracteres. Entre eles estão o ponto final (.), o ponto de interrogação (?), o sinal de mais (+), o colchete de abertura ([], os parênteses, o acento circunflexo (^), o cifrão (\$), a barra vertical (|), a barra invertida (\), a barra normal (/) e o asterisco (\*).

Se quiser usar esses caracteres na sua expressão regular, com o significado deles, em vez dos metacaracteres ou quantificadores que eles normalmente representam, você precisa "escapá-los", colocando uma barra invertida antes deles.

Por exemplo, se quiser encontrar parênteses em um número telefônico, você não poderia fazer só isto:

**(555)636-4652**



Estes serão simplesmente  
tratados como um grupo.  
/^(\\d{3})\\d{3}-\\d{4}\$/

Em vez disso, os parênteses, tanto o de abertura quanto o de fechamento, têm de ser precedidos por barras invertidas para indicar que eles devem ser interpretados realmente como parênteses:

**(555)636-4652**



Agora o PHP sabe que estes são  
parênteses literais.  
/^\\(\\d{3}\\)\\d{3}-\\d{4}\$/



Crie uma string que bata com cada padrão mostrado abaixo.

`/^ [3-6] {4} /      /^ ([A-Z] \d) {2} $/`

Suponha que queiramos aprimorar o esquema de validação do Risky Jobs para números telefônicos, com o objetivo de permitir que os usuários submetam os seus números em mais alguns formatos. Escreva uma única expressão regular que corresponda a TODAS as strings de texto abaixo, e que não permita um 0 ou 1 como o primeiro dígito. O seu padrão só deve permitir dígitos, parênteses, espaços e hifens.

**555-636-4652    555 636-4652**

**(555)-636-4652    (555) 636-4652**

## exercício solução



### Exercício Solução

Crie uma string que bata com cada padrão mostrado abaixo.

A string deve começar com...

3 a 6.... e repetir essa classe de caracteres quatro vezes.

/^ [3-6] {4} /

A string deve começar uma letra maiúscula... e um dígi... duas vezes... e então acaba...

/^ ([A-Z] \d) {2} \$/

Qualquer string que comece com quatro dígitos na faixa de 3 a 6 irá corresponder a esta expressão.

Todas estas strings corresponderão:

"5533", "3456 é um número.", "b533xyz"

Qualquer string que comece com uma letra maiúscula e depois um dígi..., seguido de outra letra maiúscula, um dígi... e só:

"B5C9", "R2D2"

Suponha que queremos aprimorar o esquema de validação do Risky Jobs para números telefônicos, com o objetivo de permitir que os usuários submetam os seus números em mais alguns formatos.

Escreva uma única expressão regular que corresponda a TODAS as strings de texto abaixo, e que não permita um 0 ou 1 como o primeiro dígi... O seu padrão só deve permitir dígitos, parênteses, espaços e hifens.

555-636-4652    555 636-4652

(555)-636-4652    (555) 636-4652

A string deve começar com...

2 a 9....

duas vezes...

um hifen ou um espaço...

outro hifen

e um dígi..., quatro vezes...

/^ (([2-9]\d{2})[-\s]?([-\s]\d{3})-\d{4})/

um parêntese de abertura opcional, que pode aparecer 0 ou 1 vez...

e um dígi...

um parêntese de fechamento opcional...

e um dígi..., três vezes...

e então acaba.

Já me enchi dos dados inválidos do banco de currículos do Risky Jobs.  
Qual a utilidade dessas expressões se não as estamos utilizando?

**O Risky Jobs precisa colocar as expressões regulares para trabalhar, validando dados do formulário!**



a função `the php preg_match()`

## Verifique os padrões com `preg_match()`

Nós não vimos desenvolvendo padrões só porque é divertido. Você pode usar esses padrões com a função `preg_match()` do PHP. Esta função toma um padrão regex, igual àqueles que vimos construindo, e uma string de texto. Retorna false se não houver uma correspondência, e true se houver.

**`preg_match($regex, $my_string)`**

Coloque a sua regex aqui. A função  
espera receber uma string, o que  
significa que a regex deve ficar entre  
aspas simples.

A string que você está  
comparando fica aqui..

Eis um exemplo da função `preg_match()` em ação, usando uma regex que procura, em uma string de texto, por um padrão de quatro caracteres de letras maiúsculas e dígitos alternando-se:

Quando regexes são passadas para `preg_match()`, elas devem ser colocadas entre aspas..

`preg_match('/^\\d{3}-\\d{2}-\\d{4}$/', '555-02-9983')`

Retorna um número inteiro:  
1 caso a string bata com o  
padrão e 0 caso contrário.

Você pode colocar o padrão  
dentro da função desta forma,  
mas em geral, é melhor armazená-lo  
em uma variável.

Esta string bate com a regex,  
e a função retornará 1.

A função `preg_match()`  
fica aninhada dentro  
da condição, portanto  
o seu resultado  
determina qual código  
será executado.

Podemos tirar proveito da função `preg_match()` para habilitar uma funcionalidade de validação mais sofisticada nos scripts PHP, colocando uma declaração if em torno do valor de retorno.

```
if (preg_match('/^\\d{3}-\\d{2}-\\d{4}$/', '555-02-9983')) {
```

echo 'Número de seguro social válido';

```
} else {
```

echo 'Esse número de seguro social não é válido!';

} Se não ocorrer correspondência,  
`preg_match()` retorna false, o que  
faz a condição avaliar como false.  
Portanto, este código é executado.

Se houver correspondência, `preg_match()`  
retorna true, o que indica para o PHP  
que a condição é verdadeira. Portanto,  
este código é executado.



Reescreva a parte salientada do script PHP do Risky Jobs que verifica os dados de registro abaixo para validar o texto digitado no campo phone, usando preg\_match() em vez de empty(). Use a regex que você criou anteriormente na função preg\_match().

```
if (empty($phone)) {  
    // $phone está em branco  
    echo '<p class="error">O seu número de telefone não é  
válido.</p>';  
  
    $output_form = 'yes';  
}
```

.....  
.....  
.....  
.....



## Exercício Solução

Reescreva a parte salientada do script PHP do Risky Jobs que verifica os dados de registro abaixo para validar o texto digitado no campo phone, usando preg\_match() em vez de empty(). Use a regex que você criou anteriormente na função preg\_match().

```
if (empty($phone)) {
    // $phone está em branco
    echo '<p class="error">O seu número de telefone não é
válido.</p>';
    $output_form = 'yes';
}
```

Em vez de empty(), nós usamos uma preg\_match para validar o número de telefone. Nós colocamos antes dela o operador NOT (!), porque queremos exibir um erro sempre que os dados digitados NÃO baterem com o padrão.

A expressão regular que já vimos anteriormente.

```
if (!preg_match('/^(?E2-9]\d{2})\)?[-\s]\d{3}-\d{4}$/, $phone)) {
    // $phone não é válido
    echo '<p class="error">Seu número de telefone é inválido</p>';
    $output_form = 'yes';
}
```

O echo precisa ser modificado um pouco, uma vez que não estamos apenas verificando se dados foram digitados, mas se eles batem com um padrão que estabelecemos para os números telefônicos.

Colocamos \$output como 'yes'(sim), igual antes

Recebi um erro e depois digitei meu número telefônico inteiro. E então consegui um trabalho como ninja!

First Name: Jimmy
Last Name: Swift
Email: JS@sim-u-duck.com
Phone: (555) 636 4652
Desired Job: Ninja





## TEST DRIVE

E não apenas se o usuário digitou o número!

### Verifique se os números telefônicos são válidos no script de registro do Risky Jobs.

Baixe o script registration.php no site da Alta Books, em [www.altabooks.com.br](http://www.altabooks.com.br), junto com a folha de estilo do Risky Jobs (style.css) e as imagens (riskyjobs\_title.gif e riskyjobs\_fireman.png). Em seguida, modifique o script registration.php para que ele use a função preg\_match() para validar os números telefônicos em relação à expressão regular. Certifique-se de ajustar a mensagem de erro para que os usuários saibam que o número telefônico não é válido, e não apenas que está em branco.

Envie o script modificado para o seu servidor web e então, abra-o em um navegador. Experimente digitar alguns números telefônicos com diversos formatos, e observe como o script captura os erros.

Agora nós não podemos digitar accidentalmente números telefônicos inválidos. Com isso, não perderemos mais oportunidades de emprego!

**Risky Jobs**  
Danger! Your dream job is out there.  
Do you have the guts to go find it?

**Risky Jobs - Registration**

Your phone number is invalid.

Register with Risky Jobs, and post your resume.

First Name:	Swashy
Last Name:	McLane
Email:	swashymlane@starbuzzcoffee.com
Phone:	555-903-6386 <span style="color: red;">X</span>
Desired Job:	Penner

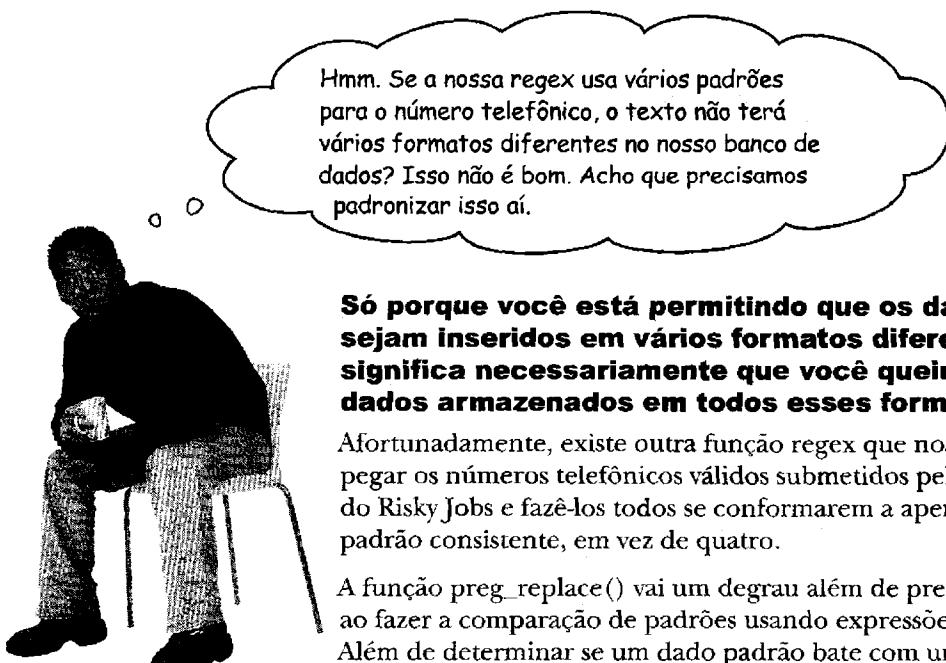
Paste your resume here:  
I'm quite the swashbuckler, been doing it for years. I started out young with broomsticks but quickly moved on to real swords. Now I'll fence anything. I am based certified from the Royal Fencing Institute.

O script agora exibe uma mensagem de erro quando o número de telefone é digitado incorretamente, neste caso, com pontos em vez de hifens.

**Faça o Download!**

O código-fonte completo da aplicação Risky Jobs está disponível para download no site da Alta Books:

[www.altabooks.com.br](http://www.altabooks.com.br)



**Só porque você está permitindo que os dados sejam inseridos em vários formatos diferentes, não significa necessariamente que você queira esses dados armazenados em todos esses formatos.**

Afortunadamente, existe outra função regex que nos permitirá pegar os números telefônicos válidos submetidos pelos usuários do Risky Jobs e fazê-los todos se conformarem a apenas um padrão consistente, em vez de quatro.

A função `preg_replace()` vai um degrau além de `preg_match()`, ao fazer a comparação de padrões usando expressões regulares. Além de determinar se um dado padrão bate com uma dada string de texto, ela lhe permite especificar um padrão para ser colocado na string, no lugar do texto encontrado. Funciona de modo bastante semelhante com a função `str_replace()` que já usamos, exceto pelo fato de que ela faz a comparação usando uma expressão regular, e não uma string.

### `preg_replace($padrão, $substituição, $minha_string)`

Nós precisamos encontrar  
esses caracteres  
indesejados.

Quando encontramos um  
caráter indesejado, queremos  
que ele seja transformado nisto.

A string onde estamos  
fazendo a operação de  
localizar-e-substituir.

Eis um exemplo da função `preg_replace()` em ação:

```
$ano_novo = preg_replace('/200[0-9]/', '2010', 'Estamos em 2009.');
```

O resultado da função  
`Preg_replace()`, que  
é a nossa string de  
texto revisada depois  
de feita a operação de  
localizar-e-substituir,  
fica armazenado em  
`ano_novo`.

Esta regex diz a `preg_replace` para procurar  
uma correspondência que  
seja um número entre  
2000 e 2009.

Quando uma  
correspondência é  
encontrada, ela é  
substituída por 2010.

Cada vez que um ano  
de 2000-2009  
for encontrado na  
nossa string, ele será  
substituído por 2010.



## Exercício

Padronize os dados de números telefônicos digitados no formulário do Risky Jobs, escrevendo cada um dos seguintes números na coluna phone da tabela abaixo. Certifique-se de usar um formato que armazene o mínimo de dados possível para representar cada número de telefone.

**(555) 935-2659**

**(555)672-0953**

**555-343-8263**

**555-441-9005**

**555.903.6386**

**555-612-8527-8724**

Risky Jobs

Danger! Your dream job is out there.  
Do you have the guts to go find it?

Risky Jobs - Registration

Register with Risky Jobs, and post your resume.

First Name: \_\_\_\_\_

Last Name: \_\_\_\_\_

Email: \_\_\_\_\_

Phone: \_\_\_\_\_

Desired Job: \_\_\_\_\_

Paste your resume here:

phone

**exercício solução**



**Exercício  
Solução**

Padronize os dados de números telefônicos digitados no formulário do Risky Jobs, escrevendo cada um dos seguintes números na coluna phone da tabela abaixo. Certifique-se de usar um formato que armazene o mínimo de dados possível para representar cada número de telefone.

**(555) 935-2659**

**(555)672-0953**

**555-343-8263**

**555-441-9005**

**555.903.6386**

**555-612-8527-8724**

A forma mais fácil e rápida de armazenar um número telefônico é retirar tudo, exceto os dígitos.

**Risky Jobs**  
Danger! Your dream job is out there.  
Do you have the guts to go find it?

Risky Jobs - Registration

Register with Risky Jobs, and post your resume.

First Name: \_\_\_\_\_  
Last Name: \_\_\_\_\_  
Email: \_\_\_\_\_  
Phone: \_\_\_\_\_  
Desired Job: \_\_\_\_\_

Paste your resume here:

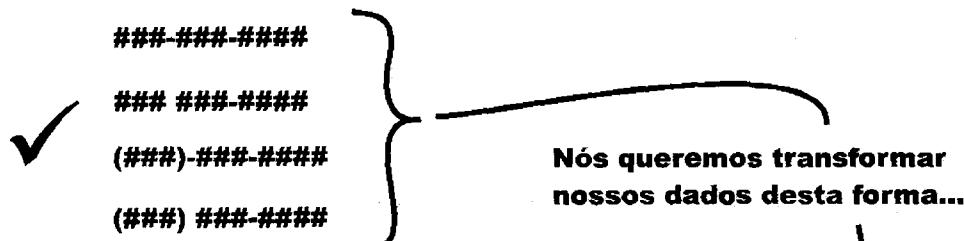
phone
5559352659
5556720953
5553438263
5554419005
5559036386
5556128527

## Padronize os dados dos números de telefone

Da forma como está agora, o Risky Jobs está usando a seguinte expressão regular para validar os números telefônicos fornecidos pelos usuários no formulário de registro:

```
/^\(?[2-9]\d{2}\)?[-\s]\d{3}-\d{4}$/
```

Isto encontrará números que recaiam em um destes quatro padrões:



Embora esses formatos sejam facilmente interpretados pelas pessoas, eles dificultam o trabalho das consultas SQL, de classificar os resultados da forma como queremos. Os parênteses muito provavelmente irão atrapalhar as nossas tentativas de agrupar os números por DDD, por exemplo, o que poderia ser importante para o Risky Jobs caso nós queiramos analisar quantos dos usuários do site são de uma determinada localização geográfica.

Para tornar esses tipos de consultas possíveis, precisamos padronizar os números em um só formato, usando preg\_replace() antes de usarmos INSERT para colocar os dados no banco. A nossa melhor opção é nos livrarmos de todos os caracteres, exceto os dígitos. Dessa forma, nós simplesmente armazenamos 10 dígitos na nossa tabela, sem nenhum outro caractere. Queremos que os nossos números sejam armazenados desta forma, na tabela:

#####

**...para esta.**

Isto nos deixa com quatro caracteres para localizar e substituir. Nós queremos achar e remover parênteses, espaços e hífens. E queremos encontrar esses caracteres independentemente de onde eles estejam na string, portanto, não precisamos do circunflexo (^) e nem do cifrão (\$). Nós sabemos que estamos procurando por qualquer caractere de um tipo específico, logo, podemos usar uma classe de caracteres. A ordem da busca não importa. Eis a regex que podemos usar:

/[\(\)\)-\s]/

abre parênteses

fecha parênteses

hifen

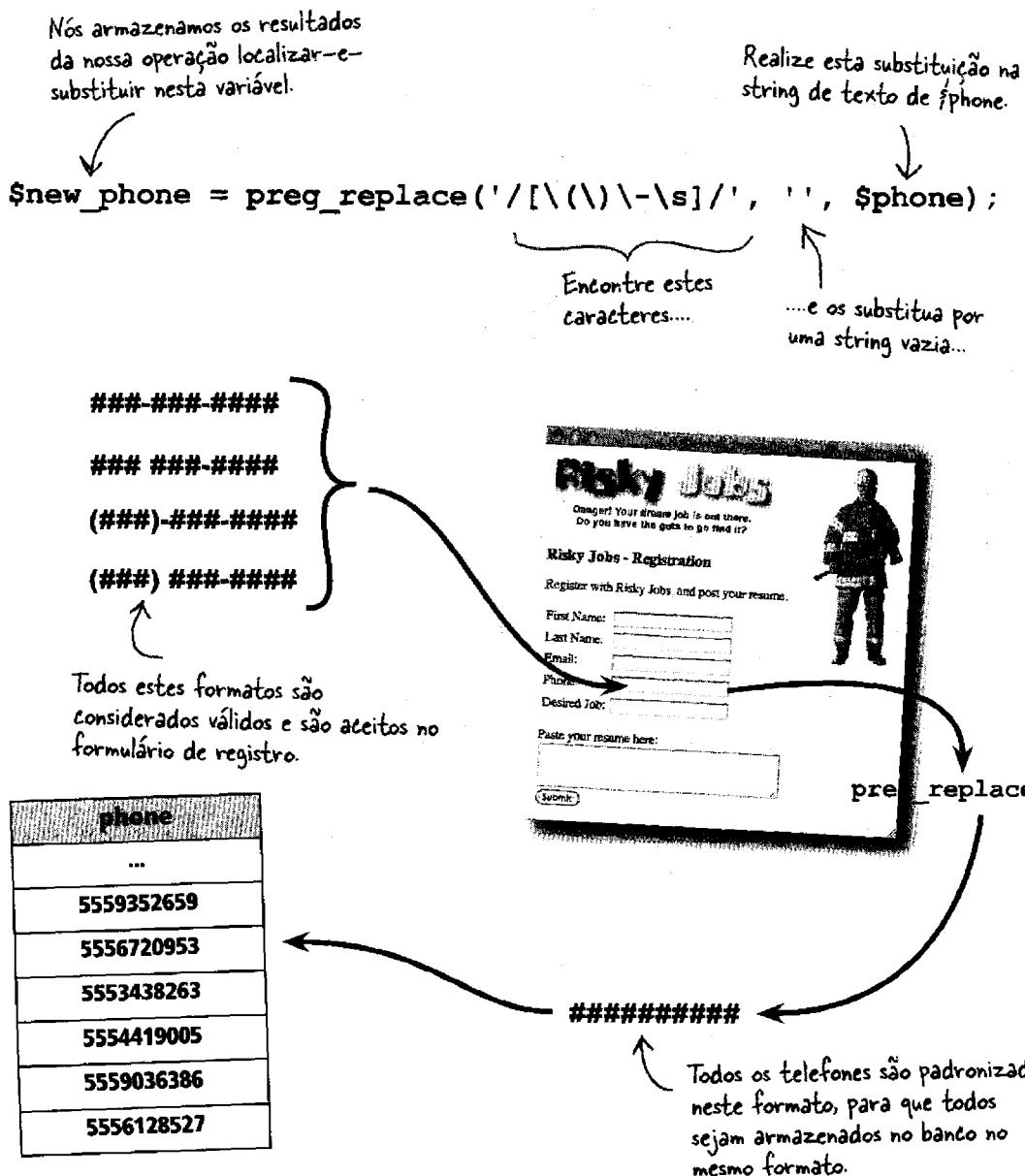
espaço

**Padronizar os seus dados lhe dá melhores resultados em consultas SQL.**

*(removendo caracteres com preg\_replace())*

## Livre-se dos caracteres indesejados

Agora que temos o nosso padrão para encontrar esses caracteres indesejados, podemos aplicá-lo aos números telefônicos para ajustá-los antes de os armazenarmos no banco. Mas como? E aí que a função preg\_replace() realmente vem em nosso auxílio. A questão aqui é que não queremos substituir os caracteres indesejados, queremos apenas nos livrar deles. Assim, nós simplesmente enviamos uma string vazia para preg\_replace() como o valor de substituição. Eis um exemplo que encontra caracteres indesejados nos números telefônicos e os substitui por strings vazias, na prática eliminando-os:





Não sei não, isso parece muito trabalho apenas para termos strings de 10 dígitos no nosso banco de dados. Não poderíamos simplesmente exigir que os usuários digitassem logo desse jeito, no formulário de registro?

**Certamente, mas isso acabaria causando problemas depois, uma vez que as consultas com números telefônicos não funcionariam como esperado.**

A maioria dos usuários está acostumada a digitar números telefônicos com alguma combinação de hífens, parênteses e espaços, de modo que tentar forçar o uso de apenas dígitos nos telefones poderia não funcionar como esperado. É muito melhor tentar alcançar algum equilíbrio, dando aos usuários opções razoavelmente flexíveis para a inserção dos dados, ao mesmo tempo nos certificando de que os dados armazenados sejam o mais consistentes possível.

Além disso, só é preciso uma chamada a preg\_replace() para resolver o problema, o que não é grande coisa. Mas se estivéssemos falando em escrever algum tipo de função personalizada com um monte de código, seria outra história. Mas melhorar a usabilidade e a integridade dos dados com uma única linha de código não tem nenhuma dificuldade!



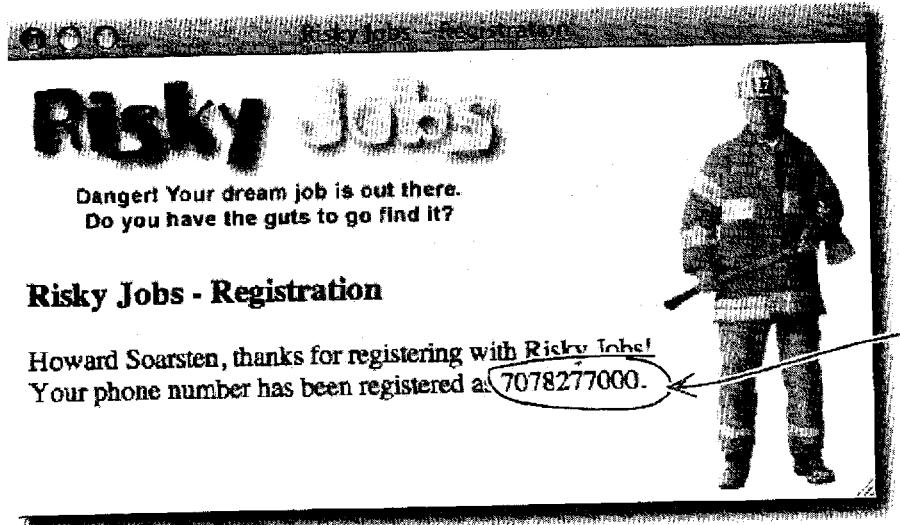
## TEST DRIVE

### Ajuste os números de telefone no script de registro.

Modifique o script registration.php para ajustar os números de telefone, adicionando as seguintes linhas de código ao script, logo depois da linha que agradece ao usuário por se registrar com o Risky Jobs:

```
$pattern = '/[\(\)\)-\s]/';
$replacement = '';
$new_phone = preg_replace($pattern, $replacement, $phone);
echo 'Seu número telefônico foi registrado como ' . $new_phone . '.</p>';
```

Envie o script para o seu servidor web e depois abra-o em um navegador. Preencha o formulário, sem se esquecer de colocar um número telefônico com caracteres extra, como por exemplo, (707) 827-7000. Submeta o formulário e verifique os resultados.



Experimente colocar algumas outras variantes do número, como: 707-827-7000, (707)-827-7000, 707 827 7000. Repare em como a expressão regular e preg\_replace() se livram dos caracteres extras.



**Da mesma forma que os números telefônicos, os endereços de email têm um formato específico o suficiente para que possamos validá-los, verificando mais do que simplesmente se os usuários os digitaram.**

Assim como na validação dos números telefônicos, primeiro nós temos de determinar as regras que os endereços de email válidos têm de seguir. Então poderemos formalizá-las como uma expressão regular, e implementá-las no nosso script PHP. Assim, vamos primeiro dar uma olhada do que exatamente se compõe um endereço email.

Nós sabemos que os endereços de email devem conter estes dois caracteres:

**NomeLocal@PrefixoDoDominio.SufixoDoDominio**

Aqui poderá haver caracteres alfanuméricos, onde NomeLocal tem no mínimo um carácter e PrefixoDoDominio tem no mínimo dois caracteres.

Isto geralmente se compõe de 3 caracteres alfanuméricos.

## PODER DO CÉREBRO

Veja se você consegue bolar uma expressão regular flexível o suficiente para encontrar os endereços de email à direita. Escreva-a abaixo:

aviator.howard@bannerocity.com

cube\_lovers@youcube.ca

rocky@i-rock.biz

uma regex para endereços de email

## Conferir endereços de email pode ser complicado

Parece que deveria ser bem simples encontrar endereços de email porque, à primeira vista, não parece haver tantas restrições sobre os caracteres que você pode usar quanto há para os números telefônicos.

Por exemplo, não parece ser grande dificuldade encontrar a parte NomeLocal de um endereço de email (tudo que vem antes do sinal @). Uma vez que ela é composta de caracteres alfanuméricos, deveremos poder usar o seguinte padrão:

/^\w+/  
Começa com...  
...um ou mais caracteres  
alfanuméricos.

Isso permitiria qualquer caractere alfanumérico no nome local, mas, infelizmente, ele não incluiria caracteres que também são permitidos em endereços de email.

Acredite se quiser, mas endereços de email válidos podem conter qualquer um destes caracteres na parte NomeLocal, embora alguns deles não possam ser usados como o primeiro caractere:

Todos estes caracteres  
podem aparecer na  
parte NomeLocal de  
um endereço de email.

! \$ & \* - = ^ ` | ~ # % ' + / ? \_ { }

Se quisermos permitir que os usuários registrem endereços de email com esses caracteres, realmente precisamos de uma regex parecida com essa:

/^ [a-zA-Z0-9] [a-zA-Z0-9\.\_\-\&!?=#]\* /

O primeiro caractere  
tem de ser um destes.

Os demais caracteres  
podem ser quaisquer  
destes...

...e podemos ter zero ou  
mais deles.

!\$&\*-=^`|~#%'+/?\_{}  
()

Isso não encontrará absolutamente todo NomeLocal válido, uma vez que ainda estamos pulando alguns dos caracteres realmente obscuros, mas é bastante prático de usar e, de qualquer forma, ainda deverá encontrar os endereços de email da maioria dos usuários do Risky Jobs.





Esse negócio dos emails é fácil.  
Basta usarmos o mesmo padrão que  
usamos para o nome local para validar o  
nome do domínio... nada demais!

### **Isso funcionaria para uma parte do domínio, o prefixo, mas não para o sufixo.**

Enquanto o prefixo do domínio pode conter praticamente qualquer combinação de alfanuméricos e alguns caracteres especiais, como o NomeLocal, as restrições sobre os sufixos dos domínios são maiores.

A maioria dos endereços de email terminam em algum sufixo de domínio comum: .com, .edu, .org, .gov e assim por diante. Nós precisamos nos certificar de que os endereços de email terminem em um sufixo de domínio válido também.

não existem

## **Perguntas Idiotas**

**P:** E se eu quiser permitir absolutamente todos os endereços de email possíveis?

**R:** Você pode, e se for o ideal para o seu site, certamente deve. Mas às vezes é melhor usar formatos comumente aceitos, e não necessariamente aceitar qualquer variação possível. Você precisa decidir qual será o formato dos emails de 99.9% dos seus usuários, e precisa estar disposto a não validar o .1% restante simplesmente porque isso lhe permitirá escrever um código melhor. A validação é, na realidade, um equilíbrio entre o que é permitido e o que é prático aceitar. Se quiser implementar uma validação de emails mais robusta no seu site, você poderá encontrar alguns excelentes códigos PHP open source (ou seja, gratuitos) aqui: <http://code.google.com/p/php-email-address-validation/>.

**P:** Mas as pessoas não vão ficar irritadas se elas tiverem um endereço de email que eu me recuse a validar?

**R:** É possível, mas a maioria das pessoas não terá endereços de email malucos. A maioria dos serviços de email online tem as suas próprias restrições que impedem os usuários de criarem endereços de email loucos, embora válidos, como:  
"\_eu sou louco"@gregs-list.net.

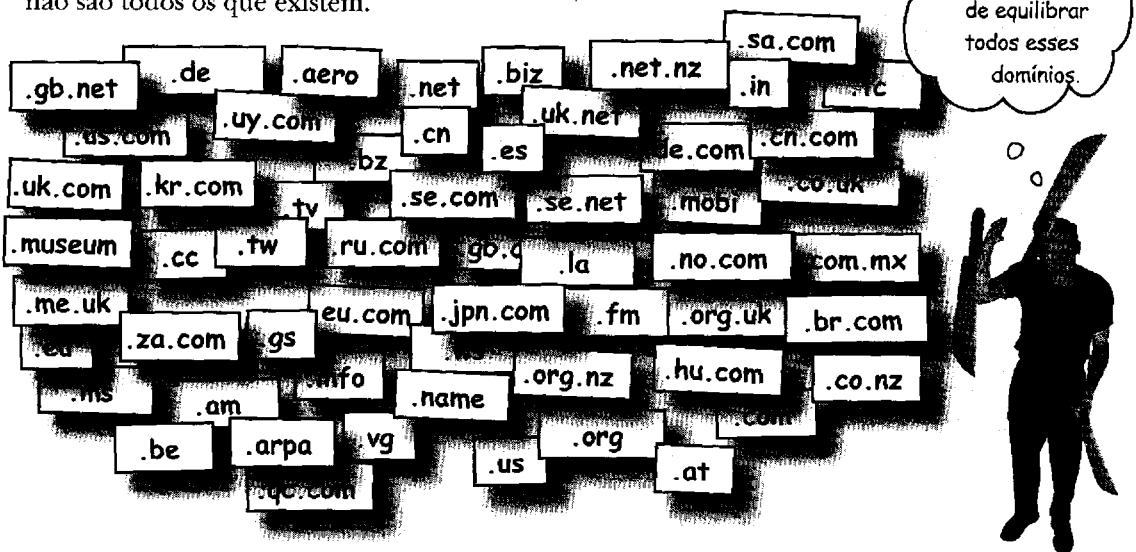
**A validação frequentemente  
é um equilíbrio entre o  
que é permitido e o que é  
prático aceitar.**

*coincidir sufixos de domínio*

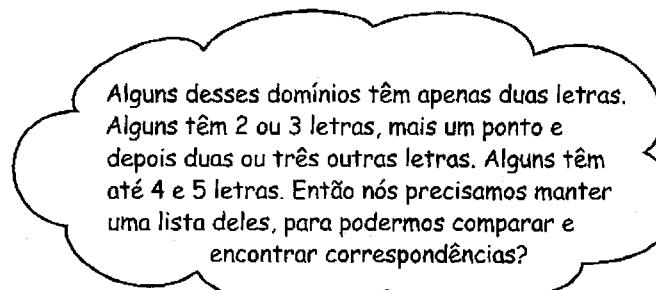
## Sufixos de domínios estão por toda a parte

Além dos ultracomuns sufixos de domínios que você vê frequentemente, como .com e .org, existem muitos, muitos outros sufixos válidos para uso em endereços de email. Outros sufixos reconhecidos como válidos pelo Sistema de Nomes de Domínios (Domain Name System, ou DNS) que você pode já ter visto incluem .biz e .info. Além desses há uma série de sufixos que correspondem aos diferentes países, como .ca para o Canadá e .tj para o Tadjiquistão.

Eis uma lista com apenas alguns dos sufixos de domínios existentes. Estes não são todos os que existem.



Nem eu gostaria  
de ter  
de equilibrar  
todos esses  
domínios.



**Nota Geek**

O Sistema de Nomes de Domínios é um serviço de dados distribuído que fornece um diretório mundial de domínios e os seus endereços IP. Ele torna possível o uso de nomes de domínio. Sem o DNS, teríamos de digitar 208.201.239.36 em vez de oreilly.com.

### Poderíamos fazer isso, e funcionaria.

Mas há um jeito mais fácil. Em vez de manter registro de todos os domínios possíveis e ter de modificar nosso código caso um novo domínio seja adicionado, podemos verificar a parte do domínio de um endereço de email usando a função `checkdnsrr()` do PHP. Esta função se conecta ao DNS e verifica a validade dos domínios.



## Use PHP para verificar o domínio

O PHP fornece a função `checkdnsrr()` para verificarmos se um domínio é válido ou não. Esse método é ainda melhor do que usar expressões regulares para encontrar o padrão de um endereço email, porque em vez de apenas verificar se uma string de texto poderia ser um domínio válido, essa função na verdade consulta os registros do DNS e descobre se o domínio realmente está registrado. Assim, por exemplo, enquanto que uma expressão regular poderia lhe dizer que `gzsdlkjdfalkjaf.com` é válido, `checkdnsrr()` pode dar um passo adiante e lhe dizer que, na verdade, esse domínio não está registrado, e que nós provavelmente deveríamos rejeitá-lo caso ele seja digitado no nosso formulário de registro.

A sintaxe de `checkdnsrr()` é bem simples:

Retorna 1 caso o domínio seja real, e 0 se não for.

`checkdnsrr()` espera receber uma string contendo um nome de domínio. Esse nome é tudo o que vem depois do sinal @.

`checkdnsrr('headfirstlabs.com')`

Isto só é um problema se o seu servidor web usar Windows. Caso esteja usando um computador Windows para criar seu site, mas o hospedando em um servidor UNIX/Linux, então não será problema.



Watch it!

**Se você estiver rodando o PHP em um servidor Windows, esse comando não funcionará para você.**

Em vez dele, você pode usar este código:

```
function win_checkdnsrr($domain, $recType='') {
    if (!empty($domain)) {
        if ($recType=='') $recType="MX";
        exec("nslookup -type=$recType $domain", $output);
        foreach($output as $line) {
            if (preg_match("/^$domain/", $line)) {
                return true;
            }
        }
        return false;
    }
    return false;
}
```

Este comando exec chama um programa externo, que roda no servidor, para verificar o domínio.

Só por diversão, tente executar `$line` logo depois do loop `foreach`. Você verá algo como isto:

```
Server:68.87.64.146#Address:  
68.87.64.146#53Non-authoritative answer:  
oreilly.com mail exchanger =  
20 smtp.  
oreilly.com.
```

## Validação de email: montando o quadro

Agora nós sabemos como validar tanto a parte NomeLocal de um endereço de email, usando expressões regulares, quanto a parte de domínio do endereço, usando checkdnsrr(). Vejamos o passo a passo de como podemos juntar essas duas partes para adicionar uma validação completa de endereços de email ao formulário de registro do Risky Jobs:

- 1 Use preg\_match() para determinar se a parte NomeLocal do nosso endereço de email contém um padrão de caracteres válido.

Podemos usar a seguinte regex para fazer isso:

```
/^ [a-zA-Z0-9] [a-zA-Z0-9\._\-\&!\?=#]* @ /
```

O email deve começar com um caractere alfanumérico, e depois disso pode conter qualquer número de alfanuméricos, bem como alguns caracteres especiais.

Repare que não há nenhum cifrão no final desta regex, uma vez que haverá caracteres após o @.

Desta vez, também iremos procurar por um símbolo arroba (@), para nos certificarmos de que o endereço de email contém esse símbolo antes do domínio.

- 2 Se a validação do *NomeLocal* falhar, ecoe uma mensagem de erro para o usuário e recarregue o formulário.

Se a validação do *NomeLocal* tiver sucesso, envie a checkdnsrr() parte da string de texto submetida pelo usuário referente ao domínio.

- 4 Se checkdnsrr() retornar 0, então o domínio não está registrado, portanto, nós ecoamos uma mensagem de erro para o usuário e recarregamos o formulário.

- 5 Se checkdnsrr() retornar 1, então o domínio está registrado, e podemos ter razoável confiança de que teremos um endereço de email válido. Podemos prosseguir com a validação do restante dos campos do formulário.



Segue abaixo o novo código PHP para validar os endereços de email dos usuários, mas alguns pedaços desapareceram. Preencha os espaços em branco para completar o código.

```
if (!preg_match(' ..... ', $email)) {  
    // $email é inválido porque NomeLocal é incorreto  
    echo 'Seu endereço de email não é válido.<br />';  
    $output_form = 'yes';  
}  
  
else {  
    // Retira tudo do email, exceto o domínio  
    $domain = preg_replace('.....', ' ', '.....');  
    // Agora verifica se $domain está registrado  
    if (.....) {  
        echo 'Seu endereço de email é inválido. <br />';  
        $output_form = 'yes';  
    }  
}
```

## exercício solução



### Exercício Solução

Segue abaixo o novo código PHP para validar os endereços de email dos usuários, mas alguns pedaços desapareceram. Preencha os espaços em branco para completar o código.

Esta é a nossa regex para encontrar a parte NomeLocal do endereço de email, terminando com um sinal @.

```
if (!preg_match(' [a-zA-Z0-9][a-zA-Z0-9]_\&?=#]*@/ ', $email)) {  
    // $email é inválido porque NomeLocal é incorreto  
    echo 'Seu endereço de email não é válido.<br />';  
    $output_form = 'yes';  
}  
  
else {  
    // Retira tudo do email, exceto o domínio.  
    $domain = preg_replace(' ^[a-zA-Z0-9][a-zA-Z0-9]_\&?=#]*@/ ', "", $email);  
  
    // Agora verifica se $domain está registrado  
    if (!checkdnsrr($domain)) {  
        echo 'Seu endereço de email é inválido. <br />';  
        $output_form = 'yes';  
    }  
    // checkdnsrr() retorna true se o domínio não estiver registrado.  
}
```

Para retirar o NomeLocal e o símbolo @, especifique a string vazia ("") como string de substituição.

Realize a substituição no valor de \$email.

Se estiver usando um servidor Windows, não se esqueça de incluir o código para win\_checkdnsrr() e então chamar-lo aqui.

## PONTOS DE BALA

- **preg\_match()** localiza correspondências para padrões em strings.
- **preg\_replace()** modifica strings encontradas..
- **Quantificadores** (Quantifiers) lhe permitem controlar quantas vezes um caractere ou conjunto de caracteres pode aparecer sequidamente.
- Você pode especificar um conjunto de caracteres a serem permitidos no seu padrão, usando uma **classe de caracteres**.
- No seu padrão, \d, \w e \s significam dígitos, caracteres alfanuméricos e espaço em branco, respectivamente.
- **checkdnsrr()** verifica a validade de nomes de domínio.



## TEST DRIVE

### Adicione validação de emails ao script de registro do Risky Jobs.

Use o código da página anterior para adicionar a validação de emails ao script registration.php. Depois envie o script para o seu servidor web, e abra-o em um navegador. Tente submeter um endereço de email inválido, e perceba como o código da nova expressão regular rejeita a submissão do formulário e exibe uma mensagem de erro para explicar o que aconteceu.

**Risky Jobs**

Danger! Your dream job is out there.  
Do you have the guts to go find it?

Risky Jobs - Registration

Your email address is invalid.

Register with Risky Jobs, and post your resume.

Name:

Name:

Email:

Phone:

Desired Job:

Paste your resume here:

I really love surfing, been doing it for years and years. And I have to tell you I'm quite good at spotting sharks and luring them to shore. I spent three years as a lifeguard here at Squeaky Beach.

Submit

Excelente! Eu cumprí a minha cota de contratações para empregos arriscados. Não preciso fazer mais nada, além de contar todo o dinheiro que ganhei.

Uma mensagem de erro informa que o endereço de email do usuário não é válido (ele tem um espaço onde deveria haver um símbolo @).

Com a ajuda da validação no formulário de registro do Risky Jobs, entrar em contato com candidatos promissores não é mais um problema, e é possível fazer as contratações em tempo recorde.

## Sua Caixa de Ferramentas do PHP & MySQL



Procurar por padrões em textos pode ser bastante útil no que se refere a validar dados digitados pelo usuário em formulários web. Eis aqui algumas das técnicas PHP usadas para se validar dados com a ajuda de expressões regulares:

### Expressão regular

Regras que são usadas para se encontrar padrões de texto em strings. O PHP inclui funções que lhe permitem usar expressões regulares para verificar se uma string segue um determinado padrão, bem como para localizar e-substituir padrões de texto dentro de uma string.

#### `preg_match()`

Esta função PHP verifica uma string de texto para ver se ela bate com uma expressão regular. A função retorna true se houver uma correspondência, e false caso contrário.

#### `preg_replace()`

Use esta função PHP para substituir uma substring dentro de uma string com base em uma expressão regular. A função localiza-e-substitui usando uma expressão regular para a localização e substitui por uma string que você fornece à ela.

#### `\d, \w, \s, ^, $`

As expressões regulares são criadas usando-se metacaracteres, os quais representam expressões de texto, como por exemplo, três dígitos numéricos (`\d\d\d`) ou espaço em branco (`\w`).

### Classe de caracteres

Um conjunto de regras para se encontrar um único caractere dentro de uma expressão regular. Por exemplo, `[A-D]` encontra os caracteres A, B, C ou D.

#### `checkdnsrr()`

Esta função PHP verifica um nome de domínio para ver se ele realmente existe. É útil quando estamos validando um endereço de email, porque você vai querer ter certeza de que a parte do domínio do email é real.

## 11 Visualizando seus Dados... e Mais!

# Desenhando Gráficos Dinâmicos

Fique parado. Espere, não se move. Agora olhe diretamente para mim e sorria. Não, não você, os seus dados. Ok; vamos tentar cruzar as suas colunas e inclinar a sua chave primária só um pouquinho para a esquerda. Isso, perfeito!

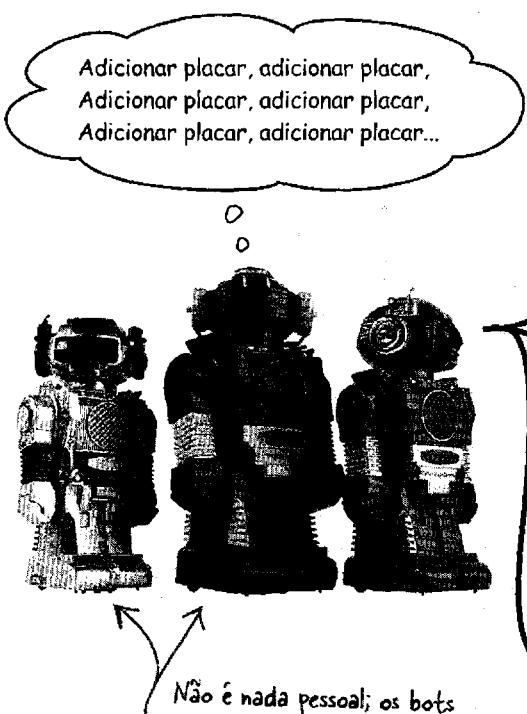


**É claro que todos nós conhecemos o poder de uma boa consulta e os consequentes resultados satisfatórios.** Mas os resultados de consultas nem sempre falam por si mesmos. As vezes é útil apresentar os dados de uma forma diferente, uma forma mais visual. O PHP possibilita o fornecimento de uma representação gráfica dos dados: gráficos pizza, gráficos em barra, diagramas de Venn, desenhos Rorschach, qualquer coisa. Vale de tudo para ajudar os usuários a compreender os dados que fluem através da sua aplicação. Mas nem todos os gráficos úteis, em aplicações PHP, se originam do seu banco de dados. Por exemplo, você sabia que é possível evitar ataques de spam de preenchimento de formulário robotizado com imagens geradas dinamicamente?

# Guitar Wars Reloaded: A Vingança das Máquinas

O futuro já chegou. Os robôs foram soltos no mundo virtual e não há muito que se possa fazer para impedi-los, exceto usar alguma vigilância através de código PHP. Esses robôs são os spam bots, os quais infestam a Web procurando por formulários de input que os permitem publicar anúncios às centenas. Esses robôs são incrivelmente eficientes e não estão nem aí para o que deveria ser o uso legítimo dos formulários que atacam. O seu único objetivo é substituir o seu conteúdo pelo deles, em uma tentativa desenfreada de conseguir receitas publicitárias para os seus mestres. Infelizmente, a aplicação de pontuações do Guitar Wars acabou caindo vítima dos bots.

Todos  
formulários  
web tem o risco  
de ataques de  
spam bots



Não é nada pessoal; os bots  
apenas querem a atenção dos seus  
usuários, para aumentar as receitas  
publicitárias dos seus mestres.

Os bots de spam são excelentes para repetição burra, neste caso preenchendo e submetendo um formulário do Guitar Wars atrás do outro, com dados que na verdade contêm anúncios, em vez de pontuações.

## Guitar Wars - Add Your High Score

Thanks for adding your new high score! It will be reviewed and added to the high score list as soon as possible.

Name: www.classshares.com  
Score: 99999999

## CLASSSHARES.COM

Reconnect and relive  
awful experiences  
with present

## Guitar Wars - Add Your High Score

Thanks for adding your new high score! It will be reviewed and added to the high score list as soon as possible.

Name: www.frownycentral.com  
Score: 99999999

## FROWNY CENTRAL



## Guitar Wars - Add Your High Score

Thanks for adding your new high score! It will be reviewed and added to the high score list as soon as possible.

Name: www.headlastlabs.com  
Score: 99999999

## Head Last Labs

Because sometimes it's just better not to think

## Nenhum formulário de input está a salvo

Felizmente para o Guitar Wars, os ataques dos bots de spam ficam invisíveis para o usuário final, graças ao recurso de moderação humana adicionado lá no Capítulo 6. No entanto, o moderador agora está completamente soterrado pelo enorme volume de posts dos bots de spam, o que lhe dificulta a tarefa de "peneirar" e aprovar as pontuações legítimas. A moderação humana é um excelente recurso, mas os humanos ficam em desvantagem ao se depararem com um adversário automatizado que nunca se cansa.

### Guitar Wars - High Scores Administration

Below is a list of all Guitar Wars high scores. Use this page to remove them if needed.

Name	Date	Score	Action
www.classhates.com	2008-06-23 11:44:56	9999999999	Remove / Approve
www.classhates.com	2008-06-23 11:45:15	9999999999	Remove / Approve
www.classhates.com	2008-06-23 11:45:29	9999999999	Remove / Approve
www.frowneycentral.com	2008-06-23 11:45:53	9999999999	Remove / Approve
www.frowneycentral.com	2008-06-23 11:46:06	9999999999	Remove / Approve
www.frowneycentral.com	2008-06-23 11:46:19	9999999999	Remove / Approve
www.frowneycentral.com	2008-06-23 11:47:26	9999999999	Remove / Approve
www.frowneycentral.com	2008-06-23 11:47:42	9999999999	Remove / Approve
www.headlastlabs.com	2008-06-23 11:47:55	9999999999	Remove / Approve
www.headlastlabs.com	2008-06-23 11:48:12	9999999999	Remove / Approve
www.headlastlabs.com	2008-06-23 11:50:24	9999999999	Remove / Approve
www.headlastlabs.com	2008-06-23 11:52:20	9999999999	Remove / Approve
www.headlastlabs.com	2008-06-23 11:52:32	9999999999	Remove / Approve

Isto é ridículo.  
Não consigo moderar  
todos estes posts, a maioria  
dos quais parecem ser falsos.  
Eu nem sequer sei o que é um  
frowney!

O nosso destemido  
moderador do  
Guitar Wars se  
encontra em uma  
difícil batalha  
contra bots  
que nunca se  
cansam de postar  
pontuações falsas,  
que na verdade  
são spam.

Está claro que a moderação humana dos posts não é o suficiente. O que nós precisamos é de uma forma de evitar que os robôs consigam submeter as pontuações – cortar o acesso logo no início, por assim dizer. Mas para isso será preciso distinguir, de alguma forma, entre um software automatizado e um ser humano com um cérebro real... um problema difícil, mas que pode ser solucionado.

Escreva abaixo três perguntas que você poderia fazer para distinguir entre um ser humano real e o cérebro artificial de um robô:

.....

.....

.....

## Precisamos separar os humanos das máquinas

Para descobrir como detectar se há um ser humano real no outro lado da página Adicionar Pontuação do Guitar Wars, você precisa primeiramente avaliar o que exatamente o bot de spam está fazendo quando ele preenche o formulário com dados falsos.

Para um bot de spam, é a coisa mais fácil do mundo bombardear um mesmo formulário dezenas, centenas, e até mesmo milhares de vezes... ugh!

**Guitar Wars - Add Your High Score**

Name:  Score:  Screen shot:

**Guitar Wars - Add Your High Score**

Name:  Score:  Screen shot:

O banco de dados do Guitar Wars está sendo inundado com pontuações inválidas porque os bots estão se aproveitando do formulário Adicionar Pontuação.

O formulário Adicionar Pontuação não tem nada para distinguir entre um post de um ser humano real e um post feito automaticamente por um robô.

O formulário Adicionar Pontuação precisa de um novo campo que exija a confirmação de que o usuário é um ser humano, antes de permitir que uma pontuação seja submetida.

**Guitar Wars - Add Your High Score**

Name:  Score:  Screen shot:

O problema com o formulário Adicionar Pontuação é que ele não faz nada para impedir submissões automatizadas, significando que qualquer programador habilidoso poderá criar um bot que preencha com anúncios e submeta o formulário repetidas vezes. É claro que esse spam nunca chega à página principal do Guitar Wars, graças ao recurso de moderação, mas ele torna a moderação inútil em muitos aspectos, porque o moderador acaba perdendo tempo removendo manualmente centenas de posts falsos que contêm anúncios.

O formulário precisa de um novo campo de verificação que precise ser informado com sucesso para que a pontuação seja submetida. E a verificação específica desse campo precisa ser algo fácil para um ser humano real, mas difícil para uma máquina.



Seguem abaixo algumas ideias de campos para o formulário que poderiam ser usados para impedir que bots de spam submetam posts. Circule os campos que achar que poderiam, de forma simples e eficaz, permitir que apenas seres humanos submetam posts, e não se esqueça de escrever por quê.

**Você é um robô?**  Sim  Não

**Qual era o prato favorito de Elvis?**

#### **Scaneamento de retina:**

**Digite as letras mostradas:**

**Quanto é 7 + 5?**

## **Que tipo de animal é este?**



**Digite as letras mostradas:**

#### **Scaneamento de impressão digital:**

## exercício solução



### Exercício Solução

Muito fácil de adivinhar — mesmo com uma taxa de sucesso de 50% na base do chute, você ainda acabaria tendo de apagar toneladas de posts falsos.

Seguem abaixo algumas ideias de campos para o formulário que poderiam ser usados para impedir que bots de spam submetam posts. Circule os campos que achar que poderiam, de forma simples e eficaz, permitir que apenas seres humanos submetam posts, e não se esqueça de escrever por quê.

**Você é um robô?**  Sim  Não

Certamente difícil para os robôs, mas potencialmente difícil para alguns seres humanos também. Nem todo mundo sabe que Elvis adorava sanduíches de pasta de amendoim com banana. Seria também preciso usar um grande banco de dados contendo as diversas perguntas e respostas.

**Qual era o prato favorito de Elvis??**

**Scaneamento de retina:**

**Digite as letras mostradas:**  kdyqmc

Excelente para impedir os bots, mas tecnicamente difícil e caro de implementar.

Não é má ideia, assumindo-se que as letras a serem digitadas apareçam como uma imagem, e não como texto, mas ainda é possível de ser quebrada por bots inteligentes o suficiente para usarem o reconhecimento óptico de caracteres (OCR).

**Quanto é  $7 + 5$ ?**

Simples e eficiente, pois a maioria dos bots não é esperta o suficiente para conseguir interpretar que está sendo pedido o resultado de uma expressão matemática — só temos de torcer para que a maioria dos seres humanos o consigam!

**Que tipo de animal é este?**



Lembra-se da Fang, o cão que foi abduzido por alienígenas no começo deste livro?

Bastante eficiente — os bots têm dificuldade em interpretar o conteúdo de imagens. Mas será preciso um banco de dados contendo as imagens e as respectivas respostas.

**Digite as letras mostradas:**  kdyqmc

Um interessante aprimoramento do verificador com letras aleatórias: aqui, as letras são obscurecidas com linhas e pontos, para confundir os bots que tenham OCR.

Não tão problemático quanto a verificação de retina, mas ainda exige hardware e software especiais.

**Scaneamento de impressão digital:**

# Podemos vencer a automação usando a automação

O teste usado para se verificar se a criatura do outro lado do formulário é uma pessoa real é conhecido como CAPTCHA, do termo inglês Completely Automated Public Turing Test to Tell Computers and Humans Apart ("Teste Turing Público Completamente Automatizado para Diferenciar Computadores e Humanos"). Isso é uma forma meio prolixa de se referir a qualquer "teste", em um formulário, em que apenas seres humanos podem passar. Já foram inventados diversos CAPTCHAs interessantes, mas um dos mais populares envolve a geração de uma senha aleatória, que o usuário precisa então digitar. Para ajudar a impedir que bots com reconhecimento óptico de caracteres (OCR) consigam vencer o sistema, as letras da senha são distorcidas ou parcialmente obscurecidas com linhas e pontos aleatórios.

Um CAPTCHA é um programa que protege um site contra bots automatizados, usando algum tipo de teste.

Uma vez que as letras da senha são geradas aleatoriamente, a senha é diferente a cada vez que o formulário é exibido.

**Digite as letras exibidas:**

Um campo de texto normal é usado para permitir que o usuário digite a senha CAPTCHA.

Linhas e pontos aleatórios ajudam a obscurecer o texto apenas o suficiente para impedir o reconhecimento óptico de caracteres, mas ao mesmo tempo permitindo que seres humanos o possam ler.

Um campo CAPTCHA é igual a qualquer outro campo do formulário, exceto pelo fato de que o seu propósito é impedir que o formulário seja submetido se o teste não tiver sido completado com sucesso. Assim, ao contrário de outros campos, que geralmente enviam dados ao servidor no momento da submissão, o campo CAPTCHA é verificado e usado para controlar todo o processo de submissão.

Uma vez que o bot de spam não é capaz de identificar a senha, a única coisa que ele pode fazer é tentar adivinhar.

**Digite as letras exibidas:**

**Errado!**




Para um ser humano real, é fácil identificar a senha.

**Digite as letras exibidas:**

**Correto!**




É muito importante que a senha CAPTCHA seja exibida no formulário como uma imagem, e não apenas como texto; caso contrário, seria muito mais fácil para os bots identificar as letras.

## ***não existem perguntas idiotas: edição captcha***

# **não existem Perguntas Ídotas**

**P:** Aquele CAPTCHA gráfico com a imagem do cachorro é bem legal. Eu poderia usá-lo em vez deste com senha?

**R:** Perfeitamente. Apenas tenha em mente que você terá de manter um banco de dados com imagens e descrições do que elas são, porque uma das chaves para o sucesso de qualquer CAPTCHA é a variedade. Um bom CAPTCHA precisa ter um repositório de conteúdo amplo o suficiente para que o formulário raramente mostre o mesmo teste duas vezes. Esse é o benefício do CAPTCHA com senha: uma vez que a senha é gerada a partir de letras aleatórias, é muito improvável que o mesmo teste apareça duas vezes para um mesmo usuário, mesmo com muitas tentativas repetidas.

**P:** Como o CAPTCHA funciona para os deficientes visuais? E se eles não conseguirem passar no teste visual?

**R:** Os CAPTCHAs visuais obviamente não são a melhor solução para os usuários com deficiência visual. Uma solução ideal poderia incluir uma alternativa sonora aos CAPTCHAs visuais. Por exemplo, existe um CAPTCHA de áudio no qual uma série de números é lida em voz alta, após o que o usuário precisa digitá-los para passar no teste. Mas ainda há o problema de bots inteligentes usarem reconhecimento de voz para fraudar o teste, que é o motivo pelo qual algumas dessas soluções usam áudio altamente distorcido, que faz a voz soar um pouco macabra. Os CAPTCHAs de áudio são tecnicamente semelhantes aos visuais, no sentido de requerem um banco de dados com clipes de áudio e as suas

respectivas respostas. Existem serviços que oferecem CAPTCHAs flexíveis, que utilizam tanto vídeo quanto áudio, como por exemplo [wwwcaptcha.net](http://wwwcaptcha.net). Tais serviços são excelentes para oferecer as tecnologias mais atuais, mas em geral eles não se integram tão bem à sua aplicação web quanto um CAPTCHA personalizado feito especificamente para ela.

**P:** Mas há também pessoas com pouca acuidade visual, e pessoas com deficiência auditiva. E quanto a elas?

**R:** No fim das contas, você precisará pesar a conveniência de impedir os ataques dos bots contra o risco de alienar alguns usuários. De forma semelhante ao que ocorre entre vírus e softwares antivírus, os bots de spam e os CAPTCHAs provavelmente continuarão nesse jogo de gato e rato, em que bots são criados para vencer um determinado CAPTCHA, o que fará surgir um CAPTCHA mais sofisticado, e assim por diante. Pegos no fogo cruzado estão os usuários que poderão acabar excluídos por causa da acessibilidade limitada de alguns CAPTCHAs. Fica a cargo de cada desenvolvedor web pesar os riscos de um ataque de bots contra a potencial perda de usuários que poderão não conseguir acessar partes do site. Se servir de consolo, tenha em mente que os bots mais sofisticados geralmente miram nos alvos mais graúdos, onde poderão obter um bom retorno em termos de receita publicitária, o que significa que você poderá não encontrar um bot realmente maléfico até que o seu site cresça ao ponto de se tornar um alvo grande o suficiente para os bots mais poderosos.

OK, então a senha do CAPTCHA precisa ser exibida como uma imagem, com linhas e pontos aleatórios. Tudo bem, mas como isso pode ser criado com o PHP? O PHP só é capaz de gerar código HTML, certo?

**O PHP possui capacidades gráficas que podem gerar imagens dinamicamente, as quais você pode então exibir usando código HTML.**

Com a ajuda de uma biblioteca de gráficos chamada GD (Graphics Draw), os nossos scripts PHP são capazes de gerar imagens dinamicamente em formatos populares, tais como GIF, JPEG e PNG, e ou retorná-las a um navegador para serem exibidas ou escrevê-las em um arquivo no servidor. Esse recurso do PHP é extremamente importante, porque não existe a possibilidade de se "desenhar" em uma página web usando unicamente HTML. O PHP lhe permite "desenhar" em uma parte da página, realizando operações gráficas em uma imagem e depois exibindo essa imagem na página com a familiar tag <img>.



## Gere o texto da senha de CAPTCHA

Antes de sequer podermos pensar no aspecto gráfico de um CAPTCHA com senha, precisamos descobrir como gerar a própria senha aleatória, que começa como uma sequência de caracteres de texto. A senha pode ter qualquer número de caracteres, mas algo entre seis e oito caracteres em geral é o suficiente. Podemos usar uma constante para a extensão da senha, o que nos permite modificar facilmente o número de caracteres posteriormente, se for preciso.

```
define('CAPTCHA_NUMCHARS', 6);
```

Uma senha CAPTCHA  
com seis caracteres  
provavelmente é suficiente  
para impedir os bots sem  
atrapalhar os humanos.



`$pass_phrase`

Então como exatamente nós vamos gerar uma string aleatória de texto com seis caracteres? É aqui que duas funções internas do PHP entram em cena: rand() e chr(). A função rand() retorna um número aleatório na faixa especificada pelos seus dois argumentos, enquanto que chr() converte um código de caracter ASCII numérico em um caracter propriamente dito. O ASCII (American Standard Code for Information Interchange ou Código Padrão Americano para Intercâmbio de Informações) é uma codificação de caracteres padrão, que representa os caracteres através de números. Nós só precisamos dos códigos ASCII na faixa de 97-122, que representam as letras minúsculas a-z. Se gerarmos um código nessa faixa seis vezes, iremos obter uma senha de seis caracteres aleatórios, todos os quais sendo letras minúsculas.

```
// Gera a senha aleatória
$pass_phrase = "";
for ($i = 0; $i < CAPTCHA_NUMCHARS; $i++) {
    $pass_phrase .= chr(rand(97, 122));
}
```

Mais adiante, este  
código será colocado em  
um script reutilizável  
próprio, captcha.php.

Um loop para cada caracter  
da senha.

A senha é construída um  
caracter aleatório de  
cada vez.

### rand()

Esta função interna retorna um número inteiro aleatório ou dentro de uma faixa especificada ou entre 0 e a constante interna RAND\_MAX (dependente do servidor). Para obter um número aleatório dentro de uma determinada faixa, basta enviar os limites inferior e superior da faixa como argumentos para rand().

**chr()**

Esta função interna converte um número no seu caracter ASCII equivalente. Como exemplo, o número 97 é o código ASCII para a letra 'a' minúscula. Assim, chamar chr(97) retorna o caracter 'a'.

A função rand() retorna um número inteiro aleatório, dentro de uma determinada faixa.

## Visualizando a imagem CAPTCHA

Criada a senha aleatória, nós podemos passar para a tarefa de gerar uma imagem consistindo do texto da senha juntamente com linhas e pontos aleatórios, para ajudar a obscurecer o texto. Mas por onde começar? A primeira coisa a fazer é decidir qual será o tamanho da imagem CAPTCHA. Sabendo que essa imagem será exibida em um formulário, ao lado de um campo de input, faz sentido mantê-la relativamente pequena. Vamos tentar 100x25 e vamos colocar esses valores em constantes, para que o tamanho da imagem seja definido em apenas um lugar e, portanto, seja fácil de modificar depois, se necessário.

```
define('CAPTCHA_WIDTH', 100);  
define('CAPTCHA_HEIGHT', 25);
```

O tamanho da imagem CAPTCHA fica armazenado em constantes, para que seja mais fácil ajustá-lo depois, se for preciso.

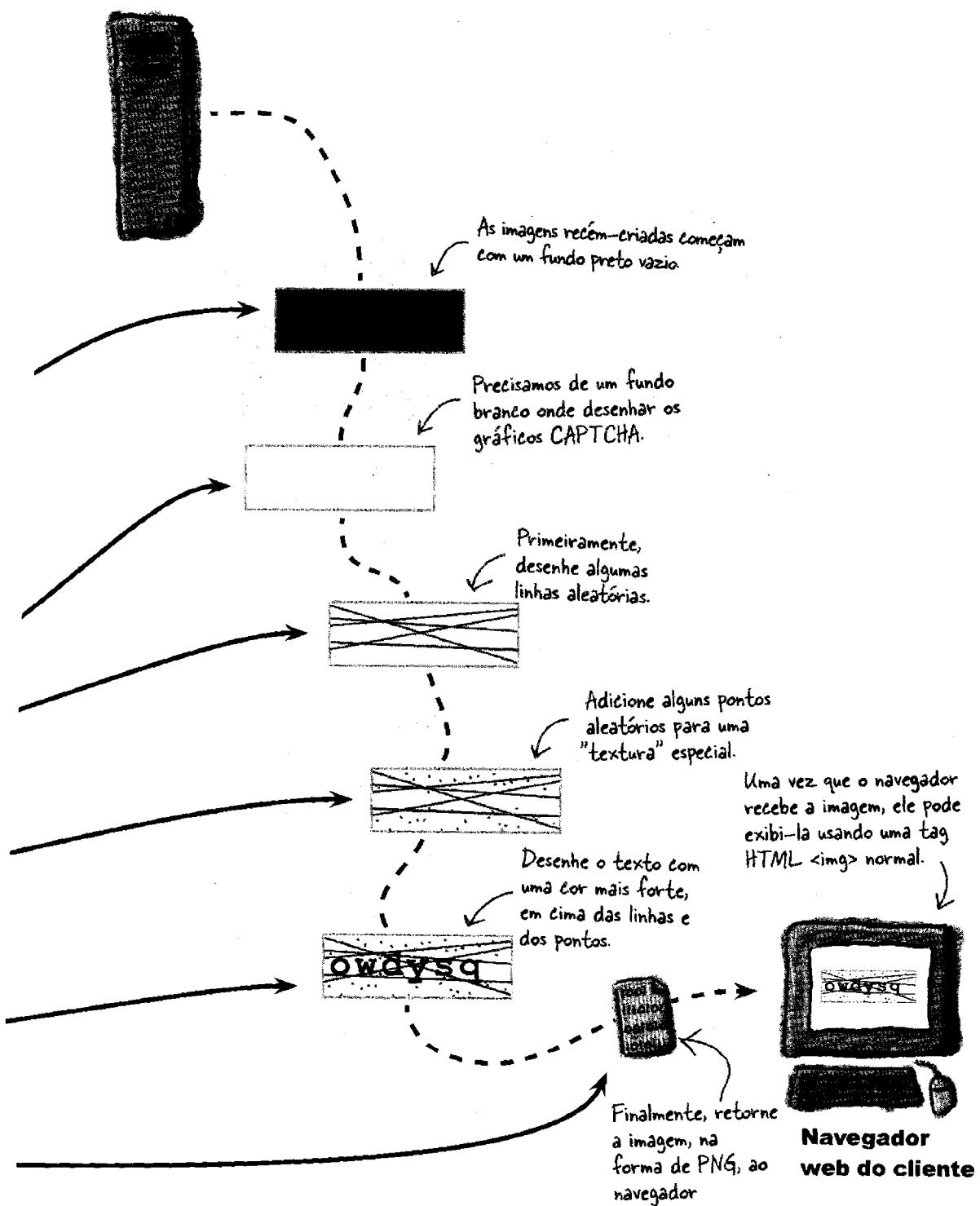
Para se desenhar a imagem CAPTCHA, é preciso chamar algumas funções da biblioteca GD, todas as quais operam sobre uma imagem carregada na memória. Em outras palavras, você cria uma imagem na memória e então desenha em cima dela; depois, quando tiver terminado, você a envia para o navegador para ser exibida.

```
// Cria a imagem  
$img = imagecreate(truecolor(CAPTCHA_WIDTH, CAPTCHA_HEIGHT));  
  
// Define um fundo branco com texto preto e gráficos cinza  
$bg_color = imagecolorallocate($img, 255, 255, 255); // branco  
$text_color = imagecolorallocate($img, 0, 0, 0); // preto  
$graphic_color = imagecolorallocate($img, 64, 64, 64); // cinza escuro  
  
// Preenche o fundo  
imagefilledrectangle($img, 0, 0, CAPTCHA_WIDTH, CAPTCHA_HEIGHT, $bg_color);  
  
// Desenha algumas linhas aleatórias  
for ($i = 0; $i < 5; $i++) {  
    imagedline($img, rand(0, rand() % CAPTCHA_HEIGHT, CAPTCHA_WIDTH,  
        rand() % CAPTCHA_HEIGHT, $graphic_color));  
}  
  
// Insere alguns pontos aleatórios  
for ($i = 0; $i < 50; $i++) {  
    imagesetpixel($img, rand() % CAPTCHA_WIDTH,  
        rand() % CAPTCHA_HEIGHT, $graphic_color);  
}  
  
// Desenha a string da senha  
imagettftext($img, 18, 0, 5, CAPTCHA_HEIGHT - 5, $text_color,  
    "Courier New Bold Italic", $passphrase);  
  
// Faz output da imagem como PNG, usando um cabeçalho  
header("Content-type: image/png");  
imagepng($img);
```

Para se desenhar uma imagem dinâmica em PHP, é preciso usar funções da biblioteca GD.

Este código cria cores para serem usadas por outras funções GD.

## Servidor web



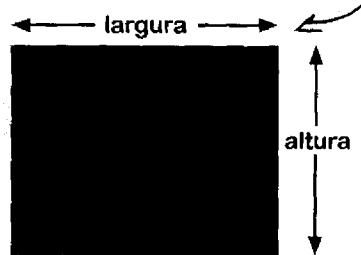
## Por dentro das funções gráficas GD

A mágica por trás da criação de imagens CAPTCHA é possibilitada pela biblioteca de gráficos GD, que, como você já aprendeu, oferece funções para se desenhar gráficos dinamicamente usando-se código PHP. Vamos examinar algumas dessas funções com mais detalhes e ver como elas se relacionam à geração da imagem CAPTCHA.

### `imagecreatetruecolor()`

Esta função cria uma imagem em branco na memória, pronta para ser editada por outras funções GD. Os dois argumentos de `imagecreatetruecolor()` são a largura e a altura da imagem. A imagem no início é apenas um fundo preto, de modo que geralmente você vai querer preenchê-la com uma cor de fundo, por exemplo, branco, antes de desenhar qualquer coisa. Isso pode ser feito chamando-se a função `imagefilledrectangle()`. O valor de retorno de `imagecreatetruecolor()` é um **identificador de imagem**, o qual a maioria das funções GD requerem como primeiro argumento para a identificação da imagem que está sendo desenhada.

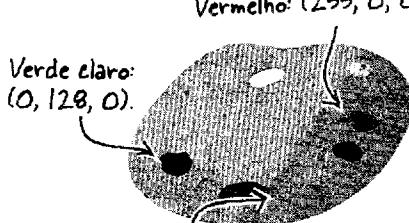
As imagens são criadas inicialmente com fundos pretos.



```
$img = imagecreatetruecolor(CAPTCHA_WIDTH, CAPTCHA_HEIGHT);
```

A função retorna um identificador para a imagem, o qual é requerido pelas outras funções para que você possa usá-las para desenhar algo na imagem.

Este código cria uma imagem com tamanho 100x25, definido nas nossas constantes.



Azul: (0, 0, 255).

O valor de retorno é um identificador da cor, o qual você pode usar em outras funções de desenho para determinar a cor a ser usada, como por exemplo, a cor do texto CAPTCHA.

### `imagecolorallocate()`

Use esta função para alocar uma cor a ser usada em outras funções de desenho. O primeiro argumento é o identificador da imagem, seguido de três argumentos representando os três componentes numéricos do código RGB (do inglês Red-Green-Blue, ou "Vermelho-Verde-Azul"). Cada um desses valores pode ir de 0 a 255. O valor de retorno é um identificador da cor, o qual pode ser usado para identificá-lo em outras funções de desenho, frequentemente como o último argumento.

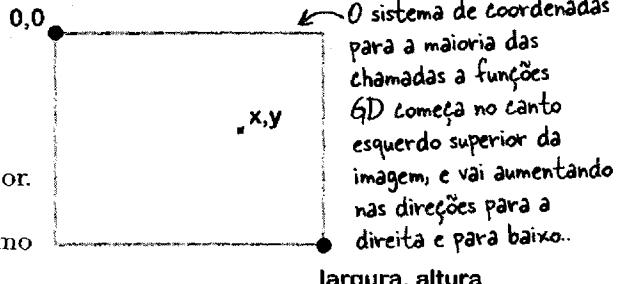
O identificador da imagem na qual a cor irá ser usada.

Os componentes vermelho, verde e azul da cor, neste caso preta.

```
$text_color = imagecolorallocate($img, 0, 0, 0);
```

**imagesetpixel()**

Esta função desenha um único pixel em uma coordenada especificada, dentro da imagem. As coordenadas começam em 0,0 para o canto esquerdo superior da imagem, e vão aumentando até chegar ao canto direito inferior. Como na maioria das funções GD, o pixel é desenhado usando-se a cor que foi passada como o último argumento para a função.



largura, altura

```
imagesetpixel($img, rand() % CAPTCHA_WIDTH, rand() % CAPTCHA_HEIGHT, $graphic_color);
```

A imagem (identificador) onde o pixel está sendo desenhado.

A coordenada XY do pixel, relativa ao canto esquerdo superior da imagem, que neste caso acaba sendo uma localização aleatória dentro da imagem CAPTCHA.

A cor (identificador) do pixel.

**imageline()**

Chame esta função para desenhar uma linha entre duas coordenadas ( $x_1, y_1$  e  $x_2, y_2$ ). As coordenadas são especificadas relativamente ao canto esquerdo superior da imagem, e a linha é desenhada na cor que foi passada como o último argumento para a função.



As coordenadas XY do início da linha, neste caso no canto esquerdo da imagem CAPTCHA.

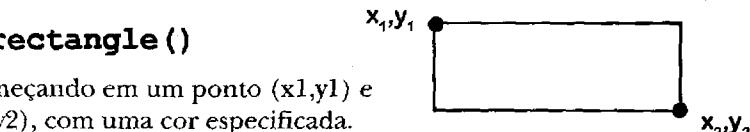
```
imageline($img, 0, rand() % CAPTCHA_HEIGHT, CAPTCHA_WIDTH, rand() % CAPTCHA_HEIGHT, $graphic_color);
```

O ponto final XY da linha, que neste caso, localiza-se na borda direita da imagem CAPTCHA.

**imagerectangle()**

Desenha um retângulo começando em um ponto ( $x_1, y_1$ ) e terminando em outro ( $x_2, y_2$ ), com uma cor especificada. Os dois pontos e a cor são fornecidos como argumentos, do segundo até o sexto, para a função, vindo logo após o argumento com o identificador da imagem.

A função imagerectangle() usa exatamente os mesmos argumentos que imagefilledrectangle().

**imagefilledrectangle()**

Semelhante a imagerectangle(), esta função desenha um retângulo cujo interior é preenchido com a cor especificada.

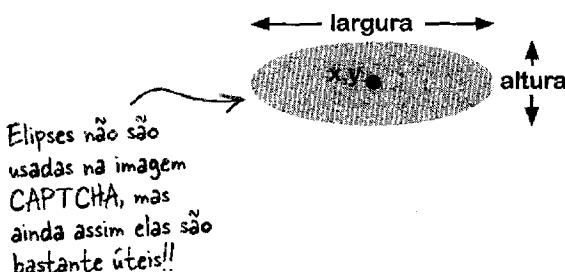
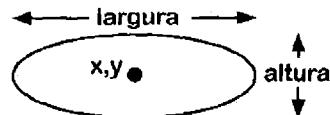
```
imagefilledrectangle($img, 0, 0, CAPTCHA_WIDTH, CAPTCHA_HEIGHT, $bg_color);
```

As coordenadas XY dos pontos inicial e final - aqui, a imagem CAPTCHA inteira é preenchida.

## Continuando as funções gráficas GD...

### `imageellipse()`

Usada para se desenhar círculos e elipses, esta função aceita um ponto central, uma largura e uma altura. Um círculo perfeito é apenas uma elipse com largura e altura iguais. A cor da elipse ou círculo é passada como o último argumento para a função.



### `imagefilledellipse()`

Precisa de uma elipse preenchida? Basta chamar `imagefilledellipse()`, que funciona da mesma forma que `imageellipse()` exceto pelo fato de que a cor especificada é usada para se preencher a elipse, em vez de contorná-la.

→ `imagefilledellipse($img, 0, 0, 320, 240, $color);`

Tanto `imageellipse()` quanto `imagefilledellipse()` aceitam os mesmos argumentos.

A largura e a altura da elipse  
- defina estes com o mesmo  
número para desenhar um  
círculo perfeito.

A coordenada XY  
do centro da elipse.

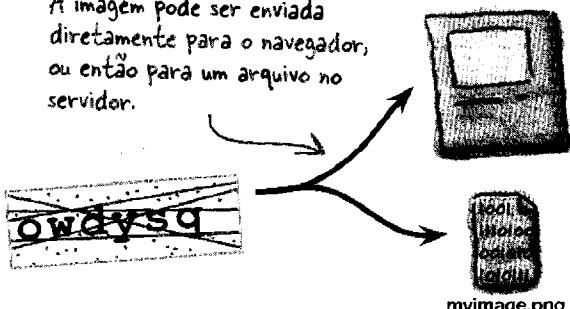
### `imagepng()`

Quando tiver terminado de desenhar em uma imagem, você pode enviá-la diretamente ao navegador do cliente, ou então, para um arquivo no servidor chamando esta função. De um jeito ou de outro, o resultado final é uma imagem que pode ser usada com a tag HTML `<img>` para exibição em uma página web. Se preferir gerar uma imagem PNG diretamente na memória (ou seja, sem um arquivo), então você precisará chamar também a função `header()` para que a imagem seja entregue ao navegador através de um cabeçalho.

A função retorna true ou false dependendo de se a imagem foi criada com sucesso ou não.

→ `imagepng($img);`

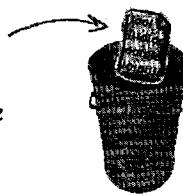
A imagem pode ser enviada diretamente para o navegador, ou então para um arquivo no servidor.



O identificador de imagem  
que você vem usando nas  
outras funções de desenho.

Você pode passar um nome de arquivo como um segundo argumento opcional - sem ele, a função gera uma imagem na memória que pode ser enviada de volta ao navegador, em um cabeçalho.

Apagar as suas imagens quando elas não são mais necessárias é uma boa ideia, para impedir o desperdício de recursos do servidor.



### **imagedestroy()**

O trabalho com imagens usando-se a biblioteca GD consome recursos do sistema, e esta função se encarrega de liberá-los depois que você tiver terminado de trabalhar com a imagem. Basta chamá-la depois de fazer o output com imagepng().

Semelhante a imagepng(), esta função retorna true se a operação tiver sucesso, e false, caso contrário.

**Libere sempre as imagens da memória, depois de usá-las, com imagedestroy()**

→ **imagedestroy(\$img);**

O identificador da imagem que você deseja destruir.

Tente sempre escrever uma chamada a esta função para cada imagem que você criar, para que todas as imagens sejam destruídas depois de serem usadas.

### **imagestring()**

Esta função desenha uma string de texto usando a fonte interna do PHP, na cor especificada. Além do identificador da imagem, você passa à função o tamanho da fonte (na forma de um número de 1 a 5), junto com as coordenadas para o canto esquerdo superior da string, a própria string, e finalmente, a cor.

O tamanho da fonte a ser usado, na faixa de 1 a 5.

→ **imagestring(\$img, 3, 75, 75, 'Exemplo de texto', \$color);**

A coordenada XY do canto esquerdo superior da string.

O texto desenhado com imagestringup() é girado em 90 graus no sentido anti-horário, para aparecer na vertical.



O número na faixa de 1 a 5 define o tamanho da fonte usada para se desenhar a string de texto, com 5 sendo o maior tamanho.

x,y Exemplo de texto

↑ A fonte interna é adequada para o desenho de texto básico, mas é limitada em termos de tamanho.

Está é a string de texto a ser desenhada.

↑ A cor do texto.

### **imagestringup()**

Semelhante a imagestring(), esta função desenha uma string de texto usando a fonte interna, mas o desenha verticalmente, como se ele fosse girado 90 graus no sentido anti-horário. A função é chamada com os mesmos argumentos que imagestring().

a função `imagettftext()`

## Desenhando texto com uma fonte

A função `imagestring()` é fácil de usar para desenhos, mas é um tanto limitada em termos do controle que você tem sobre a aparência do texto. Para conseguir uma aparência específica, você precisa usar a sua própria fonte True Type. A imagem com a senha CAPTCHA é um bom exemplo dessa necessidade, uma vez que os caracteres precisam ser desenhados em tamanho razoavelmente grande, e de preferência em negrito. Para conseguir esse look personalizado, você precisa da ajuda de mais uma função GD, que desenha o texto usando uma fonte True Type que você fornece no servidor.

### `imagettfttext()`

Para desenhar um texto realmente personalizado, coloque uma fonte True Type no seu servidor web e depois chame esta função. Você não apenas pode usar qualquer fonte que quiser; como também obtém mais flexibilidade quanto ao tamanho da fonte, e mesmo quanto ao ângulo no qual o texto é desenhado. Diferentemente de `imagestring()`, as coordenadas passadas a esta função especificam o "ponto-base" do primeiro caracter do texto, que é aproximadamente o canto esquerdo inferior do primeiro caracter.

Esta função requer que você coloque um arquivo contendo uma fonte True Type no seu servidor, e depois especifique esse arquivo como o último argumento. Os arquivos de fontes True Type geralmente têm a extensão `.ttf`.

Para se desenhar um texto altamente personalizado, é preciso usar uma fonte True Type e a função `imagettfttext()`.

### x,y Exemplo de texto

Diferentemente de `imagestring()`, as coordenadas usadas para se desenhar texto com `imagettfttext()` apontam para o canto esquerdo inferior do texto.

O tamanho da fonte, geralmente especificado em "pontos".

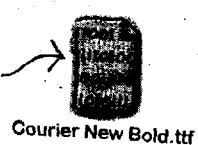
O ângulo da fonte, especificado em graus no sentido anti-horário (0 é o texto normal).

```
imagettfttext($img, 18, 0, 5, CAPTCHA_HEIGHT - 5, $text_color,  
'Courier New Bold.ttf', $pass_phrase);
```

As coordenadas XY do canto esquerdo inferior do texto.

O texto que está sendo desenhado.

Você precisa colocar a fonte True Type no seu servidor web, para que a biblioteca de gráficos GD possa encontrá-la.



### Nota Geek



Se quiser tentar criar a sua própria fonte TrueType, para personalizar ainda mais o seu CAPTCHA, dê uma olhada em [www.fontstruct.com](http://www.fontstruct.com). Essa é uma comunidade online de criadores de fontes, incluindo uma ferramenta web para que você possa criar fontes personalizadas.

Use a função `imagettfttext()` para desenhar um texto altamente personalizado, com a sua própria fonte True Type.

\* \* \* \* \*  
DESENHA  
COMO FAZ O CÓDIGO?

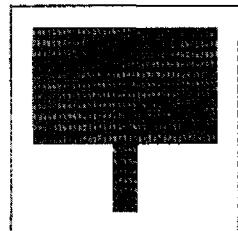
```
imagecolorallocate($img, 128, 128, 128);
```

Ligue cada bloco de código PHP à imagem gráfica gerada por ele. Assuma que a imagem (\$img) e as cores (\$black\_color, \$white\_color e \$gray\_color) já tenham sido criadas.

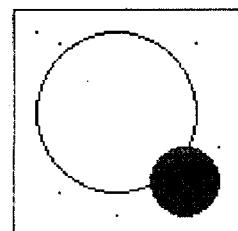
```
imagecolorallocate($img, 0, 0, 0); imagecolorallocate($img,
```

```
255, 255, 255);
```

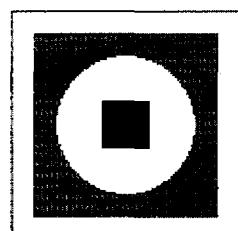
```
imagefilledrectangle($img, 10, 10, 90, 90, $gray_color);
imagefilledellipse($img, 50, 50, 60, 60, $white_color);
imagefilledrectangle($img, 40, 40, 60, 60, $black_color);
```



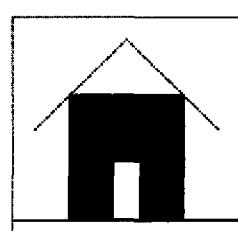
```
imageline($img, 15, 15, 50, 50, $black_color);
imageline($img, 15, 85, 50, 50, $black_color);
imageline($img, 50, 50, 85, 50, $black_color);
imagefilledellipse($img, 15, 15, 20, 20, $gray_color);
imagefilledellipse($img, 15, 85, 20, 20, $gray_color);
imagefilledellipse($img, 50, 50, 20, 20, $gray_color);
imagefilledellipse($img, 85, 50, 20, 20, $gray_color);
```



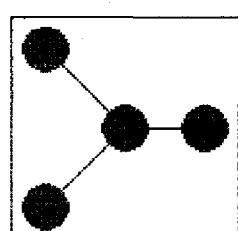
```
imagefilledrectangle($img, 10, 10, 90, 60, $gray_color);
imagesetpixel($img, 30, 25, $black_color);
imagesetpixel($img, 70, 25, $black_color);
imageline($img, 35, 45, 65, 45, $black_color);
imagefilledrectangle($img, 45, 50, 55, 90, $gray_color);
```



```
imageellipse($img, 45, 45, 70, 70, $black_color);
imagefilledellipse($img, 75, 75, 30, 30, $gray_color);
imagesetpixel($img, 10, 10, $black_color);
imagesetpixel($img, 80, 15, $black_color);
imagesetpixel($img, 20, 15, $black_color);
imagesetpixel($img, 90, 60, $black_color);
imagesetpixel($img, 20, 80, $black_color);
imagesetpixel($img, 45, 90, $black_color);
```



```
imagefilledrectangle($img, 25, 35, 75, 90, $black_color);
imageline($img, 10, 50, 50, 10, $black_color);
imageline($img, 50, 10, 90, 50, $black_color);
imagefilledrectangle($img, 45, 65, 55, 90, $white_color);
imageline($img, 0, 90, 100, 90, $black_color);
```



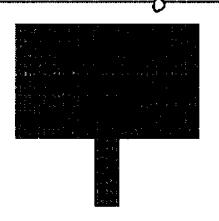
**quem desenha o que solução**

**DESENHA**

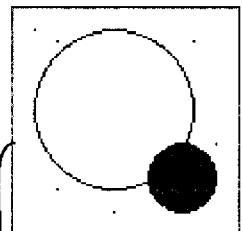
Ligue cada bloco de código PHP à imagem gráfica gerada por ele.  
Assuma que a imagem (\$img) e as cores (\$black\_color, \$white\_color e \$gray\_color) já tenham sido criadas.

Eu  
sou um andróide,  
e não um robô.

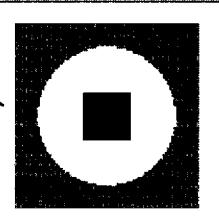
```
imagefilledrectangle($img, 10, 10, 90, 90, $gray_color);  
imagefilledellipse($img, 50, 50, 60, 60, $white_color);  
imagefilledrectangle($img, 40, 40, 60, 60, $black_color);
```



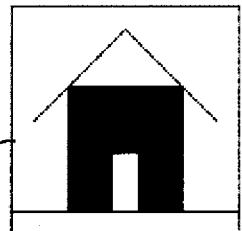
```
imageline($img, 15, 15, 50, 50, $black_color);  
imageline($img, 15, 85, 50, 50, $black_color);  
imageline($img, 50, 50, 85, 50, $black_color);  
imagefilledellipse($img, 15, 15, 20, 20, $gray_color);  
imagefilledellipse($img, 15, 85, 20, 20, $gray_color);  
imagefilledellipse($img, 50, 50, 20, 20, $gray_color);  
imagefilledellipse($img, 85, 50, 20, 20, $gray_color);
```



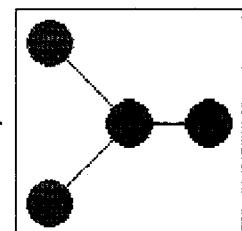
```
imagefilledrectangle($img, 10, 10, 90, 60, $gray_color);  
imagesetpixel($img, 30, 25, $black_color);  
imagesetpixel($img, 70, 25, $black_color);  
imageline($img, 35, 45, 65, 45, $black_color);  
imagefilledrectangle($img, 45, 50, 55, 90, $gray_color);
```



```
imageellipse($img, 45, 45, 70, 70, $black_color);  
imagefilledellipse($img, 75, 75, 30, 30, $gray_color);  
imagesetpixel($img, 10, 10, $black_color);  
imagesetpixel($img, 80, 15, $black_color);  
imagesetpixel($img, 20, 15, $black_color);  
imagesetpixel($img, 90, 60, $black_color);  
imagesetpixel($img, 20, 80, $black_color);  
imagesetpixel($img, 45, 90, $black_color);
```



```
imagefilledrectangle($img, 25, 35, 75, 90, $black_color);  
imageline($img, 10, 50, 50, 10, $black_color);  
imageline($img, 50, 10, 90, 50, $black_color);  
imagefilledrectangle($img, 45, 65, 55, 90, $white_color);  
imageline($img, 0, 90, 100, 90, $black_color);
```



# Gere uma imagem CAPTCHA aleatória

Agora, nós juntamos todo o código CAPTCHA em um script `captcha.php`, o qual se encarrega de gerar uma senha aleatória e depois retornar uma imagem PNG para o navegador.

O script `captcha.php` é completamente autônomo  
- você pode abri-lo no seu navegador e ver a imagem que ele gera.

```
<?php
session_start();

// Define algumas constantes importantes
define('CAPTCHA_NUMCHARS', 6); // número de caracteres na senha
define('CAPTCHA_WIDTH', 100); // largura da imagem
define('CAPTCHA_HEIGHT', 25); // altura da imagem

// Gera a senha aleatória
$pass_phrase = "";
for ($i = 0; $i < CAPTCHA_NUMCHARS; $i++) {
    $pass_phrase .= chr(rand(97, 122));
}

// Armazena a senha criptografada em uma variável de sessão
$_SESSION['pass_phrase'] = sha1($pass_phrase);

// Cria a imagem
$img = imagecreatetruecolor(CAPTCHA_WIDTH, CAPTCHA_HEIGHT);

// Define um fundo branco com texto preto e gráficos cinza
$bg_color = imagecolorallocate($img, 255, 255, 255); // branco
$text_color = imagecolorallocate($img, 0, 0, 0); // preto
$graphic_color = imagecolorallocate($img, 64, 64, 64); // cinza escuro

// Preenche o fundo
imagefilledrectangle($img, 0, 0, CAPTCHA_WIDTH, CAPTCHA_HEIGHT, $bg_color);

// Desenha algumas linhas aleatórias
for ($i = 0; $i < 5; $i++) {
    imageline($img, 0, rand() % CAPTCHA_HEIGHT, CAPTCHA_WIDTH, rand() % CAPTCHA_HEIGHT, $graphic_color);
}

// Insere alguns pontos aleatórios
for ($i = 0; $i < 50; $i++) {
    imagesetpixel($img, rand() % CAPTCHA_WIDTH, rand() % CAPTCHA_HEIGHT, $graphic_color);
}

// Desenha a string da senha
imagettftext($img, 18, 0, 5, CAPTCHA_HEIGHT - 5, $text_color, "Courier New Bold.ttf",
$pass_phrase);
// Faz output da imagem como PNG, usando um cabeçalho
header("Content-type: image/png");
imagepng($img);

// Apaga a imagem
imagedestroy($img);
?>
```

**Crie constantes para armazenar o número de caracteres no CAPTCHA e defina a altura e largura da imagem.**

**Embora você possa armazenar a senha criptografada no banco de dados, é mais simples colocá-la apenas em uma variável de sessão - nós precisamos armazená-la em algum lugar para que o script Adicionar Pontuação possa ter acesso a ela.**

**Algumas versões da biblioteca GD requerem que você use um caminho relativo até o arquivo da fonte, como "/Courier New Bold.ttf".**

**Gere uma imagem PNG contendo tudo o que foi desenhado.**

**A imagem PNG é enviada ao navegador através de um cabeçalho.**

Finalizando, apague a imagem da memória (ela ainda é enviada ao navegador, através do cabeçalho).



**teste captcha.php**

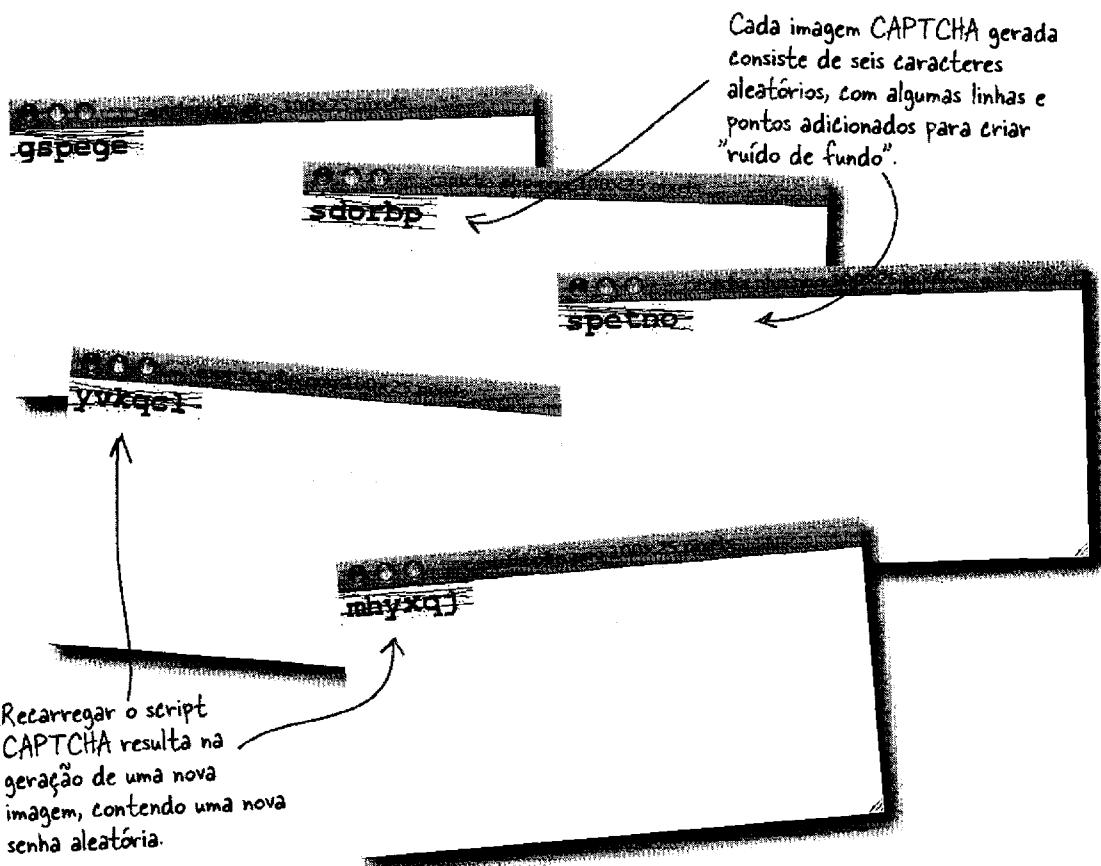


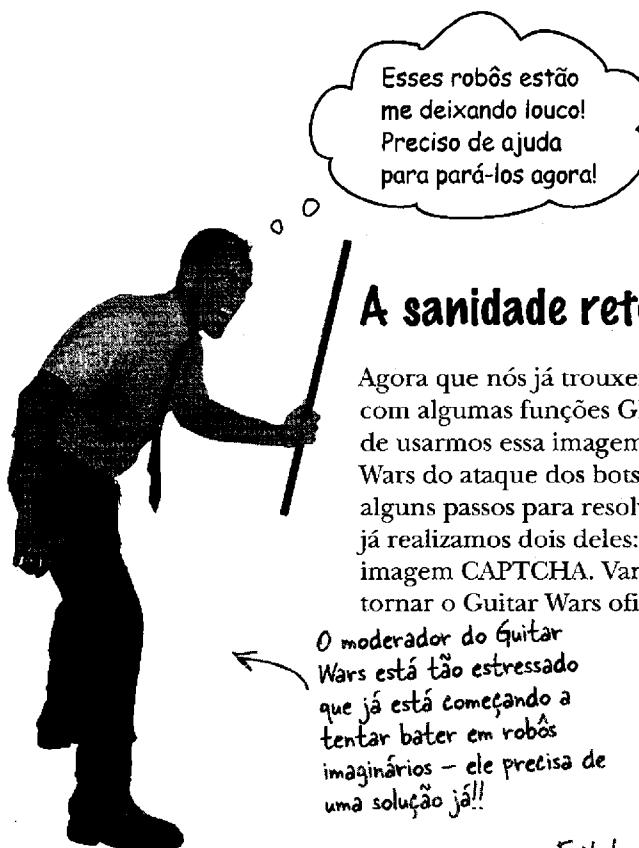
## TEST DRIVE

### Crie o script CAPTCHA e teste-o.

Crie um arquivo de texto chamado captcha.php, e digite nele o código para o script CAPTCHA mostrado na página anterior (ou baixe o script no site da Alta Books em [www.altabooks.com.br](http://www.altabooks.com.br)).

Envie o script para o seu servidor web, e depois abra-o em um navegador. Você verá imediatamente a imagem CAPTCHA com a senha aleatória no navegador. Para gerar uma nova senha, recarregue o navegador.





## A sanidade retorna ao Guitar Wars

Agora que nós já trouxemos à tona o seu artista PHP interior, com algumas funções GD e uma imagem CAPTCHA, é hora de usarmos essa imagem para salvar o moderador do Guitar Wars do ataque dos bots. Na verdade, será preciso seguir alguns passos para resolver o problema. A boa notícia é que já realizamos dois deles: gerar a senha aleatória e desenhar a imagem CAPTCHA. Vamos eliminar os passos restantes para tornar o Guitar Wars oficialmente livre de bots!

O moderador do Guitar Wars está tão estressado que já está começando a tentar bater em robôs imaginários – ele precisa de uma solução já!!

*Feito!*  
**1 Gerar uma senha aleatória**

*Desenho pronto!*  
**2 Desenhar uma imagem CAPTCHA usando a senha**

**3 Exibir a imagem CAPTCHA no formulário**  
Adicionar Pontuação do Guitar Wars e pedir ao usuário que digite a senha.

**4 Verificar a senha em relação ao input do usuário.**



Complete o Passo 3 do CAPTCHA do Guitar Wars, escrevendo o código HTML para um novo campo de input de texto chamado Verificação, o qual pedirá ao usuário para digitar a senha do CAPTCHA. Certifique-se de dar a esse campo um rótulo, e ponha uma tag `<img>` depois dele para exibir a imagem CAPTCHA gerada pelo script `captcha.php`.

## exercício solução



### Exercício Solução

Complete o Passo 3 do CAPTCHA do Guitar Wars, escrevendo o código HTML para um novo campo de input de texto chamado Verificação, o qual pedirá ao usuário para digitar a senha do CAPTCHA. Certifique-se de dar a esse campo um rótulo, e ponha uma tag depois dele para exibir a imagem CAPTCHA gerada pelo script captcha.php.

Uma tag <label> é usada para se rotular o novo campo texto Verificação.

<label for="verify">Verificação: </label>

<input type="text" id="verify" name="verify" value="Enter the pass-phrase." />



Este campo de texto é onde o usuário irá digitar a senha revelada na imagem CAPTCHA.

A "fonte" da imagem é o nome do script PHP que gera dinamicamente a imagem CAPTCHA. Isso funciona porque o script captcha.php retorna uma imagem diretamente para o navegador, através de imagepng() e um cabeçalho.

A imagem CAPTCHA é exibida no formulário, ao lado de um campo de texto.

The screenshot shows a web form titled "Guitar Wars - Add Your High Score". It has fields for "Name", "Score", and "Screen shot" (with a file selection button). Below these is a "Verification" field containing the text "guitarwars" and a "Verify" button. Arrows point from the explanatory text above to the "Verification" field and the "Verify" button.

Pronto! Só mais um passo para terminarmos.

- 3 Exibir a imagem CAPTCHA no formulário Adicionar Pontuação do Guitar Wars e pedir ao usuário que digite a senha.

# Adicione CAPTCHA ao script Adicionar Pontuação

No lado do cliente, o script addscore.php contém o novo campo de texto Verificação com a imagem CAPTCHA ao seu lado. A modificação mais importante, porém, é a nova declaração if no script Adicionar Pontuação (Passo 4), que verifica se a senha digitada pelo usuário bate com a de CAPTCHA.

```

<?php
    session_start();
?>

<html>
<head>
    <title>Guitar Wars - Adicione o seu Recorde</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
    <h2>Guitar Wars - Adicione o seu Recorde</h2>

<?php
    require_once('appvars.php');
    require_once('connectvars.php');

    if (isset($_POST['submit'])) {
        // Conecta-se ao banco de dados
        $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

        // Pega os dados em POST
        $name = mysqli_real_escape_string($dbc, trim($_POST['name']));
        $score = mysqli_real_escape_string($dbc, trim($_POST['score']));
        $screenshot = mysqli_real_escape_string($dbc, trim($_FILES['screenshot']['name']));
        $screenshot_type = $_FILES['screenshot']['type'];
        $screenshot_size = $_FILES['screenshot']['size'];

        // Verifica a senha CAPTCHA
        $user_pass_phrase = sha1($_POST['verify']);
        if ($_SESSION['pass_phrase'] == $user_pass_phrase) {
            ...
        } else {
            echo '<p class="error">Por favor, digite o código de verificação exatamente como  
mostrado </p>';
        }
    }
?>

<hr />
<form enctype="multipart/form-data" method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
    <input type="hidden" name="MAX_FILE_SIZE" value="<?php echo GW_MAXFILESIZE; ?>" />
    <label for="name">Nome: </label>
    <input type="text" id="name" name="name" value="<?php if (!empty($name)) echo $name; ?>" /><br />
    <label for="score">Pontuação: </label>
    <input type="text" id="score" name="score" value="<?php if (!empty($score)) echo  
$score; ?>" /><br />
    <label for="screenshot">Captura de tela: </label>
    <input type="file" id="screenshot" name="screenshot" /><br />
    <label for="verify">Verificação: </label>
    <input type="text" id="verify" name="verify" value="Enter the pass-phrase." />
    
    <hr />
    <input type="submit" value="Add" name="submit" />
</form>
</body>
</html>

```

4 Verificar a senha em relação ao input do usuário.

Tudo pronto!

É aqui que a senha criptografada é lida, a partir de uma variável de sessão e verificada para ver se o usuário digitou corretamente.

É aqui que o script CAPTCHA foi "ligado" ao script Adicionar Pontuação, no Passo 3, o que resulta na imagem CAPTCHA ser exibida na página.

teste drive addscore.php com a funcionalidade captcha



## TEST DRIVE

### Modifique o script Adicionar Pontuação para implementar o suporte a CAPTCHA.

Modifique o script addscore.php de modo que ele passe a ter um campo Verificação, bem como a usar o script captcha.php para exibir uma imagem CAPTCHA. Adicione também o código para verificar se o usuário digitou a senha correta antes de adicionar a pontuação.

Envie ambos os scripts ao seu servidor web, e depois, abra addscore.php em um navegador. Tente adicionar uma nova pontuação sem digitar a senha CAPTCHA. Depois tente novamente, agora digitando a senha mostrada na imagem CAPTCHA.

Guitar Wars - Add Your High Score

Please enter the verification pass-phrase exactly as shown.

Name: www.hackertabs.com  
Score: 99999999  
Screen shot: Choose File no file selected  
Verification: Enter the pass-phrase.

**Add**

Guitar Wars - Add Your High Score

Please enter the verification pass-phrase exactly as shown.

Name: www.hackertabs.com  
Score: 99999999  
Screen shot: Choose File no file selected  
Verification: Enter the pass-phrase.

**Add**

Guitar Wars - Add Your High Score

Please enter the verification pass-phrase exactly as shown.

Name: www.classshates.com  
Score: 99999999  
Screen shot: Choose File no file selected  
Verification: Enter the pass-phrase.

**Add**

Uma senha CAPTCHA que se modifica constantemente dificulta aos bots a tarefa de fazer spam no formulário do Guitar Wars.

Ah, como é bom ser humano.

O nosso moderador humano finalmente tem paz, graças a um pouco da nossa própria automação!

nao existem  
**Perguntas Ídiotas**

**P:** Posso usar as funções GD para criar imagens em outros formatos além de PNG?

**R:** Sim. As funções `imagegif()` e `imagejpeg()` funcionam de modo bastante semelhante a `imagepng()`, mas criam imagens GIF e JPEG, respectivamente.

**P:** As funções de criação de imagens podem criar imagens com transparência?

**R:** Sim! Existe uma função chamada `imagecolortransparent()` que define uma cor para ser a cor transparente dentro de uma imagem. Essa precisa ser uma cor que você já tenha criado com a função `imagecolorallocate()`. Após defini-la como transparente, qualquer coisa desenhada nessa cor será considerada como transparente. Para

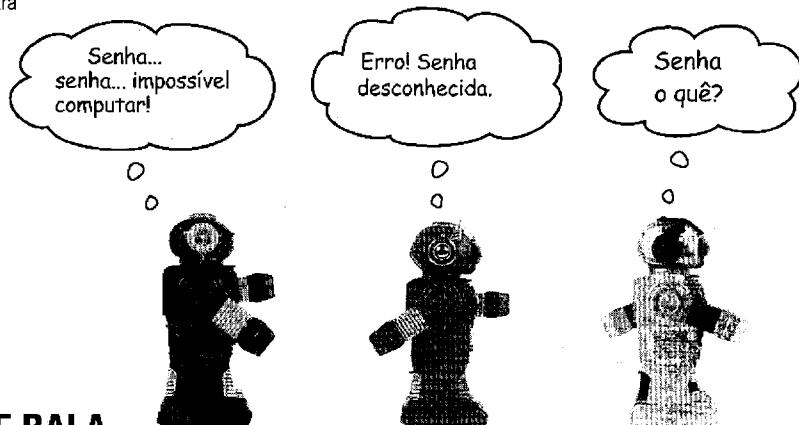
gerar a imagem com transparência, chame `imagegif()` ou `imagepng()`; você não pode usar `imagejpeg()` porque as imagens JPEG não têm suporte ao recurso de transparência.

**P:** Quando usamos `imagepng()` para enviar uma imagem PNG diretamente ao navegador do cliente, onde o arquivo .png da imagem é armazenado, e qual é o seu nome?

**R:** Não há nenhum arquivo .png para a imagem, pelo simples motivo que ela não é armazenada em um arquivo. Em vez disso, a função `imagepng()` gera uma imagem PNG binária na memória, no servidor, e então a envia diretamente ao navegador através de um cabeçalho (por meio da função `imagepng()`), não se usa um arquivo. E o navegador sabe que deve conectar a imagem do cabeçalho à tag `<img>` porque o script está especificado no atributo `src`.

**P:** É por isso que eu consigo colocar o nome do script CAPTCHA diretamente no atributo `src` de uma tag `<img>`?

**R:** Correto. Referenciar um script PHP no atributo `src` de uma tag `<img>`, como fizemos no script `captcha.php` para o Guitar Wars, resulta na imagem ser entregue diretamente pelo script. Isso é diferente do modo normal como a tag `<img>` funciona, onde o nome de um arquivo gráfico é especificado no atributo `src`. Uma vez que o script está enviando a imagem diretamente para o navegador através de um cabeçalho (por meio da função `imagepng()`), não se usa um arquivo. E o navegador sabe que deve conectar a imagem do cabeçalho à tag `<img>` porque o script está especificado no atributo `src`.



## PONTOS DE BALA

- Todo formulário web corre o risco de ser atacado por bots de spam, mas todo bot de spam está a mercê de ser combatido por programadores PHP inteligentes, que usam técnicas como CAPTCHA para derrotá-los.
- GD é uma biblioteca de gráficos padrão do PHP, que lhe permite criar dinamicamente imagens e desenhar todo tipo de gráficos ou textos nelas.

- A função GD `createtrcolorimage()` é usada para se criar uma imagem em branco.
- Para enviar uma imagem PNG ao navegador ou a um arquivo no servidor, chame a função GD `imagepng()`.
- Ao terminar de trabalhar com uma imagem, chame `imagedestroy()` para apagá-la da memória.

## Cinco graus de oposição

Uma vez que o Mismatch é uma comunidade de usuários (humanos!) registrados, bots de spam não têm sido problema. Porém, os usuários querem um pouco mais do recurso de encontrar pares imperfeitos do site, principalmente o conceito de "cinco graus de oposição" de que têm ouvido falar. Os usuários do Mismatch querem mais do que apenas uma lista de tópicos do seu par imperfeito ideal – eles querem algum tipo de contextualização visual de como esses tópicos se relacionam com cada categoria principal de "desencontrabilidade"

O conceito de "cinco graus de oposição" do Mismatch envolve a avaliação dos tópicos desencontrados por categoria.

Eu vejo um monte de tópicos, mas realmente não sei exatamente como nós nos desencontramos nas diferentes categorias. Eu gosto do conceito de "cinco graus de oposição", mas não consigo ver como isso se relaciona ao meu par imperfeito. E agora?

- Belita é uma pessoa visual, e quer ver mais do que apenas uma lista de tópicos referentes ao seu par imperfeito ideal.



Mismatch - My Mismatch

Home ♥ View Profile ♥ Edit Profile ♥ Questionnaire ♥ My Mismatch ♥ Log Out (belita)

Jason Filmington  
Hollywood, CA

You are mismatched on the following 18 topics:

- Tattoos
- Cowboy boots
- Long hair
- Reality TV
- Horror movies
- Easy listening music
- The opera
- Sushi
- Spicy food
- Peanut butter & banana sandwiches
- Martini
- Bill Gates
- Hugh Hefner
- Yoga
- Weightlifting
- Cube puzzles
- Karaoke
- Hiking

[View Jason's profile.](#)

Copyright ©2006 Mismatch Enterprises, Inc.

A lista de tópicos desencontrados é interessante, mas o que os usuários realmente querem é uma perspectiva mais visual sobre como eles se desencontram uns dos outros.

# Pondo a desencontrabilidade em tabela

Como você deve se lembrar, o Mismatch inclui um questionário categorizado, onde os usuários selecionam as opções Adoro ou Odeio para uma série de tópicos. São essas respostas que determinam os tópicos que compõem um par imperfeito ideal. Ao apresentar o par imperfeito ideal para o usuário, o script My Mismatch exibe uma lista de tópicos desencontrados, que ele cria como um array a partir do banco de dados do Mismatch. Mas os usuários agora querem mais do que uma lista de tópicos... eles querem uma representação visual categorizada das suas "desencontrabilidades", talvez em forma de gráfico.

A screenshot of a computer application window titled "Projeto". It displays a list of topics under several categories. The categories include "Música", "Atividades", "Personagens", and "Atrizes". Each topic has two options: "Adoro" (radio button) and "Odeio" (radio button). The topics listed are:

Categoria	Opção	Outra Opção
Música	Adoro	Odeio
Atividades	Adoro	Odeio
Personagens	Adoro	Odeio
Atrizes	Adoro	Odeio



## Exercício

Desenhe um gráfico de barras, usando os dados do Mismatch, que mostre visualmente os "cinco graus de oposição" para Belita e Jason. Escreva o que as informações do gráfico significam.

A vertical list of topics grouped into categories, with arrows pointing from the "Topicos" section of the screenshot above to this list. The categories and topics are:

Categoria	Topicos
Tatuagens	Tattoos
Botas de cowboy	Cowboy boots
Cabelo longo	Long hair
TV realidade	Reality TV
Filmes de terror	Horror movies
Música fácil de escutar	Easy listening music
Ópera	The opera
Sushi	Sushi
Alimentos picantes	Spicy food
Peanut butter & banana sandwiches	Peanut butter & banana sandwiches
Martini	Martinis
Bill Gates	Bill Gates
Hugh Hefner	Hugh Hefner
Yoga	Yoga
Weightlifting	Weightlifting
Cube puzzles	Cube puzzles
Karaoke	Karaoke
Hiking	Hiking

Precisamos transformar, de algum modo, esta lista de tópicos em um gráfico de barras contendo as categorias.

## gravando dados gráficos no array

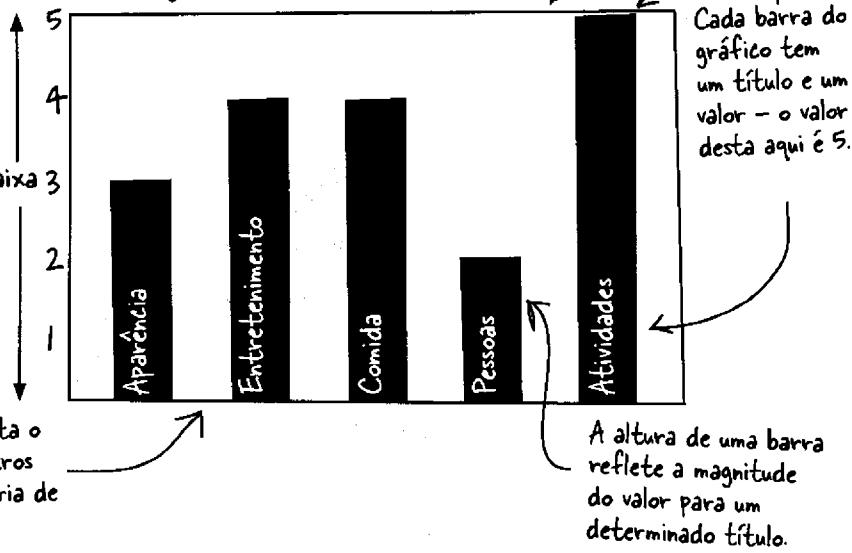


### Exercício Solução

A faixa de valores de um gráfico de barras estabelece os valores possíveis para cada barra.

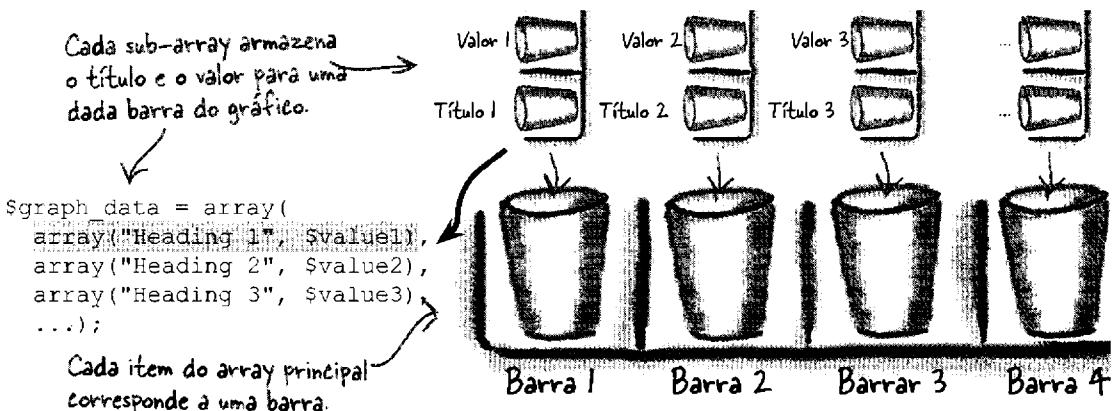
Desenhe um gráfico de barras, usando os dados do Mismatch, que mostre visualmente os "cinco graus de oposição" para Belita e Jason. Escreva o que as informações do gráfico significam.

Embora haja muitas formas diferentes de representar visualmente os dados do Mismatch, um gráfico de barras não é uma má opção, uma vez que as categorias têm, cada uma, um número igual de tópicos.



## Armazenando dados do gráfico de barras

No fim das contas, os dados por trás de um gráfico de barras são talvez mais importantes do que os gráficos. Sabendo que esse tipo de gráfico na verdade é apenas um conjunto de títulos e valores, podemos visualizar os dados de um gráfico de barras com um array bidimensional, em que o array principal armazena as barras, enquanto que cada sub-array armazena o par título/valor para cada barra.



não existem

## Perguntas Ídiotas

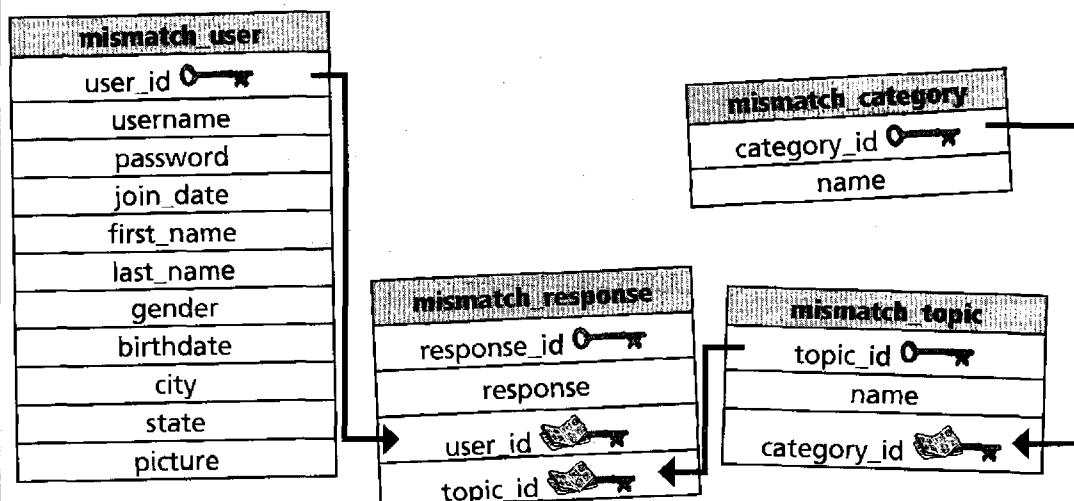
**Q:** O gráfico de barras precisa ser alimentado a partir de um array bidimensional?

**A:** Não, não necessariamente. Mas tenha em mente que cada barra do gráfico geralmente envolve duas informações: um título e um valor. E cada gráfico contém várias barras, de modo que um array bidimensional é uma forma lógica e eficiente de se armazenar dados usados para preencher um gráfico de barras. Como diz o ditado, "se você só consegue ver uma solução, você

não entendeu realmente o problema". Neste caso, o problema é achar a melhor forma de armazenar os dados a serem injetados em um gráfico de barras, e uma solução que funciona bem é o array bidimensional. É claro que o desafio ainda é como exatamente criar esse array bidimensional com dados das categorias do Mismatch. O primeiro passo é isoler quais dados do banco entram na solução.



O esquema do banco de dados da aplicação Mismatch é mostrado abaixo. Circule todos os dados que precisam ser usados para a geração dinâmica do gráfico de barras "cinco graus de oposição", e não se esqueça de escrever como eles são usados para se criar o gráfico





## Exercício Solução

O esquema do banco de dados da aplicação Mismatch é mostrado abaixo. Circule todos os dados que precisam ser usados para a geração dinâmica do gráfico de barras "cinco graus de oposição", e não se esqueça de escrever como eles são usados para se criar o gráfico..

A coluna `user_id` é usada para se consultar respostas do questionário, para que se possa determinar o melhor par imperfeito para um usuário.

A coluna `response_id` é usada para se combinar as respostas de dois usuários, para determinar se elas estão definidas com valores opostos - um desencontro!

A coluna `category_id` é usada para se associar uma categoria com um tópico, o que leva a uma resposta desencontrada. O nome da categoria é importante porque é ele que fornece os títulos para o gráfico de barras.

mismatch_user
<code>user_id</code>
<code>username</code>
<code>password</code>
<code>join_date</code>
<code>first_name</code>
<code>last_name</code>
<code>gender</code>
<code>birthdate</code>
<code>city</code>
<code>state</code>
<code>picture</code>

mismatch_response
<code>response_id</code>
<code>response</code>
<code>user_id</code>
<code>topic_id</code>

mismatch_topic
<code>topic_id</code>
<code>name</code>
<code>category_id</code>

A coluna `topic_id` serve como intermediário para as categorias e as respostas, que é o que permite a você descobrir a categoria de cada resposta desencontrada.

O gráfico de barras requer, em última instância, uma contagem de quantos desencontros existem para cada categoria; a contagem é o valor para cada barra, enquanto que o nome da categoria é o seu título.



## Gráfico de Barras Exposto

**Na entrevista desta semana:**  
**Lendo entre as linhas com o senhor**  
**dos gráficos**

**Use a Cabeça:** Então você é o cara que as pessoas chamam quando elas precisam de uma representação visual de alguns dados. É isso mesmo?

**Gráfico de Barras:** Exato. Eu sou expert em todos os tipos de visualização de dados, principalmente no tipo retangular.

**Use a Cabeça!:** Então as suas capacidades de desenho se limitam principalmente a retângulos

**Gráfico de Barras:** Acho que "se limitam" é uma expressão forte neste caso. É uma daquelas situações em que quanto mais simples, melhor – as pessoas simplesmente parecem mais acostumadas com as barras, talvez porque elas estejam acostumadas a ver as coisas medidas dessa forma. É como aquela barrinha nos celulares que lhe diz o quanto forte está o sinal. "Está conseguindo me ouvir agora?" Eu adoro isso.

**Use a Cabeça!:** Certo. Mas eu já vi alguns gráficos redondos bem eficientes. Me lembram de coisas boas... como torta de maçã, entende o que eu quero dizer?

**Gráfico de Barras:** Entendi o que você está insinuando, e conheço o Gráfico Torta. Veja, são duas maneiras diferentes de se pensar sobre a mesma coisa. O Gráfico Torta vê o mundo em curvas; eu vejo um pouco mais reto, só isso.

**Use a Cabeça!:** Mas as pessoas não se relacionam inherentemente melhor com uma torta do que com um monte de barras?

**Gráfico de Barras:** Não, a não ser que elas estejam com fome. Veja, o Gráfico Torta é realmente bom para mostrar as partes que compõem o todo, onde os dados compõem algo que realmente importa: 100%, 32 times, 50 estados. São 50 estados nos EUA, certo?

**Use a Cabeça!:** Sim. Bem, assumindo-se que você conte Washington, D.C. como "distrito federal" e lugares como Porto Rico e Guam como "territórios". Mas, de qualquer forma, entendo o que você está dizendo sobre o Gráfico Torta ser mais adequado para revelar partes de um todo, mas você não faz a mesma coisa?

**Gráfico de Barras:** Sim, mas tenha em mente que eu sou muito mais flexível do que o Gráfico Torta. Você pode me adicionar quantas barras quiser, e eu não terei problema nenhuma para exibi-las.

No caso do Gráfico Torta, quanto mais coisas você adiciona, menores as fatias ficam. Chega um ponto em que as partes ficam difíceis de visualizar. No meu caso, tudo o que importa é que as barras têm valores que podem ser mostrados na mesma escala.

**Use a Cabeça!:** O que isso significa?

**Gráfico de Barras:** Bem, eu tenho dificuldades em mostrar coisas que tenham valores altamente diferentes – a não ser, é claro, que você não se importe que as barras sejam altamente diferentes. O meu ponto forte é mostrar a diferença entre valores que estejam dentro da mesma faixa. Por exemplo, talvez você queira que eu mostre o preço da gasolina ao longo de um período de um ano, em cujo caso, todos os valores estariam dentro de uma faixa razoavelmente restrita, por exemplo, com apenas alguns reais de diferença uns para os outros.

**Use a Cabeça!:** Tem certeza disso?

**Gráfico de Barras:** Eu sei, o preço da gasolina parece ser um valor que varia sem limites, mas na verdade não é tão ilimitado assim, dentro do tipo de situações com que eu lido.

**Use a Cabeça!:** Então você está acostumado a ver coisas bem esquisitas, não?

**Gráfico de Barras:** Você não acreditaria. Uma vez, um cara criou uma aplicação web que mantinha registro de quantas milhas ele arrastava o seu mouse por mês. Ele tinha até um blog sobre isso, e me usava para representar as suas "viagens" em forma de gráfico. Bem louco, mas as pessoas adoraram esse tipo de coisa.

**Use a Cabeça!:** Então esse é o seu campo de atuação – fornecer representações visuais dos dados das pessoas?

**Gráfico de Barras:** Sim, creio que sim. Sempre que eu posso ser colocado em uma página para fornecer um visual extra para dados que, sem mim, ficariam um pouco chatos e difíceis de entender, considero meu trabalho feito.

**Use a Cabeça!:** Fico feliz de ouvi-lo. Obrigado por conversar conosco, espero que possamos fazer isto novamente.

**Gráfico de Barras:** O prazer foi todo meu. E não se preocupe, você continuará me vendendo por aí.

## construindo um array para categorias

### De um array para outro

Da última vez que vimos o Mismatch, tínhamos uma lista de tópicos que correspondiam a desencontros entre dois usuários. Mais especificamente, nós tínhamos na verdade um array de tópicos. O problema é que o gráfico de barras que estamos tentando desenhar não é exatamente sobre os tópicos – é sobre as categorias associadas com os tópicos. Assim, estamos separados em um nível dos dados que realmente precisamos. Parece que precisamos de mais algumas consultas SQL. Não apenas precisamos do array de tópicos desencontrados, mas precisamos também de um array paralelo semelhante, de categorias desencontradas.

Um alias é usado para se eliminar a confusão ao se referir à coluna name da tabela mismatch\_category.

Um join múltiplo

conecta a tabela das categorias à tabela das respostas, de modo que o nome da categoria possa ser extraído.

```
$query = "SELECT mr.response_id, mr.topic_id, mr.response,
    mt.name AS topic_name, mc.name AS category_name
    FROM mismatch_response AS mr "
    . "INNER JOIN mismatch_topic AS mt USING (topic_id) "
    . "INNER JOIN mismatch_category AS mc USING (category_id) "
    . "WHERE mr.user_id = '" . $_SESSION['user_id'] . "'";
```

O join adicional nessa consulta faz o nome da categoria correspondente a cada tópico de resposta ser anexado aos dados do resultado, acabando no array \$user\_responses.

Mas lembre-se, nós só precisamos das categorias desencontradas, e não de todas as categorias. Precisamos criar outro array, contendo apenas as categorias desencontradas para as respostas.

Nós ainda precisamos de um novo array, contendo apenas as categorias desencontradas para este par de usuários.

A nova "coluna" de dados de resultado armazena o nome da categoria de cada resposta.

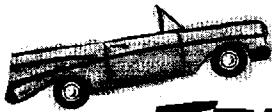
Appearance
Entertainment
Food
People
People
People
People

Nós precisamos extrair apenas os nomes das categorias referentes a respostas desencontradas, e colocá-los em um array.

topic_name	category_name
	Appearance
	Entertainment
	Food
	People
	People
	People
	People

Lá no Capítulo 8, o array bidimensional \$user\_responses foi criado e preenchido com dados de resultado correspondentes às respostas do usuário atual.

Mas ainda não estamos atingindo o nosso objetivo de criar um array de categorias desencontradas. Para fazer isso, precisamos revisitar o código que cria o array de tópicos desencontrados...



## TEST DRIVE

### Teste a nova consulta para obter tópicos e categorias desencontrados.

Usando uma ferramenta MySQL, emita a seguinte consulta para selecionar tópicos e categorias desencontrados para um determinado usuário. Certifique-se de especificar a ID de um usuário que não só exista no banco de dados, mas que tenha também preenchido o questionário do Mismatch:

```
SELECT mr.response_id, mr.topic_id, mr.response,
       mt.name AS topic_name, mc.name AS category_name
    FROM mismatch_response AS mr
   INNER JOIN mismatch_topic AS mt USING (topic_id)
   INNER JOIN mismatch_category AS mc USING (category_id)
   WHERE mr.user_id = 3;
```

A ID deve ser a de um usuário válido, que tenha respondido o questionário do Mismatch.

```
File Edit Window Help Oppose
mysql> SELECT mr.response_id, mr.topic_id, mr.response,
       mt.name AS topic_name, mc.name AS category_name
    FROM mismatch_response AS mr
   INNER JOIN mismatch_topic AS mt USING (topic_id)
   INNER JOIN mismatch_category AS mc USING (category_id)
   WHERE mr.user_id = 3;
+-----+-----+-----+-----+
| response_id | topic_id | response | topic_name | category_name |
+-----+-----+-----+-----+
| 26 | 1 | 1 | Tattoos | Appearance |
| 27 | 2 | 2 | Gold chains | Appearance |
| 28 | 3 | 1 | Body piercings | Appearance |
| 29 | 4 | 2 | Cowboy boots | Appearance |
| 30 | 5 | 1 | Long hair | Appearance |
| 31 | 6 | 2 | Reality TV | Entertainment |
| 32 | 7 | 1 | Professional wrestling | Entertainment |
| 33 | 8 | 1 | Horror movies | Entertainment |
| 34 | 9 | 2 | Easy listening music | Entertainment |
```

Rpare que os resultados desta consulta batem com o array `fuser_responses` da página anterior, que é o que nós queremos.

É sempre uma boa ideia testar as consultas em uma ferramenta MySQL antes de colocá-las no código PHP.

## Crie um array de tópicos desencontrados

Agora nós temos uma consulta que realiza um join múltiplo para obter a categoria de cada resposta, além do tópico, que é então colocada no array \$user\_responses. Lembre-se que outra consulta, parecida com esta, também está obtendo dados para cada outro usuário do banco, para que as comparações de desencontrabilidade possam ser feitas. Assim, \$user\_responses armazena os dados de resposta para o usuário logado no Mismatch, enquanto que \$mismatch\_responses armazena um dos outros usuários do sistema. Isso nos permite fazer um loop através de todos os usuários, e atualizar \$mismatch\_responses em cada comparação.

Nós já estamos usando esses dois arrays para encontrar pares imperfeitos e para criar um array de tópicos desencontrados. Agora, podemos adicionar uma nova linha de código para construir também um array de categorias desencontradas – este array conterá a categoria de cada tópico desencontrado entre dois usuários.

Este é o mesmo código da versão anterior do Mismatch,

exceto pelo fato de que agora ele cria um array de categorias desencontradas, além do array de tópicos.

```
$categories = array();
for ($i = 0; $i < count($user_responses); $i++) {
    if ($user_responses[$i]['response'] + $mismatch_responses[$i]['response']
== 3) {
        $score += 1;
        array_push($topics, $user_responses[$i]['topic name']);
        array_push($categories, $user_responses[$i]['category name']);
    }
}
```

Este código resulta em um array contendo apenas as categorias desencontradas.

No fim das contas, o array \$categories contém uma categoria para cada desencontro.

Um array de categorias desencontradas é criado armazenando-se a categoria associada com cada resposta desencontrada..

## não existem Perguntas Ídiotas

**P:** Estou um pouco confuso. Qual é a diferença entre um conjunto de resultados do MySQL e um array do PHP?

**R:** Uma grande diferença é o acesso. Um conjunto de resultados só disponibiliza uma linha de dados de cada vez, enquanto que um array pode armazenar várias "linhas" de dados, graças às múltiplas dimensões. Colocar um conjunto de resultados em um array bidimensional permite que nos movamos através das linhas de dados de forma eficiente, sem termos que ficar voltando ao servidor do banco de dados para obter linhas várias vezes. Isso não funciona com conjuntos enormes de dados, uma vez que para isso você precisaria criar arrays enormes; mas, no caso das respostas do Mismatch, os arrays nunca são maiores do que o número total de tópicos no sistema.

**P:** Mas nós ainda precisamos contar quantas vezes uma categoria foi desencontrada para podermos gerar o gráfico de barras, certo?

**R:** Certo. Um array de categorias desencontradas não é suficiente. Lembre-se de que a ideia por trás do gráfico de barras do Mismatch é que cada barra representa uma categoria desencontrada, e que a altura da barra representa quantas vezes a categoria foi desencontrada. Assim, precisamos de uma forma de contar quantas vezes cada categoria foi desencontrada. Mas provavelmente vale a pena dar um passo atrás, para formularmos um plano geral de ataque.....

# Formulando um plano para execução dos gráficos de barras

Com um array de categorias desencontradas e um monte de ideias sobre como usá-las para gerar um gráfico de barras para a página My Mismatch, só o que nos falta é um plano. Como veremos logo abaixo, só são necessários três passos para se gerar dinamicamente o gráfico de barras, e já realizamos um deles.

## 1 Consultar o banco de dados do Mismatch em busca dos nomes das categorias desencontradas

Este passo nos fornece a lista das categorias desencontradas.

## 2 Calcular os totais de desencontros para cada categoria.

A lista de categorias precisa ser convertida em uma lista de totais.

## 3 Desenhar o gráfico de barras usando-se os totais das categorias.

Tendo os totais das categorias, nós podemos passar para a parte divertida, que é desenhar o gráfico de barras com funções GD.

Para completar o Passo 2 do nosso plano, precisamos de alguma forma de pegar o array de categorias desencontradas e transformá-lo em um conjunto de totais para as categorias, ou seja, uma contagem de quantas vezes cada categoria aparece no array. Como você deve se lembrar, esse é, precisamente, o tipo de dados requeridos para se criar um gráfico de barras, onde as categorias são os títulos e a contagem para cada categoria é o valor de cada barra. Um array bidimensional pode ser usado para se combinar categorias e totais em uma única estrutura de dados.



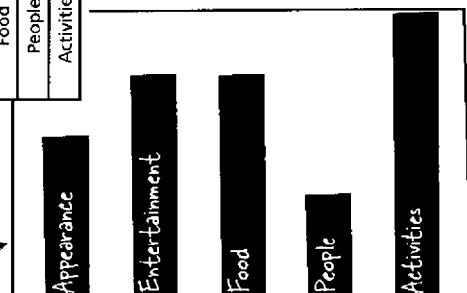
Appearance	3
Entertainment	4
Food	4
People	2
Activities	5

Este array contém o nome de cada categoria, junto com o total de vezes que ela foi desencontrada.

Girando!

3	4	4	2	5
Appearance	Entertainment	Food	People	Activities

Os novos dados das categorias desencontradas são exatamente o que precisamos para o gráfico de barras.



Olhar o novo array das categorias por um ângulo diferente revela como ele é usado para fornecer os dados ao gráfico de barras.

Uma vez criado esse array de totais, estaremos prontos para seguirmos para o Passo 3, onde usaremos algumas funções GD para gerar o visual do gráfico de barras.

*um pouco de matemática de array*

## Amassando as categorias

O desafio agora é totalizar o array de categorias e colocá-lo em um array bidimensional de títulos e valores. Nós temos **um array de categorias desencontradas** armazenado em `$categories`. Precisamos criar um novo array chamado `$category_totals`, que irá conter uma entrada para cada categoria, junto com o número de desencontros para cada uma.

**Precisamos  
partir disto...**

Appearance
Appearance
Appearance
Entertainment
Entertainment
Entertainment
Food
Food
Food
Food
People
People
Activities

`$categories`

O número total de ocorrências de uma categoria no array `$categories` aparece como um total no array `$category_totals`.

**...e chegar a isto!**

Appearance	3
Entertainment	4
Food	4
People	2
Activities	5

`$category_totals`



Como você faria para totalizar as categorias desencontradas no array `$categories`, para criar o array bidimensional `$category_totals`?

## A matemática das categorias

Passar de um array unidimensional de categorias desencontradas para um array bidimensional de totais das categorias pode ser um pouco mais complicado do que você poderia pensar, à primeira vista. Por esse motivo, é útil trabalhar na solução através de pseudocódigo antes de realmente escrever qualquer PHP. O pseudocódigo nos livra dos detalhes sintáticos e nos permite focalizar nas ideias centrais envolvidas em uma determinada solução de codificação.

**Criar um novo array bidimensional para armazenar os totais das categorias, certificando-se de inicializar o primeiro elemento com a primeira categoria desencontrada e uma contagem 0.**

**Fazer loop através do array de categorias desencontradas. Para cada categoria do array...**

**O último elemento do array contendo os totais é de uma categoria diferente do desencontro atual?**

- ◆ Sim! Temos uma nova categoria, portanto vamos adicioná-la ao array com os totais das categorias e inicializar a sua contagem como 0.

- ◆ **Não. Esta é apenas uma nova instância da categoria atual, portanto vamos incrementar a contagem do último elemento do array com os totais das categorias.**

O produto deste código é um array bidimensional de totais, no qual o array principal armazena uma única categoria, enquanto que cada sub-array contém o nome da categoria e o seu valor.



## **Exercício**

Traduza o pseudocódigo para completar o código PHP real que cria o array bidimensional de dados de categorias do Mismatch, chamado \$category\_totals.

```
$category_totals = array(array($mismatch_categories[0], 0));  
  
foreach ($mismatch_categories as $category) {
```

## exercício solução



### Exercício Solução

Tradua o pseudocódigo para completar o código PHP real que cria o array bidimensional de dados de categorias do Mismatch, chamado \$category\_totals.

Os arrays são indexados a partir do zero, portanto o último elemento de um array é sempre count() menos um.

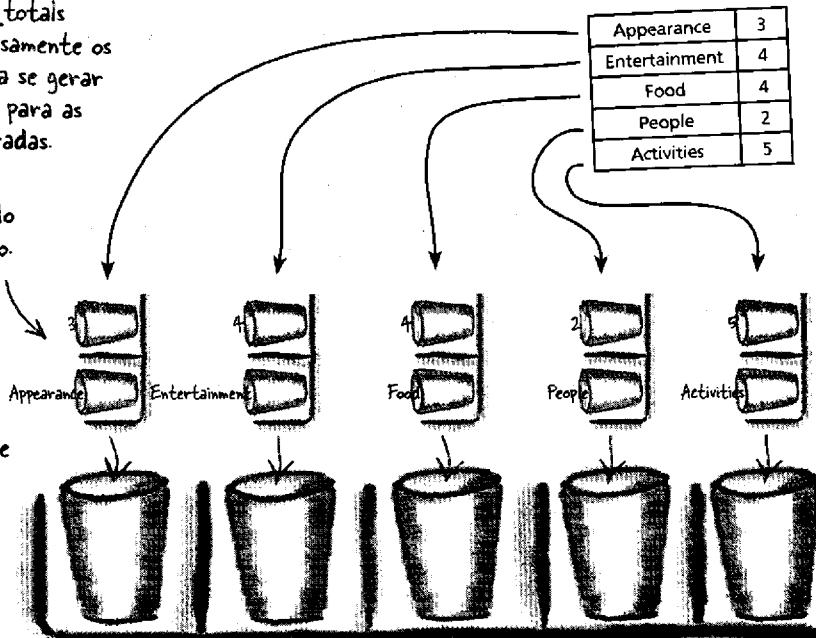
```
$category_totals = array(array($mismatch_categories[0], 0));  
  
foreach ($mismatch_categories as $category) {  
    if (!category_totals[count(category_totals) - 1][0] == $category) {  
        array_push(category_totals, array($category, 1));  
    }  
    else {  
        category_totals[count(category_totals) - 1][1]++;  
    }  
}
```

**\$category\_totals**

A variável `category_totals` agora armazena precisamente os dados necessários para se gerar um gráfico de barras para as categorias desencontradas.

Este é o resultado final deste código.

Agora nós podemos riscar este passo da lista, o que nos deixa apenas com a tarefa de desenhar o gráfico de barras.



- 3 Calcular os totais de desencontros para cada categoria

não existem  
**Perguntas Ídiotas**

**P:** O que acontece com o código de totalização se as categorias não estiverem em ordem no array \$mismatch\_categories?

**R:** Grande problema. O código é inteiramente dependente das categorias estarem em ordem no array \$mismatch\_categories. Isso se vê no modo como o código assume que qualquer modificação na categoria é o início de uma nova categoria, o que funciona desde que as categorias estejam agrupadas. Felizmente, a consulta no script do questionário, que seleciona originalmente os tópicos para inserção na tabela mismatch\_response, é inteligente o suficiente para ordenar as respostas por categoria.

`SELECT topic_id FROM mismatch_topic ORDER BY category_id, topic_id`

É esta consulta que obtém primeiramente os tópicos do banco de dados, e depois os insere como respostas vazias para um determinado usuário. Isso garante que as respostas do usuário sejam armazenadas no banco ordenadas por categoria, o que permite que o código totalizador das categorias funcione corretamente.

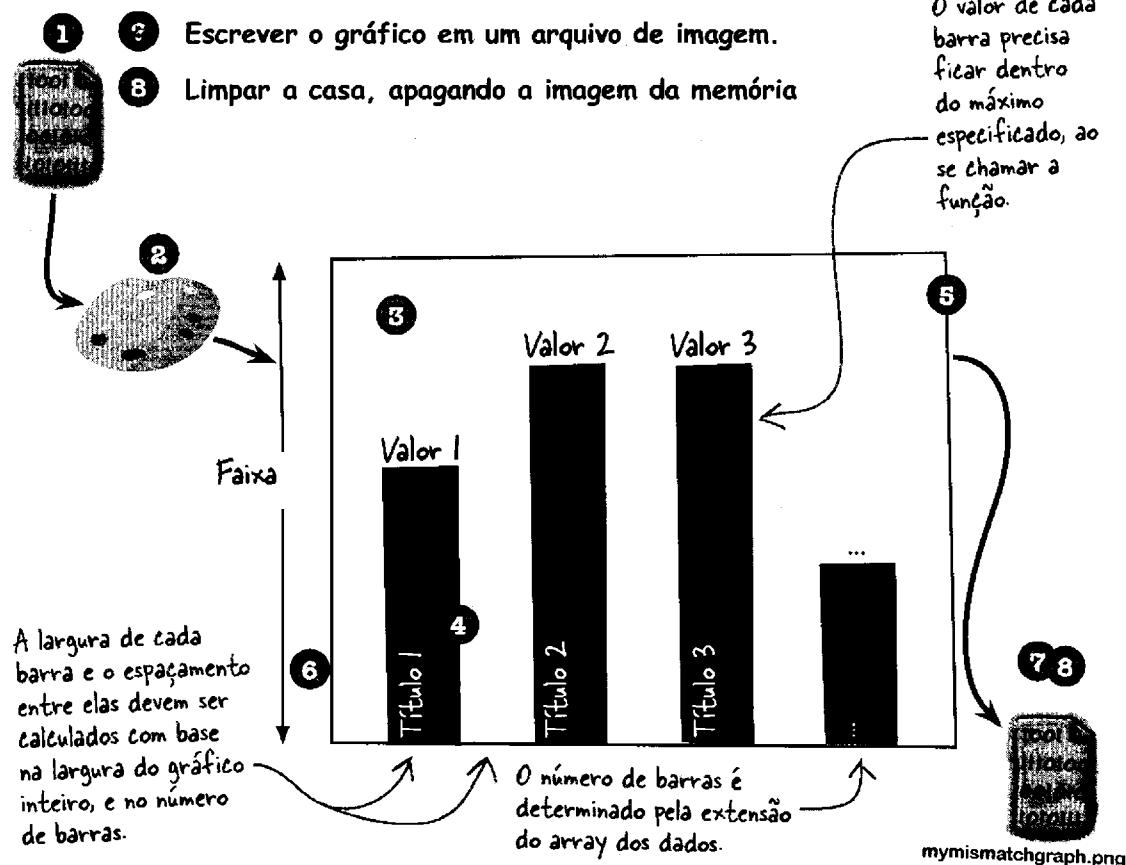
**P:** Mas não é arriscado escrever código que dependa da ordem dos dados armazenados em uma tabela do banco?

**R:** Sim e não. Lembre-se de que este banco de dados é controlado inteiramente através de código que você escreve, portanto a ordem dos dados na verdade só se modifica se você escrever algum código que a modifique. Mesmo assim, certamente se poderia defender que é uma boa ideia ordenar a consulta de join, no script My Mismatch, por categoria, para termos certeza absoluta que a lista das categorias desencontradas estará na ordem certa.

## Fundamentos dos gráficos de barras

Com um array bidimensional novinho, contendo os dados das categorias desencontradas, é hora de passarmos à tarefa de desenhar o gráfico de barras. Mas em vez de nos concentrarmos nos detalhes de como desenhar o gráfico do Mismatch, por que não usarmos uma abordagem mais genérica? Se você elaborar e criar uma função geral para desenho de gráficos de barra, é possível usá-la para o Mismatch e tê-la à disposição para qualquer necessidade futura de desenho de gráficos. Em outras palavras, ela é reutilizável. Esta nova função terá de realizar uma série de passos para gerar com sucesso o gráfico de barras a partir de um array de dados bidimensional.

- 1 Criar a imagem.
- 2 Criar as cores que você irá usar para desenhar os gráficos e o texto.
- 3 Preencher o fundo com alguma cor.
- 4 Desenhar as barras e os títulos.
- 5 Desenhar um retângulo em torno do gráfico de barras inteiro
- 6 Desenhar a faixa dos valores no lado esquerdo do gráfico.
- 7 Escrever o gráfico em um arquivo de imagem.
- 8 Limpar a casa, apagando a imagem da memória





## Ímãs de Geladeira do PHP

O script My Mismatch contém uma nova função, `draw_bar_graph()`, que se encarrega de desenhar um gráfico de barras tendo uma largura, uma altura, um array bidimensional contendo os dados para o gráfico, um valor máximo para a faixa e um nome de arquivo para a imagem PNG resultante. Use os ímãs para completar as chamadas às funções GD de desenho que estão faltando.

```

function draw_bar_graph($width, $height, $data, $max_value, $filename) {
    // Cria a imagem vazia
    $img = ..... ($width, $height);

    // Define um fundo branco com texto preto e gráficos cinza
    $bg_color = ..... ($img, 255, 255, 255); // branco
    $text_color = ..... ($img, 255, 255, 255); // branco
    $bar_color = ..... ($img, 0, 0, 0); // preto
    $border_color = ..... ($img, 192, 192, 192); // cinza claro

    // Preenche o fundo
    ..... ($img, 0, 0, $width, $height, $bg_color);

    // Desenha as barras
    $bar_width = $width / ((count($data) * 2) + 1);
    for ($i = 0; $i < count($data); $i++) {
        ..... ($img, ($i * $bar_width * 2) + $bar_width, $height,
               ($i * $bar_width * 2) + ($bar_width * 2), $height - (($height / $max_value) *
               $data[$i][1]), $bar_color);
        ..... ($img, 5, ($i * $bar_width * 2) + ($bar_width), $height - 5,
               $data[$i][0],
               $text_color);
    }

    // Desenha um retângulo em torno da imagem inteira
    ..... ($img, 0, 0, $width - 1, $height - 1, $border_color);

    // Desenha a faixa de valores no lado esquerdo do gráfico
    for ($i = 1; $i <= $max_value; $i++) {
        ..... ($img, 5, 0, $height - ($i * ($height / $max_value)), $i, $bar_
               color);
    }

    // Escreve a imagem em um arquivo
    ..... ($img, $filename, 5);
    ..... ($img);
}

```

`imagecreatetruecolor`

`imagestringup`

`imagerectangle`

`imagecolorallocate`

`imagefilledrectangle`

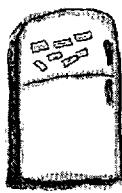
`imagecolorallocate`

`imagepng`

`imagestring`

`imagecolorallocate`

`imagecolorallocate`



## Ímãs de Geladeira do PHP - Solução

O script My Mismatch contém uma nova função, `draw_bar_graph()`, que se encarrega de desenhar um gráfico de barras tendo uma largura, uma altura, um array bidimensional contendo os dados para o gráfico, um valor máximo para a faixa e um nome de arquivo para a imagem PNG resultante. Use os ímãs para completar as chamadas às funções GD de desenho que estão faltando.

```

1 function draw_bar_graph($width, $height, $data, $max_value, $filename) {
    // Cria a imagem vazia
    $img = imagecreatetruecolor($width, $height);
    // Em primeiro lugar, crie
    // uma imagem em branco.

2    // Define um fundo branco com texto preto e gráficos cinza
    $bg_color = imagecolorallocate($img, 255, 255, 255); // branco
    $text_color = imagecolorallocate($img, 255, 255, 255); // branco
    $bar_color = imagecolorallocate($img, 0, 0, 0); // preto
    $border_color = imagecolorallocate($img, 192, 192, 192); // cinza claro
    // Crie algumas cores
    // a serem usadas
    // para desenhar as
    // partes do gráfico.

3    // Preenche o fundo
    imagefilledrectangle($img, 0, 0, $width, $height, $bg_color);
    // Clareie o fundo, para deixá-lo pronto
    // para os desenhos no gráfico.

4    // Desenha a barra
    $bar_width = $width / ((count($data) * 2) + 1);
    for ($i = 0; $i < count($data); $i++) {
        imagefilledrectangle($img, ($i * $bar_width * 2) + $bar_width, $height,
        ($i * $bar_width * 2) + ($bar_width * 2), $height - (($height / $max_value) * $data[$i][1]),
        $bar_color);
        imagestringup($img, 5, ($i * $bar_width * 2) + ($bar_width), $height - 5, $data[$i][0],
        $text_color);
        // Desenhe a barra como um
        // retângulo preenchido.
        // Desenhe o título da barra como uma string de
        // texto orientada verticalmente.
    }

5    // Desenha um retângulo em torno da imagem inteira
    imagerectangle($img, 0, 0, $width - 1, $height - 1, $border_color);
    // Desenhe um retângulo em
    // torno de todo o gráfico.

6    // Desenha a faixa de valores no lado esquerdo do gráfico
    for ($i = 1; $i <= $max_value; $i++) {
        imagestring($img, 5, 0, $height - ($i * ($height / $max_value)), $i, $bar_color);
        // Desenhe um retângulo em
        // torno de todo o gráfico.
        // Desenhe a faixa de valores no lado
        // esquerdo do gráfico, na forma de
        // strings de texto normais, na horizontal.

7    }
    // Escreve a imagem em um arquivo
    imagepng($img, $filename, 5);
    imagedestroy($img);
    // Escreva a imagem em um arquivo
    // PNG com o nome especificado e o
    // nível de compressão 5 (médio).

8    // Apague a imagem da
    // memória.
}

A pasta do servidor onde o arquivo deve ser escrito precisa estar liberada
para escrita, caso contrário esta função não poderá operar corretamente.

```

## Perguntas Ídotas

**Q:** Por que a função draw\_bar\_graph() escreve a imagem do gráfico de barras em um arquivo, em vez de retorná-lo diretamente para o navegador?

**A:** Porque a função não fica contida em um script próprio, que possa retornar a imagem para o navegador através de um cabeçalho. Lembre-se, a única forma de retornar uma imagem dinamicamente gerada diretamente para o navegador é com o script usando um cabeçalho, o que significa que o propósito único do script tem de ser a geração da imagem.

**Q:** Então por que a função draw\_bar\_graph() não é colocada em um script próprio, para que possa retornar a imagem diretamente para o navegador usando um cabeçalho?

**A:** Embora seja uma boa ideia colocar a função em um script próprio, para torná-la reutilizável, ainda há um problema quanto a se retornar uma imagem através de um cabeçalho. O problema tem a ver com o modo como você reutiliza código. Quando algum código é incluído em um script usando-se include, include\_once, require ou require\_once, esse código é colocado no script como se ele existisse lá, originalmente. Isso

funciona muito bem para códigos que não façam nada que manipule o navegador. Mas enviar um cabeçalho tem impacto sobre o output do script, o que pode ser problemático para código incluído. Não é que você não possa enviar cabeçalhos a partir de código incluído; na verdade, nós fizemos isso em alguns exemplos anteriores. O problema é que você precisa ser extremamente cuidadoso, e em alguns casos, não é seguro assumir que outros cabeçalhos já não tenham sido enviados. O script My Mismatch, por exemplo, não é capaz de retornar uma imagem ao navegador porque a sua tarefa é fazer output de código HTML contendo resultados do Mismatch. Incluir algum código que gere dinamicamente e retorne uma imagem causaria um conflito de cabeçalhos.

**Q:** Ok, então eu posso apenas referenciar o código do gráfico de barras, como o script captcha.php do Guitar Wars. Ele funcionou bem sem usarmos um include, certo?

**A:** Certo, e referenciou o script captcha.php diretamente a partir do atributo src de uma tag <img>. O problema aqui é que temos um monte de dados que precisam ser passados para o código do gráfico de barras, e seria bastante bagunçado tentar fazê-lo através de GET ou POST.

## Desenhe e mostre a imagem do gráfico de barras

A função draw\_bar\_graph() torna possível gerar dinamicamente um gráfico de barras, desde que você forneça a ela as informações necessárias. No caso do gráfico do Mismatch, é preciso enviar uma largura e uma altura que funcionem na página My Mismatch (480x 240), o array bidimensional com os dados das categorias desencontradas, 5 como o valor máximo (número máximo de tópicos desencontrados por categoria), e um caminho e nome de arquivo adequados para o gráfico de barras resultante. Após chamarmos a função, a imagem é gerada e pode ser exibida usando-se uma tag HTML <img>.

O arquivo gráfico gerado por esta chamada a função recebe o nome mymismatchgraph.png, e é armazenado no servidor no caminho identificado por MM\_UPLOADPATH.

```
echo '<h4>Detalhes das categorias desencontradas:</h4>';
```

```
draw_bar_graph(480, 240, $category_totals, 5, MM_UPLOADPATH, 'mymismatchgraph.png');
```

Com a ajuda de uma função reutilizável, é possível criar um gráfico de barras, armazenado em um arquivo, a partir dos dados presentes no banco.

O mesmo caminho e nome de arquivo são especificados no atributo src da tag <img>.

Appearance	3
Entertainment	4
Food	4
People	2
Activities	5



mymismatchgraph.png

teste o mismatch, agora com gráficos de barras



## TEST DRIVE

### Crie o script My Mismatch e teste-o.

Crie um arquivo de texto chamado mymismatch.php, e digite nele o código para o script My Mismatch (ou baixe o script no site da Alta Books em [www.altabooks.com.br](http://www.altabooks.com.br)). Além disso, adicione um item de menu para My Mismatch ao script navmenu.php.

Envie os scripts para o seu servidor web, e depois abra a página principal do Mismatch (index.php) em um navegador web. Faça login caso ainda não o tenha feito, e clique em "My Mismatch" no menu de navegação principal. Parabéns, você encontrou seu par imperfeito ideal!

Mismatch - My Mismatch

Home ▶ View Profile ▶ Edit Profile ▶ Questionnaire ▶ My Mismatch ▶ Log Out

Jason Flamington  
Hollywood, CA

You are mismatched on the following 18 topics:

Tattoos	Cowboy boots	Long hair	Reality TV
Horror movies	Easy listening music	The opera	Sushi
Spicy food	Peanut butter & banana sandwiches	Martini's	Bill Gates
Hugh Hefner	Yoga		Weightlifting Cube puzzles
Karaoke	Hiking		

Mismatched category breakdown:

A bar chart titled "Mismatched category breakdown" showing the count of topics for each category. The categories are: Relationships, Career / Education, Food, Hobbies, and Activities. The counts are approximately: Relationships (~3), Career / Education (~4.5), Food (~4.5), Hobbies (~2.5), and Activities (~5).

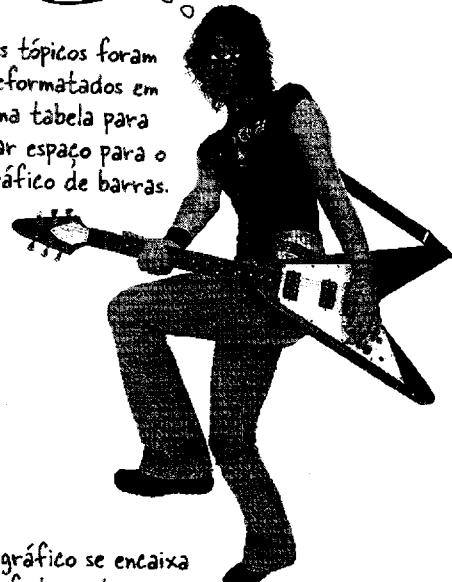
Category	Count
Relationships	3
Career / Education	4.5
Food	4.5
Hobbies	2.5
Activities	5

[View Jason's profile](#)

Copyright ©2008 Mismatch Enterprises, Inc.

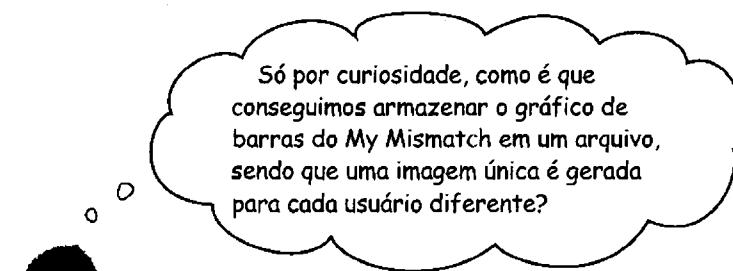
Agora sim! A representação visual das categorias é tudo o que eu precisava para me convencer de que Jason é o cara certo para mim!

Os tópicos foram reformatados em uma tabela para dar espaço para o gráfico de barras.



O gráfico se encaixa perfeitamente na página My Mismatch, junto com a lista de tópicos desencontrados.

- ➊ Desenhar o gráfico de barras usando-se os totais das categorias.

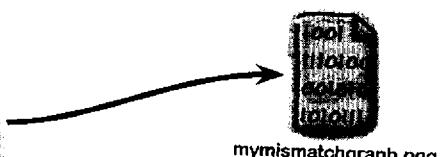
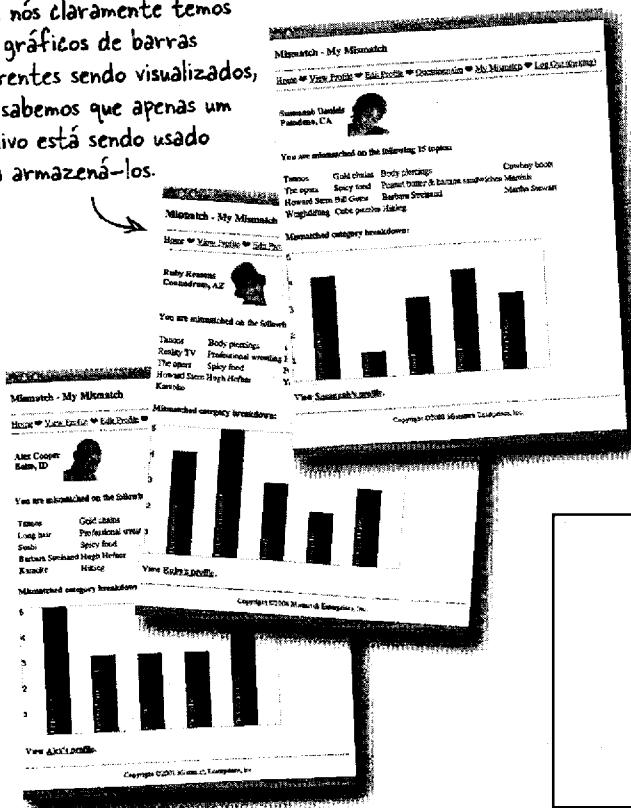


### Um pouco de sorte, na verdade.

É verdade que só existe um gráfico de barras de cada vez, não importando quantos usuários existam. Isso poderia ser um problema se dois usuários tentassem ver a página My Mismatch exatamente no mesmo momento. Nós corremos o risco de gerar imagens separadas para as duas pessoas, e depois escrevê-las em um único arquivo.

No mundo real, esse problema é razoavelmente improvável, mas à medida que o Mismatch crescer em popularidade e se expandir para milhares e milhares de usuários, poderia se tornar significativo. O fato de que cada usuário pensa no gráfico de barras como sendo unicamente seu é sinal de que há um ponto fraco no design com um só gráfico de barras.

Aqui, nós claramente temos três gráficos de barras diferentes sendo visualizados, mas sabemos que apenas um arquivo está sendo usado para armazená-los.



**Uma mesma imagem é continuamente gerada e re-gerada para o gráfico de barras de cada usuário.**

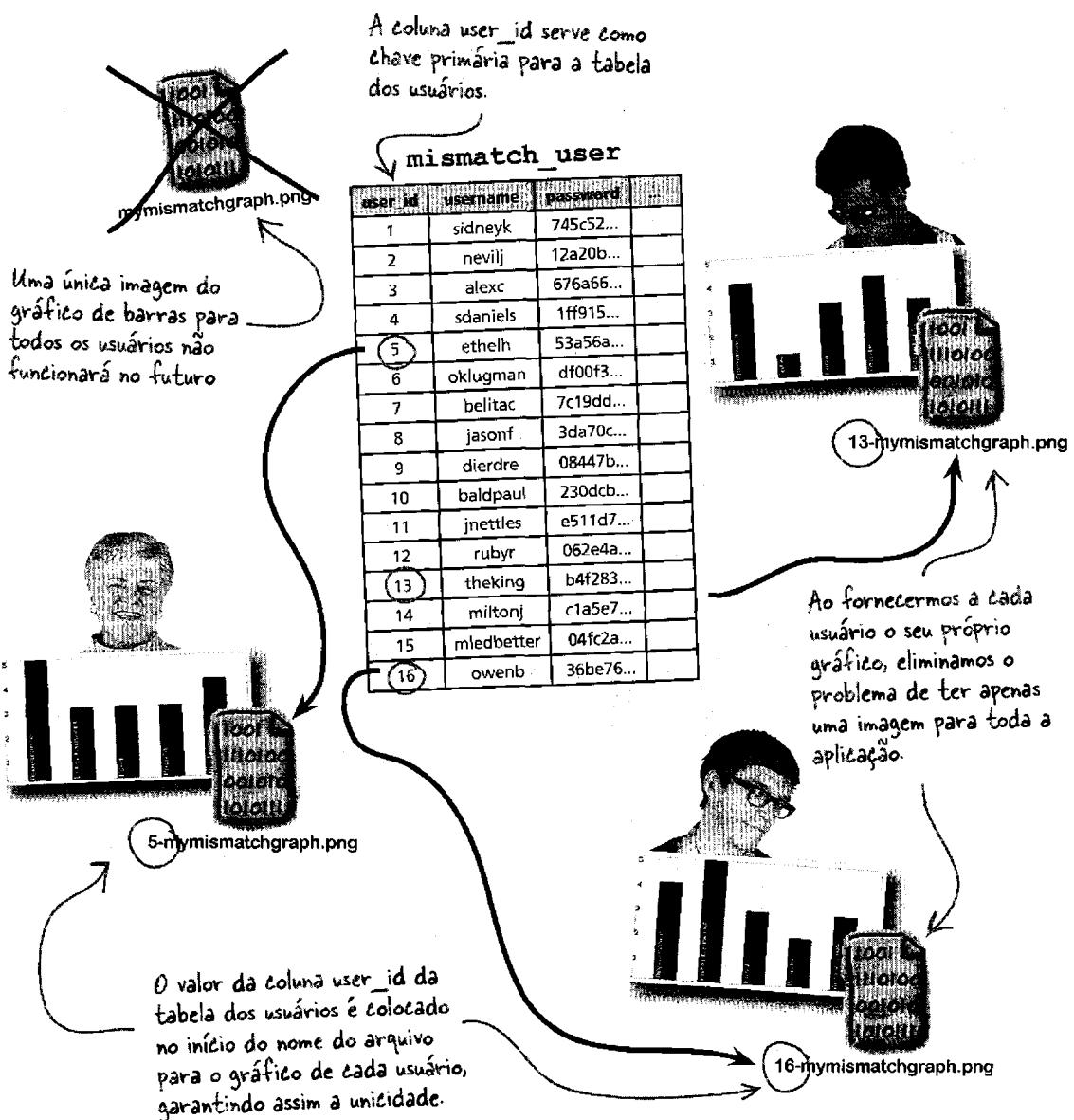
**PODER DO CÉREBRO**

Como você modificaria o código do Mismatch para garantir que jamais dois usuários compartilhem o mesmo gráfico de barras?

um gráfico de barras para cada usuário

## Imagens individuais do gráfico de barras para todos

A solução para o problema dos gráficos de barras compartilhados reside em se gerar múltiplas imagens, na verdade uma para cada usuário. Mas ainda temos de garantir que cada uma dessas imagens fique vinculada a exatamente um usuário, e não mais que um. É aí que um elemento familiar dos projetos de bancos de dados entra em ação... a chave primária! A chave primária da tabela `mismatch_user`, `user_id`, identifica individualmente cada usuário, e portanto, nos fornece uma excelente maneira de nomear de forma única cada gráfico de barras, além de associá-lo com um usuário. Tudo o que temos de fazer é juntar as IDs dos usuários ao nome de arquivo dos seus gráficos.



## Perguntas Ídotas

**P:** Existe alguma vantagem em se fazer o output de imagens geradas dinamicamente como PNG, em vez de GIFs ou JPEGs?

**R:** Não, nenhuma além dos motivos pelos quais você normalmente preferiria um formato a outro. Por exemplo, GIFs e PNGs são melhores para gráficos vetoriais, enquanto que JPEGs são melhores para gráficos foto-realísticos. No caso do Mismatch, estamos lidando com gráficos vetoriais, portanto PNG ou GIF funcionariam bem. PNG é um padrão mais moderno, e por isso foi usado, mas GIF teria funcionado também. Para fazer o output de uma imagem GD no formato GIF ou JPEG, respectivamente, chame as funções imagegif() ou imagejpeg().

**P:** Como eu sei qual nível de compressão usar ao fazer o output de imagens PNG para um arquivo?

**R:** As configurações de nível de compressão para imagepng() entram em cena ao se enviar uma imagem PNG para um arquivo, e elas vão de 0 (nenhuma compressão) até 9 (compressão máxima). Não existem regras fixas sobre quanta compressão usar, portanto, você poderá querer experimentar diferentes valores. O Mismatch usa o nível 5 para os gráficos de barras, o que parece ser um equilíbrio razoável entre qualidade e eficiência.



Segue abaixo o código do Mismatch que gera dinamicamente um gráfico de barras e o exibe na página. Reescreva o código de modo que ele passe a gerar uma imagem única para cada usuário. Dica: use `S_SESSION['user_id']` para criar um nome de arquivo único para cada usuário.

```
echo '<h4>Detalhes das categorias desencontradas:</h4>';

draw_bar_graph(480, 240, $category_totals, 5, MM_UPLOADPATH . 'mymismatchgraph.png');

echo '<br />';

.....
.....
.....
.....
```

**P:** Existem problemas de armazenamento introduzidos pelo fato de estarmos gerando um gráfico para cada usuário?

**R:** Na verdade não. Essa questão tem um pouco a ver com o nível de compressão acima, mas é improvável que você sobrecarregue seu servidor com arquivos demais ou arquivos grandes demais, a não ser que realmente fique louco e saia gerando milhares de arquivos gráficos enormes. Como exemplo, considere que os gráficos de barra do Mismatch têm uma média de 2 KB cada um, portanto, mesmo que o site estoure e passe a ter 50.000 usuários, estamos falando de um total de 100 MB em gráficos de barras. É claro que isso é uma boa quantia de espaço em disco, mas um site com 50.000 deverá ser capaz de gerar receita mais que suficiente para pagar por esse tipo de armazenamento.



## exercício solução



### Exercício Solução

Segue abaixo o código do Mismatch que gera dinamicamente um gráfico de barras e o exibe na página. Reescreva o código de modo que ele passe a gerar uma imagem única para cada usuário. Dica: use `S_SESSION['user_id']` para criar um nome de arquivo único para cada usuário.

```
echo '<h4>Detalhes das categorias desencontradas:</h4>';  
  
draw_bar_graph(480, 240, $category_totals, 5, MM_UPLOADPATH . 'mymismatchgraph.png');  
echo '<img src="" . MM_UPLOADPATH . 'mymismatchgraph.png" alt="Mismatch category graph" /><br />';  
  
Certifique-se de que esta pasta do servidor está liberada para escrita, para que o arquivo possa ser gravado nela.  
  
echo '<h4>Mismatched category breakdown:</h4>';  
draw_bar_graph(480, 240, $category_totals, 5,  
    MM_UPLOADPATH . S_SESSION['user_id'] . '-mymismatchgraph.png');  
echo '<img src="" . MM_UPLOADPATH . S_SESSION['user_id'] . '-mymismatchgraph.png"'.  
    ' alt="Mismatch category graph" /><br />';
```

O caminho padrão para upload de arquivos ainda é usado para garantir que a imagem seja armazenada no lugar desejado, no servidor.

O nome único do arquivo segue a forma X-mymismatchgraph.png, onde X é a ID do usuário

Podemos usar a ID que já temos armazenada em uma variável de sessão.

O mesmo nome de arquivo é usado para se definir o atributo `src` da tag `<img>` correspondente ao gráfico de barras, no código HTML.



## TEST DRIVE

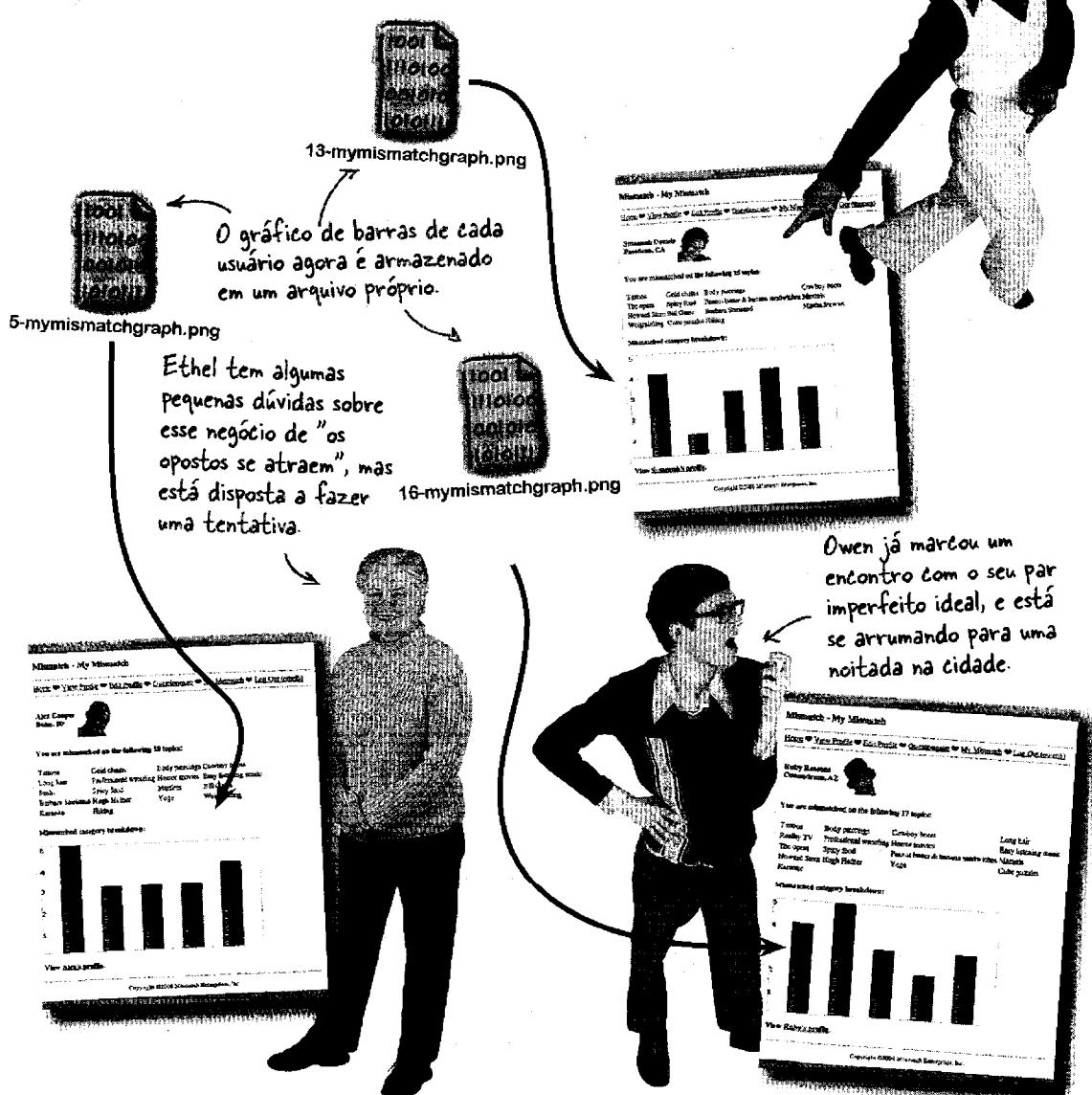
### Modifique o script My Mismatch para que ele gere gráficos de barras únicos.

Modifique o script My Mismatch para que ele gere um gráfico de barras único para cada usuário. Envie o script mymismatch.php para o seu servidor web, e depois abra-o em um navegador. A página não terá nenhuma diferença visível, mas você pode visualizar o código-fonte dela para verificar que o gráfico de barras agora tem um nome de arquivo único.

# Os usuários do Mismatch estão gostando dos gráficos de barras

Resolvido o problema dos gráficos de barras compartilhados, você ajudou a eliminar um potencial gargalo de desempenho no longo prazo, quando mais e mais usuários se juntarem ao Mismatch e tirarem proveito do gráfico de "cinco graus de oposição". Agora, cada usuário gera o seu próprio gráfico de barras ao visualizar o seu par imperfeito ideal. Felizmente, esse conserto pode ser feito nos bastidores, sem que os usuários, que são realmente as pessoas tirando proveito dos dados do Mismatch para encontrarem seus pares, ficasssem sabendo.

Elmer agora está praticando alguns novos passos de dança para compartilhar com o seu par.



sua caixa de ferramentas do php &amp; mysql



## Sua Caixa de Ferramentas do PHP & MySQL

**Os gráficos dinâmicos abrem todo tipo de possibilidades interessantes no que diz respeito à criação de scripts PHP para criar imagens automaticamente.**

**Vamos recapitular o que torna tudo isso possível.**

### CAPTCHA

Um programa que protege um site web de bots de spam automatizados, usando algum tipo de teste. Por exemplo, o teste do CAPTCHA poderia exigir que o usuário distinguisse letras em uma imagem distorcida, identificasse o conteúdo da imagem ou talvez analisar uma equação para realizar um cálculo matemático simples.

### Biblioteca GD

Um conjunto de funções PHP usadas para se desenhar gráficos em uma imagem. A biblioteca GD lhe permite criar imagens dinamicamente e desenhar nelas, e depois enviar-las diretamente para o navegador ou escrevê-las em um arquivo, no servidor.

### imageLine(), imagerectangle()

A biblioteca de gráficos GD oferece muitas funções para se desenhar fundamentos de gráficos, tais como linhas, retângulos, elipses e até mesmo pixels individuais. Cada função opera sobre uma imagem que já tenha sido criada com `imagecreatetruecolor()`.

### `imagecreatetruecolor()`

Esta função faz parte da biblioteca de gráficos GD, e é usada para se criar uma nova imagem. A imagem é criada inicialmente na memória, e não é enviada para exibição até que você chame outra função, como por exemplo, `imagepng()`.

### `imagestring(), imagestringup(), imagettftext()`

A biblioteca de gráficos GD também lhe permite desenhar texto ou com uma fonte interna ou com uma fonte True Type escolhida por você.

### `imagedestroy()`

Após desenhar em sua imagem e exibi-la conforme desejados, é uma boa ideia destruir os recursos associados com ela por meio da chamada à esta função.

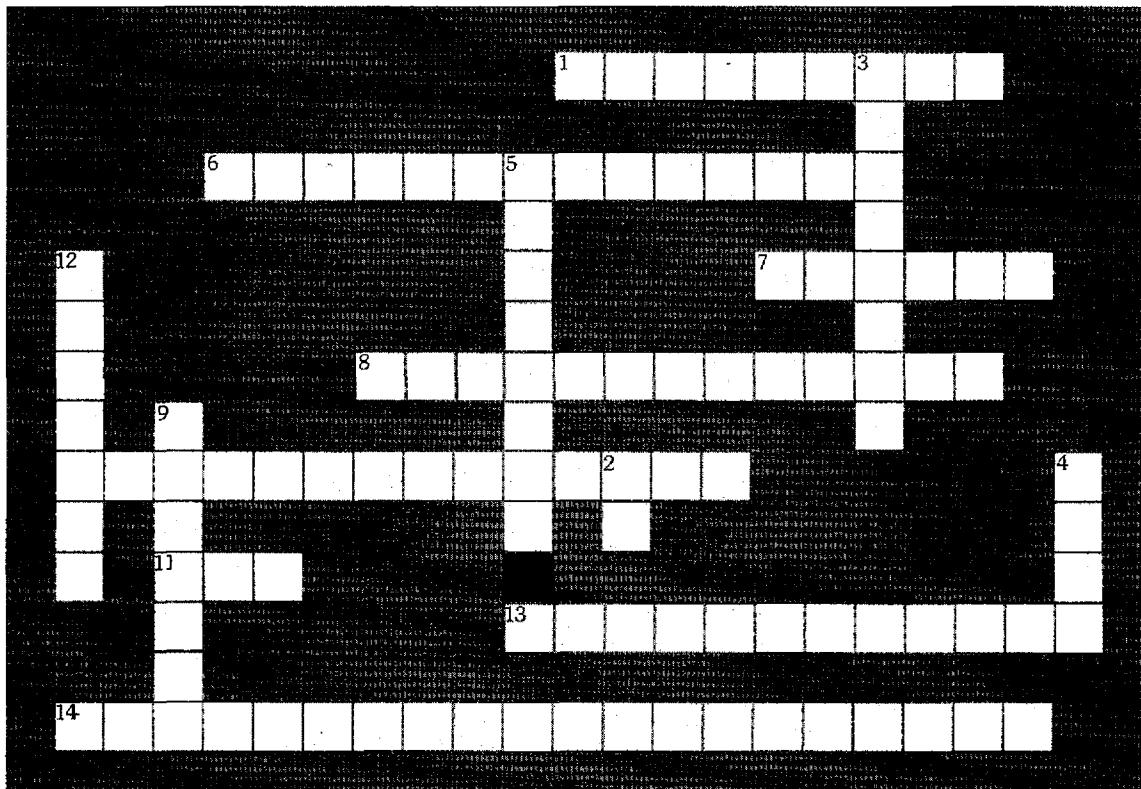
### `imagepng()`

Ao terminar de desenhar em uma imagem usando as funções gráficas GD, esta função faz o output da imagem. Para que ela possa ser exibida, você pode optar por enviar a imagem diretamente para o navegador web ou então para um arquivo no servidor.



# Palavras Cruzadas PHP & MySQL

Quando você realmente precisa de um robô, não tem nenhum por perto. Bem, de qualquer forma o seu cérebro analógico será capaz de resolver estas palavras cruzadas.



## Horizontais

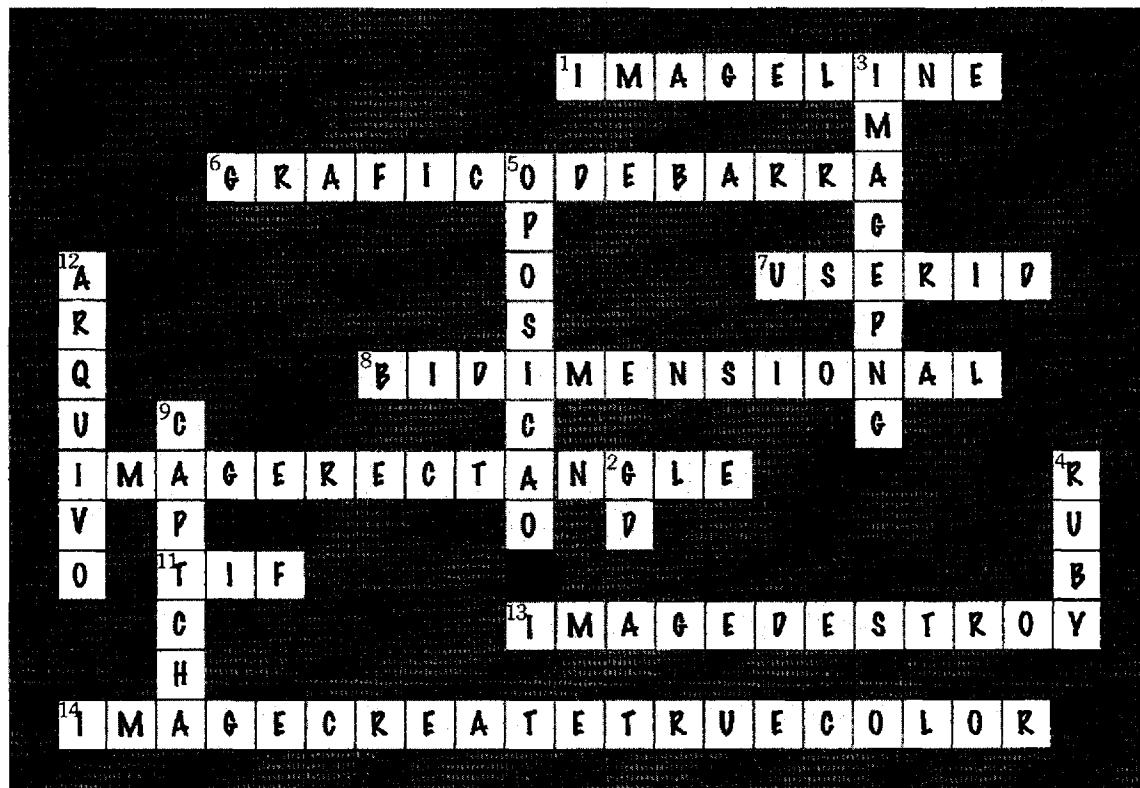
1. Esta função gráfica do PHP desenha uma linha.
6. A representação visual usada para mostrar como os usuários do Mismatch comparam-se uns com os outros, em termos das respostas às categorias.
7. Para gerarmos gráficos de barras individuais para cada usuário do Mismatch, esta informação é usada como parte do nome do arquivo para a imagem.
8. O Mismatch usa este tipo de array para armazenar os dados para o gráfico de barras.
10. Forneça-lhe dois pontos e esta função gráfica desenhará um retângulo.
11. Para desenhar texto com uma determinada fonte, chame a função `image...text()`.
13. Sempre apague a imagem da memória após terminar de trabalhar com ela no PHP, chamando esta função.
14. Chame esta função gráfica para criar uma nova imagem.

## Verticais

2. O nome da biblioteca de gráficos do PHP.
3. Chame esta função para fazer o output de uma imagem como PNG.
4. O nome do par imperfeito ideal de Owen.
5. O Mismatch usa um gráfico de barras para comparar usuários com base no conceito de "cinco graus de .....".
9. Um teste usado para se distinguir entre pessoas reais e bots de spam automatizados.
12. Quando o PHP faz output de uma imagem, esta é enviada diretamente para o navegador cliente ou então é armazenada em um .....



# Palavras Cruzadas PHP & MySQL - Solução



## 12 Repúblicação e Serviços Web

# Interfaces com o Mundo

Isto é fascinante. Em vez de termos de viajar por aí perguntando às pessoas quais são as novidades, podemos ter as novidades enviadas para nós... brilhante!

De fato. A tecnologia traz o mundo para as nossas mãos sujas de tinta!



**Existe um grande mundo lá fora, mundo esse que a sua aplicação web não pode se dar ao luxo de ignorar. Ou que talvez seja mais, você é que não gostaria que o mundo ignorasse a sua aplicação. Uma excelente maneira de fazer o mundo notá-la é tornar os seus dados disponíveis para repúblicação, o que significa que os usuários poderão assinar o conteúdo do seu site, em vez de ter que visitá-lo diretamente para encontrar novas informações. Não apenas isso, a sua aplicação pode fazer interfaces com outras aplicações, através dos serviços web, e pode tirar proveito dos dados de outras pessoas para fornecer uma experiência mais rica.**

*espalhando a palavra além do website do owen*

## Owen precisa avisar ao mundo sobre Fang

Um dos grandes problemas com que qualquer site precisa se deparar é como fazer as pessoas continuarem voltando. Uma coisa é atrair um visitante; mas fazê-lo voltar ao site depois já é outra história. Até mesmo os sites com o conteúdo mais atraente podem acabar saindo da mente do usuário, simplesmente porque é difícil se lembrar de visitar um site regularmente. Sabendo disso, Owen deseja oferecer um meio alternativo de visualizar os relatos de abdução alienígena – ele deseja enviar automaticamente os relatórios para as pessoas (algo conhecido como tecnologia "push"), para que elas não precisem se lembrar de visitar o site regularmente.

**Aliens Abducted Me - Report an Abduction**

Share your story of alien abduction:

First name: Belta  
Last name: Chevy  
What is your email address? beltagir@rockin.net  
When did it happen? 2008-06-21  
How long were you gone? almost a week  
How many did you see? 27  
Describe them: Clumsy little buggers, had no rhythm.  
What did they do to you? Tried to get me to play bad music.  
Have you seen my dog Fang? Yes  No

Anything else you want to add?  
Report Abduction

Looking forward to playing some *Catstar Wars*.

Os relatos de abduções estão chegando, mas ainda não encontrei o Fang!

Parece que Fang foi avistado algumas vezes, mas essas informações ainda não permitirão que Owen o localizasse.



O formulário Relatar uma Abdução está funcionando perfeitamente, mas Owen acha que o site precisa de mais publicidade.

Desde a última vez que vimos Owen, ele criou uma página para a visualização dos relatos de abduções submetidos pelos usuários.

With extraterrestrials? Were you abducted? Have you seen my abducted dog, Fang?

Recent reported abductions:	
2008-06-21 : Belta Chevy	Fang spotted: no
Abducted for: Alien description: almost a week Clumsy little buggers, had no rhythm.	Fang spotted: yes
2008-05-11 : Sally Jones	Fang spotted: no
Abducted for: Alien description: 1 day green with six tentacles	Fang spotted: yes
2000-07-12 : Alf Nader	Fang spotted: no
Abducted for: Alien description: one week It was a big non-recyclable shiny disc full of what appeared to be mutated labor union officials.	Fang spotted: yes
1991-09-14 : Don Quayle	Fang spotted: no
Abducted for: Alien description: 37 seconds They looked like donkeys made out of metal with some kind of jet packs attached to them.	Fang spotted: yes
1969-01-21 : Rick Nixon	Fang spotted: no
Abducted for: Alien description: nearly 4 years They were pasty and pandering, and not very forgiving.	

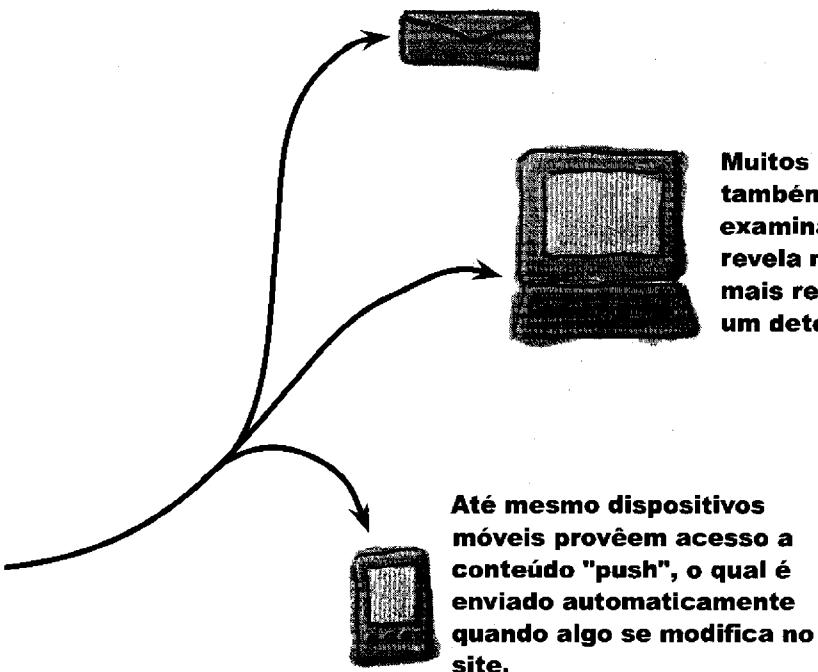
Owen espera que uma maior publicidade para o site aumente suas chances de encontrar Fang.

## Envie dados sobre abduções alienígenas para as pessoas

Ao enviar o conteúdo sobre as abduções para os usuários, Owen estará, na prática, criando uma equipe virtual de pessoas que podem ajudá-lo a monitorar os relatos. Com mais pessoas trabalhando no caso, aumentam as chances de identificar mais avistamentos de Fang, e possivelmente de encontrá-lo.

**Alguns clientes de email têm suporte ao conteúdo "push", permitindo que você receba atualizações do site da mesma forma que recebe mensagens de email.**

Enviar conteúdo web para os usuários é uma excelente forma de obter mais publicidade para um site.



Muitos navegadores web também lhe permitem examinar conteúdo "push" que revela rapidamente as notícias mais recentes publicadas em um determinado site.

Owen ainda não sabe exatamente como enviar conteúdo para os usuários, mas gostou muito da ideia.

Owen espera que a sua equipe virtual de analistas de abduções aumente as suas chances de encontrar Fang.



**use RSS para republicar seu site**

## O RSS envia conteúdo web para as pessoas

A ideia por trás do conceito de publicar conteúdo HTML na web é que esse conteúdo seja visto pelas pessoas que visitarem o site. Mas e se quiséssemos que os usuários recebessem nosso conteúdo sem ter de requisitá-lo? Isso é possível com o RSS, um formato de dados que permite aos usuários descobrir conteúdos web sem ter que visitar o site.

**O RSS funciona mais ou menos como o equivalente web de um aparelho gravador de vídeo digital (digital video recorder ou DVR).**

Os DVRs lhe permitem "assinar" determinados programas de TV, automaticamente gravando cada episódio assim que ele vai ao ar.

Por que ficar zapeando pelos canais em busca do seu programa favorito, quando pode simplesmente deixar que eles sejam gravados e entregues a você por meio do DVR? Embora o RSS na verdade não grave nada, ele é semelhante ao DVR, no sentido que traz o conteúdo web para você, em vez de você ter que ir atrás desse conteúdo.

Criando um feed ou transmissão, RSS para os seus dados referentes às abduções, Owen deseja notificar os usuários sempre que novos relatos forem inseridos. Isso ajudará a garantir que as pessoas permaneçam interessadas, resultando em mais gente analisando os dados. O legal é que o mesmo banco de dados pode alimentar tanto a página web quanto o feed RSS

O HTML serve para visualizar; o RSS serve para notificar.

Um leitor de notícias permite que você assine determinados newsfeeds, os quais contêm itens originários do conteúdo do seu site.

**Aliens Abducted Me**

Welcome, here you had an encounter with extraterrestrials? Were you abducted? Have you seen my abd Report it here!

Most recent reported abductions:

2008-06-21 : Belita Chevy  
Abducted for: Alien description:  
Abducted for: Aliens description:  
almost a week Chumy little buggers, had no rhythm.

2008-05-11 : Sally Jones  
Abducted for: Alien description:  
1 day green with six tentacles

2000-07-12 : Alf Nader  
Abducted for: Alien description:  
one week It was a big non-recyclable shiny disc full of what appeared to be mutated labor union

1991-09-14 : Don Quayle  
Abducted for: Alien description:  
37 seconds They looked like donkeys made out of metal with some kind of jet packs attached

1969-01-21 : Rick Nixon  
Abducted for: Alien description:  
nearly 4 years They were pasty and pandering, and not very forgiving.

**Aliens Abducted Me - Newsfeed**

Belita Chevy - Chumy little buggers, had no rhythm... Jun 21, 2008 AM  
Tried to get me to play bad music. [Read more...](#)

Sally Jones - green with six tentacles... May 11, 2008 AM  
We just talked and played with a dog. [Read more...](#)

Alf Nader - It was a big non-recyclable shiny... Jun 22, 2008 AM  
Shoved down from the sky and snatched me up with no warning. [Read more...](#)

Don Quayle - They looked like donkeys made ou... Sep 14, 2001 10:30 AM  
I was being there during a break pastie when "Zooper" this beam of light took me away. [Read more...](#)

Rick Nixon - They were pasty and pandering, a... Jun 23, 2008 1:00 AM  
Impaled me, of course, when they picked me. [Read more...](#)

Recent Articles:  
Today  
Yesterday  
Last Seven Days  
This Month  
Last Month

Sources:  
Aliens Abducted Me...

Address:  
Update Now  
Mail Link To This Page  
Subscribe In Atom  
Add Bookmark...

Embora a página web seja gerada dinamicamente a partir de dados do banco, você precisa revisitá-la para ver se existem novos dados publicados.

Aqui, o leitor de notícias interno do navegador Safari está sendo usado.

O RSS oferece uma visualização de dados web para os usuários automaticamente, logo que algum novo conteúdo é disponibilizado. Uma visualização RSS sobre um determinado conjunto de dados chama-se feed RSS ou newsfeed. Os usuários assinam o feed e recebem novo conteúdo logo que este é postado no site – não é preciso visitar o site.

Para visualizar um feed RSS, a pessoa só precisa ter um leitor de notícias (newsread) RSS. A maioria dos navegadores e clientes de email mais populares oferece o recurso de assinar feeds RSS. Basta fornecer ao leitor de news a URL do feed, e ele faz o resto

# O RSS é na verdade XML

O RSS é parecido com o HTML, no sentido que é uma simples linguagem de markup de texto, que usa tags e atributos para descrever o conteúdo. O RSS é baseado no XML, que é uma linguagem de markup geral que pode ser usada para descrever qualquer tipo de dados. O poder do XML advém da sua flexibilidade – ele não define tags nem atributos específicos; apenas estabelece as regras para o modo como tags e atributos são criados e usados. Fica a cargo de linguagens específicas, como o HTML e o RSS, estabelecer os detalhes referentes à quais tags e atributos podem ser usados, e como.

Para ser proficiente com o RSS, você precisa primeiro entender as regras fundamentais do XML. Essas regras se aplicam a todas as linguagens baseadas no XML, incluindo o RSS e a versão moderna do HTML, conhecida como XHTML. Essas regras são simples, porém importantes – o seu código XML (RSS) não funcionará se você as violar! Aí vão elas:

Tags que possuam conteúdo precisam aparecer em pares.

Incorreto! Não há uma tag final formando o par.  
~~<p>Phone home!</p>~~

<p>Phone home!</p>



Correto.

Tags vazias (sem conteúdo) precisam ser programadas com um espaço e uma barra ao final, antes do fechamento do comando.

Incorreto! A tag vazia precisa de um espaço e uma barra antes do >.  
~~<br />~~

<br>

Correto.

Todos os valores dos atributos precisam estar dentro de aspas duplas.

Ao contrário do PHP, que lhe permite usar aspas duplas ou simples na maioria das situações, o XML é rígido ao só permitir aspas duplas para os valores dos atributos.

~~~~



Correto.

O XML é uma linguagem markup usada para descrever qualquer tipo de dados.

**não existem perguntas idiotas sobre RSS**

não existem  
**Perguntas Idiotas**

**P:** Por que usar o RSS é tão melhor do que ter as pessoas acessando o meu site?

**R:** Se as pessoas regularmente visitarem o seu site e procurarem o conteúdo mais recente, então usar o RSS não teria nenhum benefício adicional em relação à simplesmente exibir o conteúdo no site. Mas a maioria das pessoas se esquece dos sites, até mesmo dos que elas gostam. Assim, o RSS fornece um meio eficiente de levar o seu conteúdo diretamente às pessoas, em vez de exigir que elas o procurem.

**P:** O que significa a sigla RSS?

**R:** Hoje em dia, RSS significa Really Simple Syndication ("Repubicação Realmente Simples"). Ao longo da sua história, tem havido diferentes versões para a sigla, mas a encarnação mais recente do RSS (versão 2.0) significa "Repubicação Realmente Simples", que é a única coisa com que você precisa se preocupar.

**P:** Do que consiste o RSS?

**R:** O RSS é um formato de dados. Assim como o HTML é um formato de dados que lhe permite descrever conteúdo web para visualização em um navegador, o RSS é um formato de dados que descreve um conteúdo que fica acessível como um newsfeed. Semelhante ao HTML, o formato de dados do RSS é puro texto, e consiste de tags e atributos usados para se descrever o conteúdo de um newsfeed.

**P:** Onde eu consigo um leitor de RSS?

**R:** A maioria dos navegadores web tem um leitor de RSS interno. Alguns clientes de email também incluem leitores de RSS, em cujo caso os itens RSS aparecem como mensagens de email em uma pasta especial para eles. Existem também leitores de RSS autônomos.



Segue abaixo um código RSS para o newsfeed Fui Abduzido por Aliens. Escreva comentários para as partes marcadas do código, explicando o que você acha que cada tag está fazendo.

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
  <channel>
    <title>Fui Abduzido por Aliens - Newsfeed</title>
    <link>http://aliensabductedme.com/</link>
    <description>Relatos de abduções alienígenas em todo o mundo, cortesia
de Owen e seu cão abduzido Fang.</description>
    <language>en-us</language>

    <item>
      <title>Belita Chevy - Uns carinhos bem bagunçados, sem rit...</title>
      <link>http://www.aliensabductedme.com/index.php?abduction_id=7</link>
      <pubDate>Sat, 21 Jun 2008 00:00:00 EST</pubDate>
      <description>Tentaram me fazer tocar música ruim.</description>
    </item>

    <item>
      <title>Sally... = Sally Jones - verdes com seis tentáculos...</title>
      <link>http://www.aliensabductedme.com/index.php?abduction_id=8</link>
      <pubDate>Sun, 11 May 2008 00:00:00 EST</pubDate>
      <description>Nós só conversamos e brincamos com um cão</description>
    </item>

  ...
</channel>
</rss>
```

## **o código RSS comentado**



### **Exercício Solução**

Segue abaixo um código RSS para o newsfeed Fui Abduzido por Aliens. Escreva comentários para as partes marcadas do código, explicando o que você acha que cada tag está fazendo.

Esta linha de código não é uma tag - é uma "diretiva" XML que indica que este documento contém código XML.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<rss version="2.0">
```

```
<channel>
```

```
<title>Fui Abduzido por Aliens - Newsfeed</title>
```

```
<link>http://aliensabductedme.com/</link>
```

<description>Relatos de abduções alienígenas em todo o mundo, cortesia de Owen e seu cão abduzido Fang.

```
abducted dog Fang.</description>
```

Esta tag <title> aplica-se ao canal como um todo.

O link para o canal normalmente aponta para o site web associado com o newsfeed.

```
<language>en-us</language>
```

Todo canal precisa de uma descrição para explicar que tipo de conteúdo ele oferece.

```
<item>
```

```
<title>Belita Chevy - Uns carinhos bem bagunçados, sem rit...</title>
```

```
<link>http://www.aliensabductedme.com/index.php?abduction_id=7</link>
```

```
<pubDate>Sat, 21 Jun 2008 00:00:00 EST</pubDate>
```

```
<description>Tentaram me fazer tocar música ruim.</description>
```

```
</item>
```

O nome do abduzido e os dados referentes à descrição dos aliens são combinados para formar o título de cada item de notícias.

```
<item>
```

```
<title>Sally Jones - verdes com seis tentáculos...</title>
```

```
<link>http://www.aliensabductedme.com/index.php?abduction_id=8</link>
```

```
<pubDate>Sun, 11 May 2008 00:00:00 EST</pubDate>
```

```
<description>Nós só conversamos e brincamos </description>
```

```
</item>
```

Este documento RSS contém apenas um canal, o que é perfeitamente aceitável caso você não precise dividir os itens de notícias em diferentes categorias.

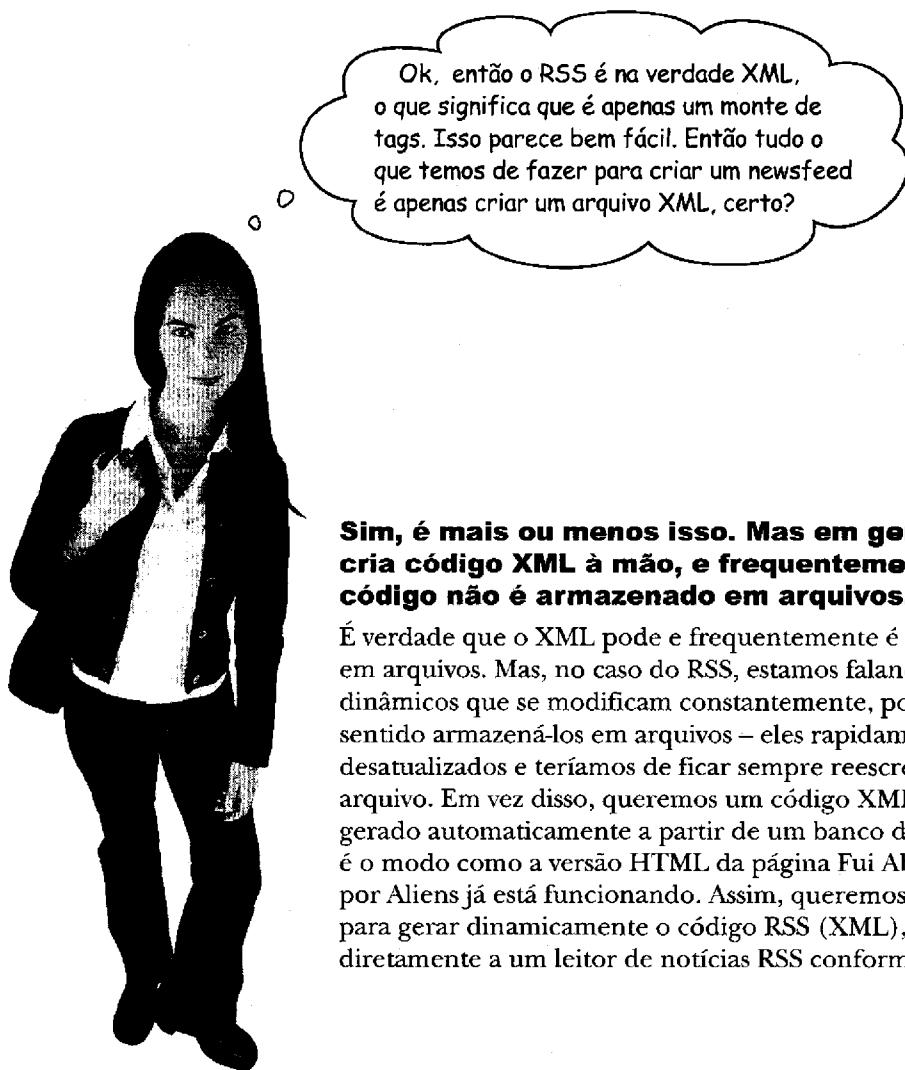
```
</channel>
```

```
</rss>
```

A data especificada na tag <pubDate> segue o formato RFC-822 para a data e hora, que é um padrão para se representar essas informações como texto.

Toda tag XML precisa ter uma abertura e um fechamento - esta tag de fechamento finaliza o documento RSS.

O link para um determinado item de notícias geralmente aponta para o conteúdo completo do item, presente no site associado com o newsfeed.



Ok, então o RSS é na verdade XML, o que significa que é apenas um monte de tags. Isso parece bem fácil. Então tudo o que temos de fazer para criar um newsfeed é apenas criar um arquivo XML, certo?

**Sim, é mais ou menos isso. Mas em geral não se cria código XML à mão, e frequentemente esse código não é armazenado em arquivos.**

É verdade que o XML pode e frequentemente é armazenado em arquivos. Mas, no caso do RSS, estamos falando de dados dinâmicos que se modificam constantemente, portanto não faz sentido armazená-los em arquivos – eles rapidamente ficariam desatualizados e teríamos de ficar sempre reescrevendo no arquivo. Em vez disso, queremos um código XML que seja gerado automaticamente a partir de um banco de dados, que é o modo como a versão HTML da página Fui Abduzido por Aliens já está funcionando. Assim, queremos usar o PHP para gerar dinamicamente o código RSS (XML), e enviá-lo diretamente a um leitor de notícias RSS conforme desejado.

## Do banco de dados para o news reader

Para fornecer um newsfeed dos dados de abduções, Owen precisa gerar dinamicamente o código RSS a partir do seu banco de dados MySQL. Esse código forma um documento RSS completo, pronto para ser consumido pelos news readers (leitores de notícias). Assim, o PHP é usado para traduzir os dados puros referentes às abduções para o formato RSS, que então pode ser processado pelos news readers (leitores de notícias) e disponibilizado para os usuários. A parte realmente legal desse processo é que, uma vez disponibilizado o newsfeed em formato RSS, tudo o mais é automático – é função do news readers (leitores de notícias) mostrar os itens atualizados assim que eles aparecem.

`aliens_abduction`

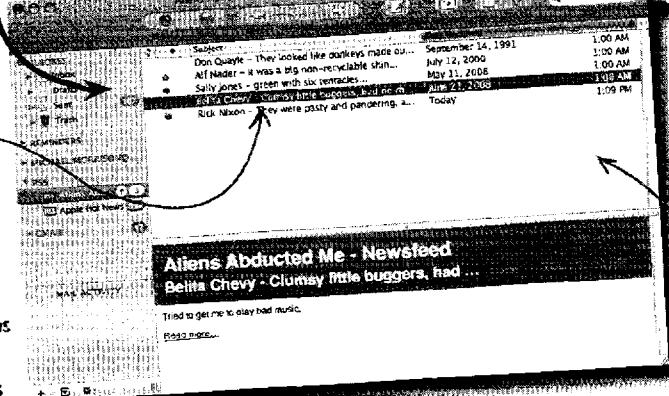
| abduction_id | first_name | last_name | when it happened | how long       | how many | at least | alien description                                 | what they did                                |
|--------------|------------|-----------|------------------|----------------|----------|----------|---------------------------------------------------|----------------------------------------------|
| 1            | Aif        | Nader     | 200-07-12        | one week       |          | 12       | It was a big non-recyclable shiny disc full of... | Swooped down from the sky and...             |
| 2            | Don        | Quayle    | 1991-09-14       | 37 seconds     |          | dunno    | They looked like donkeys made out of metal...     | I was sitting there eating a baked...        |
| 3            | Rick       | Nixon     | 1969-01-21       | nearly 4 years |          | just one | They were pasty and wandering, and not very...    | Impeached me, of course, then they probed... |
| 4            | Belita     | Chevy     | 2008-06-21       | almost a week  | 27       |          | Clumsy little buggers, had no rhythm.             | Tried to get me to play bad music.           |
| 5            | Sally      | Jones     | 2008-06-21       | 1 day          | four     |          | green with six tentacles                          | We just talked and played                    |

O leitor sabe como interpretar os itens de notícias individuais no código XML, e como mostrá-los para o usuário.

Este leitor de notícias é parte integrante do aplicativo Mail do Mac OS X. Muitos outros aplicativos de email populares também incluem leitores de notícias.

A função do news reader RSS é consumir os dados disponibilizados por um newsfeed RSS.

O PHP é usado para gerar um documento RSS a partir de um banco de dados MySQL.



Cada item de notícias individual tem a sua própria seção no documento RSS.

Cada leitor de notícias apresenta os itens de uma forma diferente – neste aqui, os itens são exibidos praticamente da mesma forma que mensagens de email.



Para se criar feeds RSS, é preciso entender a linguagem RSS, e para isso é preciso conhecer as tags usadas para se descreverem os itens de notícias. Ligue cada tag RSS à sua descrição

- |                                  |                                                                                                                                                                    |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>&lt;rss&gt;</code>         | Esta tag não tem nada a ver com o RSS. Mas pelo menos ela seria um nome legal para algum item de dados!                                                            |
| <code>&lt;channel&gt;</code>     | A data de publicação é uma informação importante para qualquer item de notícias, e esta tag é usada para especificá-la.                                            |
| <code>&lt;enclosure&gt;</code>   | Esta tag representa um canal em um feed RSS, e age como um container para dados descritivos e itens de news individuais.                                           |
| <code>&lt;title&gt;</code>       | Representa um item de notícias individual, ou história, que é descrito com mais detalhes por elementos subordinados.                                               |
| <code>&lt;language&gt;</code>    | Esta tag sempre contém uma URL que serve como o link para um canal ou item de notícias.                                                                            |
| <code>&lt;link&gt;</code>        | Delimita o feed RSS inteiro – todas as demais tags precisam estar dentro desta.                                                                                    |
| <code>&lt;description&gt;</code> | Esta tag armazena o título de um canal ou item de notícias, e geralmente é usada dentro das tags <code>&lt;channel&gt;</code> e <code>&lt;item&gt;</code> .        |
| <code>&lt;pubDate&gt;</code>     | Usada para se fornecer uma breve descrição de um canal ou item de notícias, aparecendo dentro das tags <code>&lt;channel&gt;</code> ou <code>&lt;item&gt;</code> . |
| <code>&lt;item&gt;</code>        | Esta tag se aplica a um canal, e especifica a linguagem usada por ele, como por exemplo, en-us para o inglês americano.                                            |

# COMO FAZER SOLUÇÃO

Todo feed RSS consiste de pelo menos um canal, que é basicamente um grupo de itens de notícias relacionados.

Para se criar feeds RSS, é preciso entender a linguagem RSS, e para isso é preciso conhecer as tags usadas para se descreverem os itens de notícias. Ligue cada tag RSS à sua descrição

A tag <rss> é a "raiz" do documento RSS – todas as outras tags precisam aparecer dentro dela.

<rss>

Esta tag não tem nada a ver com o RSS. Mas pelo menos ela seria um nome legal para algum item de dados!

<channel>

A data de publicação é uma informação importante para qualquer item de notícias, e esta tag é usada para especificá-la.

<title>

Esta tag só é usada nos canais.

<language>

Esta tag representa um canal em um feed RSS, e age como um container para dados descritivos e itens de notícias individuais.

Representa um item de notícias individual ou história, que é descrito com mais detalhes por elementos subordinados.

<link>

Esta tag sempre contém uma URL que serve como o link para um canal ou item de notícias.

<description>

Esta tag só se aplica a itens de news.

<pubDate>

Delimita o feed RSS inteiro – todas as demais tags precisam estar dentro desta.

Esta tag armazena o título de um canal ou item de notícias, e geralmente é usada dentro das tags <channel> e <item>.

Usada para se fornecer uma breve descrição de um canal ou item de notícias, aparecendo dentro das tags <channel> ou <item>.

<item>

Esta tag se aplica a um canal, e especifica a linguagem usada por ele, como por exemplo,enus para o inglês americano.

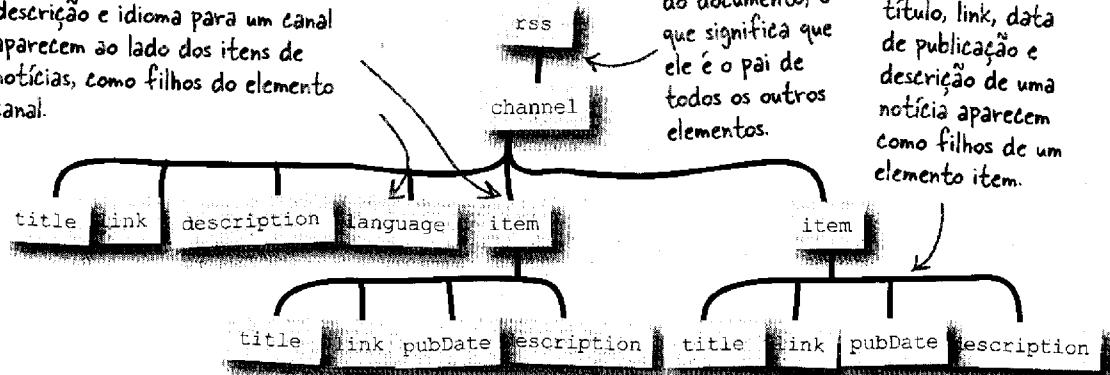
As tags <title>, <link>, <pubDate> e <description> são usadas dentro de <item> para descrever um item de notícias.

# RSS

## Visualizando XML

Você já aprendeu que o código XML consiste de tags, às quais são também conhecidas como elementos, que formam relacionamento pai - filho (parent - child), dentro do contexto de um documento XML completo. É bastante útil visualizar essa relação, à medida que você trabalha com código XML. Como um exemplo, o documento RSS da página anterior pode ser visualizado como uma hierarquia de elementos, mais ou menos como uma árvore genealógica dos dados do newsfeed, com os elementos "pais" no alto, descendo até os elementos "filhos".

Os elementos de título, link, descrição e idioma para um canal aparecem ao lado dos itens de notícias, como filhos do elemento canal.



**Exercício**

Segue abaixo um relato de abdução novinho em folha, que foi adicionado ao banco de dados aliens\_abduction. Escreva o código XML para uma tag <item> RSS para este relato, certificando-se de obedecer ao formato RSS para newsfeeds.

### aliens\_abduction

| abduction_id | first_name | last_name | when_it_happened | how_long | how_many | alien_description                 | what_they_did                                              | ... |
|--------------|------------|-----------|------------------|----------|----------|-----------------------------------|------------------------------------------------------------|-----|
| 14           | Shill      | Watner    | 2008-07-05       | 2 horas  | não sei  | uma luz<br>brilhante no<br>céu... | Eles me<br>transportaram<br>até um posto<br>de gasolina... | ... |

## exercício solução



### Exercício Solução

Segue abaixo um relato de abdução novinho em folha, que foi adicionado ao banco de dados aliens\_abduction. Escreva o código XML para uma tag <item> RSS para este relato, certificando-se de obedecer ao formato RSS para newsfeeds.

#### aliens\_abduction

| incident_id | first_name | last_name | when_it_happened | how_long | how_many | alien_description                     | where_it_did                                       | ... |
|-------------|------------|-----------|------------------|----------|----------|---------------------------------------|----------------------------------------------------|-----|
| ...         |            |           |                  |          |          |                                       |                                                    |     |
| 14          | Shill      | Watner    | 2008-07-05       | 2 horas  | não sei  | Apareceu uma luz brilhante no céu.... | Eles me transportaram até um posto de gasolina.... | ... |

A tag <item> delimita o item de notícias.  
 As tags <title>, <link>, <pubDate> e <description> descrevem os detalhes de cada item de news.  
 <item>  
 <title>Shill Watner - Apareceu uma luz brilhante no céu.</title>  
 <link>http://www.aliensabductedme.com/index.php?abduction\_id=14</link> A tag <pubDate> precisa ser escrita com o D maiúsculo, portanto você não pode escrever <pubdate>, nem <PUBDATE>.  
 <pubDate>Sat, 05 Jul 2008 00:00:00 EST</pubDate>  
 <description>Eles me transportaram até um posto de gasolina...</description>  
 </item>

não existem

## Perguntas Ídiotas

P: O XML é case-sensitive?

R: Sim, a linguagem XML é case-sensitive, portanto faz diferença se o texto está em maiúsculas ou minúsculas quando você especifica tags e atributos XML. Um bom exemplo é a tag RSS <pubDate>, que precisa aparecer exatamente dessa forma, com o D maiúsculo. A maioria das tags XML é escrita totalmente em minúsculas ou então misturando minúsculas com maiúsculas.

P: E quanto ao espaço em branco? Como ele é tratado em XML?

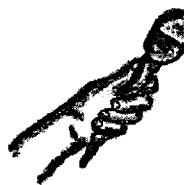
R: Antes de mais nada, o espaço em branco consiste, em XML, de carriage returns (\r), newlines (\n), tabs (\t) e espaços (' '). A maioria dos espaços em branco em quase todos os documentos XML serve apenas para melhor estética, como por exemplo, a indentação de tags subordinadas. Esse espaço em branco "insignificante" é ignorado pelas aplicações que processam dados XML, tais como os leitores de notícias RSS. Porém, o espaço em branco que apareça dentro de uma tag é considerado significativo, e geralmente é interpretado exatamente como aparece. Isso permite que coisas como poemas, onde existem espaços em branco significativos, possam ser representados de forma precisa em XML.

P: Um feed RSS pode conter imagens?

R: Sim. Apenas tenha em mente que nem todo leitor de notícias é capaz de exibi-las. Além disso, no RSS 2.0, você só pode adicionar imagens a um canal, e não a itens de notícias individuais. Para isso, usa-se a tag <image>, que precisa aparecer dentro da tag <channel>. Eis um exemplo:

```
<image>
  <url>http://www.aliensabductedme.com/fang.jpg</url>
  <title>My dog Fang</title>
  <link>http://www.aliensabductedme.com</link>
</image>
```

É tecnicamente possível incluir uma imagem em um item de notícias no RSS 2.0; o truque é usar a tag HTML <img> dentro da descrição do item. Embora seja possível, isso requer que você codifique a tag HTML usando entidades XML e, de muitas formas, vai contra a premissa dos itens RSS serem conteúdo de puro texto.



## RSS Revelado

**Na entrevista desta semana:  
O que interessa a um comunicador**

**Use a Cabeça:** Fiquei sabendo que, quando as pessoas procuram por notícias na web, elas consultam você. Isso é verdade?

**RSS:** Suponho que depende do que você considera como "notícias". A minha função principal é empacotar informações em um formato prontamente acessível para os leitores de notícias. Agora, se esse conteúdo qualifica como "notícias" ou não... é algo fora do meu controle. Fica a cargo das pessoas decidirem.

**Use a Cabeça:** E quando se refere a "leitores de notícias", você está falando dos indivíduos que lêem o conteúdo, certo?

**RSS:** Não, eu me refiro a ferramentas de software que entendem o que eu sou e como represento os dados. Por exemplo, vários programas de email têm suporte ao meu formato, o que significa que você pode assinar um notíciasfeed e receber atualizações praticamente da mesma forma como recebe mensagens de email.

**Use a Cabeça:** Interessante. Mas então de que formas você é diferente do email?

**RSS:** Ah, eu sou muito diferente do email. Em primeiro lugar, as mensagens de email são enviadas de pessoa para pessoa, e geralmente fazem parte de um diálogo bilateral. Assim, você pode responder a uma mensagem de email, receber uma resposta a ela, etc. Eu apenas comunico unilateralmente, de um site web para um indivíduo.

**Use a Cabeça:** Como isso torna a comunicação unilateral?

**RSS:** Bem, quando uma pessoa decide receber um notíciasfeed, assinando-o no seu software leitor de notícias, ela está dizendo basicamente que deseja ser informada sobre novos conteúdos que sejam publicados em um dado site. Quando esse novo conteúdo é publicado, eu me certifico de que ele seja traduzido para um formato tal que o software leitor de notícias fique sabendo que ele existe, e o mostre para a pessoa. Mas essa pessoa não tem a chance de responder a um item de notícias, e por isso se trata de uma comunicação unilateral, de um site para um indivíduo.

**Use a Cabeça:** Entendi. Então o que você é exatamente?

**RSS:** Na realidade sou apenas um formato de dados, uma forma de armazenar conteúdo de modo que ele possa ser reconhecido e

consumido por leitores de notícias. Use-me para armazenar dados, e os leitores de notícias poderão acessá-los na forma de notíciasfeeds.

**Use a Cabeça:** Ok, então de que formas você é diferente do HTML?

**RSS:** Bem, ambos somos formatos de dados em modo texto que se baseiam, em última análise, no XML, o que significa que ambos usamos tags e atributos para descrever dados. Porém, enquanto que o HTML foi elaborado especificamente para ser processado e interpretado por navegadores web, eu fui criado para ser processado e interpretado por leitores de notícias. Você poderia dizer que nós oferecemos diferentes pontos de vista sobre os mesmos dados.

**Use a Cabeça:** Mas eu já vi alguns navegadores web capazes de exibir notíciasfeeds. Como isso funciona?

**RSS:** Boa pergunta. Ocorre que alguns navegadores incluem leitores de notícias internos, portanto, eles na verdade, são duas ferramentas em uma só. Mas quando você visualiza um notíciasfeed em um navegador, está olhando para algo completamente diferente de uma página HTML.

**Use a Cabeça:** Mas a maioria dos notíciasfeeds tem links para páginas HTML, certo?

**RSS:** É verdade. Portanto eu trabalho lado a lado com o HTML para fornecer um melhor acesso a conteúdos web. A ideia é que você pode me usar para saber mais sobre novos conteúdos sem precisar ir até o site diretamente. Então, se vir alguma coisa sobre a qual queria ler mais, você clica para acessar a página propriamente dita. É por isso que cada item de notícias tem um link.

**Use a Cabeça:** Então você é uma espécie de preview para páginas web.

**RSS:** Sim, algo assim. Mas lembre-se de que eu vou até você, você não precisa vir até mim. É isso que as pessoas realmente apreciam a meu respeito – eu lhes economizo o trabalho de ter de revisitar os sites para se manterem informadas sobre novos conteúdos.

**Use a Cabeça:** Entendi. Isso realmente é conveniente. Obrigado por nos esclarecer sobre o seu papel na Web.

**RSS:** O prazer foi meu. Até mais.

## Gere um feed RSS dinamicamente

Entender o formato de dados do RSS é muito bom, mas Owen ainda precisa de um newsfeed para levar os relatos de abduções alienígenas para as pessoas. É hora de colocar o PHP em ação e gerar dinamicamente um newsfeed cheio de dados de abduções, retirados do banco de dados MySQL de Owen. Isso pode ser realizado através de uma série de passos:

O newsfeed resultante não é armazenado em um arquivo, mas sim em um documento XML.

### 1 Definir o tipo de conteúdo do documento como XML.

```
<?php header('Content-Type: text/xml'); ?>
```

Nós temos de definir o tipo de conteúdo do documento RSS como XML, usando um cabeçalho.



### 2 Gerar a diretiva XML para indicar que este é um documento XML.

```
<?php echo '<?xml version="1.0" encoding="utf-8"?>'; ?>
```

### 3 Gerar o código RSS estático que não vem do banco de dados, como a tag <rss> e as informações do canal.

```
<rss version="2.0">
  <channel>
    <title>...
    <link>...
    <description>...
    <language>...
```

Este código não é afetado pelo banco de dados - ele é sempre o mesmo para este newsfeed.

### 4 Consultar o banco de dados aliens\_abduction para obter os dados.

```
abduction_id
        first_name   last_name
when_it_happened      alien_description
                    what_they_did
```



Antes de gerar o código RSS para itens de notícias, precisamos consultar o banco de dados MySQL para obter os dados referentes às abduções.

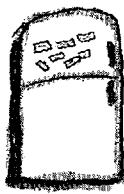
### 5 Fazer um loop através dos dados, gerando código RSS para cada item de notícias.

```
<item>
  <title>...
  <link>...
  <pubDate>...
  <description>...
</item>
```

Este código contém dados extraídos do banco, e portanto, ele precisa ser elaborado cuidadosamente.

### 6 Gerar o código RSS estático necessário para finalizar o documento, incluindo as tags de fechamento </channel> e </rss>.

```
</channel>
</rss>
```



# Ímãs de Geladeira do PHP & MySQL (& XML!)

O script de newsfeed RSS de Owen (newsfeed.php) ainda precisa de algumas partes importantes do código. Escolha com cuidado os ímãs apropriados para completar o código e gerar o newsfeed dinamicamente.

```

1 <?php header('Content-Type: text/xml'); ?>
2 <?php echo '<?xml version="1.0" encoding="utf-8"?>'; ?>
<rss version="2.0">
.....
<title>Fui Abduzido por Aliens - Newsfeed</title>
..... http://aliensabductedme.com/ .....
3 <description>Relatos de abduções alienígenas em todo o mundo, cortesia de Owen e seu
cão abduzido Fang.</description>
..... en-us ......

<?php
require_once('connectvars.php');

// Conecta-se ao banco de dados
$dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

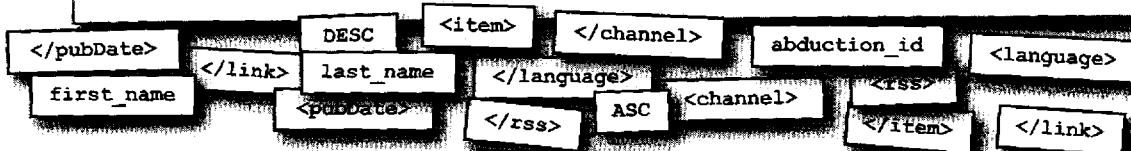
// Obtém os dados dos avistamentos de OVNIs a partir do MySQL
$query = "SELECT abduction_id, first_name, last_name, "
    . "DATE_FORMAT(when_it_happened, '%a, %d %b %Y %T') AS when_it_happened_rfc, "
    . "alien_description, what_they_did "
    . "FROM aliens_abduction "
    . "ORDER BY when_it_happened .....";
$data = mysqli_query($dbc, $query);

// Faz um loop através do array dos dados dos avistamentos, formatando-os como RSS
while ($row = mysqli_fetch_array($data)) {
    // Exibe cada linha como um item RSS
    echo '.....';
    echo ' <title>' . $row['first_name'] . ' ' . $row['last_name'] . ' - ' .
        substr($row['alien_description'], 0, 32) . '...</title>';
    echo ' <link>http://www.aliensabductedme.com/index.php?abduction_id=' .
        $row['.....'] . '</link>';
    echo ' .....' . $row['when_it_happened_rfc'] . ' ' . date('T') . ' .....';
    echo ' <description>' . $row['what_they_did'] . '</description>';
    echo '</item>';
}
?>
</channel>
.....

```



newsfeed.php



## Ímãs do php & mysql & xml Solução



# Ímãs de Geladeira do PHP & MySQL (& XML!) - Solução

O script de newsfeed RSS de Owen (newsfeed.php) ainda precisa de algumas partes importantes do código. Escolha com cuidado os ímãs apropriados para completar o código e gerar o newsfeed dinamicamente.

```
<?php header('Content-Type: text/xml'); ?> ← De forma semelhante ao que usamos anteriormente no exemplo do CAPTCHA para fazer output da imagem PNG, este cabeçalho faz com que o output do script seja um documento XML
<?php echo '<?xml version="1.0" encoding="utf-8"?>'; ?>
<rss version="2.0">

<channel>
    <title>Fui Abduzido por Aliens - Newsfeed</title>
    <link> http://aliensabductedme.com/ ... </link>
    <description>Relatos de abduções alienígenas em todo o mundo, cortesia de Owen e seu cão abduzido Fang.</description>
    <language> pt-BR... </language>

<?php
require_once('connectvars.php');

// Conecta-se ao banco de dados
$dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);

// Obtém os dados dos avistamentos de OVNIs a partir do MySQL
$query = "SELECT abduction_id, first_name, last_name, ".
    "DATE_FORMAT(when_it_happened, '%a, %d %b %Y %T') AS when_it_happened_rfc, ".
    "'alien_description', what_they_did".
    "FROM aliens_abduction ".
    "ORDER BY when_it_happened DESC ";
$data = mysqli_query($dbc, $query);

// Faz um loop através do array dos dados dos avistamentos, formatando-os como RSS
while ($row = mysqli_fetch_array($data)) {
    // Exibe cada linha como um item RSS

    echo ' <item> ';
    echo ' <title>' . $row['first_name'] . ' ' . $row['last_name'] . ' - ' .
        substr($row['alien_description'], 0, 32) . '...</title>';
    echo ' <link>http://www.aliensabductedme.com/index.php?abduction_id=' .
        $row['abduction_id'] . '</link>';
    echo ' <pubDate>' . $row['when_it_happened_rfc'] . ' ' . date(' ', ..... </pubDate>';

    echo ' <description>' . $row['what_they_did'] . '</description>';
    echo '</item>';

}
?>

</channel>
</rss>
```

first\_name

last\_name

</channel>

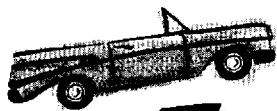
ASC

<rss>

</item>



newsfeed.php



## TEST DRIVE

### Adicione o script do Newsfeed RSS à página Fui Abduzido por Aliens.

Crie um arquivo de texto chamado newsfeed.php e digite o código para o script de Newsfeed RSS de Owen, que você viu no exercício dos Ímãs (ou baixe o script no site da Alta Books, em [www.altabooks.com.br](http://www.altabooks.com.br)).

Envie o script para o seu servidor web, e então abra-o em um leitor de news. A maioria dos navegadores web e alguns clientes de email lhe permitem visualizar newsfeeds, portanto você pode tentar um deles primeiro, caso não tenha um aplicativo de leitura de notícias (newsread) autônomo. O script Newsfeed deverá mostrar as abduções mais recentes, retiradas diretamente do banco de dados.

Se o seu navegador encontrar dificuldades para visualizar o newsfeed, tente usar feed:// na URL, no lugar de http://.

**Aliens Abducted Me - Newsfeed**

8 Total

| Search Articles:                |  |
|---------------------------------|--|
| <input type="text"/> G          |  |
| Article Length:                 |  |
| <input type="checkbox"/> All    |  |
| <input type="checkbox"/> 1000   |  |
| <input type="checkbox"/> 2000   |  |
| <input type="checkbox"/> 3000   |  |
| <input type="checkbox"/> 4000   |  |
| <input type="checkbox"/> 5000   |  |
| <input type="checkbox"/> 6000   |  |
| <input type="checkbox"/> 7000   |  |
| <input type="checkbox"/> 8000   |  |
| <input type="checkbox"/> 9000   |  |
| <input type="checkbox"/> 10000  |  |
| <input type="checkbox"/> 11000  |  |
| <input type="checkbox"/> 12000  |  |
| <input type="checkbox"/> 13000  |  |
| <input type="checkbox"/> 14000  |  |
| <input type="checkbox"/> 15000  |  |
| <input type="checkbox"/> 16000  |  |
| <input type="checkbox"/> 17000  |  |
| <input type="checkbox"/> 18000  |  |
| <input type="checkbox"/> 19000  |  |
| <input type="checkbox"/> 20000  |  |
| <input type="checkbox"/> 21000  |  |
| <input type="checkbox"/> 22000  |  |
| <input type="checkbox"/> 23000  |  |
| <input type="checkbox"/> 24000  |  |
| <input type="checkbox"/> 25000  |  |
| <input type="checkbox"/> 26000  |  |
| <input type="checkbox"/> 27000  |  |
| <input type="checkbox"/> 28000  |  |
| <input type="checkbox"/> 29000  |  |
| <input type="checkbox"/> 30000  |  |
| <input type="checkbox"/> 31000  |  |
| <input type="checkbox"/> 32000  |  |
| <input type="checkbox"/> 33000  |  |
| <input type="checkbox"/> 34000  |  |
| <input type="checkbox"/> 35000  |  |
| <input type="checkbox"/> 36000  |  |
| <input type="checkbox"/> 37000  |  |
| <input type="checkbox"/> 38000  |  |
| <input type="checkbox"/> 39000  |  |
| <input type="checkbox"/> 40000  |  |
| <input type="checkbox"/> 41000  |  |
| <input type="checkbox"/> 42000  |  |
| <input type="checkbox"/> 43000  |  |
| <input type="checkbox"/> 44000  |  |
| <input type="checkbox"/> 45000  |  |
| <input type="checkbox"/> 46000  |  |
| <input type="checkbox"/> 47000  |  |
| <input type="checkbox"/> 48000  |  |
| <input type="checkbox"/> 49000  |  |
| <input type="checkbox"/> 50000  |  |
| <input type="checkbox"/> 51000  |  |
| <input type="checkbox"/> 52000  |  |
| <input type="checkbox"/> 53000  |  |
| <input type="checkbox"/> 54000  |  |
| <input type="checkbox"/> 55000  |  |
| <input type="checkbox"/> 56000  |  |
| <input type="checkbox"/> 57000  |  |
| <input type="checkbox"/> 58000  |  |
| <input type="checkbox"/> 59000  |  |
| <input type="checkbox"/> 60000  |  |
| <input type="checkbox"/> 61000  |  |
| <input type="checkbox"/> 62000  |  |
| <input type="checkbox"/> 63000  |  |
| <input type="checkbox"/> 64000  |  |
| <input type="checkbox"/> 65000  |  |
| <input type="checkbox"/> 66000  |  |
| <input type="checkbox"/> 67000  |  |
| <input type="checkbox"/> 68000  |  |
| <input type="checkbox"/> 69000  |  |
| <input type="checkbox"/> 70000  |  |
| <input type="checkbox"/> 71000  |  |
| <input type="checkbox"/> 72000  |  |
| <input type="checkbox"/> 73000  |  |
| <input type="checkbox"/> 74000  |  |
| <input type="checkbox"/> 75000  |  |
| <input type="checkbox"/> 76000  |  |
| <input type="checkbox"/> 77000  |  |
| <input type="checkbox"/> 78000  |  |
| <input type="checkbox"/> 79000  |  |
| <input type="checkbox"/> 80000  |  |
| <input type="checkbox"/> 81000  |  |
| <input type="checkbox"/> 82000  |  |
| <input type="checkbox"/> 83000  |  |
| <input type="checkbox"/> 84000  |  |
| <input type="checkbox"/> 85000  |  |
| <input type="checkbox"/> 86000  |  |
| <input type="checkbox"/> 87000  |  |
| <input type="checkbox"/> 88000  |  |
| <input type="checkbox"/> 89000  |  |
| <input type="checkbox"/> 90000  |  |
| <input type="checkbox"/> 91000  |  |
| <input type="checkbox"/> 92000  |  |
| <input type="checkbox"/> 93000  |  |
| <input type="checkbox"/> 94000  |  |
| <input type="checkbox"/> 95000  |  |
| <input type="checkbox"/> 96000  |  |
| <input type="checkbox"/> 97000  |  |
| <input type="checkbox"/> 98000  |  |
| <input type="checkbox"/> 99000  |  |
| <input type="checkbox"/> 100000 |  |
| <input type="checkbox"/> 101000 |  |
| <input type="checkbox"/> 102000 |  |
| <input type="checkbox"/> 103000 |  |
| <input type="checkbox"/> 104000 |  |
| <input type="checkbox"/> 105000 |  |
| <input type="checkbox"/> 106000 |  |
| <input type="checkbox"/> 107000 |  |
| <input type="checkbox"/> 108000 |  |
| <input type="checkbox"/> 109000 |  |
| <input type="checkbox"/> 110000 |  |
| <input type="checkbox"/> 111000 |  |
| <input type="checkbox"/> 112000 |  |
| <input type="checkbox"/> 113000 |  |
| <input type="checkbox"/> 114000 |  |
| <input type="checkbox"/> 115000 |  |
| <input type="checkbox"/> 116000 |  |
| <input type="checkbox"/> 117000 |  |
| <input type="checkbox"/> 118000 |  |
| <input type="checkbox"/> 119000 |  |
| <input type="checkbox"/> 120000 |  |
| <input type="checkbox"/> 121000 |  |
| <input type="checkbox"/> 122000 |  |
| <input type="checkbox"/> 123000 |  |
| <input type="checkbox"/> 124000 |  |
| <input type="checkbox"/> 125000 |  |
| <input type="checkbox"/> 126000 |  |
| <input type="checkbox"/> 127000 |  |
| <input type="checkbox"/> 128000 |  |
| <input type="checkbox"/> 129000 |  |
| <input type="checkbox"/> 130000 |  |
| <input type="checkbox"/> 131000 |  |
| <input type="checkbox"/> 132000 |  |
| <input type="checkbox"/> 133000 |  |
| <input type="checkbox"/> 134000 |  |
| <input type="checkbox"/> 135000 |  |
| <input type="checkbox"/> 136000 |  |
| <input type="checkbox"/> 137000 |  |
| <input type="checkbox"/> 138000 |  |
| <input type="checkbox"/> 139000 |  |
| <input type="checkbox"/> 140000 |  |
| <input type="checkbox"/> 141000 |  |
| <input type="checkbox"/> 142000 |  |
| <input type="checkbox"/> 143000 |  |
| <input type="checkbox"/> 144000 |  |
| <input type="checkbox"/> 145000 |  |
| <input type="checkbox"/> 146000 |  |
| <input type="checkbox"/> 147000 |  |
| <input type="checkbox"/> 148000 |  |
| <input type="checkbox"/> 149000 |  |
| <input type="checkbox"/> 150000 |  |
| <input type="checkbox"/> 151000 |  |
| <input type="checkbox"/> 152000 |  |
| <input type="checkbox"/> 153000 |  |
| <input type="checkbox"/> 154000 |  |
| <input type="checkbox"/> 155000 |  |
| <input type="checkbox"/> 156000 |  |
| <input type="checkbox"/> 157000 |  |
| <input type="checkbox"/> 158000 |  |
| <input type="checkbox"/> 159000 |  |
| <input type="checkbox"/> 160000 |  |
| <input type="checkbox"/> 161000 |  |
| <input type="checkbox"/> 162000 |  |
| <input type="checkbox"/> 163000 |  |
| <input type="checkbox"/> 164000 |  |
| <input type="checkbox"/> 165000 |  |
| <input type="checkbox"/> 166000 |  |
| <input type="checkbox"/> 167000 |  |
| <input type="checkbox"/> 168000 |  |
| <input type="checkbox"/> 169000 |  |
| <input type="checkbox"/> 170000 |  |
| <input type="checkbox"/> 171000 |  |
| <input type="checkbox"/> 172000 |  |
| <input type="checkbox"/> 173000 |  |
| <input type="checkbox"/> 174000 |  |
| <input type="checkbox"/> 175000 |  |
| <input type="checkbox"/> 176000 |  |
| <input type="checkbox"/> 177000 |  |
| <input type="checkbox"/> 178000 |  |
| <input type="checkbox"/> 179000 |  |
| <input type="checkbox"/> 180000 |  |
| <input type="checkbox"/> 181000 |  |
| <input type="checkbox"/> 182000 |  |
| <input type="checkbox"/> 183000 |  |
| <input type="checkbox"/> 184000 |  |
| <input type="checkbox"/> 185000 |  |
| <input type="checkbox"/> 186000 |  |
| <input type="checkbox"/> 187000 |  |
| <input type="checkbox"/> 188000 |  |
| <input type="checkbox"/> 189000 |  |
| <input type="checkbox"/> 190000 |  |
| <input type="checkbox"/> 191000 |  |
| <input type="checkbox"/> 192000 |  |
| <input type="checkbox"/> 193000 |  |
| <input type="checkbox"/> 194000 |  |
| <input type="checkbox"/> 195000 |  |
| <input type="checkbox"/> 196000 |  |
| <input type="checkbox"/> 197000 |  |
| <input type="checkbox"/> 198000 |  |
| <input type="checkbox"/> 199000 |  |
| <input type="checkbox"/> 200000 |  |
| <input type="checkbox"/> 201000 |  |
| <input type="checkbox"/> 202000 |  |
| <input type="checkbox"/> 203000 |  |
| <input type="checkbox"/> 204000 |  |
| <input type="checkbox"/> 205000 |  |
| <input type="checkbox"/> 206000 |  |
| <input type="checkbox"/> 207000 |  |
| <input type="checkbox"/> 208000 |  |
| <input type="checkbox"/> 209000 |  |
| <input type="checkbox"/> 210000 |  |
| <input type="checkbox"/> 211000 |  |
| <input type="checkbox"/> 212000 |  |
| <input type="checkbox"/> 213000 |  |
| <input type="checkbox"/> 214000 |  |
| <input type="checkbox"/> 215000 |  |
| <input type="checkbox"/> 216000 |  |
| <input type="checkbox"/> 217000 |  |
| <input type="checkbox"/> 218000 |  |
| <input type="checkbox"/> 219000 |  |
| <input type="checkbox"/> 220000 |  |
| <input type="checkbox"/> 221000 |  |
| <input type="checkbox"/> 222000 |  |
| <input type="checkbox"/> 223000 |  |
| <input type="checkbox"/> 224000 |  |
| <input type="checkbox"/> 225000 |  |
| <input type="checkbox"/> 226000 |  |
| <input type="checkbox"/> 227000 |  |
| <input type="checkbox"/> 228000 |  |
| <input type="checkbox"/> 229000 |  |
| <input type="checkbox"/> 230000 |  |
| <input type="checkbox"/> 231000 |  |
| <input type="checkbox"/> 232000 |  |
| <input type="checkbox"/> 233000 |  |
| <input type="checkbox"/> 234000 |  |
| <input type="checkbox"/> 235000 |  |
| <input type="checkbox"/> 236000 |  |
| <input type="checkbox"/> 237000 |  |
| <input type="checkbox"/> 238000 |  |
| <input type="checkbox"/> 239000 |  |
| <input type="checkbox"/> 240000 |  |
| <input type="checkbox"/> 241000 |  |
| <input type="checkbox"/> 242000 |  |
| <input type="checkbox"/> 243000 |  |
| <input type="checkbox"/> 244000 |  |
| <input type="checkbox"/> 245000 |  |
| <input type="checkbox"/> 246000 |  |
| <input type="checkbox"/> 247000 |  |
| <input type="checkbox"/> 248000 |  |
| <input type="checkbox"/> 249000 |  |
| <input type="checkbox"/> 250000 |  |
| <input type="checkbox"/> 251000 |  |
| <input type="checkbox"/> 252000 |  |
| <input type="checkbox"/> 253000 |  |
| <input type="checkbox"/> 254000 |  |
| <input type="checkbox"/> 255000 |  |
| <input type="checkbox"/> 256000 |  |
| <input type="checkbox"/> 257000 |  |
| <input type="checkbox"/> 258000 |  |
| <input type="checkbox"/> 259000 |  |
| <input type="checkbox"/> 260000 |  |
| <input type="checkbox"/> 261000 |  |
| <input type="checkbox"/> 262000 |  |
| <input type="checkbox"/> 263000 |  |
| <input type="checkbox"/> 264000 |  |
| <input type="checkbox"/> 265000 |  |
| <input type="checkbox"/> 266000 |  |
| <input type="checkbox"/> 267000 |  |
| <input type="checkbox"/> 268000 |  |
| <input type="checkbox"/> 269000 |  |
| <input type="checkbox"/> 270000 |  |
| <input type="checkbox"/> 271000 |  |
| <input type="checkbox"/> 272000 |  |
| <input type="checkbox"/> 273000 |  |
| <input type="checkbox"/> 274000 |  |
| <input type="checkbox"/> 275000 |  |
| <input type="checkbox"/> 276000 |  |
| <input type="checkbox"/> 277000 |  |
| <input type="checkbox"/> 278000 |  |
| <input type="checkbox"/> 279000 |  |
| <input type="checkbox"/> 280000 |  |
| <input type="checkbox"/> 281000 |  |
| <input type="checkbox"/> 282000 |  |
| <input type="checkbox"/> 283000 |  |
| <input type="checkbox"/> 284000 |  |
| <input type="checkbox"/> 285000 |  |
| <input type="checkbox"/> 286000 |  |
| <input type="checkbox"/> 287000 |  |
| <input type="checkbox"/> 288000 |  |
| <input type="checkbox"/> 289000 |  |
| <input type="checkbox"/> 290000 |  |
| <input type="checkbox"/> 291000 |  |
| <input type="checkbox"/> 292000 |  |
| <input type="checkbox"/> 293000 |  |
| <input type="checkbox"/> 294000 |  |
| <input type="checkbox"/> 295000 |  |
| <input type="checkbox"/> 296000 |  |
| <input type="checkbox"/> 297000 |  |
| <input type="checkbox"/> 298000 |  |
| <input type="checkbox"/> 299000 |  |
| <input type="checkbox"/> 300000 |  |
| <input type="checkbox"/> 301000 |  |
| <input type="checkbox"/> 302000 |  |
| <input type="checkbox"/> 303000 |  |
| <input type="checkbox"/> 304000 |  |
| <input type="checkbox"/> 305000 |  |
| <input type="checkbox"/> 306000 |  |
| <input type="checkbox"/> 307000 |  |
| <input type="checkbox"/> 308000 |  |
| <input type="checkbox"/> 309000 |  |
| <input type="checkbox"/> 310000 |  |
| <input type="checkbox"/> 311000 |  |
| <input type="checkbox"/> 312000 |  |
| <input type="checkbox"/> 313000 |  |
| <input type="checkbox"/> 314000 |  |
| <input type="checkbox"/> 315000 |  |
| <input type="checkbox"/> 316000 |  |
| <input type="checkbox"/> 317000 |  |
| <input type="checkbox"/> 318000 |  |
| <input type="checkbox"/> 319000 |  |
| <input type="checkbox"/> 320000 |  |
| <input type="checkbox"/> 321000 |  |
| <input type="checkbox"/> 322000 |  |
| <input type="checkbox"/> 323000 |  |
| <input type="checkbox"/> 324000 |  |
| <input type="checkbox"/> 325000 |  |
| <input type="checkbox"/> 326000 |  |
| <input type="checkbox"/> 327000 |  |
| <input type="checkbox"/> 328000 |  |
| <input type="checkbox"/> 329000 |  |
| <input type="checkbox"/> 330000 |  |
| <input type="checkbox"/> 331000 |  |
| <input type="checkbox"/> 332000 |  |
| <input type="checkbox"/> 333000 |  |
| <input type="checkbox"/> 334000 |  |
| <input type="checkbox"/> 335000 |  |
| <input type="checkbox"/> 336000 |  |
| <input type="checkbox"/> 337000 |  |
| <input type="checkbox"/> 338000 |  |
| <input type="checkbox"/> 339000 |  |
| <input type="checkbox"/> 340000 |  |
| <input type="checkbox"/> 341000 |  |
| <input type="checkbox"/> 342000 |  |
| <input type="checkbox"/> 343000 |  |
| <input type="checkbox"/> 344000 |  |
| <input type="checkbox"/> 345000 |  |
| <input type="checkbox"/> 346000 |  |
| <input type="checkbox"/> 347000 |  |
| <input type="checkbox"/> 348000 |  |
| <input type="checkbox"/> 349000 |  |
| <input type="checkbox"/> 350000 |  |
| <input type="checkbox"/> 351000 |  |
| <input type="checkbox"/> 352000 |  |
| <input type="checkbox"/> 353000 |  |
| <input type="checkbox"/> 354000 |  |
| <input type="checkbox"/> 355000 |  |
| <input type="checkbox"/> 356000 |  |
| <input type="checkbox"/> 357000 |  |
| <input type="checkbox"/> 358000 |  |
| <input type="checkbox"/> 359000 |  |
| <input type="checkbox"/> 360000 |  |
| <input type="checkbox"/> 361000 |  |
| <input type="checkbox"/> 362000 |  |
| <input type="checkbox"/> 363000 |  |
| <input type="checkbox"/> 364000 |  |
| <input type="checkbox"/> 365000 |  |
| <input type="checkbox"/> 366000 |  |
| <input type="checkbox"/> 367000 |  |
| <input type="checkbox"/> 368000 |  |
| <input type="checkbox"/> 369000 |  |
| <input type="checkbox"/> 370000 |  |
| <input type="checkbox"/> 371000 |  |
| <input type="checkbox"/> 372000 |  |
| <input type="checkbox"/> 373000 |  |
| <input type="checkbox"/> 374000 |  |
| <input type="checkbox"/> 375000 |  |
| <input type="checkbox"/> 376000 |  |
| <input type="checkbox"/> 377000 |  |
| <input type="checkbox"/> 378000 |  |
| <input type="checkbox"/> 379000 |  |
| <input type="checkbox"/> 380000 |  |
| <input type="checkbox"/> 381000 |  |
| <input type="checkbox"/> 382000 |  |
| <input type="checkbox"/> 383000 |  |
| <input type="checkbox"/> 384000 |  |
| <input type="checkbox"/> 385000 |  |
| <input type="checkbox"/> 386000 |  |
| <input type="checkbox"/> 387000 |  |
| <input type="checkbox"/> 388000 |  |
| <input type="checkbox"/> 389000 |  |
| <input type="checkbox"/> 390000 |  |
| <input type="checkbox"/> 391000 |  |
| <input type="checkbox"/> 392000 |  |
| <input type="checkbox"/> 393000 |  |
| <input type="checkbox"/> 394000 |  |
| <input type="checkbox"/> 395000 |  |
| <input type="checkbox"/> 396000 |  |
| <input type="checkbox"/> 397000 |  |
| <input type="checkbox"/> 398000 |  |
| <input type="checkbox"/> 399000 |  |
| <input type="checkbox"/> 400000 |  |

**Basta colocar um link para ele na home page.**

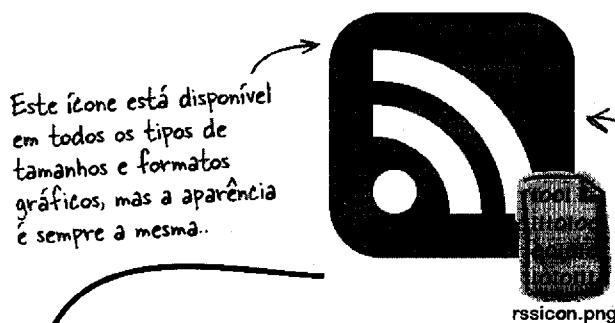
Não se esqueça de que newsfeed.php é um simples script PHP. A única diferença entre ele e a maioria dos outros scripts PHP que você viu ao longo deste livro é que ele gera um documento RSS, em vez de um documento HTML. Mas você ainda o acessa da mesma forma que qualquer outro script PHP – basta especificar o nome dele em uma URL. O que Owen ainda precisa é de uma forma de compartilhar essa URL com as pessoas que visitam o seu site. Isso é feito sem muito esforço, colocando-se um link para o script newsfeed.php, no servidor de Owen.



fornecendo um link rss

## Crie um link para o feed RSS

É importante fornecer um link bem chamativo para o newsfeed do seu site, porque muitos usuários irão apreciar que você esteja oferecendo esse tipo de serviço. Para ajudar os usuários a acharem facilmente o feed RSS do site, existe um ícone padrão que você pode usar para chamar a atenção para o feed. Podemos usar esse ícone para criar um link newsfeed na parte de baixo da home page (index.php) de Owen.



Este ícone está disponível em todos os tipos de tamanhos e formatos gráficos, mas a aparência é sempre a mesma.

```
<p>
<a href="newsfeed.php">
    
    Click to syndicate the abduction news feed.
</a>
</p>
```

A URL do newsfeed é apenas o script newsfeed.php, que só funcionará se o script estiver armazenado na mesma pasta que a página web principal.

Para baixar uma coleção de ícones RSS de diferentes cores e formatos, acesse [www.feedicons.com](http://feedicons.com).

O link HTML para o newsfeed inclui um ícone RSS e um texto desritivo.

**Aliens Abducted Me**

Welcome, have you had an encounter with extraterrestrials? Were you abducted? Have you seen my abducted dog, Fang?

Report it here!

Most recent reported abductions:

|                                                                                                             |                      |
|-------------------------------------------------------------------------------------------------------------|----------------------|
| 2008-08-10 : Meinhold Ressner                                                                               | Fang spotted:<br>no  |
| Abducted for: Alien description:<br>3 hours They were in a ship the size of a full moon.                    |                      |
| 2008-07-11 : Mickey Mikens                                                                                  | Fang spotted:<br>yes |
| Abducted for: Alien description:<br>45 minutes Huge heads, skinny arms and legs                             |                      |
| 2008-07-05 : Shill Wirtner                                                                                  | Fang spotted:<br>yes |
| Abducted for: Alien description:<br>2 hours There was a bright light in the sky, followed by a bark or two. |                      |
| 2008-06-21 : Belita Chevy                                                                                   | Fang spotted:<br>no  |
| Abducted for: Alien description:<br>almost a week Chumsy little buggers, had no rhythm.                     |                      |
| 2008-05-11 : Sally Jones                                                                                    | Fang spotted:<br>yes |
| Abducted for: Alien description:<br>1 day green with six tentacles                                          |                      |

Click to syndicate the abduction news feed.

Um link chamativo na página Fui Abduzido por Aliens oferece aos visitantes uma forma rápida de acessar o newsfeed de Owen.



## TEST DRIVE

### Adicione o link newsfeed à home page do site Fui Abduzido por Aliens.

Modifique o script index.php do site Fui Abduzido por Aliens, de modo que ele passe a ter um link para o newsfeed na parte de baixo da página. Baixe também a imagem rssicon.png, como parte do código para este capítulo, no site da Alta Books em [www.altabooks.com.br](http://www.altabooks.com.br).

Envie o script index.php e a imagem rssicon.php para o seu servidor web, e depois abra o script em um navegador. Clique no novo link para visualizar o newsfeed RSS.

**Com todas essas abduções acontecendo, estou sempre atenta para os OVNIs.**

**Eu não vi Fang, mas esses relatos são fascinantes.**

Graças ao RSS, os novos relatos de abduções são enviados para os assinantes, sem que eles tenham de visitar o site Fui Abduzido por Aliens diretamente.

Aliens Abducted Me - Newsfeed

Meinhold Ressner - They were in a s...

Carried me to the top of a mountain and dropped me off.

Read more...

Será que esse é o mesmo cão que eu vi naquele vídeo do YouTube...?

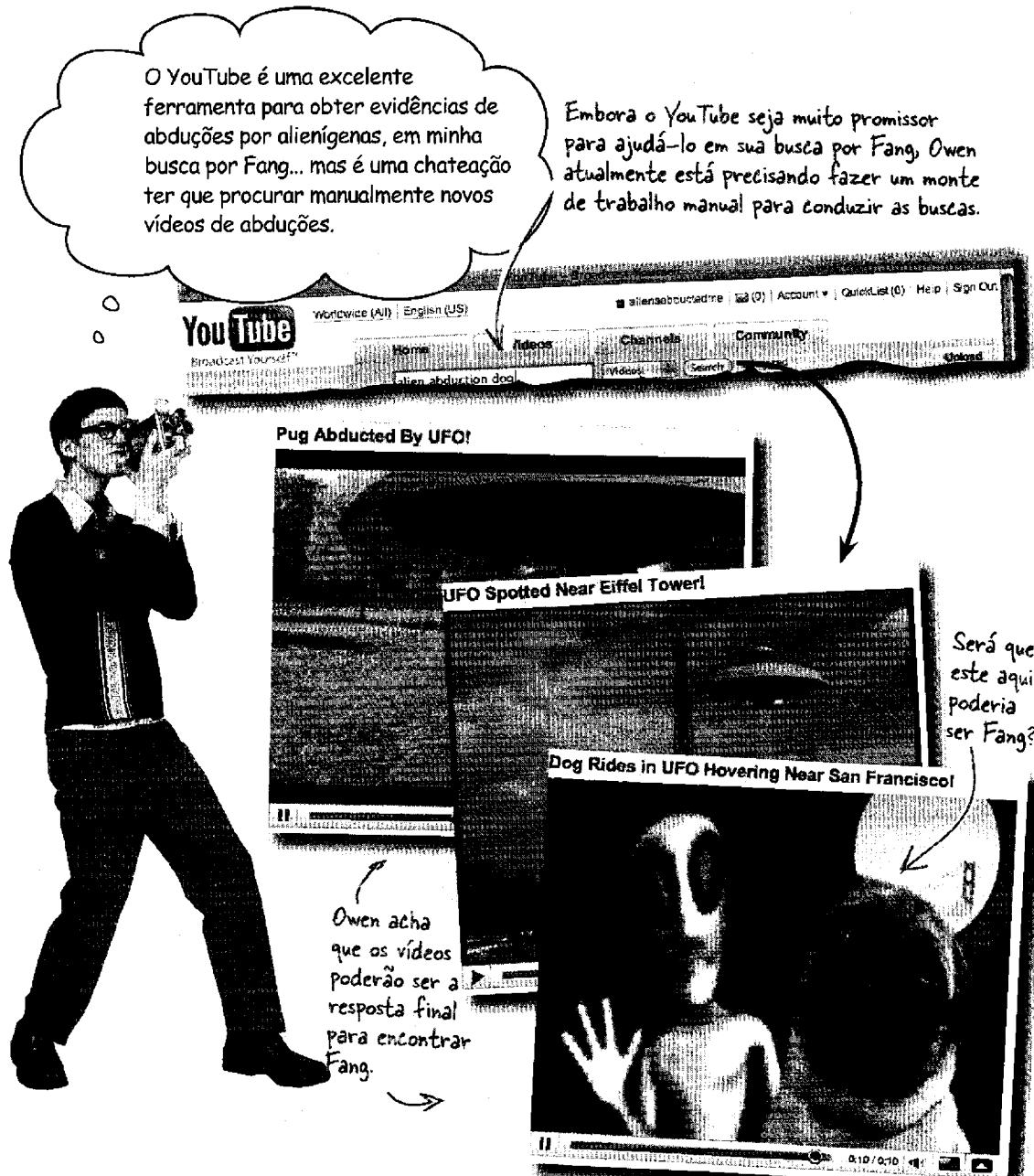
Esta é Chloe, ávida leitora do newsfeed do site Fui Abduzido por Aliens, que pensa ter visto Fang em um vídeo do YouTube.

Você está aqui ▶

adicionando conteúdo do youtube no site do oween

## Uma vídeo vale mais que milhões de palavras

Depois que uma assinante do newsfeed alertou Owen sobre um vídeo do YouTube com um cão parecido com Fang, Owen percebeu que é uma boa ideia usar outras tecnologias para ampliar a sua busca por Fang. Mas como? Se ele pudesse incorporar vídeos do YouTube no seu próprio site, todos os seus usuários poderiam ficar de olho para tentar encontrar Fang. E não apenas isso, mas ele realmente precisa bolar uma forma de não precisar ficar fazendo buscas manuais no YouTube o tempo todo.





- 1 Visite os vídeos de Owen no YouTube, acessando [www.youtube.com/user/aliensabductedme](http://www.youtube.com/user/aliensabductedme).
- 2 Assista a alguns dos vídeos de abduções que Owen encontrou. Você acha que o cachorro nos vídeos é Fang?

Não seria um sonho se eu pudesse ver os vídeos diretamente no site Fui Abduzido por Aliens, em vez de ter de procurar por eles no YouTube? Se pelo menos houvesse uma maneira de poder ir a uma página e ter a busca já feita para mim. Mas isso é apenas um sonho...



obter conteúdo é diferente de enviar conteúdo

## Obtendo conteúdo web de terceiros

A ideia central do newsfeed RSS é que ele envia o seu conteúdo para outras pessoas, para que elas não precisem ficar visitando o seu site constantemente, em busca de conteúdo novo. Essa é uma excelente forma de tornar as coisas mais convenientes para as pessoas se manterem em dia com o que acontece no seu site, como Owen já descobriu. Mas existe outro lado nessa tecnologia de transmissão, que envolve obter conteúdo a partir de outro site e colocá-lo no seu próprio. Dessa maneira, você se torna o consumidor, e outro site age como o provedor de conteúdo. No caso de Owen, que está colocando vídeos do YouTube em seu site, o YouTube se torna o fornecedor.

### O site Fui Abduzido por Aliens é o consumidor dos vídeos.

**O YouTube é o fornecedor dos vídeos.**

**Aliens Abducted Me**

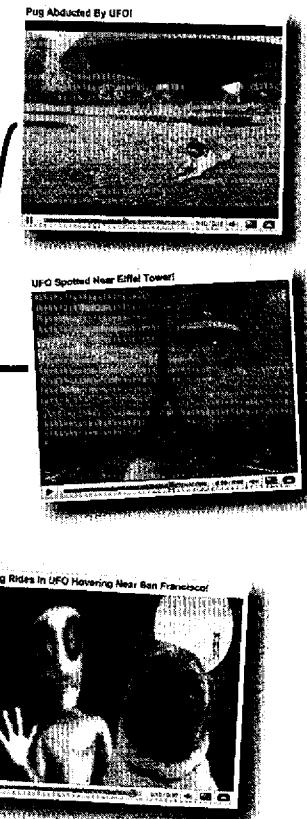
Welcome, have you had an encounter with extraterrestrials? Were you abducted? Have you seen my abducted dog, Fang? Report it here!

Most recent reported abductions:

|                               |                                                                 |                      |
|-------------------------------|-----------------------------------------------------------------|----------------------|
| 2008-08-10 : Meinhold Ressner | Alien description:<br>3 hours                                   | Fang spotted:<br>no  |
| Abducted for:                 | They were in a ship the size of a full moon.                    |                      |
| 2008-07-11 : Mickey Mikens    | Alien description:<br>45 minutes                                | Fang spotted:<br>yes |
| Abducted for:                 | Huge heads, skinny arms and legs                                |                      |
| 2008-07-05 : Shilf Wathne     | Alien description:<br>2 hours                                   | Fang spotted:<br>yes |
| Abducted for:                 | There was a bright light in the sky, followed by a bark or two. |                      |
| 2008-06-21 : Belita Chevy     | Alien description:<br>almost a week                             | Fang spotted:<br>no  |
| Abducted for:                 | Clumsy little buggers, had no rhythm.                           |                      |
| 2008-05-11 : Sally Jones      | Alien description:<br>1 day                                     | Fang spotted:<br>yes |
| Abducted for:                 | green with six tentacles                                        |                      |

Click to syndicate the abduction news feed.

Clique para ver os resultados da busca por vídeos.



O design da home page do site Fui Abduzido por Aliens terá de se modificar ligeiramente para acomodar os resultados da busca por vídeos.

É importante entender que Owen não quer simplesmente inserir um vídeo ou link para um vídeo, específico do YouTube. Isso é fácil de fazer, bastando copiar e colar algum código HTML disponibilizado pelo próprio YouTube. O que ele quer é realizar uma busca nos vídeos do YouTube e exibir os resultados dessa busca. Assim, o site Fui Abduzido por Aliens precisa realizar uma consulta em tempo real aos dados do YouTube, e depois exibir dinamicamente os resultados. Isso permite que Owen e seus assistentes na busca por Fang estejam sempre atualizados sobre vídeos de abduções que tenham sido postados no YouTube.

Os vídeos que aparecem como resultado de uma busca no YouTube são retornados por esse site e colocados na página principal de Owen..

# Republicando vídeos do YouTube

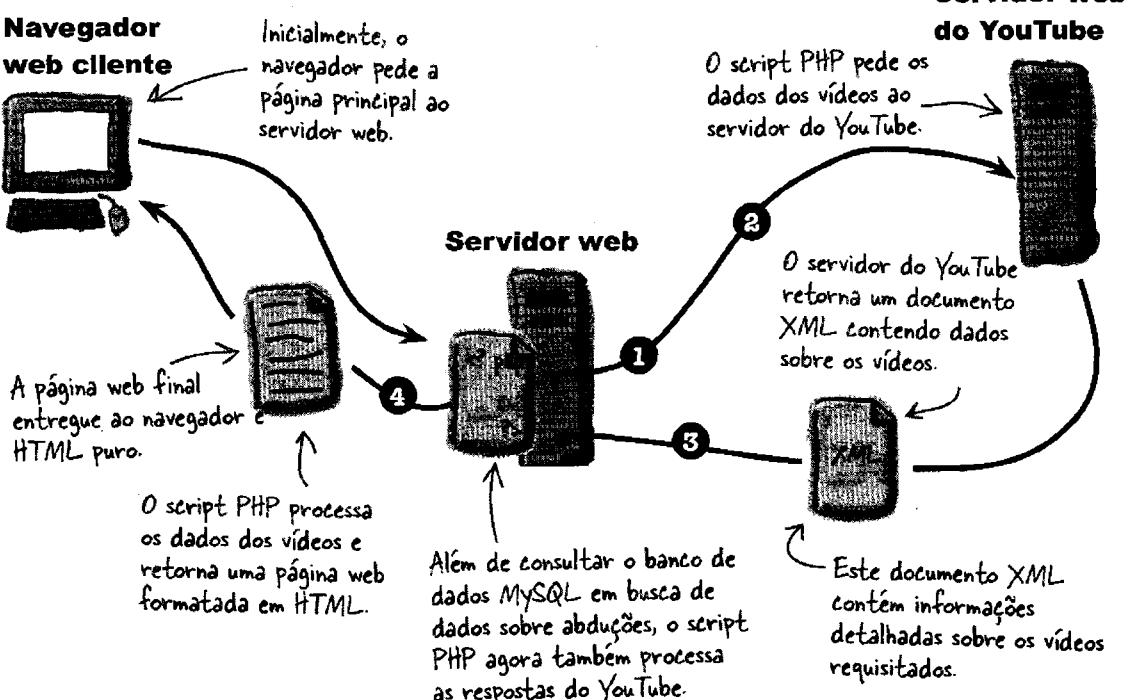
Para obter vídeos a partir do YouTube, nós precisamos aprender exatamente como esse site disponibiliza os vídeos para serem republicados em outros lugares. O YouTube oferece os vídeos através de um processo de requisição/resposta, onde você requisita determinados vídeos e então recebe informações sobre eles em uma resposta dos servidores do YouTube. Você é responsável tanto por emitir a requisição no formato esperado pelo YouTube quanto por manipular a resposta, o que inclui peneirar os dados da resposta para obter os dados específicos de que você precisa (título do vídeo, imagem thumbnail, link, etc.).

Seguem abaixo os passos necessários para se obter vídeos do YouTube e exibi-los:

- 1 Criar uma requisição por vídeos do YouTube.
- 2 Emitir a requisição do vídeo para o YouTube.
- 3 Receber a resposta do YouTube, contendo informações sobre os vídeos.
- 4 Processar os dados da resposta e formatá-los como código HTML.

Esta requisição  
frequentemente é feita  
na forma de uma URL.

O YouTube usa XML  
para responder a  
requisições por vídeos.



## Fazendo uma requisição de vídeo no YouTube

A operação de obter vídeos do YouTube e incorporá-los às suas próprias páginas web começa com uma requisição. O YouTube espera que os vídeos sejam consultados mediante o uso de uma requisição REST, que é uma URL personalizada que leva a recursos específicos, tais como dados de vídeos do YouTube. Você constrói uma URL identificando os vídeos que deseja receber, e então o YouTube retorna informações sobre eles através de um documento XML.

Os detalhes da URL para uma requisição ao YouTube são determinados por quais vídeos você deseja acessar. Por exemplo, você pode requisitar os vídeos favoritos de um determinado usuário. No caso de Owen, a melhor abordagem é provavelmente realizar uma busca com palavras-chaves por todos os vídeos do YouTube. A URL requerida para cada um desses tipos de requisições REST varia ligeiramente, mas a base dessa URL sempre começa desta forma:

`http://gdata.youtube.com/feeds/api/` ← Esta URL-base é usada para todas as requisições REST ao YouTube.

### Requisitando vídeos por usuário

Para se requisitar os vídeos favoritos de um determinado usuário do YouTube, é preciso fazer algumas adições à URL-base, dentre as quais o nome do usuário no YouTube.

`http://gdata.youtube.com/feeds/api/users/username/favorites`

Para requisitar os vídeos favoritos do usuário elmerpriestley, use a seguinte URL:

`http://gdata.youtube.com/feeds/api/users/elmerpriestley/favorites` ←

O nome de um usuário do YouTube fornece acesso aos vídeos favoritos desse usuário.

### Requisitando vídeos com uma busca por palavras-chaves

Um tipo mais poderoso e frequentemente mais útil de requisições de vídeos no YouTube é o que utiliza uma busca por palavras-chaves, independentemente dos usuários. Você pode usar mais de uma palavra-chave, desde que as separe por meio de barras, no final da URL

`http://gdata.youtube.com/feeds/api/videos/-/keyword1/keyword2/...`

A URL começa da mesma forma que quando requisitamos vídeos de usuários, mas aqui você usa "videos" em vez de "users".

Não se esqueça das barras e do hífen!

O resultado desta requisição REST são os vídeos favoritos do usuário elmerpriestley.

Múltiplas palavras-chaves podem ser usadas em uma busca por vídeos, separando-as com barras.

Para requisitar os vídeos correspondentes às palavras-chaves "elvis" e "imitador", use a seguinte URL:

`http://gdata.youtube.com/feeds/api/videos/-/elvis/impersonator`

As palavras-chaves são case-insensitive, de modo que "elvis", "Elvis" e "Elvls" retornam todas o mesmo resultado.

Aqui, as palavras-chaves de busca "elvis" e "imitador" são usadas para se procurar por vídeos.

## Perguntas Ídiotas

**P:** O que significa a sigla REST?

**R:** Representational State Transfer, ou "Transferência de Estado Representacional". Este é definitivamente um daqueles acrônimos que soam muito mais chiques e mais técnicos do que realmente são. A ideia central do REST é que os recursos web devem ficar acessíveis mediante links únicos, o que significa que você deve poder acessar dados REST simplesmente criando uma URL para eles. Em termos do YouTube, isso significa que você deve poder realizar consultas sobre vídeos puramente através de uma URL que contenha os critérios de busca.

## SINTA-SE como a Requisição REST do YouTube



Sua tarefa é entrar na mente do YouTube e se tornar uma requisição de vídeo REST. Use os ímãs abaixo para montar requisições REST para os seguintes vídeos do YouTube, e depois teste-as no seu navegador:

**Todos os vídeos contendo a palavra-chave “Roswell”:**

.....

**Todos os vídeos contendo as palavras-chaves “alien” e “abduction”:**

.....

**Todos os vídeos marcados como favoritos pelo usuário headfirstmork:**

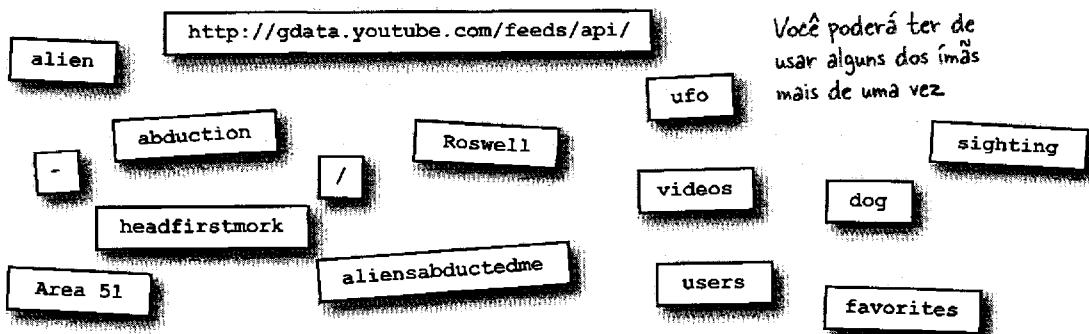
.....

**Todos os vídeos contendo as palavras-chaves “ufo”, “sighting” e “dog”:**

.....

**Todos os vídeos marcados como favoritos pelo usuário aliensabductedme:**

.....



sinta-se como a requisição REST do youtube solução

## SINTA-SE como a Requisição REST do YouTube Solução



Sua tarefa é entrar na mente do YouTube e se tornar uma requisição de vídeo REST. Use os ímãs abaixo para montar requisições REST para os seguintes vídeos do YouTube, e depois teste-as no seu navegador:

Você poderá ter usado alguns dos ímãs mais de uma vez.

A mesma URL-base do YouTube é usada para todas as requisições REST.

A palavra-chave aparece por último na URL.

Todos os vídeos contendo a palavra-chave "Roswell":

`http://gdata.youtube.com/feeds/api/` videos / - / Roswell

Cada uma das palavras-chaves de busca aparece no final da URL, separadas por barras.

Todos os vídeos contendo as palavras-chaves "alien" e "abduction":

`http://gdata.youtube.com/feeds/api/` videos / - / alien / abduction

Todos os vídeos marcados como favoritos pelo usuário headfirstmork:

`http://gdata.youtube.com/feeds/api/` users / headfirstmork / favorites

A URL para os favoritos de um usuário requer a palavra "users" aqui, em vez de "videos".

Todos os vídeos contendo as palavras-chaves "ufo", "sighting" e "dog":

`http://gdata.youtube.com/feeds/api/` videos / - / ufo / sighting / dog

A URL termina com a palavra "favoritos".

Todos os vídeos marcados como favoritos pelo usuário aliensabductedme:

`http://gdata.youtube.com/feeds/api/` users / aliensabductedme / favorites

Area 51

Este ímã não foi usado...  
é uma conspiração!

Este é o nome do usuário cujos vídeos favoritos você deseja acessar.

não existem

## Perguntas Idiotas

**Q:** Em que o REST é diferente, digamos, de uma requisição GET?

**A:** Ele não é. Sempre que você usa uma requisição GET, por exemplo, ao simplesmente requisitar uma página web, está usando REST. Você pode pensar nas páginas web normais como um recurso REST, no sentido de que ele pode ser acessado mediante uma URL, e GET é a "ação" REST usada para se acessar o recurso. O REST fica mais interessante quando é usado para criar consultas, por exemplo, as requisições por vídeos do YouTube. Neste caso, você ainda está lidando com requisições REST, mas elas estão consultando o banco em busca de dados, em vez de simplesmente pedirem uma página web estática.

**Q:** A ordem dos argumentos faz diferença, ao se realizar uma busca por palavras-chaves no YouTube?

**A:** Sim. As primeiras palavras-chaves recebem prioridade em relação às demais, portanto certifique-se de listá-las em ordem decrescente de importância.

**Q:** Quando são encontrados vários resultados para uma busca, como o YouTube determina quais vídeos serão retornados?

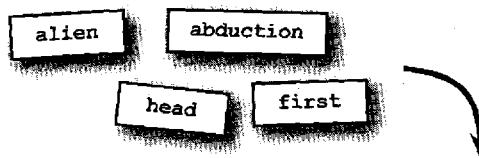
**A:** As buscas no YouTube retornam vídeos com base na relevância, o que significa que você obterá os vídeos que melhor tiverem correspondido às palavras-chaves, independentemente de quando esses vídeos foram postados no YouTube.

Estou pronto para  
ver os resultados...



## Owen está pronto para criar uma requisição REST

Uma vez que o objetivo de Owen é procurar por vídeos de abduções alienígenas nos quais Fang possa ter aparecido, uma busca por palavras-chaves é o tipo mais apropriado de requisição REST a ser submetida ao YouTube. Há diversas combinações de palavras-chaves que poderiam ser usadas para se procurar por possíveis vídeos de Fang, mas uma em particular ajudará a obter vídeos especificamente relacionados a ele:



<http://gdata.youtube.com/feeds/api/videos/-/alien/abduction/head/first>

Embora você provavelmente nunca fosse pensar em colocar o título de uma série de livros ao executar uma busca por vídeos no YouTube, isso é uma boa ideia, neste caso em particular. Digamos que, por coincidência, muitos dos vídeos de abduções foram feitos por fãs norte-americanos da série Use a Cabeça!, cujo título em inglês é Head First! Com a URL da requisição REST em mãos, Owen pode riscar o Passo 1 da lista de tarefas para republicação de vídeos do YouTube.

↑ ↑  
Estas duas últimas palavras-chaves lhe ajudarão a encontrar os vídeos de abduções relacionados com Owen e Fang!

O primeiro passo já  
foi, graças à URL de  
requisição no YouTube.

- ① Criar uma requisição por vídeos do YouTube
- ② Emitir a requisição do vídeo para o YouTube.
- ③ Receber a resposta do YouTube,  
contendo informações sobre os vídeos.
- ④ Processar os dados da resposta e  
formatá-los como código HTML.



## TEST DRIVE

### Digite a URL da requisição de Owen em um navegador:

Digite a URL da requisição de Owen em um navegador:

<http://gdata.youtube.com/feeds/api/videos/-/alien/abduction/head/first>

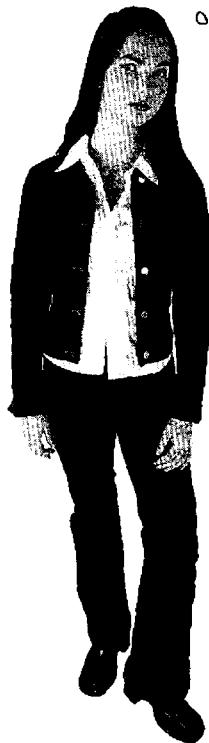
O que o navegador mostra? Tente visualizar o código-fonte da página para ver o código que foi retornado pelo YouTube.

The screenshot shows a web page titled "YouTube Videos" with a search bar and filters for "Search Articles", "Article Length", and "Sort By". The results list several items:

- UFO Spotted Crashing Party at Graceland...** alienabductedme Today, 12:54 AM  
In an unusual conspiratorial twist of fate, a UFO was spotted crashing a raucous party at Graceland, the home of the undisputed King of Rock 'n Roll.  
Read Head First PHP & MySQL!  
<http://www.headfirstlabs.com> [Read more...](#)
- Aliens Turn Face of Sphinx Into a Dog!** alienabductedme Today, 12:45 AM  
Watch as a UFO laser chisels the face of the Sphinx into a Pug.  
Read Head First PHP & MySQL!  
<http://www.headfirstlabs.com> [Read more...](#)
- Dog Rides in UFO Hovering Near San...** alienabductedme Yesterday, 10:08 PM  
This shocking video shows a pug taking a ride in a UFO as it flies around near the Golden Gate Bridge.  
[Read more...](#)
- UFO Spotted Near Eiffel Tower!** alienabductedme Yesterday, 10:03 PM  
Check out this video of a UFO zapping cats in Paris.  
Read Head First PHP & MySQL!  
<http://www.headfirstlabs.com> [Read more...](#)
- Pug Abducted By UFO!** alienabductedme Yesterday, 9:51 PM  
Help! My dog was abducted by aliens. [Read more...](#)

O navegador web vê os dados XML retornados pelo YouTube como um newsfeed, com a diferença que, neste caso, cada item é na verdade um vídeo.

## fazendo requisição REST no script php



Esse negócio de requisitar vídeos do YouTube digitando uma URL em um navegador é legal e tudo mais, mas o que isso tem a ver com o PHP? Por que não podemos acessar os resultados a partir de um script?

A extensão SimpleXML, que oferece a função `simplexml_load_file()`, foi adicionada ao PHP na versão 5. Assim, versões anteriores do PHP não têm suporte interno ao processamento de XML.

**Nós podemos; só precisamos de uma função PHP que nos permita submeter uma requisição REST e receber a resposta.**

A função `simplexml_load_file()` do PHP nos permite submeter requisições REST que resultam em respostas XML, como no caso das requisições/respostas do YouTube. A função, na verdade, carrega um documento XML para dentro de um objeto PHP, que podemos então usar para acessar os dados XML e extrair qualquer informação específica que queiramos. Então, como isso afeta a requisição de Owen no YouTube? Dê uma olhada neste código, que cria uma constante para armazenar uma URL do YouTube e depois emite uma requisição REST usando a função `simplexml_load_file()`:

```
define('YOUTUBE_URL', 'http://gdata.youtube.com/feeds/api/videos/-/alien/abduction/head/first');
```

```
$xml = simplexml_load_file(YOUTUBE_URL);
```

Embora não seja estritamente necessário, geralmente é uma boa ideia armazenar URLs estáticas em constantes, para que você saiba onde modificá-la, caso isso seja necessário.

- ① Criar uma requisição por vídeos do YouTube..
- ② Emitir a requisição do vídeo para o YouTube.
- ③ Receber a resposta do YouTube, contendo informações sobre os vídeos.
- ④ Processar os dados da resposta e formatá-los como código HTML.

Já realizamos  
estes dois passos!



Não se preocupe  
se você não souber  
o que é um objeto,  
principalmente no  
contexto do PHP.

Um objeto PHP é um tipo especial de dados que permite que sejam empacotados junto com funções em um único construto. Tudo o que precisa saber por ora é que é muito mais fácil processar dados XML no PHP usando objetos. Você aprenderá mais sobre como isso é possível logo mais adiante.

Este é o arquivo XML  
retornado pela função  
`simplexml_load_file()`,  
que consiste dos dados  
XML do YouTube  
referentes aos vídeos  
requisitados.

Ótimo... um problema ainda maior!  
O que é que nós vamos fazer com  
todos esses dados XML obscuros? Não  
é possível um script PHP entender isso  
aí



**Ah, é possível sim! O código XML retornado pelo YouTube na verdade não é tão obscuro quanto parece... você só precisa saber onde procurar.**

a requisição retorna xml

## O YouTube fala XML

O vídeo que você recebe na resposta do YouTube não é exatamente um DVD empacotado em uma caixinha de plástico e entregue na sua porta. Ele é, na verdade, um documento XML contendo informações detalhadas sobre os vídeos que você requisitou, mas não contendo os vídeos propriamente ditos.

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns='http://www.w3.org/2005/Atom'
      xmlns:openSearch='http://a9.com/-/spec/opensearchrss/1.0/'
      xmlns:gml='http://www.opengis.net/gml'
      xmlns:georss='http://www.georss.org/georss'
      xmlns:media='http://search.yahoo.com/mrss/'
      xmlns:batch='http://schemas.google.com/gdata/batch'
      xmlns:yt='http://gdata.youtube.com/schemas/2007'
      xmlns:gd='http://schemas.google.com/g/2005'>
  <id>http://gdata.youtube.com/feeds/api/users/aliensabductedme/favorites</id>
  <updated>2008-07-25T03:22:37.001Z</updated>
  <category scheme='http://schemas.google.com/g/2005#kind'
            term='http://gdata.youtube.com/schemas/2007#video'/>
  <title type='text'>Favorites of aliensabductedme</title>
  ...
  <entry>
    <id>http://gdata.youtube.com/feeds/api/videos/_6Uibqf0vtA</id>
    <published>2006-06-20T07:59:05.000-07:00</published>

    <media:group>
      <media:title type='plain'>UFO Sighting in Yosemite Park near Area 51</media:title>
      <media:description type='plain'>I went on a trip to Yosemite Park in 2007. Yosemite Park is very close to the border between California and Nevada, and close to Area 51...</media:description>
      <media:keywords>51, alien, aliens, area, ca, california, nevada, sighting, sightings, ufo</media:keywords>
      <yt:duration seconds='50' />
      <media:category label='Travel &amp; Events' scheme='http://gdata.youtube.com/schemas/2007/categories.cat'>Travel</media:category>
      <media:content url='http://www.youtube.com/v/_6Uibqf0vtA' type='application/x-shockwave-medium='video' isDefault='true' expression='full' duration='50' yt:format='5' />
      <media:content url='rtsp://rtsp2.youtube.com/CholEhNy73wIAzOnQvvSnbikI_xMYESARF0GDA==/0/video.3gp' type='video/3gp' medium='video' expression='full' duration='50' yt:format='1' />
      <media:content url='rtsp://rtsp2.youtube.com/CholEhNy73wIAzOnQvvSnbikI_xMYESARF0GDA==/0/video.3gp' type='video/3gp' medium='video' expression='full' duration='50' yt:format='6' />
      <media:player url='http://www.youtube.com/watch?v=_6Uibqf0vtA' />
      <media:thumbnail url='http://img.youtube.com/vi/_6Uibqf0vtA/2.jpg' height='97' width='130' time='00:00:25' />
      <media:thumbnail url='http://img.youtube.com/vi/_6Uibqf0vtA/1.jpg' height='97' width='130' time='00:00:12.500' />
      <media:thumbnail url='http://img.youtube.com/vi/_6Uibqf0vtA/3.jpg' height='97' width='130' time='00:00:37.500' />
      <media:thumbnail url='http://img.youtube.com/vi/_6Uibqf0vtA/0.jpg' height='240' width='32' time='00:00:25' />
    </media:group>
    <yt:statistics viewCount='2278159' favoriteCount='1897' />
    <gd:rating min='1' max='5' numRaters='11602' average='4.17' />
    <gd:comments>
      <gd:feedLink href='http://gdata.youtube.com/feeds/api/videos/_6Uibqf0vtA/comments' count='4426' />
    </gd:comments>
  </entry>
  <entry>
    <id>http://gdata.youtube.com/feeds/api/videos/YpNd-Dg6_zQ</id>
    <published>2006-11-19T16:44:43.000-08:00</published>
    ...
  </entry>
</feed>
```

O YouTube responde às requisições com dados XML que descrevem os vídeos.

Embora haja muita coisa neste código XML, um detalhe importante a se perceber é que cada vídeo aparece dentro de uma tag <entry>

← Esta tag <entry> inicia outro vídeo, dentro dos dados de resposta em XML.



## Aponte seu lápis

Estude a parte do código XML, marcada na página anterior, e responda às seguintes perguntas. Você poderá já saber mais sobre o formato XML do YouTube do que imagina!

- 1. Qual é o título do vídeo?** .....
- 2. Cite três palavras-chaves associadas com o vídeo.** .....
- 3. Qual a duração do vídeo, em segundos?** .....
- 4. A qual categoria do YouTube o vídeo pertence?** .....
- 5. Quantas vezes o vídeo foi visto?** .....
- 6. Qual a nota média que os usuários deram ao vídeo?** .....

aponte seu lápis solução



## Aponte seu lápis Solução

Estude a parte do código XML marcada, na página anterior, e responda às seguintes perguntas. Você poderá já saber mais sobre o formato XML do YouTube do que imagina!

```
<media:title type='plain'>UFO Sighting in Yosemite Park near Area 51</media:title>
```

1. Qual é o título do vídeo? ..... *UFO Sighting in Yosemite Park near Area 51* .....

```
<media:keywords>51, alien, aliens, area, ca, california, nevada, sighting, sightings, ufo</media:keywords>
```

2. Cite três palavras-chaves associadas com o vídeo. *51, aliens, nevada* .....

3. Qual a duração do vídeo, em segundos? *50* .....

O XML codifica alguns caracteres usando códigos especiais, como &#50;, que representa o caractere (').

```
<yt:duration seconds='50' />
```

```
<media:category label='Travel & Events'  
scheme='http://gdata.youtube.com/schemas/2007/categories.cat'>Travel</  
media:category>
```

4. A qual categoria do YouTube o vídeo pertence? *Travel & Events* .....

```
<yt:statistics viewCount='2478159' favoriteCount='1897' />
```

Uau, isso é um monte de visualizações... quase 2,5 milhões!

5. Quantas vezes o vídeo foi visto? *2478159* .....

```
<gd:rating min='1' max='5' numRaters='1602' average='4.17' />
```

6. Qual a nota média que os usuários deram ao vídeo? *4.17* .....



Hm, estou um pouco confuso com todas essas tags XML que têm dois nomes separados por dois-pontos. Isso é algum tipo de organização das tags? E quanto àquele código esquisito & na categoria do vídeo?

### Este código XML incomum usa namespaces e entidades, que ajudam a organizar as tags e a codificar caracteres especiais.

Quando você vir uma tag XML com dois nomes separados por dois-pontos está vendo um namespace, que é uma forma de organizar um conjunto de tags em um mesmo grupo lógico. O propósito dos namespaces é impedir que tags com o mesmo nome entrem em conflito quando múltiplos vocabulários XML são usados no mesmo documento. Como um exemplo, considere as duas seguintes tags XML:

```
<title type='text'>Favorites of aliensabductedme</title>
<media:title type='plain'>UFO Sighting in Yosemite Park near Area 51</media:title>
```

s namespaces  
o grupos de tags  
ML, enquanto  
ue as entidades  
io usadas para  
codificar caracteres  
speciais dentro de  
ocumentos XML.

Pode parecer estranho que um namespace do Yahoo! apareça no código XML do YouTube – mas isso quer dizer apenas que o YouTube depende parcialmente de um formato de dados XML criado pelo Yahoo!

Sem o namespace media na segunda tag `<title>`, seria impossível diferenciar as duas tags caso elas aparecessem no mesmo código XML. Assim, você pode pensar no namespace como uma espécie de sobrenome para tags – ele ajuda a prevenir que um documento XML cheio de "primeiros nomes" entre em conflito, criando um "sobrenome" para tags relacionadas. O código de resposta do YouTube usa vários namespaces diferentes, o que significa que está usando diversas linguagens XML diferentes ao mesmo tempo – os namespaces nos permitem diferenciá-las claramente.

Para garantir a unicidade, cada namespace XML é sempre associado a uma URL. Por exemplo, o namespace media usado nos dados XML do YouTube é estabelecido dentro da tag `<feed>`, desta maneira:

Esta URL na verdade não é uma página web – é apenas um identificador único para um namespace.

`xmlns:media='http://search.yahoo.com/mrss/'`

A outra coisa estranha no código XML do YouTube é &, que é a forma pela qual o XML representa o caracter &. Essa é uma entidade do XML, uma forma simbólica de representar um caracter especial, como &, < ou >, todos os quais têm significados especiais dentro do código XML. Seguem as cinco entidades XML pré-definidas que você provavelmente encontrará ao se aprofundar no código XML:

|                        |   |                    |                       |   |                   |                       |   |                   |                         |   |                |                         |   |                |
|------------------------|---|--------------------|-----------------------|---|-------------------|-----------------------|---|-------------------|-------------------------|---|----------------|-------------------------|---|----------------|
| <code>&amp;amp;</code> | = | <code>&amp;</code> | <code>&amp;lt;</code> | = | <code>&lt;</code> | <code>&amp;gt;</code> | = | <code>&gt;</code> | <code>&amp;quot;</code> | = | <code>"</code> | <code>&amp;apos;</code> | = | <code>'</code> |
|------------------------|---|--------------------|-----------------------|---|-------------------|-----------------------|---|-------------------|-------------------------|---|----------------|-------------------------|---|----------------|

## Desconstruindo uma resposta XML do YouTube

Uma vez que você conheça a estrutura da resposta do YouTube, fica bem fácil extrair os dados dos vídeos que você precisa. Além de entender quais tags e atributos armazenam quais dados, é importante também entender como as tags relacionam-se umas com as outras. Como você deverá se lembrar do início do capítulo, quando analisamos o feed RSS, um documento XML pode ser entendido como uma hierarquia de elementos. O mesmo é verdadeiro para os dados XML retornados em uma resposta do YouTube.

```
<entry>
  <id>http://gdata.youtube.com/feeds/api/videos/_6Uibqf0vtA</id>
  <published>2006-06-20T07:49:05.000-07:00</published>
  ...
  <media:group>
    <media:content type='video' url='http://www.youtube.com/watch?v=_6Uibqf0vtA' medium='video' ...>
      <media:description type='plain'>I went on a trip to Yosemite Park in 2002. Yosemite Park is very close to the border between California and Nevada, and close to Area 51...</media:description>
      <media:keywords>aliens, alien, aliens, area 51, california, nevada, sighting, sightings, ufo</media:keywords>
      <yt:duration seconds='150' />
      <media:category label='Travel &amp; Events' scheme='http://gdata.youtube.com/schemas/2007/categories.rdf'>Travel</media:category>
      <media:content url='http://www.youtube.com/v/_6Uibqf0vtA' type='application/x-shockwave-flash'
        medium='video' isDefault='true' expression='full' duration='50' yt:format='5' />
      <media:content url='rtsp://rtsp2.youtube.com/ChoLENy73wIaEQnQvvSnbikl_xMYDSANFEgGDA==/0/0/0/video.3gp'
        type='video/3gpp' medium='video' expression='full' duration='50' yt:format='1' />
      <media:content url='rtsp://rtsp2.youtube.com/ChoLENy73wIaEQnQvvSnbikl_xYESARFeGDA==/0/0/0/video.3gp'
        type='video/3gpp' medium='video' expression='full' duration='50' yt:format='6' />
      <media:player url='http://www.youtube.com/watch?v=_6Uibqf0vtA' />
      <media:thumbnail url='http://img.youtube.com/vi/_6Uibqf0vtA/2.jpg' height='97' width='130'
        time='00:00:25' />
      <media:thumbnail url='http://img.youtube.com/vi/_6Uibqf0vtA/1.jpg' height='97' width='130'
        time='00:00:12.500' />
      <media:thumbnail url='http://img.youtube.com/vi/_6Uibqf0vtA/3.jpg' height='97' width='130'
        time='00:00:37.500' />
      <media:thumbnail url='http://img.youtube.com/vi/_6Uibqf0vtA/0.jpg' height='240' width='320'>You
        time='00:00:25' />
    </media:group>
    <yt:statistics viewCount='2479150' favoriteCount='11897' />
    <gd:rating min='1' max='5' hrefRate='1602' average='4.177' />
    <gd:comments>
      <gd:feedLink href='http://gdata.youtube.com/feeds/api/videos/_6Uibqf0vtA/comments'
        countHint='14426' />
    </gd:comments>
  </entry>
```

O namespace "gd" significa Google Data, e inclui tags definidas pelo Google para representar vários tipos de dados - o YouTube faz parte do Google.

A tag <title> contém o título do vídeo.

Neste código, a chama-se "title" namespace é "me".

As p chav o ví A du do ví segun A catec do You na qual encaixa vídeo.

Uma imagem thumbnail do vídeo, para que o usuário o possa pré-visualizar.

O número de vezes que o vídeo foi visto.

Uma dica importante para o entendimento dos dados de vídeos escondidos neste código XML são os diferentes namespaces sendo usados. O namespace media acompanha a maioria das tags especificamente relacionadas com os dados dos vídeos, enquanto que o namespace yt é usado apenas com a tag <statistics>. Finalmente, os comentários são colocados dentro da tag <comments>, que pertencem ao namespace gd. Esses namespaces serão muito importantes quando você começar a escrever código PHP para encontrar tags específicas e os seus respectivos dados.

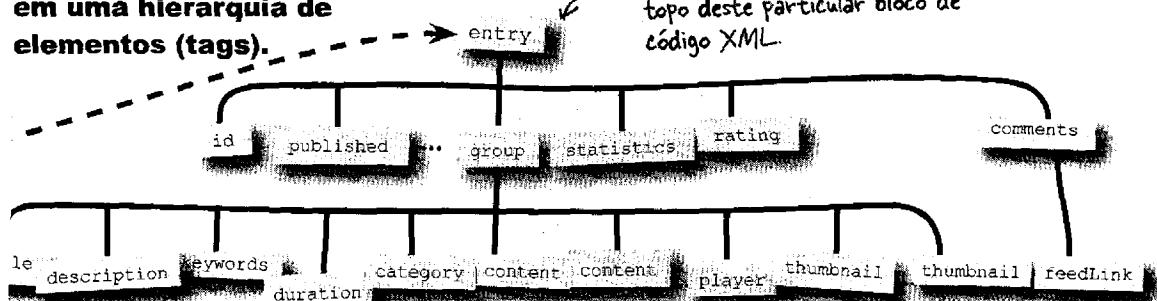
## Visualize os dados de vídeos XML

Anteriormente neste capítulo, quando trabalhamos com código RSS, foi revelado que um documento XML pode ser entendido como uma hierarquia de elementos (tags) que têm relacionamento pai-filho. Essa relação se torna cada vez mais importante à medida que você começa a processar códigos XML e acessar os dados armazenados dentro deles. Na verdade, essa habilidade de olhar um documento XML e imediatamente visualizar as relações entre os elementos pode ser preciosa.

Apenas lembre-se de que qualquer elemento colocado dentro de outro é o filho, e o elemento que o contém é o pai. Analisando-se o código XML do YouTube, na página anterior, resulta na seguinte visualização:

Um elemento é apenas uma forma abstrata de se pensar em uma tag XML e nos dados que ela contém.

**Os dados XML são organizados em uma hierarquia de elementos (tags).**



O significado desta hierarquia de elementos é que você pode navegar a partir de qualquer elemento até chegar a qualquer outro, traçando o seu caminho desde o alto da hierarquia. Por exemplo, se quisesse obter o título do vídeo, poderia traçar o seu caminho desta forma:

Para navegar até um elemento em um documento XML, você traça o caminho desde o elemento pai até o filho sendo procurado.



## Perguntas Ídiotas

**P:** Por que eu preciso me preocupar com namespaces?

**R:** Porque o código XML gerado por terceiros frequentemente usa namespaces, o que afeta o modo como você acessa os elementos XML programaticamente. Como você irá descobrir logo mais, o namespace associado com um elemento, afeta diretamente o modo como você encontra o elemento, ao escrever seu código PHP para processar dados XML. Assim, o namespace precisa ser levado em consideração ao se escrever código para tentar obter os dados de um determinado elemento.

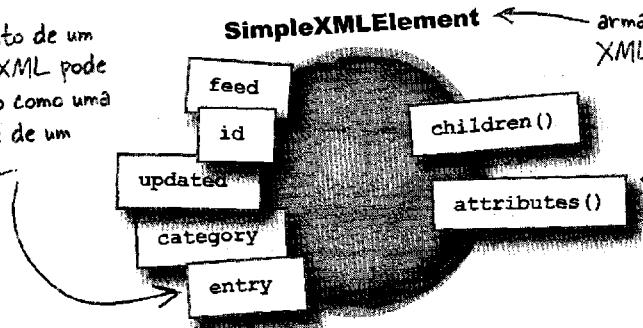
**P:** Como eu sei se uma tag faz parte de um namespace?

**R:** Embora seja possível ter um namespace padrão que não apareça explicitamente no código de uma tag, na maioria dos casos irá ver o namespace junto ao nome da tag, que é escrita como `<media:title>` em vez de apenas `<title>`. O nome à esquerda dos dois-pontos é sempre o namespace.

## Acesse os dados XML com objetos

Existem várias formas diferentes de se trabalhar com dados XML no PHP, e uma das melhores delas envolve os objetos. Um objeto é um tipo de dados especial do PHP, o qual combina dados e funções em um único construto. Mas o que isso tem a ver com o XML? Toda a hierarquia de elementos de um documento XML fica dentro de uma única variável – um objeto. Você pode, então, usar o objeto para se aprofundar nos dados e acessar elementos individuais. Os objetos também possuem métodos, que são funções vinculadas a eles, as quais nos permitem manipular ainda mais os dados do objeto. No caso de um objeto que contenha dados XML, os métodos nos permitem acessar o conjunto de elementos filhos a um pai, bem como os seus atributos.

Cada elemento de um documento XML pode ser acessado como uma propriedade de um objeto XML.



O tipo de objeto PHP usado para armazenar e manipular dados XML é o SimpleXMLElement.

O SimpleXMLElement possui métodos que lhe permitem descobrir mais sobre os elementos, como por exemplo, os seus elementos e atributos filhos.

Já vimos como criar este objeto XML para a busca de Owen no YouTube:

```
define('YOUTUBE_URL', 'http://gdata.youtube.com/feeds/api/videos/-/alien/abduction/head/first');
$xml = simplexml_load_file(YOUTUBE_URL);
```

Este código resulta em uma variável chamada \$xml que contém todos os dados XML da resposta do YouTube empacotados em um único objeto PHP. Para acessar os dados, você usa as propriedades do objeto, que são blocos individuais de dados armazenados dentro dele. Cada propriedade corresponde a um elemento XML. Dê uma olhada no seguinte exemplo, que acessa todos os elementos entry do documento:

```
$entries = $xml->entry;
```

O operador `->` lhe permite acessar uma propriedade de um objeto.

Especificando o nome do elemento (`entry`), você pode obter todos os elementos existentes nos dados XML.

Este código acessa todos os elementos entry dos dados XML usando uma propriedade. Uma vez que há muitos elementos entry nos dados, a variável \$entries armazena um array de objetos que você pode usar para acessar as entradas individuais dos vídeos. E uma vez que agora estamos lidando com um array, cada tag `<entry>` pode ser acessada indexando-se o array. Por exemplo, a primeira tag `<entry>` do documento é o primeiro item do array, a segunda tag é o segundo item e assim por diante.

Todas as entradas dos vídeos são armazenadas no array `$entries`.

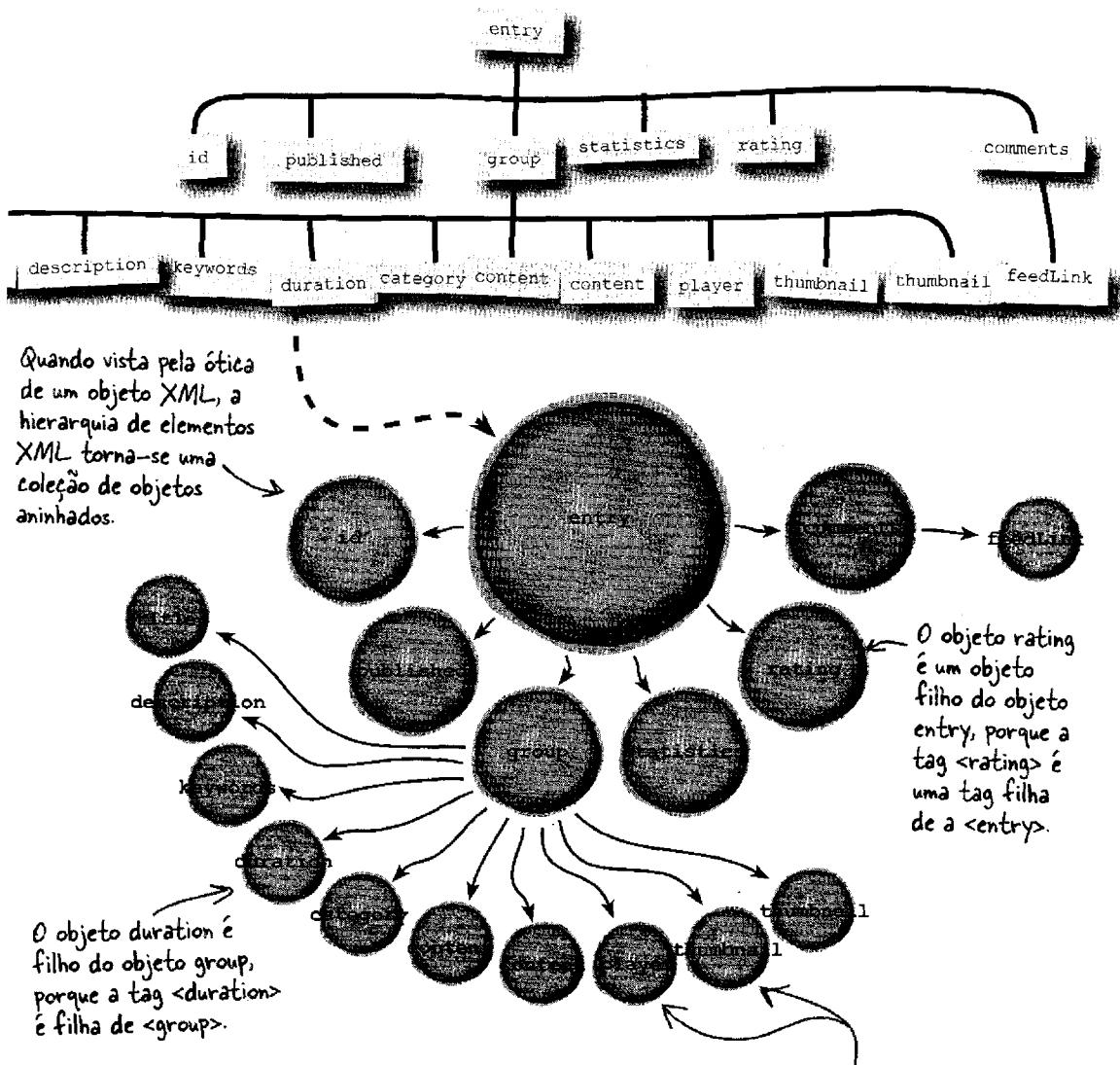


`$entries`

Os objetos são um tipo de dados especial do PHP, que combina dados e funções.

# De elementos XML para objetos PHP

No que se refere a dados XML e objetos PHP, você na verdade está lidando com um coleção (collection) de objetos. Lembra daquele negócio de visualizar um documento XML como se fosse uma hierarquia de elementos? Bom, essa mesma hierarquia é realizada na forma de uma coleção de objetos, no PHP. Dê uma olhada:



Esta hierarquia de elementos/conjunto de objetos forma a base do entendimento sobre como "peneirar" os dados XML em PHP. Com a relação entre os diversos blocos de dados XML em mente, é possível escrever código para navegar através dos dados. Então, podemos isolarmos o conteúdo armazenado em uma determinada tag ou atributo escondidos dentro de um documento XML.

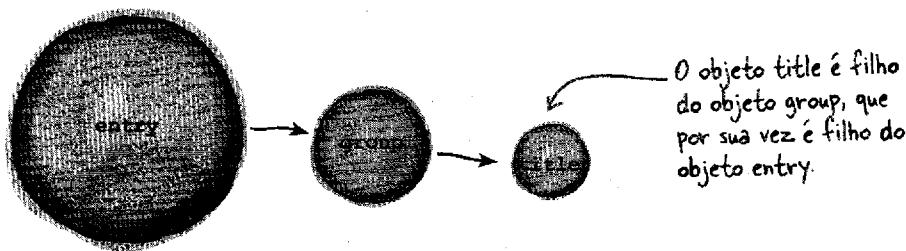
## Penetre nos dados XML com os objetos

Voltando ao Owen, nosso objetivo é obter algumas informações a partir dos vídeos retornados como parte da resposta XML do YouTube. Nós sabemos como colocar os dados XML em um objeto PHP usando a função `simplexml_load_file()`, mas a maior parte dos dados interessantes se encontra em níveis mais profundos. Como nós navegamos dentro da coleção de objetos? A resposta é o operador `->`, que é usado para referenciar uma propriedade ou método de um objeto. No caso de um objeto XML, o operador `->` acessa cada um dos objetos filhos. Assim, o seguinte código exibe o título de um vídeo, armazenado em uma variável chamada `$entry`:

```
echo $entry->group->title;
```

Aqui, o operador `->` é usado para abrir espaço dentro dos objetos filhos aninhados, para acessarmos o objeto title.

Este código é altamente dependente da relação entre os objetos `title`, `group` e `entry`, que formam o relacionamento pai-filho de um para o seguinte.



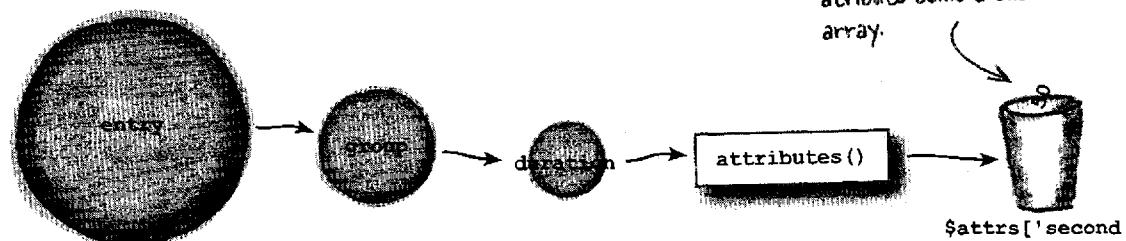
O operador `->` referencia um objeto filho a partir de um objeto principal. Assim, `title` é filho de `group`, que é filho de `entry`. Lembre-se de que o operador `->` pode ser usado para se acessar tanto propriedades quanto métodos. Um método particularmente útil é `attributes()`, o qual é capaz de obter o valor de um atributo XML de um dado elemento.

```
$attrs = $entry->group->duration->attributes();  
echo $attrs['seconds'];
```

O método `attributes()` obtém um array de atributos de um determinado objeto (elemento).

Esse código vai até o elemento `duration` e então obtém todos os seus atributos, armazenando-os na variável `$attrs`, que é um array de todos os atributos. O valor do atributo `seconds` é então obtido a partir do array.

O valor de um atributo específico pode ser obtido usando-se o nome do atributo como a chave do array.



## Não sem um namespace!

Há um pequeno problema com o código da página anterior, que acessa dados XML usando objetos, e isso tem a ver com os namespaces. Como você deverá se lembrar, os namespaces agem como sobrenomes para as tags, organizando-as em coleções significativas. Assim, em uma resposta do YouTube, a tag <duration> na verdade é codificada como <yt:duration> e o título do vídeo é codificado como <media:title>, e não <title>. Quando um elemento encontra-se associado a um namespace, você não pode referenciá-lo apenas pelo nome da tag no seu código PHP. Em vez disso, precisa primeiramente isolá-lo pelo seu namespace, chamando o método `children()` no objeto principal.

```
$media = $entry->children('http://search.yahoo.com/mrss/');
```

Esse código obtém todos os objetos filhos da entrada cujo namespace é `http://search.yahoo.com/mrss/`. Mas essa é a URL do namespace, e não o próprio namespace. Essa URL se localiza dentro da tag <feed>, no início do documento XML. E aí que você encontrará todos os namespaces sendo usados.

```
<feed xmlns='http://www.w3.org/2005/Atom'
      xmlns:openSearch='http://a9.com/-/spec/opensearchrss/1.0/'
      xmlns:gml='http://www.opengis.net/gml'
      xmlns:georss='http://www.georss.org/georss'
      xmlns:media='http://search.yahoo.com/mrss/'  

      xmlns:batch='http://schemas.google.com/gdata/batch'
      xmlns:yt='http://gdata.youtube.com/schemas/2007/'  

      xmlns:gd='http://schemas.google.com/g/2005'>
```

Esse código revela como cada namespace se associa a uma URL. Mais especificamente, ele mostra como os namespaces `media` e `yt` são especificados para uso no documento. Isso é tudo o que você precisa para encontrar tags relacionadas a esses dois namespaces.

Uma vez que tenha isolado os elementos filhos de um dado namespace, chamando o método `children()` no elemento pai, você poderá continuar a acessar os objetos filhos com o operador `->`. Por exemplo, este código obtém o título do vídeo a partir da tag <`media:group`>:

```
$title = $media->group->title;
```

A tag <title> é  
filha da tag  
<media:group>.

Os namespaces tornam um pouco mais difícil o acesso a elementos dentro dos dados XML.

O método `children()` retorna um array contendo todos os elementos filhos presentes no namespace especificado.

Todas as tags que começam com "<media:" pertencem a este namespace.

Este namespace é para as tags que começam com "<yt:"

Use o método `children()` para isolar todos os elementos associados com um dado namespace.



### Aponte seu lápis

Usando as informações de namespace e o código PHP acima, complete o código PHP que obtém a duração (em segundos) de um clipe de vídeo.

```
$yt = $media->children('.....');
$attrs = .....;
echo $attrs['.....'];
```

*não existem perguntas idiotas sobre objetos*

## Aponte seu lápis Solução

Usando as informações de namespace e o código PHP acima, complete o código PHP que obtém a duração (em segundos) de um clipe de vídeo.

```
$yt = $media->children(' http://gdata.youtube.com/schemas/2007 ');
$attrs = $yt->duration->attributes();
echo $attrs['seconds'];
```

O nome do atributo é usado como a chave para se acessar o array dos atributos.

Obtém todos os atributos da tag <yt:duration>

Esta é a URL para o namespace, conforme listado na tag <feed> do início do documento.

## *não existem* Perguntas Idiotas

P: Em que um objeto é diferente de um array? Os arrays não armazenam também coleções de dados?

R: Sim. Os arrays e os objetos na verdade são bem parecidos. Mas uma grande diferença é que os objetos podem ter código executável vinculado a eles, na forma de métodos. Os métodos são praticamente o mesmo que as funções, exceto pelo fato de que ficam vinculados a um objeto, e geralmente são elaborados para trabalhar especificamente com os dados armazenados no objeto. Os arrays servem simplesmente para armazenar um conjunto de dados relacionados, e não têm nenhuma noção de métodos. Adicionalmente, os elementos dos arrays são acessados especificando-se o índice ou chave, de um elemento entre colchetes ([ ]), enquanto que as propriedades e os métodos dos objetos são acessados por nome, usando-se o operador ->.

P: O que exatamente é um objeto? É parecido com uma variável normal?

R: Sim. Um objeto é igual a qualquer outra variável em PHP; a diferença é que ele é capaz de armazenar dados mais complexos. Assim, em vez de apenas armazenar uma string de texto ou um número, o objeto é capaz de armazenar uma combinação de strings, números, etc. A ideia é que, combinando-se dados com as funções que agem sobre eles, o design e a codificação gerais das aplicações tornam-se mais lógicos.

P: Como os objetos ajudam a processar dados XML?

R: Os objetos ajudam no processamento de dados XML porque eles são capazes de modelar a hierarquia de elementos de um documento XML em objetos filhos aninhados. O benefício dessa abordagem é que você pode navegar através dos objetos filhos usando o operador -> e acessar os dados que quiser.

P: Eu pensava que o operador -> servia para acessar propriedades dos objetos. Como ele me ajuda a acessar um objeto filhos?

R: A explicação é que, quando se lida com objetos XML em PHP, os objetos filhos na verdade são armazenados como propriedades. Assim, quando você usa o operador -> para acessar um objeto filhos, na verdade está apenas acessando uma propriedade. O objeto SimpleXMLElement é o que torna isso possível.

P: Espere aí, o que é o objeto SimpleXMLElement?

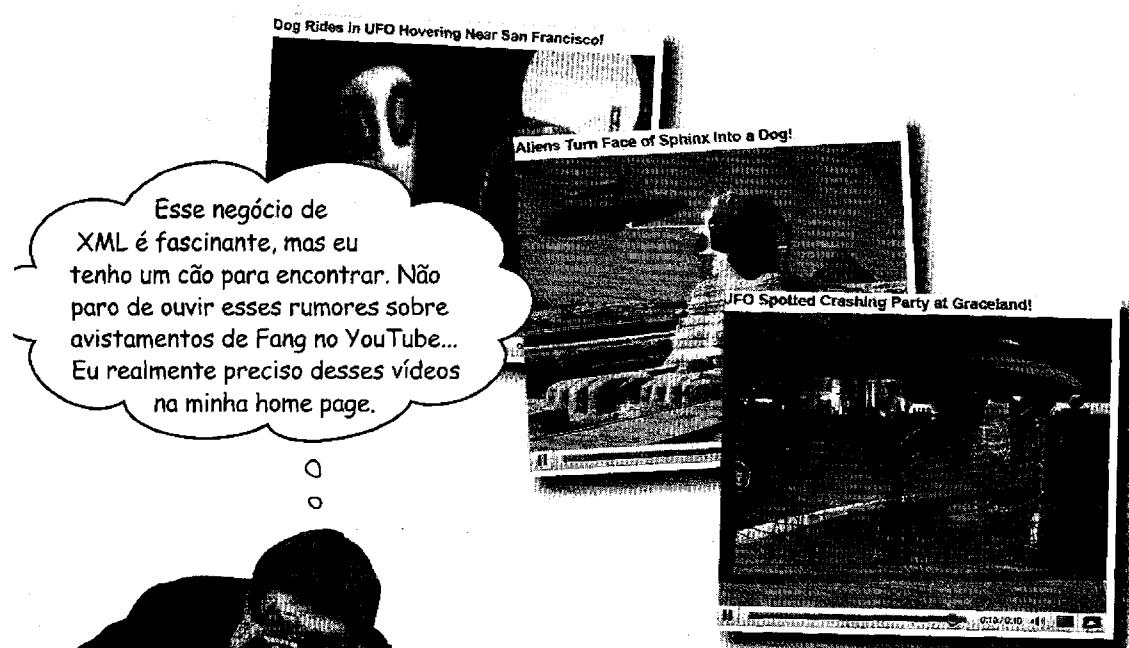
R: Todo objeto do PHP tem um tipo de dados específico, significando que "objeto" na verdade é um termo genérico. Assim, quando você cria um objeto, está criando um objeto de um tipo específico, elaborado para realizar uma tarefa específica. No caso do XML, o tipo do objeto é SimpleXMLElement, e é automaticamente retornado pela função simplexml\_load\_file(). Em outras palavras, chamar a função simplexml\_load\_file() resulta na criação de um objeto do tipo SimpleXMLElement.

P: O que eu preciso saber sobre SimpleXMLElement?

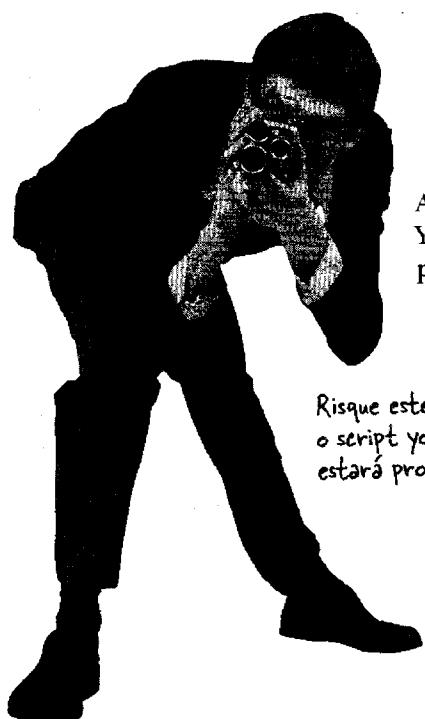
R: Surpreendentemente, não muito. A principal coisa a saber é que ele expõe os elementos de um documento XML como propriedades, e que essas propriedades levam a objetos filhos que são, eles mesmos, instâncias do objeto SimpleXMLElement, e assim por diante. O objeto SimpleXMLElement também possui métodos que lhe permitem acessar dados de um elemento, como children() e attributes().

## Cada vez mais avistamentos de Fang

Enquanto Owen esteve ocupado aprendendo a lidar com o XML e descobrindo como se comunicar com o YouTube, Fang tem estado ocupado. Numerosos avistamentos têm aparecido, com o cãozinho aparentemente trabalhando como guia turístico para os seus captores alienígenas. Owen está pronto para terminar o script do YouTube, obter alguns vídeos para colocar na sua home page e encontrar seu cão perdido.



A boa notícia é que Owen já tem quase todo o script do YouTube pronto. Na verdade, só o que falta é terminar de processar os dados XML e formatá-los como código HTML.



- ➊ Criar uma requisição por vídeos do YouTube.
- ➋ Emitir a requisição de vídeo para o YouTube.
- ➌ Receber a resposta do YouTube, contendo informações sobre os vídeos.
- ➍ Processar os dados da resposta e formatá-los como código HTML.

Desenhe abaixo como você formataria os dados de resposta do YouTube como vídeos, colocando-os na parte de baixo da página principal do site Fui Abduzido por Aliens:

## *o script completo youtube.php*

---



### **Exercício**

O script youtube.php usa código PHP para obter os cinco primeiros resultados de uma busca no YouTube. Em seguida, ele exibe imagens reduzidas (thumbnails) desses vídeos em uma linha horizontal, com links para os vídeos propriamente ditos no YouTube. Complete o código que está faltando no script, usando o exemplo de dados de resposta XML do YouTube, na página seguinte, como guia.

```
<?php

define('YOUTUBE_URL', 'http://gdata.youtube.com/feeds/api/videos/-/alien/abduction/
head/first');

define('NUM_VIDEOS', 5);

// Lê os dados XML e os coloca em um objeto
$xml = .....(YOUTUBE_URL);

$num_videos_found = count(.....);
if ($num_videos_found > 0) {
    echo '<table><tr>';
    for ($i = 0; $i < min($num_videos_found, NUM_VIDEOS); $i++) {
        // Obtém o título
        $entry = $xml->entry[$i];
        $media = $entry->children('http://search.yahoo.com/mrss/');
        $title = $media->group->....;

        // Obtém a duração em minutos e segundos, e depois a formata
        $yt = $media->children('http://gdata.youtube.com/schemas/2007');
        $attrs = $yt->duration->attributes();
        $length_min = floor($attrs['.....'] / 60);
        $length_sec = $attrs['.....'] % 60;
        $length_formatted = $length_min . (($length_min != 1) ? ' minutes, ':' minute, ') .
            $length_sec . (($length_sec != 1) ? ' seconds':' second');

        // Obtém a URL do vídeo
        $attrs = $media->group->player->.....();
        $video_url = $attrs['url'];
```

```

// Obtém a URL da imagem thumbnail
$attrs = $media->thumbnail[0]->attributes();
$thumbnail_url = $attrs['url'];

// Exibe os resultados para esta entrada
echo '<td style="vertical-align:bottom; text-align:center" width="'.(100 / NUM_VIDEOS)
.'%><a href="'.$video_url.'">.....<br /><span style="font-size:smaller">' .
$lenght_formatted.'</span><br /></a></td>';
}

echo '</tr></table>';

}
else {
echo '<p>Desculpe, nenhum vídeo encontrado.</p>';
}
?>

```

Esteja à vontade para usar este exemplo de código XML como referência ao escrever o código PHP que está faltando.

```

<entry
  id='http://gdata.youtube.com/feeds/api/videos/_6Uiqbqf0vtA'
  published='2006-06-20T07:49:05.000-07:00'
  ...
  <media:group>
    <media:title type='plain'>UFO Sighting in Yosemite Park near Area 51</media:title>
    <media:description type='plain'>I went on a trip to Yosemite Park in 2002. Yosemite Park is very close to the border between California and Nevada, and close to Area 51...</media:description>
    <media:keywords>51, alien, aliens, area, ca, california, nevada, sighting, sightings, ufo</media:keywords>
    <yt:duration seconds='50' />
    <media:category label='Travel & Events' scheme='http://gdata.youtube.com/schemas/2007/categories.cat'>Travel</media:category>
    <media:content url='http://www.youtube.com/v/_6Uiqbqf0vtA' type='application/x-shockwave-flash' medium='video' isDefault='true' expression='full' duration='50' yt:format='5' />
    <media:content url='rtsp://rtsp2.youtube.com/ChoLENy73wIaEQnQvvSnbIKl_xMYDSANFEGDAA==/0/0/0/ideo.3gp' type='video/3gpp' medium='video' expression='full' duration='50' yt:format='1' />
    <media:content url='rtsp://rtsp2.youtube.com/ChoLENy73wIaEQnQvvSnbIKl_xMYESARFEGDAA==/0/0/0/ideo.3gp' type='video/3gpp' medium='video' expression='full' duration='50' yt:format='6' />
    <media:player url='http://www.youtube.com/watch?v=_6Uiqbqf0vtA' />
    <media:thumbnail url='http://img.youtube.com/vi/_6Uiqbqf0vtA/2.jpg' height='97' width='130' time='00:00:25' />
    <media:thumbnail url='http://img.youtube.com/vi/_6Uiqbqf0vtA/1.jpg' height='97' width='130' time='00:00:12.500' />
    <media:thumbnail url='http://img.youtube.com/vi/_6Uiqbqf0vtA/3.jpg' height='97' width='130' time='00:00:37.500' />
    <media:thumbnail url='http://img.youtube.com/vi/_6Uiqbqf0vtA/0.jpg' height='240' width='320' time='00:00:25' />
  </media:group>
  <yt:statistics viewCount='2478159' favoriteCount='1897' />
  <gd:rating min='1' max='5' numRaters='1602' average='4.17' />
  <gd:comments>
    <gd:feedLink href='http://gdata.youtube.com/feeds/api/videos/_6Uiqbqf0vtA/comments' countHint='4426' />
  </gd:comments>
  ...
</entry>
<entry>
  ...
</entry>
  ...

```

O título do vídeo.

A duração do vídeo, em segundos.

A URL do link para o vídeo no YouTube.

A URL de uma imagem reduzida (thumbnail) do vídeo.

site de aliens agora com vídeos do youtube



## TEST DRIVE

### Adicione o script YouTube ao site Fui Abduzido por Aliens.

Crie um arquivo de texto chamado youtube.php e digite nele o código para o script YouTube das duas páginas anteriores (ou baixe o script no site da Alta Books, em [www.altabooks.com.br](http://www.altabooks.com.br)). Você ainda precisa incluir este script no index.php para ter os vídeos do YouTube colocados na página principal do site Fui Abduzido por Aliens. Eis as duas linhas de código PHP para fazê-lo:

```
echo '<h4>Vídeos de abduções mais recentes:</h4>';
require_once('youtube.php');
```

Envie os scripts para o seu servidor web e depois abra index.php em um navegador. A parte de baixo da página deverá mostrar uma linha, gerada dinamicamente, de links para vídeos do YouTube relacionados com abduções por alienígenas.

Incluir o script youtube.php na página principal é tudo o que você precisa fazer para adicionar a linha de vídeos de abduções.

Os vídeos do YouTube ajudaram Owen a descobrir o paradeiro de Fang.

The screenshot shows a website titled "Aliens Abducted Me". The main content area displays a list of recent reported abductions with details like date, location, duration, and alien description. To the right, there's a sidebar titled "Most recent abduction videos" featuring four thumbnail images of dog breeds (Pug, Shih Tzu, Chihuahua, and Dachshund) with captions like "UFO Spotted Crashing Party at Graceland!" and "Aliens Turn Face of Sphinx Into a Dog!".

| Date       | Name            | Abducted for:             | Alien description:                                             | Fang spotted: |
|------------|-----------------|---------------------------|----------------------------------------------------------------|---------------|
| 2008-08-16 | Melinda Ressner | 3 hours                   | they were in a ship the size of a full moon                    | no            |
| 2008-07-11 | Mickey Mikens   | 45 minutes...and counting | huge heads, skinny arms and legs                               | yes           |
| 2008-07-05 | Shih Watner     | 2 hours                   | there was a bright light in the sky, followed by a bark or two | yes           |
| 2008-06-21 | Belta Chevy     | almost a week             | clumsy little buggers, had no rhythm                           | no            |
| 2008-05-11 | Sally Jones     | 1 day                     | green with six tentacles                                       | yes           |

Click to syndicate the abduction news feed.

Most recent abduction videos:

- UFO Spotted Crashing Party at Graceland! 6 minutes, 10 seconds
- Aliens Turn Face of Sphinx Into a Dog! 6 minutes, 10 seconds
- Dog Rides in UFO Hovering Near San Francisco! 6 minutes, 11 seconds
- UFO Spotted Near Eiffel Tower! 6 minutes, 13 seconds
- Pug Abducted By UFO! 6 minutes, 16 seconds

Acho que sei  
onde Fang está...

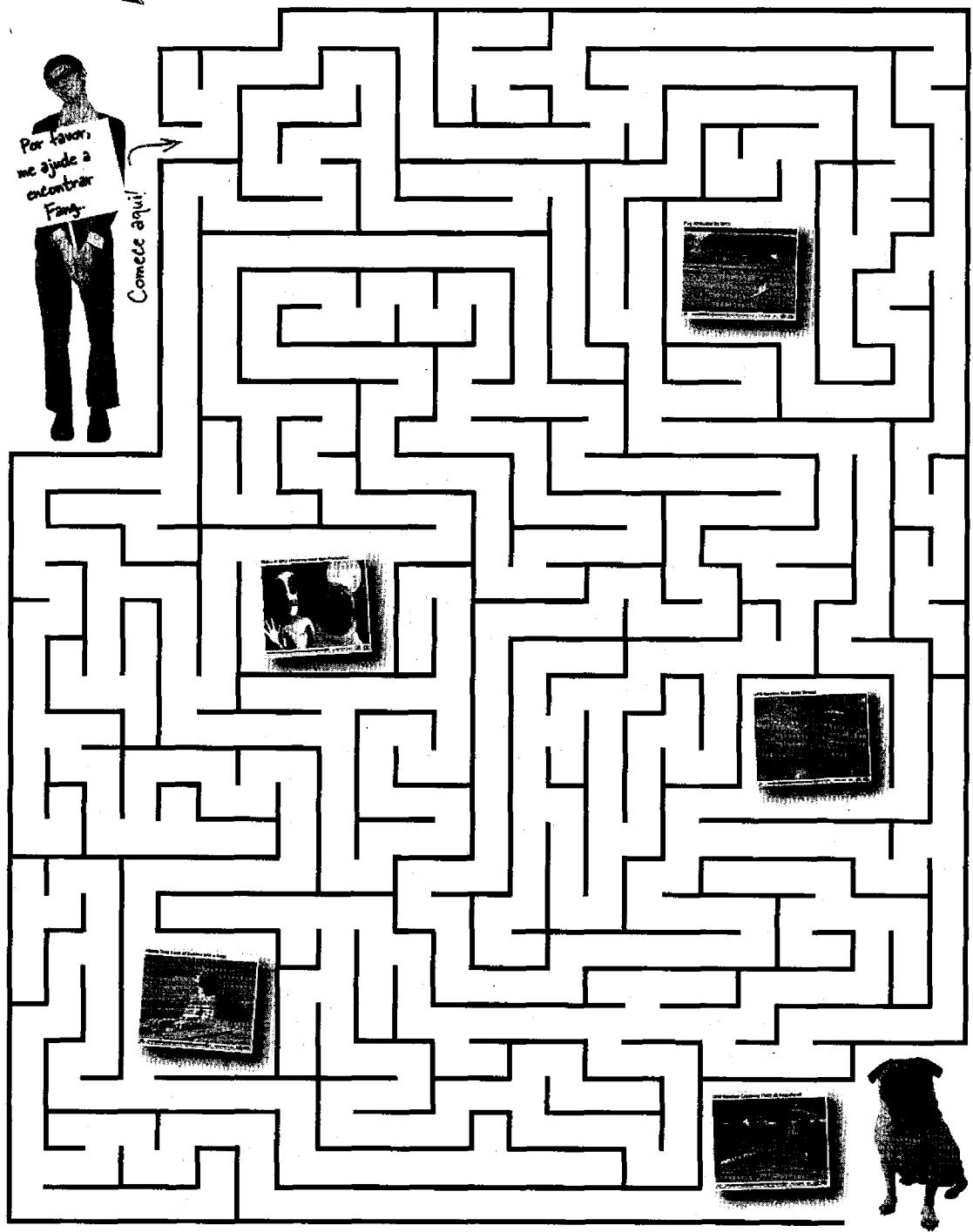


Aponte seu lápis



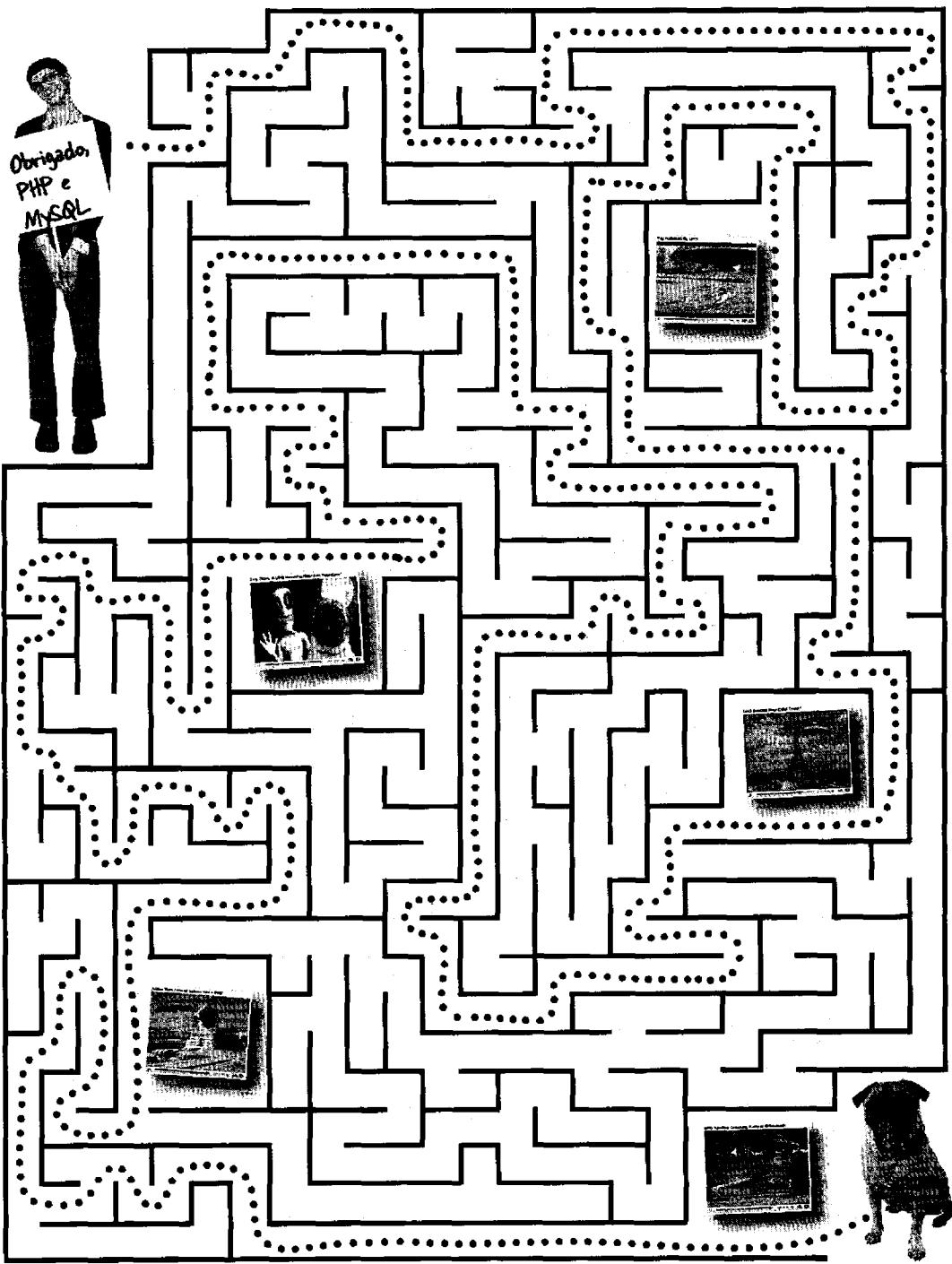
Por favor,  
me ajude a  
encontrar  
Fang.

Comece aqui!



você achou fang!

**Aponte seu lápis**  
**Solução**





## Sua Caixa de Ferramentas do PHP & MySQL

Agora que Fang foi encontrado, é possível refletir um pouco sobre tudo o que foi preciso para chegarmos até ele. Como se pode ver, o PHP e o MySQL precisaram de um pouco de ajuda de algumas outras tecnologias.

### XML

Uma linguagem markup genérica, usada para se fornecer uma estrutura previsível para os dados. Existem muitas linguagens markup diferentes, criadas a partir do XML, como por exemplo, XHTML e RSS. A ideia é que você crie um conjunto de tags específicas para os dados que está armazenando como XML.

### `simplexml_load_file()`

Esta função do PHP carrega um arquivo XML a partir de uma URL, e depois torna os dados XML resultantes acessíveis por meio de um objeto.

### RSS

Uma linguagem baseada em XML, usada para se armazenar conteúdo a ser republicado. O RSS permite que os sites disponibilizem os seus dados para outras aplicações e sites, e lhe permite tirar proveito dos dados disponibilizados por outros sites.

### `SimpleXMLElement`

Um objeto do PHP usado para acessar dados XML. O objeto é retornado pela função `simplexml_load_file()`, e contém toda a hierarquia do documento XML.

### REST

Uma forma de se acessarem informações na web puramente através de URLs. REST lhe permite fazer complexas requisições de dados simplesmente criando uma URL. Essas são, às vezes, chamadas de "requisições RESTful".

### Namespace

Uma forma de organizar um conjunto de tags XML em um grupo lógico, mais ou menos como o seu sobrenome organiza a sua família em um grupo nomeado. O namespace fica sempre associado a uma URL, o que garante a unicidade em relação a todos os outros namespaces.

FIM.

## Apendice i: Sobras



# ***Os Dez Principais Tópicos (Que Não Abordamos)***



**Mesmo depois de tudo isso, ainda tem mais um pouco**, algumas coisas que você precisa conhecer. Não seria certo as ignorarmos, mesmo que só precisem de uma breve menção. Portanto, antes de guardar o livro, dê uma lida nestes pequenos, mas importantes parágrafos sobre PHP e MySQL. Além disso, uma vez concluído este aqui, só faltarão mais dois pequenos apêndices... e o índice... e talvez algumas propagandas... e aí o livro realmente acaba. Promessa!

adpte seu código php

## #1. Adapte este livro para funções PHP4 e mysql

Com exceção das funções XML do Capítulo 12, a maioria dos códigos deste livro roda também em servidores PHP 4, com apenas uma pequena modificação. Nós usamos a família de funções mysqli neste livro, que só estão disponíveis no PHP 4.1 e acima. E, uma vez que essa biblioteca tem que ser instalada manualmente, alguns servidores não terão suporte a mysqli.

As funções mysqli em geral são mais rápidas, mas isso só começa a realmente fazer diferença quando o seu banco de dados fica muito grande. Bancos de pequeno ou médio porte não são perceptivelmente mais lentos com as funções mysql antigas. Esta seção lhe ensinará como adaptar suas funções mysqli para funcionarem como funções mysql, com versões antigas do PHP.

Quando você vir:

```
$dbc = mysqli_connect('localhost', 'mork', 'fromork');  
mysqli_select_db($dbc, 'alien_database');
```

Mude para o seguinte:

```
$dbc = mysql_connect('localhost', 'mork', 'fromork');  
mysql_select_db('alien_database', $dbc);
```

A variável de conexão ao banco de dados não é o primeiro argumento aqui, ao contrário do que acontece com mysqli\_select\_db().

Em geral, você apenas remove o i de mysqli ficando mysql e depois muda a ordem dos argumentos, de forma que a variável de conexão ao banco de dados (\$dbc neste exemplo) apareça por último.

Mas a coisa fica um pouco mais complexa quando a função mysqli\_connect() toma o lugar de mysqli\_select\_db() que usa o nome de um banco de dados. Não existe nada parecido com isso na família de funções mysql. Para reproduzir a função mysqli\_connect() que usa um nome de banco de dados, você precisará de duas funções mysql.

Quando você vir:

```
$dbc = mysqli_connect('localhost', 'mork', 'fromork', 'alien_database');
```

Precisará usar dois comandos:

```
$dbc = mysql_connect('localhost', 'mork', 'fromork');  
mysql_select_db('alien_database', $dbc);
```

Esta variável de conexão também é conhecida como "link" de conexão ao banco de dados.

Aqui, o banco de dados é selecionado como parte da operação de conexão - algo que não é possível fazer em um só passo com funções mysql.

Com funções mysql, sempre é preciso usar duas chamadas a funções para se estabelecer uma conexão a um banco de dados.

Aqui uma comparação entre as funções mysql e mysqli.

|                                                                                      |                                                                                                                                             |                                                                                                                                                           |
|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fechar a conexão do MySQL                                                            | <code>mysqli_close(conn)</code>                                                                                                             | <code>mysqli_close(conn)</code>                                                                                                                           |
| Abrir uma conexão a um servidor MySQL                                                | <code>mysql_connect(host, username, password)</code><br>Você precisa usar <code>mysql_select_db()</code> para selecionar um banco de dados. | <code>mysqli_connect(host, username, password, database)</code><br>Você não precisa de <code>mysqli_select_db()</code> para selecionar um banco de dados. |
| Retornar o texto da mensagem de erro de uma operação MySQL anterior                  | <code>mysql_error(conn)</code>                                                                                                              | <code>mysqli_error(conn)</code>                                                                                                                           |
| Escapar de uma string                                                                | <code>mysql_escape_string(string, conn)</code><br>A ordem dos argumentos é ao contrário, com a string vindo antes da conexão (link).        | <code>mysqli_escape_string(conn, string)</code><br>A ordem é conexão (link), e depois a string.                                                           |
| Obter uma linha de resultados como um array associativo, um array numérico, ou ambos | <code>mysql_fetch_row(result)</code>                                                                                                        | <code>mysqli_fetch_row(result)</code>                                                                                                                     |
| Obter o número de linhas do resultado                                                | <code>mysql_num_rows(result)</code>                                                                                                         | <code>mysqli_num_rows(result)</code>                                                                                                                      |
| Executar uma consulta MySQL                                                          | <code>mysql_query(query, conn)</code>                                                                                                       | <code>mysqli_query(query, conn)</code>                                                                                                                    |
| Escapar de caracteres especiais em uma string                                        | <code>mysql_real_escape_string(string, conn)</code><br>A ordem dos argumentos é ao contrário, com a string vindo antes da conexão (link).   | <code>mysqli_real_escape_string(conn, string)</code><br>A ordem é conexão (link), e depois a string.                                                      |
| Selecionar um banco de dados MySQL                                                   | <code>mysql_select_db(dbname, conn)</code><br>A ordem dos argumentos é ao contrário, com a string vindo antes da conexão (link).            | <code>mysqli_select_db(conn, dbname)</code><br>A ordem é conexão (link), e depois a string.                                                               |

*configurando permissão de usuário mysql*

## #2. Permissões do usuário no MySQL

Suponha que você tenha criado uma aplicação web que só permite aos visitantes selecionarem dados da sua tabela, com SELECT. Você realiza consultas nos seus dados usando um banco específico, e o MySQL lhe permite manipular os seus dados.

Mas considere o seguinte: o login e a senha que você usa na sua string de conexão mysqli, se conectados diretamente ao banco através do terminal ou GUI MySQL, permitiriam ao usuários fazer INSERT, UPDATE e DELETE nos dados.

Se a sua aplicação não precisar fazer essas coisas, não há nenhum motivo para que o nome de usuário/senha que você esteja usando para se conectar a ela tenha a capacidade de fazê-las. Com o MySQL, você pode limitar o acesso ao seu banco de dados. Você pode instruir o MySQL somente permitir que o usuário execute SELECT. Ou então SELECT e INSERT. Ou qualquer combinação que você precisar.

E o que é mais impressionante, você pode controlar o acesso a tabelas específicas. Por exemplo, se a sua aplicação só trabalha com uma tabela chamada alien\_info e não precisa acessar a tabela cyborg\_info, você pode limitá-la.

Primeiramente, você poderá querer um nome de usuário e uma senha completamente novos para serem usados na sua aplicação. Você pode fazer isso no terminal MySQL:

```
File Edit Window Help Aliens!
mysql> CREATE USER alienguy IDENTIFIED BY 'aliensRsc4ry';
Query OK, 0 rows affected (0.07 sec)
```

Então é possível usar o comando MySQL GRANT para controlar o que alienguy pode fazer no seu banco de dados. Se ele só precisar executar SELECT e INSERT no banco, isto seria o suficiente:

```
File Edit Window Help TheyLive
mysql> USE alien_database;
Database changed
mysql> GRANT SELECT, INSERT ON alien_info TO alienguy;
Query OK, 0 rows affected (0.03 sec)
```

Se não gostar de usar o terminal MySQL para criar usuários e definir privilégios, você pode baixar e instalar um útil programa chamado MySQLAdministrator. Baixe-o aqui: <http://dev.mysql.com/downloads/gui-tools/5.0.html>.

É possível definir privilégios bastante específicos para os usuários, até mesmo controlar o que o usuário pode fazer em uma coluna específica. Para aprender mais, consulte o livro Use a Cabeça! SQL.

O MySQL Administrator lhe permite controlar as contas dos seus usuários, bem como o que cada usuário pode acessar no seu banco de dados. Ele lhe permite até especificar quais tipos de consultas o usuário pode realizar em cada tabela do seu banco. Para controlar o acesso dos usuários a cada tabela e cada consulta, abra o aplicativo MySQL Administrator e clique na guia Accounts.

Segue abaixo a interface e uma explicação geral sobre como controlar o que cada usuário pode fazer. Em primeiro lugar, crie uma conta:

Primeiramente, clique em Accounts.

Em seguida, use este botão para adicionar uma nova conta.

Esta é a lista dos seus usuários. Você pode criar novos usuários para definir especificamente o controle que cada um terá sobre uma determinada aplicação. Selecione a conta que você quer modificar aqui.

Aqui está a lista de tabelas de um determinado banco de dados. Selecione aquela que a sua aplicação está usando.

Após dar um nome e uma senha para o novo usuário, clique aqui para conceder privilégios a ele.

Se observar esta lista, você reconhecerá as principais instruções MySQL que viu ao longo deste livro. Selecione apenas aquelas de que a sua aplicação precisa para funcionar.

relatando erros ao mysql

## #3. Relato de erros para o MySQL

Em muitos dos nossos exemplos de códigos, você verá linhas como a seguinte:

```
mysqli_connect('localhost', 'mork', 'fromork') or die ('não conectou.')
```

Quando esse comando falha, as palavras "não conectou." são exibidas na página web. Elas nos informam que algo deu errado, mas não nos diz nada mais além disso.

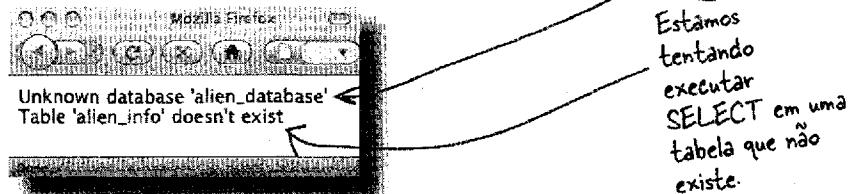
Felizmente, o PHP tem um função, `mysql_error()`, que pode nos dar uma dica sobre o que exatamente houve de errado. Considere este código, no qual estamos tentando nos conectar a um servidor MySQL que não existe:

```
<?php  
    mysqli_connect('badhostname', 'mork', 'fromork') or die (mysqli_  
    error());  
?  
'boost do servidor MYSQL 'badhostname' desconhecido(1)'  
Eis a mensagem de erro que você verá.
```

O seguinte código retorna informações claras sobre o que realmente aconteceu de errado quando a função `mysqli_connect()` falhou. Você pode usar também `mysqli_error()` com outras funções `mysqli`:

```
<?php  
    $dbc = mysqli_connect('localhost', 'mork', 'fromork');  
    mysqli_select_db($dbc, 'alien_database');  
    echo mysqli_error($dbc) . '<br />';  
    mysqli_select_db($dbc, 'alien_database');  
    mysqli_query($dbc, "SELECT * FROM alien_info");  
    echo mysqli_error($dbc);  
?  
Estamos tentando nos conectar a um banco de dados que não existe.
```

Eis o output:



Eis algumas outras mensagens de erro que você poderá ver:

Tabela 'teste.sem\_essa\_tabela' não existe

Não consegui criar a tabela

Não consegui criar o banco de dados 'seubancodedados'; banco já existe

Não consegui deletar o banco de dados 'seubancodedado'; banco não existe

Existem muitas outras e seria um desperdício de papel listá-las todas aqui. Consulte este site para obter mais informações:

<http://dev.mysql.com/doc/refman/5.0/en/error-messages-server.html>

**Caso esteja adaptando suas funções para versões antigas do PHP, como mencionado no item #1, você pode usar `mysql_error()` em vez de `mysqli_error()`.**

## #4. Erros PHP com tratamento de exceções

O tratamento de exceções lhe permite modificar o fluxo normal do seu código e executar um bloco de código especial quando uma determinada exceção ocorrer. As versões 5 e 6 do PHP oferecem o tratamento de erros. Eis aqui uma breve introdução.

Digamos que você queira sacar \$200 de um caixa automático.

Mas talvez o seu banco requeira que mantenha um saldo mínimo de \$1000, sendo que este saque o colocaria abaixo de \$1000. Isso não é permitido.

A transação falhou!

Eis aqui a forma como esse cenário poderia se desenrolar na forma de código PHP, com a ajuda do tratamento de exceções para capturar a falha.

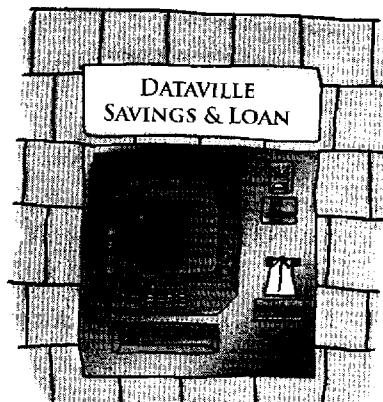
```
<?php

function checkBalance($balance) {
    if($balance < 1000) {
        throw new Exception("O saldo está abaixo de $1000.");
    }
    return true;
}

try { ←
    O bloco "try" é usado para
    testar o nosso valor sem
    terminar o fluxo do código.
    checkBalance(999); ←
    Nós verificamos
    nosso saldo aqui.
    echo 'Saldo é maior que $1000.';
}

catch(Exception $e) {
    echo 'Error: ' . $e->getMessage(); ←
}

?>
```



Eis o feedback que  
enviaremos, caso o  
saldo fique abaixo de  
1000.



Se a exceção ocorrer, nós  
executarmos o código deste  
bloco. Neste caso, echoamos  
a mensagem.

Quando o código for executado, você verá isto:

Error: O saldo está abaixo de \$1000.

## #4. Erros PHP com tratamento de exceções (continuação)

O tratamento de exceções consiste de três blocos de código:

1. Try - é neste bloco que você verifica se o valor é o que estava esperando.

Se for, está tudo bem, e o seu código prossegue. Se não, significa que ocorreu uma exceção. Na língua dos programadores, a exceção é "lançada".

E quando algo é lançado, precisa existir alguma outra coisa para pegá-lo. Se houver uma exceção, o código do bloco "catch" é executado. Se não, o código continua normalmente.

2. Throw - o "throw" comanda

o bloco "catch" e envia a ele uma mensagem de erro.

Cada "throw" possui no mínimo um "catch".

```
<?php  
function checkBalance($balance) {  
    if($balance < 1000) {  
        throw new Exception("O saldo está abaixo de $1000.");  
    }  
    return true;  
}  
  
try {  
    checkBalance(999);  
    echo 'O saldo está acima de $1000.';  
}  
catch(Exception $e) {  
    echo 'Error: ' . $e->getMessage();  
}  
?>
```

3. Catch - um objeto é criado com as informações da exceção. Para mais informações sobre os objetos, veja a página seguinte

## #5. PHP orientado a objetos

As linguagens orientadas a objetos usam um modelo de programação bastante diferente das linguagens procedurais. Você tem usado o PHP de forma procedural, mas essa linguagem também tem um lado orientado a objetos. Em vez de termos um conjunto de instruções ordenadas, as estruturas particulares se tornam objetos. Os objetos incluem não somente uma definição dos seus dados, mas também todas as operações que podem ser realizadas neles. Quando você usa PHP orientado a objetos, você cria e trabalha com objetos.

Antes de entrarmos no assunto do por que você poderia querer usar o PHP orientado a objetos, vamos escrever alguns destes:

Esta é a nossa classe Song, que define o nosso objeto.

### 1 Escreva a sua classe.

```
class Song
{
    var $title;
    var $lyrics;

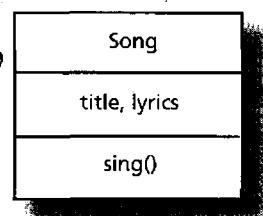
    function Song($title, $lyrics) {
        $this->title = $title;
        $this->lyrics = $lyrics;
    }

    function sing() {
        echo 'This is called ' . $this->title . '<br />';
        echo 'One, two, three...' . $this->lyrics;
    }
}
```

Estas são variáveis de instâncias.

Isto aqui define o título (title) e a letra (lyrics) da canção (song), assim que criarmos uma.

Este é um método que usa as variáveis de instância do objeto.



### 2 Crie um objeto novo.

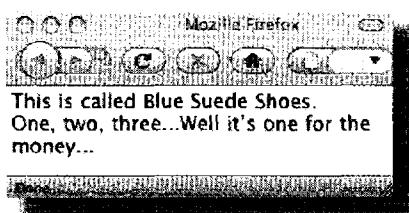
A nossa nova canção tem o valor "Blue Suede Shoes" como o nome.

```
$shoes_song = new Song('Blue Suede Shoes', 'Well it\'s one for the money...');
```

Aqui, nós chamamos o método sing() para o nosso objeto.

### 3 A sua canção pode cantar a si mesma!

Quando você executa este código, obtém o seguinte:



Mas se podemos simplesmente escrever o código echo sem nada desse negócio de objetos, por que usar o PHP orientado a objetos?

Existem algumas ótimas razões...

## #5. PHP orientado a objetos (continuação)

Em vez de um conjunto ordenado de instruções, as suas estruturas de dados tornam-se objetos. Os objetos incluem não somente a definição dos seus dados, mas também todas as operações que podem ser realizadas neles. No nosso exemplo Song, nós definimos o título e a letra da canção dentro da classe, e criamos o método sing() dentro da classe. Se precisássemos adicionar outras funcionalidades ao nosso objeto Song, só teríamos de adicionar novos métodos e variáveis à nossa classe Song. Por exemplo, se quiséssemos que o autor de cada canção ficasse associado a cada objeto, poderíamos adicionar isso, como uma variável, à nossa classe.

O poder da orientação a objetos realmente aparece quando a aplicação cresce. Suponha que decidíssemos usar a classe Song como parte de uma aplicação de karaokê, com centenas ou até milhares de objetos song individuais, cada um com o seu título, letra e autor. Agora digamos que alguém deseje escolher apenas dentre as canções escritas por Elvis. Tudo o que teríamos de fazer é olhar a variável de instância referente ao autor de cada objeto.

E para enviar a letra da canção à aplicação de karaokê? Nós poderíamos simplesmente chamar o método sing() para cada objeto executado. Embora estejamos chamando exatamente o mesmo método para cada objeto, ele estará acessando dados únicos a cada um dos objetos.

### **Assim, temos duas grandes vantagens para se usar PHP Orientado a Objetos:**

Os objetos podem ser reutilizados facilmente. Eles são elaborados para serem independentes do código onde são usados, e podem ser reutilizados conforme necessário.

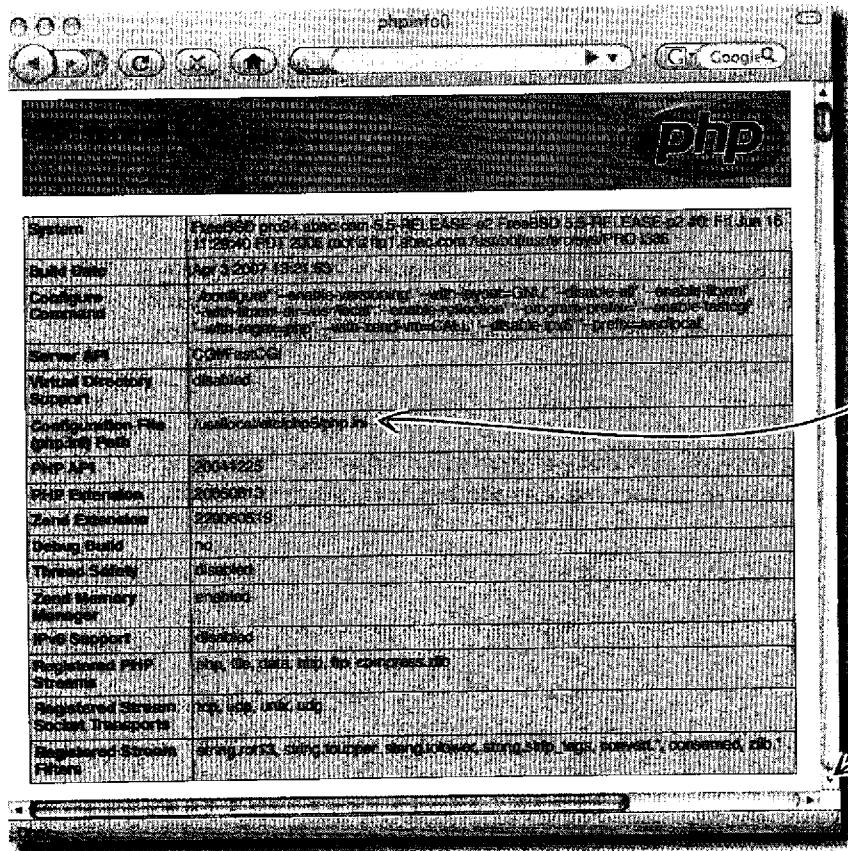
O código fica mais fácil de entender e manter. Se um tipo de dados precisar se modificar, a mudança só ocorre no objeto, e em nenhuma outra parte do código.

Uma grande desvantagem é que, de forma geral, o código OO pode ser mais longo e demorar mais tempo para ser escrito. Se você só precisa exibir a letra de uma canção, então escrever um pequeno programa procedural poderá ser a melhor opção. Mas, no futuro, se achar que poderá querer criar essa aplicação de karaokê online, considere a possibilidade de usar o PHP orientado a objetos.

## #6. Tornando segura a sua aplicação PHP

Existem alguns passos simples que você pode seguir para proteger seus scripts PHP daqueles nefastos hackers que, neste exato momento, estão a postos nos seus teclados esperando que cometa um vacilo.

- 1 Remova referências a `phpinfo()`. Quando começar a criar aplicações PHP em novos servidores web, você provavelmente irá criar um script contendo a função `phpinfo()`, para ver qual versão do PHP você está usando e se ela tem suporte ao MySQL, junto com uma lista de outras bibliotecas instaladas. Não tem problema em fazer essa checagem com `phpinfo()`, mas deve remover essa função depois de ter feito a verificação. Se não o fizer, qualquer hacker que descobrir uma nova vulnerabilidade do PHP poderá checar se o seu site é suscetível a ela.
  - 2 Se não estiver usando um serviço de web hosting e tiver acesso ao arquivo `php.ini`, há algumas modificações que pode fazer para tornar suas aplicações PHP mais seguras. Ironicamente, a localização do seu arquivo `php.ini` pode ser encontrada usando-se `phpinfo()`:



Eis o caminho até o seu arquivo php.ini. Depois de anotá-lo, lembre-se de deletar a função phpinfo().

Existem mais informações sensíveis na parte de baixo da página.

## *medidas de segurança no php*

# #6. Tornando segura a sua aplicação PHP (continuação)

Eis algumas configurações específicas que deveria considerar para o arquivo php.ini. Abra esse arquivo em um editor de texto, faça as modificações, salve-as e depois reinicie seu servidor web.

```
safe_mode = On
```

Quando você ativa o safe\_mode, nenhum script PHP pode ser chamado por outro script, pertencente a outro usuário, do mesmo servidor web. É claro que, se precisar permitir que os scripts de outros usuários chamem os seus, não poderá usar esta configuração.

```
open_basedir = directory[ : ... ]
```

Isto restringe os scripts e arquivos que o PHP poderá executar ou acessar neste diretório e nos subdiretórios dentro dele.

```
expose_php = Off
```

Com isto definido como On, todo navegador que visitar o seu site receberá informações, na forma de cabeçalhos, revelando detalhes sobre o seu servidor PHP. Definindo-o como off, você esconde essas informações e torna o seu servidor um pouco menos exposto.

```
display_errors = Off
```

Uma vez que tenha desenvolvido sua aplicação e que ela esteja rodando no seu servidor, você não precisa mais ver todas as mensagens de erro. A esta altura, espera-se que já tenha corrigido os erros, mas às vezes alguns escapam. Para esconder as mensagens de erro dos visitantes, defina este item como Off.

```
log_errors = On
```

Isto envia as suas mensagens de erro para um log. Quando quiser verificar se a sua aplicação contém erros, esse é um bom lugar para começar a procurar. Com display\_errors definido como Off e log\_errors como On, poderá ver as mensagens de erro, mas os visitantes do seu site não as verão.

```
error_log = filename
```

Você terá de verificar no seu software servidor web particular onde este arquivo se localiza. É nele que os seus erros serão escritos, quando log\_errors estiver definido como On.

## #7. Proteja sua aplicação de scripts cross-site

Você poderá ter ouvido falar dos scripts cross-site com o nome de XSS. Essa técnica refere-se a um tipo de ataque contra aplicações web, no qual um determinado código é passado para o script que processa os seus formulários, permitindo ao hacker assumir o controle do seu output. Esse é um grande problema de segurança nas aplicações web PHP. Vamos ver o que é exatamente e como nos defendermos desse tipo de ataque.

O ataque de script cross-site geralmente ocorre contra sites que exibem dados submetidos pelos usuários. Quaisquer dados que você recebe dos seus usuários e exibe no site poderiam ser corrompidos e fazer com que os visitantes do seu site fiquem vulneráveis ao hacker.

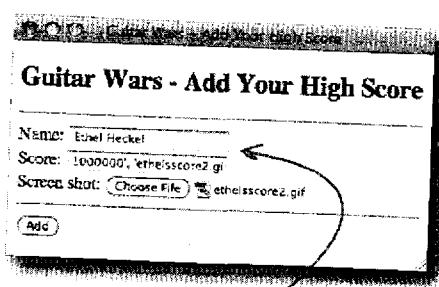
Usando um ataque XSS, o hacker pode fazer uma série de coisas. Uma das piores delas é redirecionar a sua página de resultados para uma página de um site sob o controle do hacker, onde ele poderá pedir mais informações ao usuário. O seu usuário poderá não perceber que não está mais no seu site e, uma vez que ele confia em você, poderá fornecer informações sensíveis diretamente no site do hacker.

Eis como isso poderia acontecer no site do Guitar Wars:

Ethel, em vez de escrever o nome dela no campo "Nome" do formulário, digita algum código JavaScript. No exemplo, ela está usando a função window.location para redirecionar o navegador para o site dela. E uma vez que ela controla o seu próprio site, poderá mostrar ao visitante qualquer página que quiser, inclusive uma idêntica à do Guitar Wars. Ela poderia fazer algo ainda mais nefasto no caso de sites onde as pessoas submetem informações mais importantes do que pontuações, como por exemplo, informações financeiras.

Existem outras coisas, ainda mais insidiosas, que ela poderia fazer, incluindo roubar cookies ou apresentar ao usuário uma tela que se pareça com uma tela de login. Assim que o usuário se conecta, ela passa a ter o seu nome e senha, e poderá então usar essas informações para se conectar no site original como se fosse o usuário hackeado.

Então, como evitar os ataques de scripts cross-site nas suas aplicações web?



### Guitar Wars - Add Your High Score

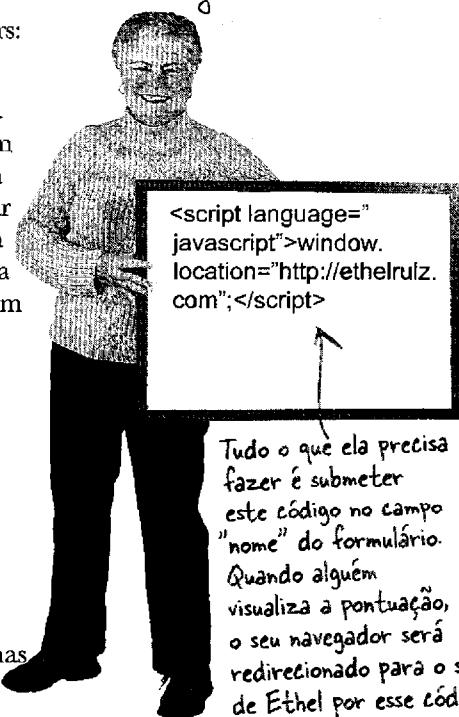
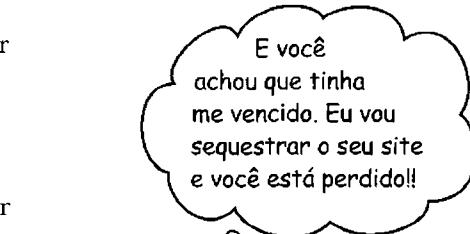
Name: Ethel Hecket

Score: 1000000; ethelsscore2.gif

Screenshot: Choose File ethelsscore2.gif

Add

Ethel não pode mais trapacear, mas ela pode redirecionar a página de pontuações para o seu próprio site, usando o ataque de script cross-site.



```
<script language="javascript">window.location="http://ethelruiz.com";</script>
```

Tudo o que ela precisa fazer é submeter este código no campo "nome" do formulário. Quando alguém visualiza a pontuação, o seu navegador será redirecionado para o site de Ethel por esse código JavaScript.

você está aqui >

725

## #7. Proteja sua aplicação de scripts cross-site (continuação)

Felizmente, se estiver validando seus dados, já está com meio caminho andado para proteger sua aplicação. Você já aprendeu como fazer isso no Guitar Wars. Eis as três orientações básicas para manter suas aplicações seguras:

### **Valide tudo**

Quaisquer dados que receber, como por exemplo input de formulários, precisam ser validados, de modo que o código do hacker seja detectado antes que possa fazer algo contra a sua aplicação. Se assumir que os dados são prejudiciais até que a validação prove que não são, você estará muito mais seguro.

### **Funções internas do PHP podem ajudar**

Use funções do PHP, como `strip_tags()`, para ajudá-lo a sanear os dados externos. `strip_tags()` é uma ótima função que remove quaisquer tags HTML presentes em uma string. Assim, se usar `strip_tags()` no `$_POST['name']` de Ethel, o resultado final será:

```
window.location='http://ethelrulz.com'
```

Embora isso ainda não seja um nome, não irá mais redirecionar o navegador, porque as tags JavaScript necessárias para isso foram removidas.

### **Os dados são culpados até provar a sua inocência**

Comece com a validação mais restrita que puder, e só facilite se for preciso. Por exemplo, se começar aceitando apenas números em um campo de número telefônico e depois aceitar hífens ou parênteses, será muito mais seguro do que se permitisse quaisquer caracteres alfanuméricos originalmente. Ou, no caso do Guitar Wars, se nós não permitirmos nada além de letras no campo do nome, nunca teremos o sinal menor que (<) que abre o maléfico código JavaScript de Ethel. As expressões regulares (Capítulo 10) podem ser bastante úteis para garantir que só sejam permitidos os dados exatos que você quer.

## #8. Precedência de operadores

Considere esta linha de código.

`$marbles = 4 / 2 - 1;` ← O resultado será 1.

O valor armazenado por \$marbles poderia ser 1 ou 4. É impossível saber só olhando o código, mas podemos assumir certas regras de precedência. Por precedência, queremos dizer a ordem em que os cálculos são executados. Os operadores do PHP são executados em uma determinada ordem. No exemplo acima, a divisão ocorre antes da subtração, e por isso \$marbles será igual a 1. Dependendo do output que precisemos do nosso código, poderíamos tê-lo escrito de duas formas diferentes:

`$marbles = (4 / 2) - 1;`  
`$marbles = 4 / (2 - 1);`

Na primeira expressão, dividimos 4 por 2 e depois subtraímos 1. No segundo caso, subtraímos 1 de 2 e depois dividimos 4 pelo resultado, que é 1. O uso de parênteses lhe permite controlar de forma precisa a ordem das operações. Mas conhecer a precedência dos operadores do PHP poderá ajudá-lo a descobrir o que está acontecendo em uma expressão complexa. E, acredite, isso lhe ajudará também a tirar os bugs do seu código quando se esquecer de usar os parênteses.

Antes de passarmos para a lista de precedência dos operadores, eis aqui outro motivo pelo qual deve usar os parênteses. Considere o seguinte:

`$marbles = 4 - 3 - 2;` ← O resultado será -1.

Nenhuma regra de precedência se aplica aqui. O resultado poderia ser 3 ou -1. Isso é bem confuso quando se está escrevendo código. Portanto, é melhor programar com parênteses, como nestas duas linhas:

`$marbles = 4 - (3 - 2);`  
`$marbles = (4 - 3) - 2;`

Agora a lista, em ordem da maior precedência (avaliada primeiro) até a menor (avaliada por último).

| Operador                                                          | Tipo de Operador     |
|-------------------------------------------------------------------|----------------------|
| <code>++ --</code>                                                | acrúscimo/decrúscimo |
| <code>* / %</code>                                                | aritmético           |
| <code>+ - .</code>                                                | aritmético e string  |
| <code>&lt; &lt;= &gt; &gt;= &lt;&gt;</code>                       | comparação           |
| <code>== != === !==</code>                                        | comparação           |
| <code>&amp;&amp;</code>                                           | lógico               |
| <code>  </code>                                                   | lógico               |
| <code>= += -= *= /= .= %= &amp;=  = ^= &lt;&lt;= &gt;&gt;=</code> | atribuição           |
| <code>and</code>                                                  | lógico               |
| <code>xor</code>                                                  | lógico               |
| <code>or</code>                                                   | lógico               |

Os operadores de comparação, como aqueles que você usa em declarações IF, também entram no ordenamento de precedência.

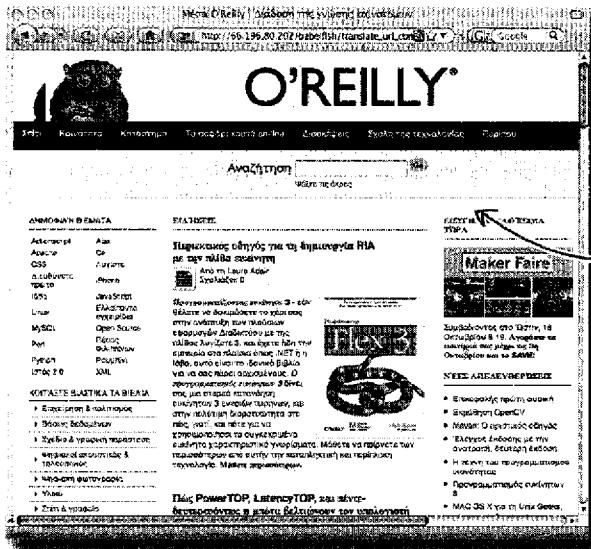
## #9. Qual é a diferença entre PHP 5 e PHP 6

No momento em que escrevemos este livro, a versão mais recente do PHP é a 5. Mas o PHP 6 está sendo elaborado e já está disponível para desenvolvedores aqui: <http://snaps.php.net/>.

As diferenças entre o PHP 4 e o 5 são muito maiores do que entre o 5 e o 6. De muitas formas, o 6 é mais um refinamento dos recursos de orientação a objetos introduzidos na versão 5. Outras modificações incluem melhor suporte a XML e Unicode.

### Maior suporte Unicode

Suponha que a sua aplicação precise fazer output de texto em grego.



Considere o tipo de coisas que você, às vezes, precisa fazer com strings quando, por exemplo, precisa saber o tamanho delas, ou precisa classificá-las. É bem simples na sua língua nativa, mas quando está trabalhando com caracteres de outros alfabetos, as operações com strings ficam muito mais complicadas.

O Unicode é um conjunto de caracteres e tecnologias para codificá-los. Em Unicode, o caracter grego que se parece com um triângulo tem um valor número específico atribuído a ele, junto com outros caracteres de outros alfabetos. O Unicode é um padrão, o que significa que ele é aceito e empregado pelos mais importantes fornecedores de tecnologias. No Unicode, todo caracter tem um número único, não importa que língua, programa ou plataforma esteja sendo usada. Antes da versão 5, o PHP não tinha suporte real ao Unicode. O PHP 6 tem um suporte melhorado a strings Unicode nas suas funções, bem como novas funções criadas especificamente para se criar e decodificar Unicode.

## #9. Qual é a diferença entre PHP 5 e PHP 6 (continuação)

### Melhorias de OO, suporte a XML, e outras modificações

O PHP 5 oferece um modelo de programação orientado a objetos, mas ainda permite a escrita no estilo procedural. O PHP 6 vai mais ao fundo do reino da orientação a objetos. Uma das grandes modificações é que as funções dinâmicas não poderão mais ser chamadas com sintaxe estática. Existem também várias modificações pequenas, porém importantes, na maneira como o PHP manipula o seu código OO, de modo a torná-lo mais consistente com outras linguagens orientadas a objetos, como C++ e Java.

Algumas outras modificações:

- » Tanto o XML Reader quanto o XML Writer serão extensões no PHP 6, facilitando o trabalho com arquivos XML.
- » As opções register\_globals, magic\_quotes e safe\_mode do arquivo php.ini não estarão mais disponíveis.
- » A extensão ereg, que fornecia outra forma de criar expressões regulares, foi removida. Felizmente, o mesmo código com preg\_match() que vimos neste livro continuará sendo a principal forma de criar expressões regulares no PHP 6.
- » Um tipo integer de 64 bits será adicionado.
- » Os arrays multidimensionais poderão usar foreach.
- » A versão 6 do PHP é, mais do que qualquer coisa, uma atualização que organiza e refina a linguagem.

Nenhum código deste livro usa funções dinâmicas, portanto não precisa se preocupar se o código vai ou não funcionar com o PHP 6.

## #10. Reutilizando o PHP de terceiros

Nem sempre é necessário escrever o seu próprio código PHP a partir do zero. Em alguns casos, é melhor reutilizar o código já escrito por terceiros. Seguem abaixo alguns pacotes de software baseados em PHP bastante populares e bem-sucedidos, que deverá pensar em usar, caso precise e prefira não ter de reinventar a roda do PHP. Ah, e são todos gratuitos!

### Drupal

Um dos projetos PHP mais impressionantes até hoje, o Drupal é um poderoso sistema de gerenciamento de conteúdo que pode ser usado para se criar praticamente qualquer tipo de site baseado em conteúdo. A NASA, o The Onion, a Electronic Frontier Foundation e o Popular Science são alguns dos que usam o Drupal nos seus web sites. Ele é flexível o suficiente para criar basicamente qualquer coisa que tenha grandes quantidades de conteúdo. Visite <http://drupal.org/>.

### phpBB

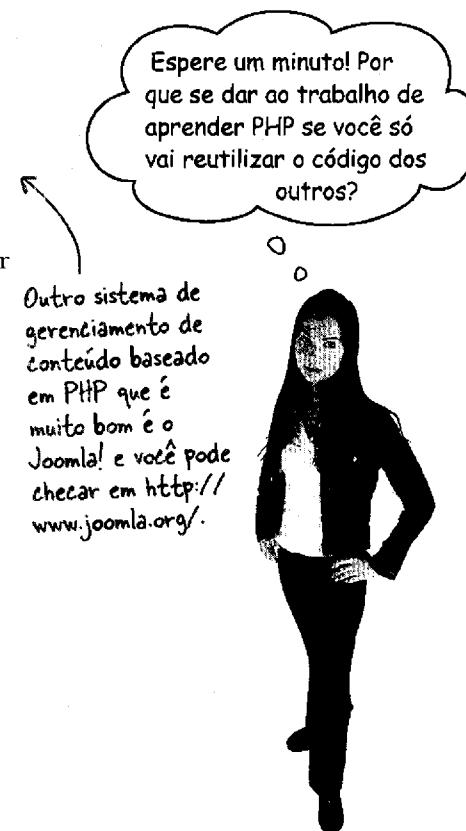
Líder da categoria dos quadros de mensagens online (forums), o phpBB é bastante fácil de usar para a criação do seu próprio forum. É extremamente flexível e difícil de ser batido naquilo que faz tão bem – gerenciar discussões temáticas. Descubra mais sobre ele em <http://www.phpbb.com/>.

### Coppermine Gallery

Se o que tem em mente é fazer hosting de imagens, a Coppermine Gallery é a aplicação PHP para o seu caso. Em uma era de Flickr, Photobucket, Shutterfly e Snapfish, manter a sua própria biblioteca de fotos parece um exagero. Mas o controle significa maior poder, e se quiser controle total sobre as suas fotos, dê uma olhada na Coppermine Gallery, em <http://coppermine-gallery.net/>.

### WordPress

Um dos campeões da blogosfera, o WordPress é um software baseado em PHP que lhe permite criar e fazer a manutenção de blogs como mínimo de esforço. Existe muita competição nessa área, de modo que poderá desejar pesquisar um pouco mais, mas o WordPress é simplesmente uma ótima opção caso queira criar um blog. Baixe-o em <http://wordpress.org/>.



**Porque reutilizar código não é sempre tão fácil quanto parece – em alguns casos, isso requer algumas habilidades em PHP.**

Muitos pacotes de software PHP ainda requerem alguma personalização, e isso frequentemente, requer sólidas habilidades de desenvolvimento PHP. Não apenas isso, mas poderá optar por reutilizar apenas um pequeno componente do código de terceiros ou então não reutilizar nada. De uma forma ou de outra, tendo conhecimento de PHP, você tem opções – e opções sempre são uma boa coisa!

## Apendice ii: Configure um Ambiente de Desenvolvimento



# Um Lugar para Brincar



Ele acha que eu sou uma ótima cozinheira, mas é que eu escondo os meus erros antes que ele os veja.

**Você precisa de um lugar para praticar suas recém-adquiridas habilidades em PHP e MySQL, porém sem tornar os seus dados vulneráveis na web.** É sempre uma boa idéia ter um local seguro para desenvolver sua aplicação PHP antes de lançá-la no mundo (ou seja, na web). Este apêndice contém instruções para a instalação de um servidor web, do MySQL e do PHP, para lhe fornecer um lugar seguro onde possa trabalhar e praticar.

## Crie um ambiente de desenvolvimento PHP

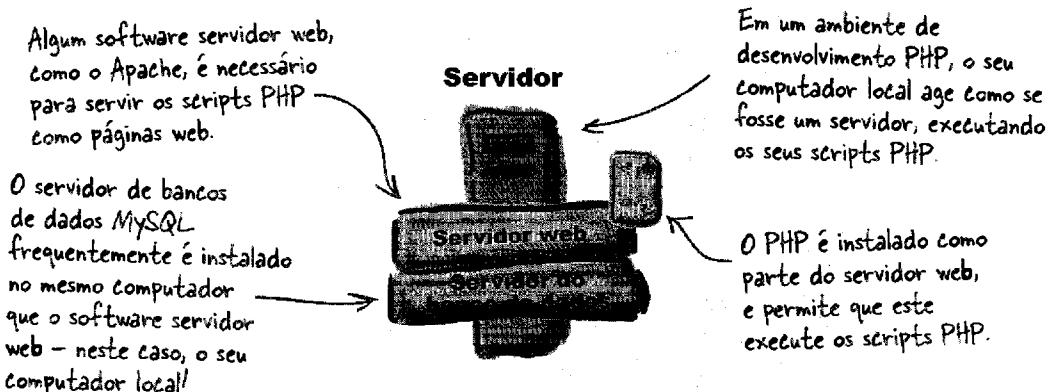
Antes de poder colocar a sua aplicação no ar, você precisa desenvolvê-la. E nunca é uma boa ideia desenvolver sua aplicação diretamente na web, onde todos possam vê-la. Você pode instalar algum software localmente para criar e testar sua aplicação antes de colocá-la online.

São três os tipos de software que precisará ter em seu computador local para criar e testar aplicações PHP:

1. Um servidor web
2. PHP
3. Um servidor de bancos de dados MySQL

O PHP não é um servidor; é um conjunto de regras que o seu servidor web entende e interpreta. Tanto o servidor web quanto o servidor MySQL são programas executáveis que rodam em um computador.

Tenha em mente que estamos falando de configurar o seu computador local como um servidor web para desenvolvimento PHP. Você ainda precisará de um servidor web online onde colocar a sua aplicação finalizada, para que as pessoas possam acessá-la e usá-la.



## Descubra o que você tem

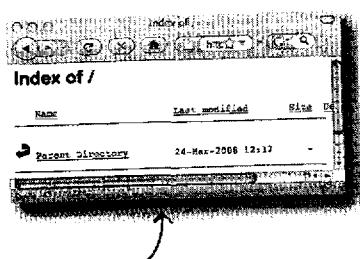
Antes de tentar instalar qualquer um dos softwares necessários para o desenvolvimento PHP, a sua melhor opção é avaliar o que já tem instalado. Vamos dar uma olhada em cada um dos tipos de software e ver como pode descobrir o que já existe no seu sistema.

A plataforma do seu computador local faz muita diferença para saber o que já está instalado. Por exemplo, o Mac OS X tem um servidor web instalado por padrão, enquanto que a maioria dos computadores Windows não tem.

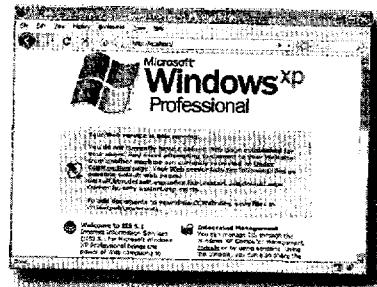
**NOTA: Este apêndice cobre o Windows 2000, XP, Vista, Windows Server 2003/2008, ou outros sistemas operacionais Windows de 32 bits. Para o Mac, aplica-se ao Mac OS X 10.3.x ou acima.**

## Você tem um servidor web?

Se estiver usando um PC ou Mac mais recente, provavelmente já tem um servidor web. Para descobrir isso rapidamente, em qualquer desses sistemas, abra uma janela do seu navegador e digite `http://localhost` na barra de endereços. Se você vir uma página introdutória, isso significa que o seu navegador web está acessando a sua própria máquina local.



Se tiver uma máquina Windows ou Mac com o servidor web Apache instalado, poderá ver algo como isto.



Se tiver uma máquina Windows com IIS, poderá ver algo como isto.

## Você tem o PHP? Qual versão?

Se tiver um servidor web, é possível verificar facilmente se tem o PHP instalado, e em caso afirmativo, qual versão. Crie um novo script chamado `info.php` e digite nele o seguinte:

```
<?php phpinfo(); ?>
```

Salve esse arquivo no diretório que o seu servidor web usa. No Windows, geralmente é:

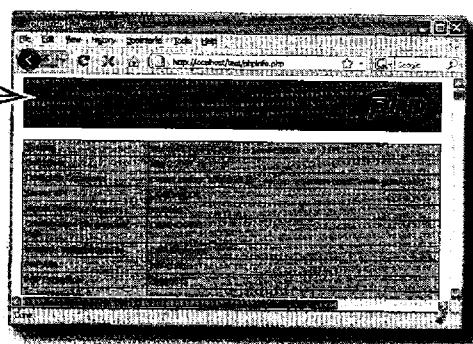
C:\inetpub/wwwroot/

No Mac, geralmente é algo como:

/Users/seunome/sites/

Se tentar abrir esse arquivo no seu navegador digitando `http://localhost/info.php`, verá algo como a figura abaixo, se tiver o PHP instalado:

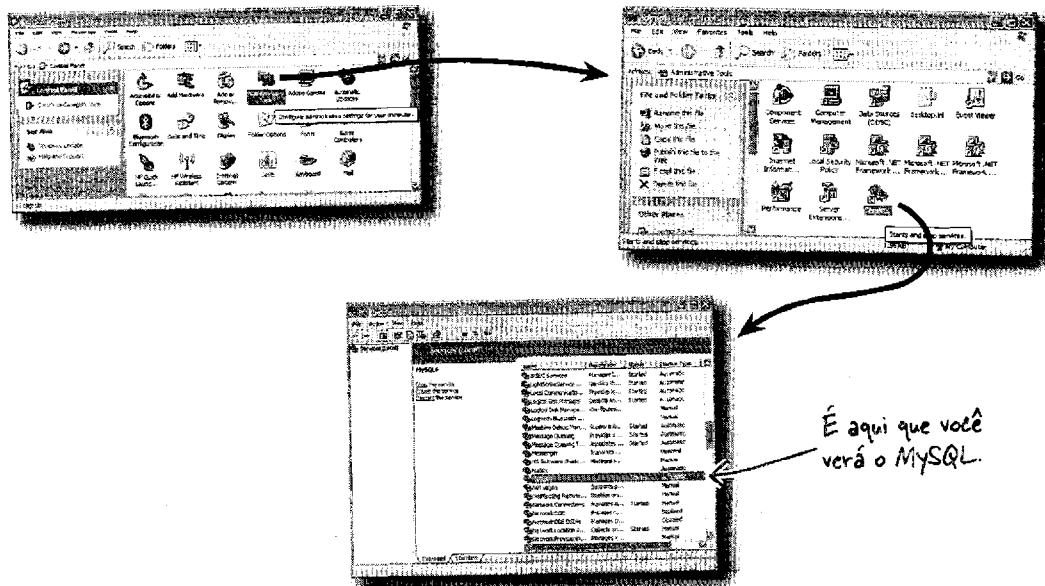
Eis a versão do PHP que você tem instalada.



verificando sua versão do mysql

## Você tem o MySQL? Qual versão?

No Windows, é possível saber abrindo o Painel de Controle -> Ferramentas Administrativas -> Serviços:



Para determinar se você tem o MySQL no MAC, abra o seu terminal e digite:

```
cd /user/local/mysql
```

Se o comando funcionar, é porque você tem o MySQL instalado.  
Para saber qual a versão, digite:

```
mysql
```

O Terminal MySQL  
também é conhecido  
como "monitor"  
MySQL.

Se este comando executar com sucesso, significa que o MySQL está instalado.

Eis a versão do MySQL que você tem instalada.

A screenshot of a Mac terminal window titled 'Terminal'. The window contains the following text:  
\$ cd /usr/local/mysql  
\$ mysql  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 3  
Server version: 5.0.51b MySQL Community Server (GPL)  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
mysql>

# Comece pelo Servidor Web

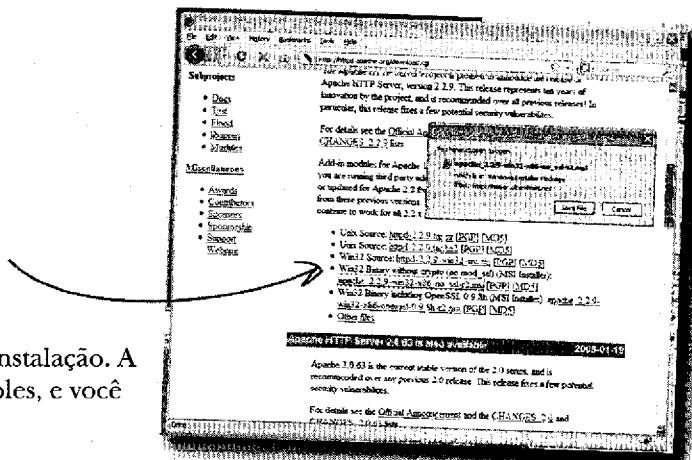
Dependendo da sua versão do Windows, é possível baixar o Servidor de Informações Internet (Internet Information Server, ou IIS) da Microsoft, ou então o servidor web de código aberto Apache. Se precisar de um servidor no Mac, a sua melhor opção é usar o Apache, uma vez que ele já estará instalado.

Seguem breves instruções para a instalação do Apache no Windows:

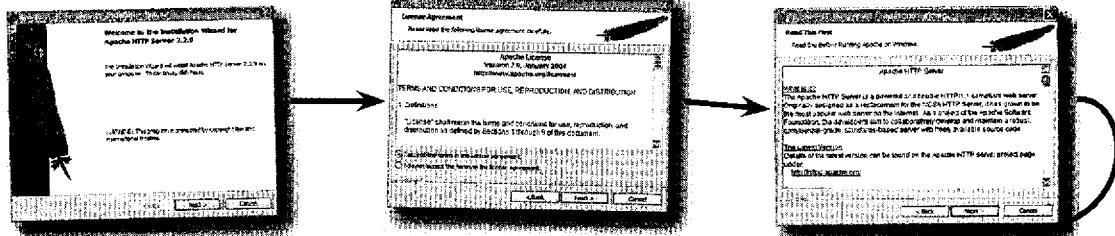
Acesse <http://httpd.apache.org/download.cgi>

Se estiver usando o Windows, sugerimos que você baixe o arquivo apache\_2.2.9-win32-x86-no\_ssl-r2.msi. Ele automaticamente instalará o Apache para você depois que o baixar e clicar duas vezes nele.

Baixe esta versão e clique duas vezes no arquivo após tê-lo baixado.



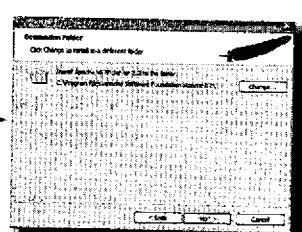
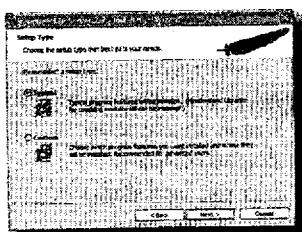
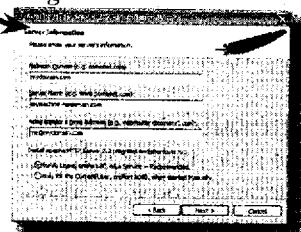
Em seguida, verá o Assistente de Instalação. A maioria das instruções é bem simples, e você pode aceitar as opções padrões.



Escolha o domínio em que o seu computador se encontra. Se você não tiver um domínio, pode digitar localhost.

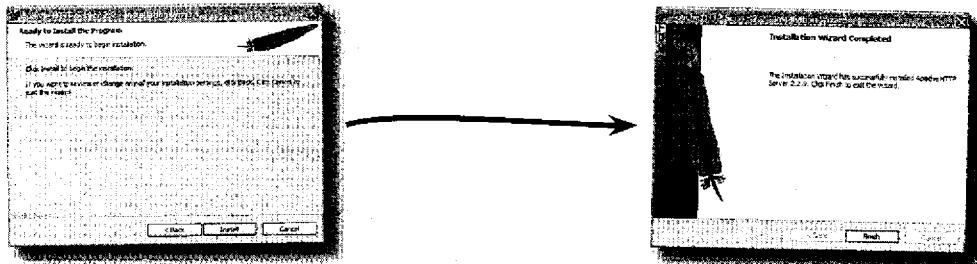
A opção de instalação típica (typical) deverá ser suficiente para as suas necessidades.

Em geral, você pode escolher o diretório padrão para a instalação do software.



## Instalação do Apache... concluída

Está quase pronto. Clique em Install e espere por mais ou menos um minuto até que a instalação se complete. Isso é tudo!

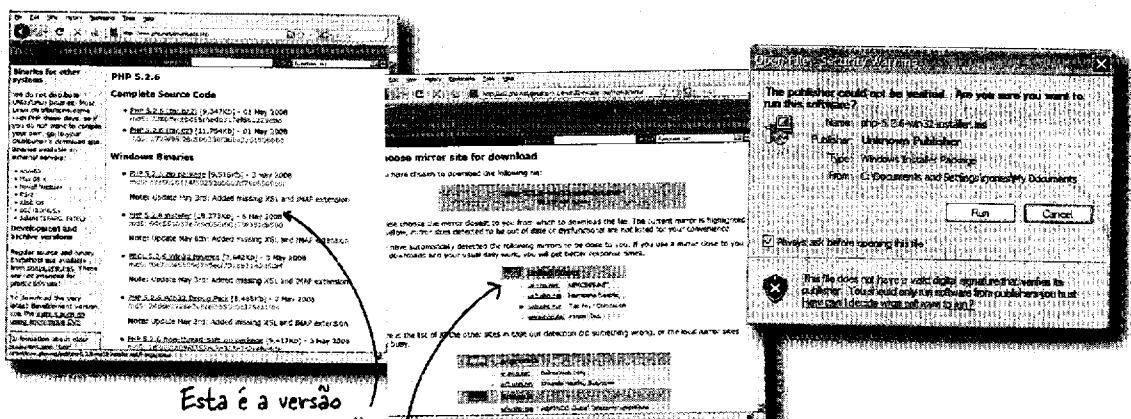


O seu servidor web é configurado para se iniciar automaticamente quando você iniciar seu computador. Mas você pode controlar esse comportamento usando o painel Service, interrompendo e iniciando o serviço na caixa de diálogo Painel de Controle -> Ferramentas Administrativas -> Serviços, onde ele agora irá aparecer.

## Instalação do PHP

Acesse <http://www.php.net/downloads.php>.

Como no caso do Apache, se estiver usando o Windows, nós recomendamos que baixe a versão com o instalador para Windows, php-5.2.6-win32-installer.msi. Esse arquivo irá automaticamente instalar o PHP para você, depois que o baixar e clicar nele duas vezes.



Esta é a versão .msi para instalação no Windows.

Depois de clicar no arquivo, clique em uma das localidades e faça o download.

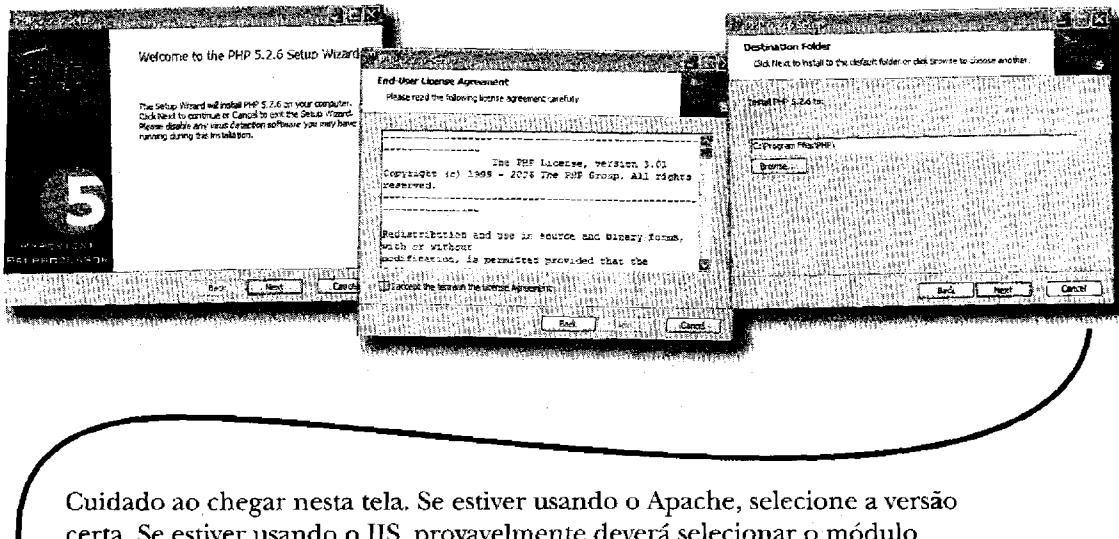
Após ter baixado o arquivo, clique nele duas vezes. Clique no botão Run para iniciar a instalação.

# Passos para a instalação do PHP

Tudo começa com um Setup básico.

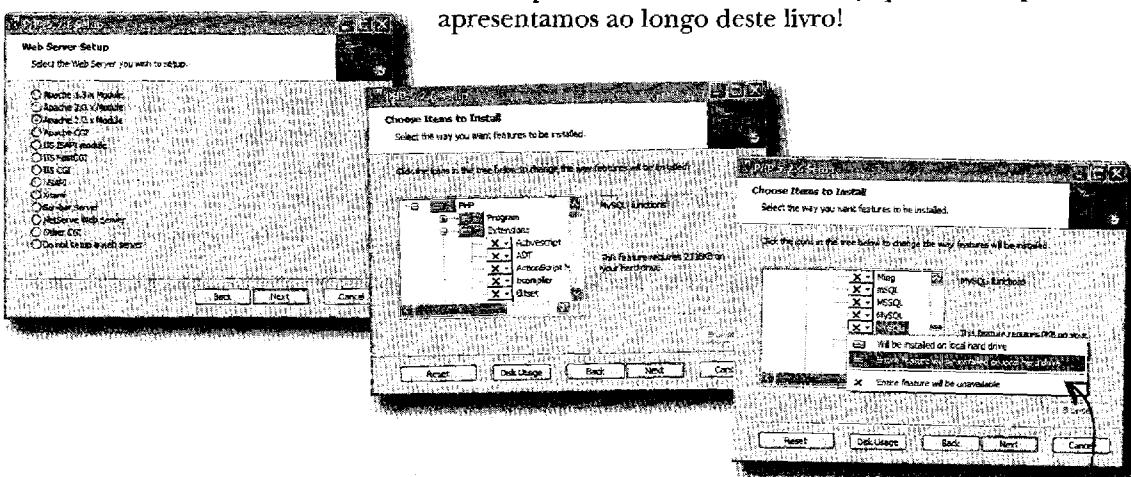
Aceite os Termos de Licença para continuar.

Selecionar a pasta de instalação padrão em geral é uma boa ideia.



Cuidado ao chegar nesta tela. Se estiver usando o Apache, selecione a versão certa. Se estiver usando o IIS, provavelmente deverá selecionar o módulo IISAPI. Verifique no seu software particular para determinar exatamente o que é que você necessita.

Esta próxima tela também é importante. Você precisa acessar as opções sob Extensions e escolher MySQLi. Isso lhe permitirá usar as funções mysqli do PHP que apresentamos ao longo deste livro!



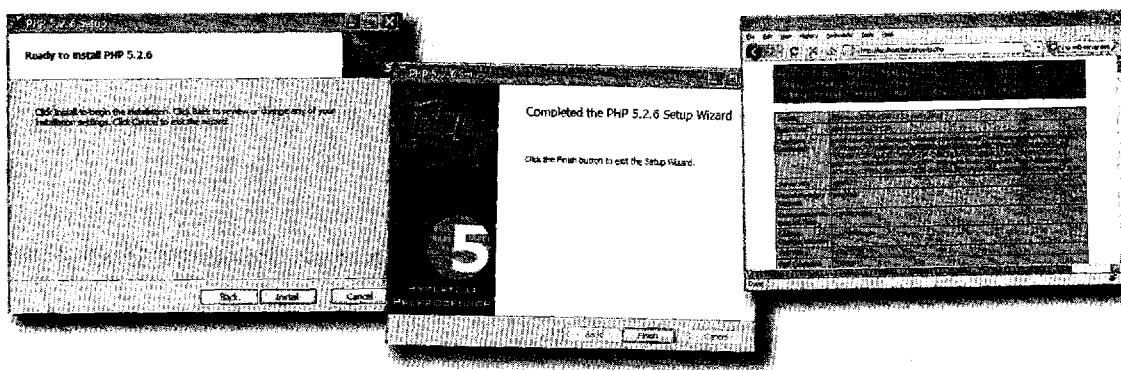
Acesse as opções para "Extensions" e clique em MySQLi. Clique na opção "Entire feature".

*instalando mysql no windows*

## Passos para a instalação do PHP... concluídos

Isso é tudo. Clique em Install, e depois em Done para fechar o instalador.

Agora experimente dar uma olhada no seu arquivo <http://localhost/info.php> no seu navegador web, e veja qual versão está aparecendo.



## Instalando o MySQL

### Instruções e Resolução de Problemas

Você ainda precisa instalar o MySQL, então vamos ver os passos para baixar e instalar o software. O nome oficial da versão gratuita do servidor MySQL RDBMS, atualmente, é MySQL Community Server.

Segue abaixo uma lista de passos para a instalação do MySQL no Windows e no Mac OS X. Isto não deve substituir as excelentes instruções encontradas no site do MySQL e nós recomendamos que acesse o site e as leia! Para instruções muito mais detalhadas, bem como um guia de resolução de problemas, acesse:

Baixe a versão 6.0 ou acima.

<http://dev.mysql.com/doc/refman/6.0/en/windows-installation.html>

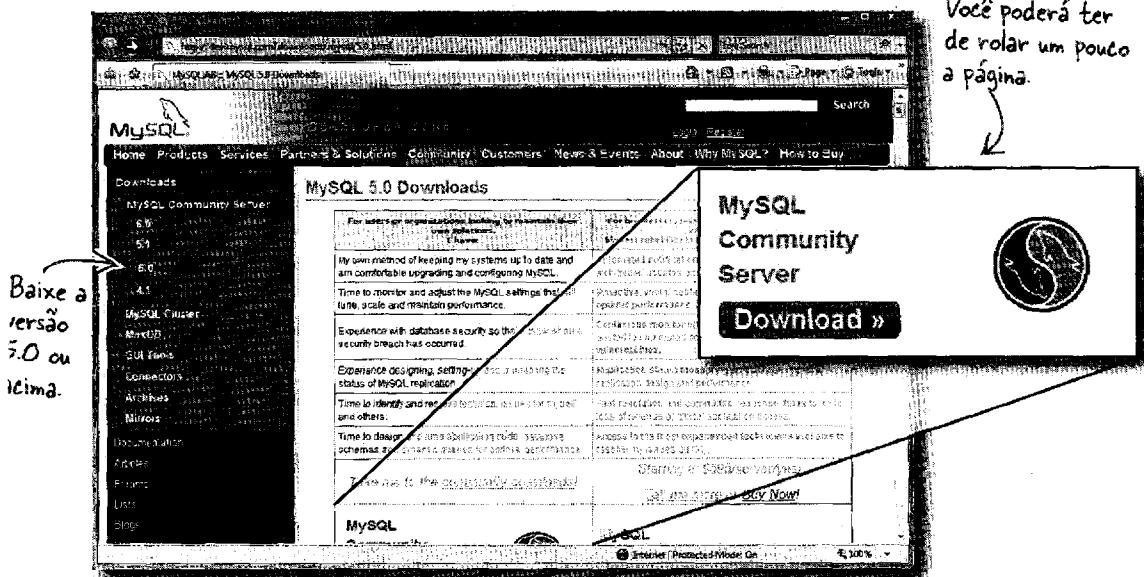
Também irá gostar do MySQL Query Browser que mencionamos anteriormente. Nele, é possível digitar suas consultas e ver os resultados dentro da interface do software, em vez de em uma janela de console.

# Passos para a instalação do MySQL no Windows

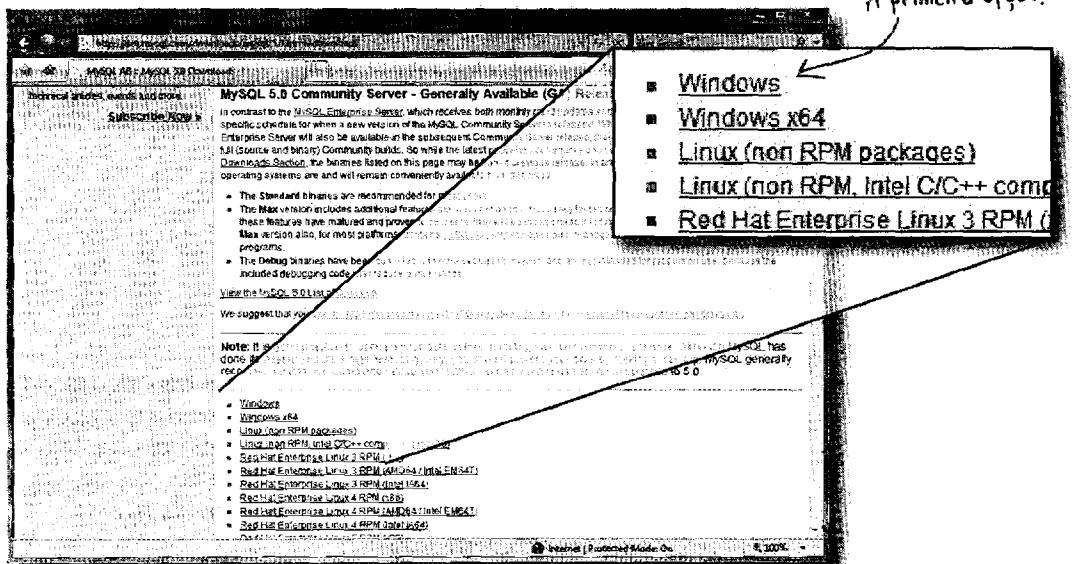
**1** Acesse:

<http://dev.mysql.com/downloads/mysql/6.0.html>

e clique no botão de download do MySQL Community Server.



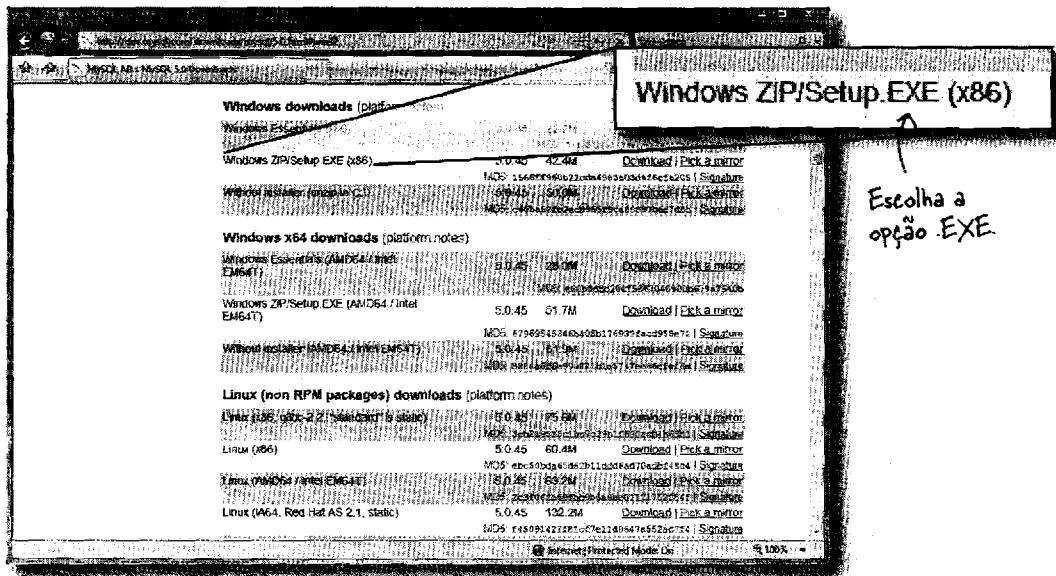
**2** Escolha Windows na lista.



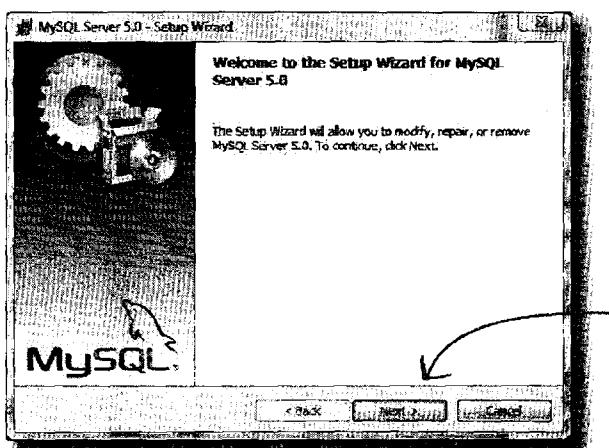
## *instalando mysql no windows (continuação)*

### Baixe o seu instalador

- 3** Em Windows downloads, nós recomendamos escolher a opção Windows ZIP/Setup.EXE, porque ela inclui um instalador que simplifica bastante o processo. Clique em Pick a Mirror.



- 4** Verá uma lista de localidades com uma cópia que pode baixar; escolha aquela mais próxima de você.  
**5** Terminado o download do arquivo, clique nele duas vezes para executá-lo. A partir deste ponto, você será guiado através da instalação pelo assistente de instalação. Clique no botão Next.



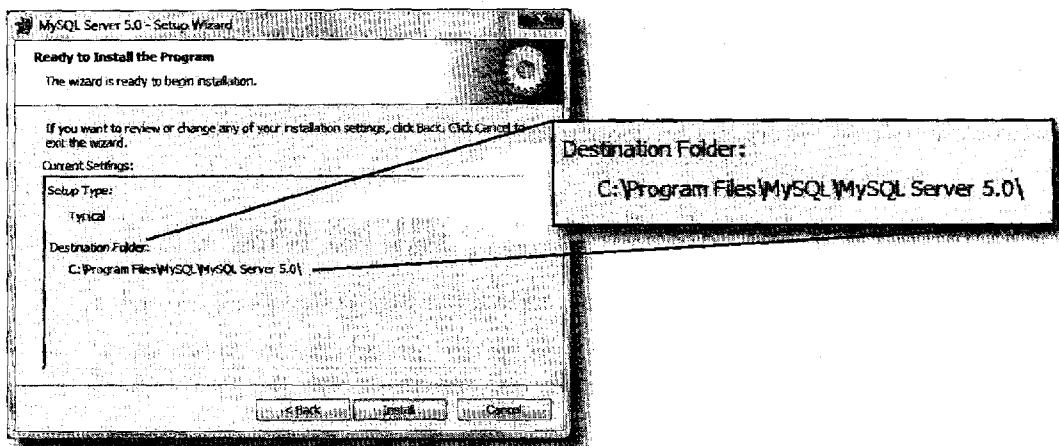
## Escolha uma pasta de destino

- 6** Será perguntado se quer a instalação Typical, Complete ou Custom. Para os nossos propósitos neste livro, escolha Typical.

Você pode mudar a localização no seu computador onde o MySQL será instalado, mas recomendamos que mantenha a localização padrão:

C:\Program Files\MySQL\MySQL Server 6.0

Clique no botão *Next*.



**Clique em "install" e pronto!**

- 7** Será visualizada a caixa de diálogo *Ready to Install* com *Destination Folder* ("pasta de destino") listada. Se estiver satisfeito com o diretório de destino, clique em *Install*. Caso contrário, clique em *Back*, depois em *Change* para mudar o diretório, e siga a partir daí.

Clique em *Install*.

## Habilitando o PHP no Mac OS X

O PHP vem incluído nos Macs, com o OS X versão 10.5+ (Leopard), mas não é habilitado por padrão. É preciso acessar o arquivo de configuração principal do Apache e ativar uma linha de código para habilitar o PHP. Esse arquivo se chama http.conf, e é um arquivo oculto, localizado dentro da pasta de instalação do Apache.

Você irá procurar pela seguinte linha de código, que tem um símbolo # na frente dela, indicando que ela está desativada como um comentário:

```
#LoadModule php5_module libexec/apache2/libphp5.so
```

É preciso remover o símbolo # e reiniciar o servidor, para habilitar o PHP. O documento http.conf é de propriedade de "root," o que significa que terá de digitar sua senha para modificá-lo. É uma boa ideia modificar o arquivo php.ini para que o Apache o utilize. Para informações mais detalhadas sobre como executar esses passos e habilitar o PHP, visite [http://foundationphp.com/tutorials/php\\_leopard.php](http://foundationphp.com/tutorials/php_leopard.php).

## Passos para a instalação do MySQL no Mac OS X

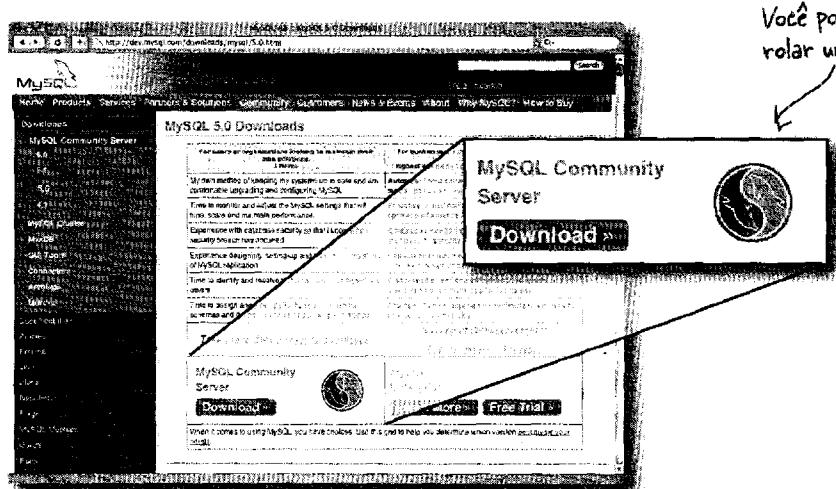
Se estiver usando o Mac OS X Server, uma versão do MySQL já deverá estar instalada.

Antes de começar, verifique se já não tem uma versão instalada. Vá até Applications/Server/MySQL Manager (Aplicações/Servidor/Gerenciador MySQL) para acessá-la.

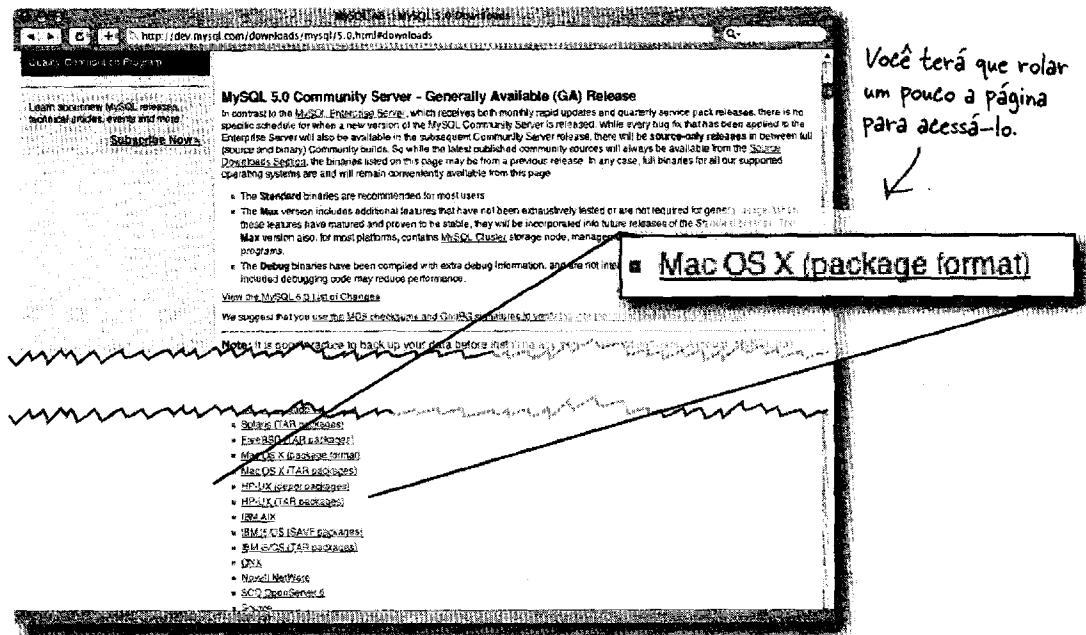
- 1 Acesse:

<http://dev.mysql.com/downloads/mysql/6.0.html>

e clique no botão Download do MySQL Community Server.



- 2 Escolha Mac OS X (package format) na lista.



- 3 Escolha o pacote apropriado para a sua versão do Mac OS X. Clique em Pick a Mirror.
- 4 Será visualizada uma lista de localidades que têm uma cópia que pode baixar; escolha aquela que estiver mais perto de você.
- 5 Terminado o download, clique duas vezes no arquivo para executá-lo. Agora é possível abrir uma janela do Terminal no seu Mac e digitar:

```
shell> cd /usr/local/mysql
shell> sudo ./bin/mysqld_safe
```

(Digite sua senha, se necessário)

(Pressione Control-Z))

```
shell> bg
```

(Pressione Control-D ou digite exit para sair do shell)

Se estiver usando uma ferramenta GUI, como o phpMyAdmin, consulte a sua documentação para saber como acessá-la depois que o MySQL estiver instalado com sucesso.

## Passando do ambiente de produção para um site no ar

Você passou dias ou semanas trabalhando no seu site, e acha que agora está tudo pronto para ir ao ar. Para passar o seu site PHP e MySQL do seu computador local para web, é necessário um pouco de planejamento, bem com algumas técnicas específicas.

Primeiramente, é necessário se certificar de que o lugar onde o seu site será colocado tem as mesmas versões do PHP e do MySQL que usou para desenvolvê-lo. Caso contrário, talvez precise adaptar seu código aos recursos disponíveis. A maioria dos códigos deste livro são adaptáveis, mas poderá ter que ajustar seu código usando as antigas funções mysql do PHP, em vez das funções mysqli que usamos no livro. Se o problema for esse, consulte o item #1 dos Dez Principais Tópicos (que não abordamos), no Apêndice 1, para mais informações,

Se o software do seu site online for compatível, então o processo de transferir seu site para lá será simples. Eis os passos:

Os seus arquivos PHP precisam ser enviados, através de FTP, para o diretório web do seu site online.

1. Envie os arquivos PHP do seu servidor de produção para o diretório web do seu servidor online. Mantenha a estrutura dos arquivos intacta, e certifique-se de manter quaisquer pastas que tenha criado para conter os seus arquivos incluídos.



É preciso obter a estrutura das suas tabelas e dos dados armazenados nela. Eis como:

2. Faça um dump do banco de dados (o que lhe mostraremos logo a seguir) para obter as instruções MySQL de que precisa para criar suas tabelas, bem como as instruções INSERT de que necessita para transferir seus dados para o servidor online.



O seu dump SQL lhe dará a sintaxe exata para as suas instruções CREATE TABLE e INSERT.

3. Faça login no seu banco de dados online, onde possa rodar as instruções MySQL CREATE e INSERT para transferir seus dados do site local para o site online.



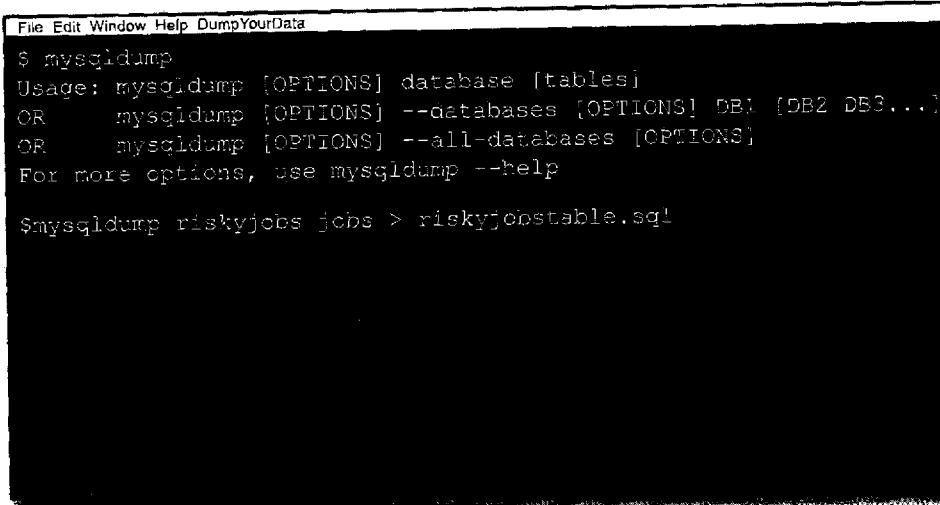
4. Modifique, conforme necessário, o código de quaisquer arquivos PHP para apontarem para o servidor online do banco de dados. Se não fizer isso, o seu código online irá tentar se conectar ao seu site de produção, e não o conseguirá.



Modifique as instruções `mysqli_connect()` para apontarem para o seu servidor MySQL online, junto com o nome de usuário e senha corretos para fazer a conexão.

## Dump seus dados (e as suas tabelas)

Você já enviou por FTP os seus arquivos PHP para o servidor online, mas os seus dados ainda não estão no servidor MySQL do site online. Quando a sua tabela contém muitos dados, a ideia de movê-la para outro servidor MySQL pode ser atemorizante. Felizmente, o MySQL oferece o programa MySQLdump, que lhe fornece uma maneira fácil de reproduzir a instrução CREATE TABLE necessária para recriar sua tabela, bem como todas as instruções INSERT para colocar seus dados na nova tabela. Você só precisa usar o programa MySQLdump. Para fazer uma cópia dos seus dados, com o propósito de transferi-los para outro servidor MySQL, digite o seguinte no seu terminal:



```
File Edit Window Help DumpYourData
$ mysqldump
Usage: mysqldump [OPTIONS] database [tables]
OR      mysqldump [OPTIONS] --databases [OPTIONS] DB1 [DB2 DB3...]
OR      mysqldump [OPTIONS] --all-databases [OPTIONS]
For more options, use mysqldump --help

$mysqldump riskyjobs jobs > riskyjobstable.sql
```

Isso enviará a instrução CREATE TABLE da tabela jobs para um arquivo de texto que acabamos de criar, chamado riskyjobstable.sql. Se você não emitir o argumento > riskyjobstable.sql, então as instruções CREATE TABLE e INSERT irão simplesmente ser exibidas no seu terminal. Tente entender o que elas fazem. Não é muito útil, mas verá todos os seus dados voando na tela, formatados elegantemente em instruções INSERT.

Uma vez que tenha enviado todos esses dados para o seu novo arquivo usando o sinal maior, você poderá usar o conteúdo desse arquivo como consultas MySQL no seu site online, para fazer a transferência das suas tabelas e seus dados.

## Prepare-se para usar seus dados

Prepare-se para transferir seus dados, executando uma instrução CREATE DATABASE no seu servidor MySQL online. Em seguida, execute um USE DATABASE no seu novo banco. Agora, está pronto para transferir seus dados do servidor de produção para o servidor online.

## Mova os dados consolidados para o servidor ativo

Você criou um arquivo chamado riskyjobstable.sql, o qual contém instruções MySQL que criam a sua tabela e inserem dados nela. O arquivo riskyjobstable.sql provavelmente se parecerá com o seguinte:



riskyjobstable.sql

```
-- MySQL dump 10.11
-- 
-- Host: localhost      Database: riskyjobs
-- 
-- Server version      5.0.51b

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/
-- 
-- Estrutura da tabela `jobs`
-- 
DROP TABLE IF EXISTS `jobs`;
CREATE TABLE `jobs` (
  `job_id` int(11) NOT NULL auto_increment,
  `title` varchar(200) default NULL,
  `description` blob,
  `city` varchar(30) default NULL,
  `state` char(2) default NULL,
  `zip` char(5) default NULL,
  `co_id` int(11) default NULL,
  PRIMARY KEY  (`job_id`)
) ENGINE=MyISAM AUTO_INCREMENT=14 DEFAULT CHARSET=utf8;

-- 
-- Dumping de dados da tabela `jobs`
-- 

LOCK TABLES `riskyjobs` WRITE;
/*!40000 ALTER TABLE `riskyjobs` DISABLE KEYS */;
INSERT INTO `riskyjobs` VALUES (8,'Custard Walker','We need
people willing to test the theory that you can walk on
custard.\r\n\r\nWe\'re going to fill a swimming pool with
custard, and you\'ll walk on it. \r\n\r\nCustard and other
kinds of starchy fluids are known as non-Newtonian fluids.
They become solid under high pressure (your feet while you
walk) while remaining in their liquid form otherwise.\r\n\r\nA
towel provided, own bathing suit, a must.\r\n\r\nNote: if
you stand on for too long on the custard\'s surface, you will
slowly sink. We are not liable for any custard sinkages;
```

Tudo isto são comentários; você pode ignorá-los.

O mysqldump sempre escreve uma instrução DROP para começar do zero, antes de emitir CREATE e INSERT.

O mysqldump cria uma única instrução INSERT que recria todas as linhas da tabela.

Se já souber que não existe uma tabela chamada "jobs", você pode ignorar este comando.

Eis a instrução CREATE TABLE.

Você pode ignorar esta instrução LOCK e apenas copiar e colar a partir da instrução INSERT.

**Copie o texto inteiro do arquivo .sql e cole-o no seu terminal do MySQL ou na janela de consultas do seu cliente gráfico MySQL (como o phpMyAdmin).**

Isso faz com que as consultas do arquivo sejam realizadas. No caso do exemplo desta página, o arquivo riskyjobstable.sql contém uma instrução CREATE TABLE e uma INSERT. Além delas, o arquivo riskyjobstable.sql instrui o seu servidor MySQL a excluir quaisquer tabelas existentes e a trancar a tabela com LOCK (impedindo que qualquer pessoa a utilize) enquanto você insere os novos dados.

## Conecte-se ao servidor ativo

Você transferiu seus arquivos PHP para o site online; traduziu as estruturas das suas tabelas na forma de instruções CREATE TABLES e os seus dados na forma de uma imensa instrução INSERT, usando o mysqldump, e as executou no seu servidor web online, o que realizou a transferência dos seus dados.

Só há mais um pequeno passo a se executar. O código PHP que você enviou por FTP ao seu web site não está se conectando com o seu servidor MySQL.

É preciso modificar a string de conexão da sua função mysqli\_connect(), para que ela aponte para o seu servidor MySQL online. Em qualquer lugar do seu código que chame a função mysqli\_connect(), você terá de modificá-la.

```
$dbc = mysqli_connect('localhost', 'myusername', 'mypassword', database)
    or die('Error connecting to MySQL server.');

Este será o nome ou endereço
IP do seu site online. Só será
"localhost" se o seu servidor
MySQL estiver na mesma
máquina que as suas páginas PHP.

E estes serão o nome de
usuário e a senha que lhe
permitirão conectar-se
ao seu servidor MySQL
online.

Este será o nome
do banco de dados
que você criou no
seu servidor online.
```

Isso é tudo!

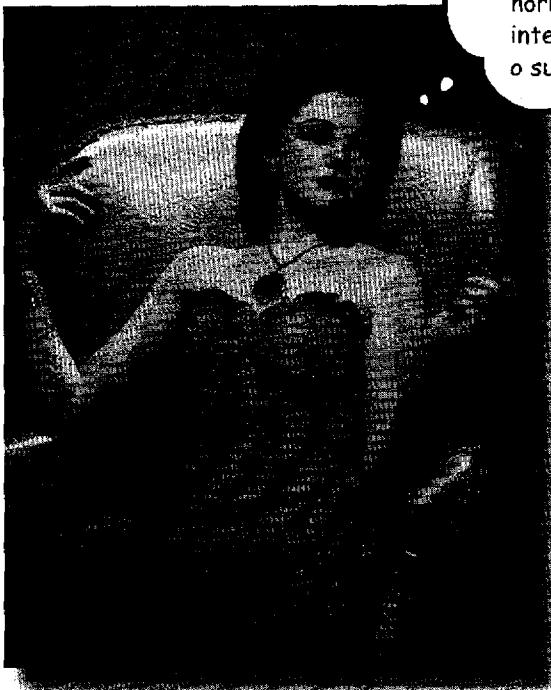
- » Você copiou seus arquivos PHP para o servidor web,
- » gerou suas tabelas e dados em um arquivo .sql,
- » executou as consultas do arquivo .sql no seu servidor MySQL online,
- » e modificou seu código PHP para que ele passe a chamar o seu servidor de banco de dados MySQL online.

Seu site agora deverá estar no ar!





## Obtenha Ainda Mais Proveito



Eu sei que já tenho tudo  
o que uma femme fatale  
normal, incrivelmente bonita e  
inteligente precisa, mas não é  
o suficiente.

**Sim, você pode programar com PHP e MySQL e criar ótimas aplicações web.** Mas você sabe que ainda deve haver mais a se explorar. E há. Este pequeno apêndice irá lhe mostrar como instalar a extensão mysqli e a extensão da biblioteca gráfica GD. Em seguida, mencionaremos mais algumas extensões do PHP que poderá querer baixar. Porque, às vezes, não tem problema em querer sempre mais.

## Estendendo o seu PHP

Este livro ensina a instalação dos módulos mysqli e GD no Windows. Nesta seção, nós lhe mostraremos como verificar quais módulos você tem, como obter o GD ou o mysqli, se não os tiver, e como instalá-los no Windows. Infelizmente, instalar esses módulos em um sistema Mac ou Linux é mais complicado. Falaremos mais sobre isso no final deste apêndice.

**NOTA: Este apêndice cobre o Windows 2000, XP, Vista, Windows Server 2003/2008 ou outros sistemas operacionais Windows de 32 bits.**

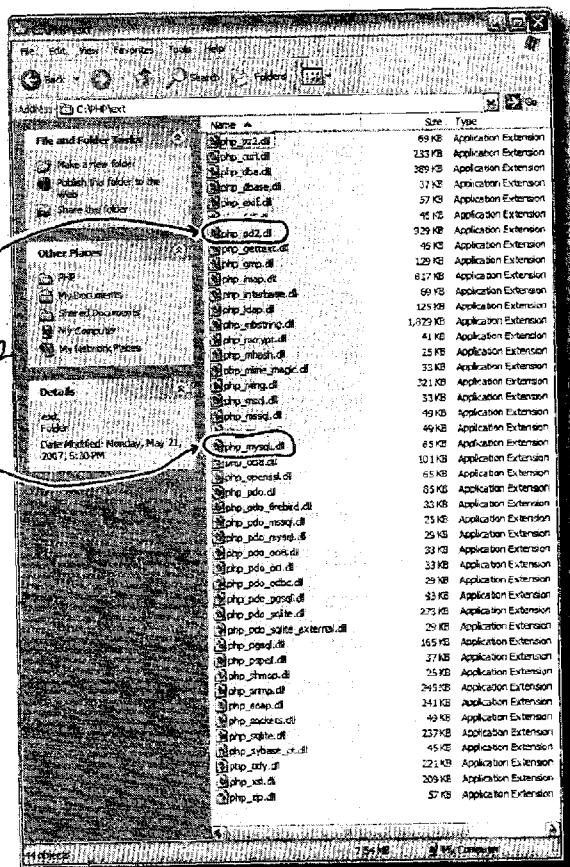
### Se estiver usando o Windows, você está com sorte

Você provavelmente já tem ambos os módulos mysqli e GD no seu computador. E mesmo se não os tiver, é relativamente fácil adicioná-los. Nós lhe mostraremos como verificar o que você tem e, caso algum deles esteja faltando, como os obter e ativar.

Vamos começar verificando o que você tem.

- 1 Primeiramente, descubra se o GD ou mysqli estão no seu sistema. Para isso, comece navegando até o diretório onde as extensões do PHP estão instaladas. Elas geralmente ficam no diretório C:/PHP/ext, embora o caminho possa ser diferente na sua máquina. Abra o diretório ext e procure por php\_gd2.dll e php\_mysql.dll. Em geral, esses módulos são instalados com o PHP 5 e mais tarde, só precisam ser ativados. Se os tiver, ótimo, passe para o item 3. Se não, vá para o item 2.

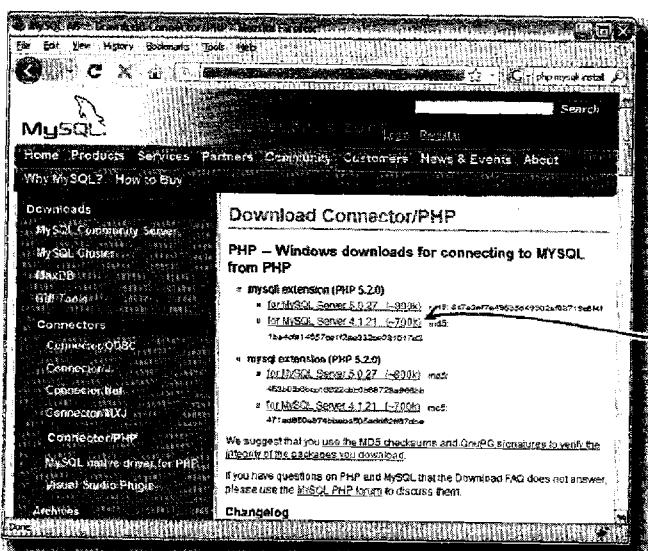
Você deverá ver  
php\_gd2.dll e  
php\_mysql.dll.



2

Caso não tenha qualquer um dos arquivos `php_gd2.dll` e `php_mysqli.dll`, você terá de obtê-lo. É provável que já tenha ambas as DLLs na sua máquina, mas se não for o caso, pode encontrar o `php_gd2.dll` em: <http://www.libgd.org/Downloads>. Baixe o arquivo e copie-o para a pasta ext dentro da sua instalação do PHP. Nos nossos exemplos, ela se localiza em `C:/PHP/ext`.

Você pode obter a extensão `mysqli` em MySQL.com. Primeiramente, acesse <http://www.mysql.com>. Clique em Downloads (no topo da página) -> Connectors (se estiver no menu da esquerda) -> MySQL native driver for PHP -> Download `php_mysqli.dll` for PHP 5.2.1 (Windows) (Certifique-se de que esta é a sua versão).

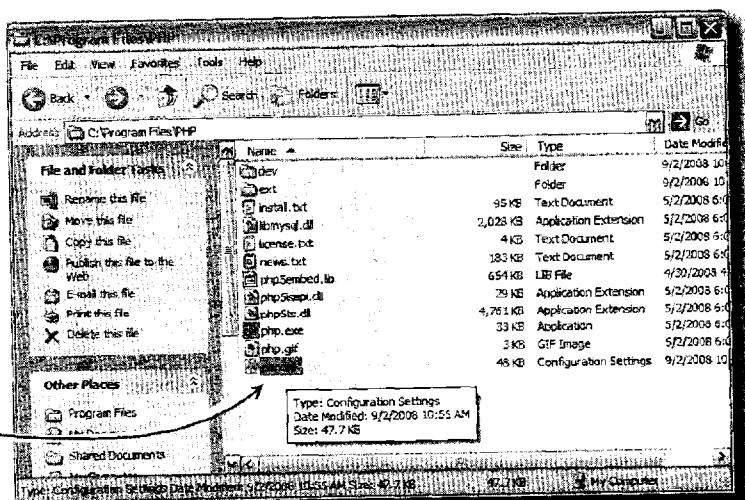


Baixe a versão do  
mysqli certa para a  
sua versão do PHP.

3

A esta altura, você deverá ter `php_mysqli.dll` e `php_gd2.dll` copiados para a sua pasta `PHP/ext`. Nós precisamos instruir o nosso arquivo `php.ini` a usar essas DLLs. Para fazer isso, vá até o diretório onde esse arquivo se encontre abra-o em um editor de texto.

Em alguns casos, a sua instalação do PHP vai parar no diretório Arquivos de Programas\PHP. Encontre o seu arquivo `php.ini` e abra-o para executar o passo seguinte.



## instalando novos módulos no php (continuação)

4

Encontre as seguintes linhas dentro do seu arquivo php.ini:

```
extension=php_gd2.dll  
e  
extension=php_mysqli.  
dll
```

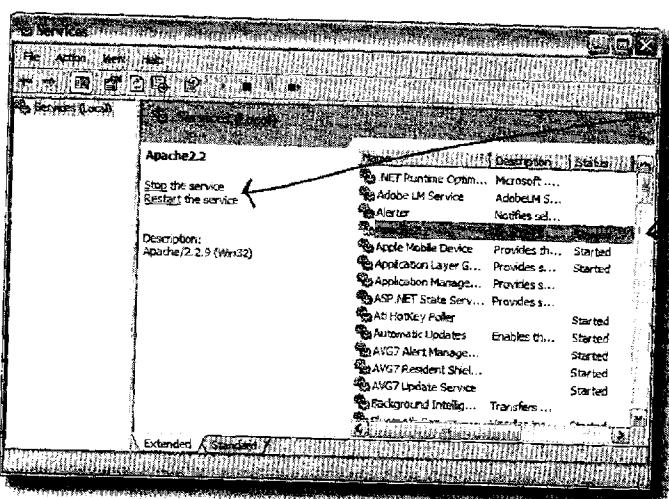
Se qualquer uma delas tiver sinais ; ou # na frente delas, isso significa que estão desativadas por comentários. Remova esses sinais e salve seu arquivo.

Apague os ponto e vírgulas que aparecerem na frente destas duas linhas, se houver. Depois, salve seu arquivo.

```
; Windows Extensions  
; Note that ODBC support is built in, so no dll is needed for  
; Note that many DLL files are located in the extensions/(Pf-  
; extension folders as well as the separate PECL DLL download  
; Be sure to appropriately set the extension_dir directive.  
extension=php_bz2.dll  
extension=php_curl.dll  
extension=php_dba.dll  
extension=php_dbase.dll  
extension=php_exif.dll  
extension=php_fdf.dll  
extension=php_gd2.dll  
extension=php_gettext.dll  
extension=php_gmp.dll  
extension=php_ifx.dll  
extension=php_imap.dll  
extension=php_interbase.dll  
extension=php_ldap.dll  
extension=php_mbstring.dll  
extension=php_mcrypt.dll  
extension=php_mhash.dll  
extension=php_mime_magic.dll  
extension=php_ming.dll  
extension=php_msqli.dll  
extension=php_myssql.dll  
extension=php_mysql.dll  
extension=php_ocl8.dll  
extension=php_openssl.dll  
extension=php_pdo.dll  
extension=php_pdo_firebird.dll  
extension=php_pdo_mssql.dll  
extension=php_pdo_mysql.dll  
extension=php_pdo_oci.dll  
extension=php_pdo_ocl8.dll  
extension=php_pdo_odbc.dll  
extension=php_pdo_pgsql.dll  
extension=php_pdo_sqlite.dll  
extension=php_pgsql.dll  
extension=php_psspell.dll  
extension=php_shmop.dll  
extension=php_snmp.dll  
extension=php_soap.dll  
extension=php_sockets.dll  
extension=php_sqlite.dll  
extension=php_sybase_ct.dll  
extension=php_tidy.dll  
extension=php_xmlrpc.dll  
extension=php_xs1.dll  
extension=php_zip.dll
```

5

O último passo é reiniciar seu servidor web Apache, para que as modificações feitas no seu arquivo php.ini tenham efeito. Para isso, vá até o Painel de Controle do Windows, clique duas vezes em Ferramentas Administrativas, e depois clique em Serviços. Você deverá ver o seguinte:



Selezione o Apache e depois clique no link Reiniciar.

Clique no serviço Apache e depois em Reiniciar, no menu da esquerda. Da próxima vez que você tentar usar as funções GD ou mysqli, elas deverão funcionar corretamente.

## E no Mac...

Infelizmente, é um pouco mais difícil. Para adicionar módulos no Mac é preciso recompilar o código-fonte do PHP, passando a ele argumentos que dizem quais módulos você quer usar. São inúmeras as combinações possíveis de sistemas operacionais Mac e versões do PHP, de modo que não é possível incluí-las neste pequeno apêndice. Existe um ótimo guia que poderá lhe ajudar a instalar o módulo GD:

<http://macoshelp.blogspot.com/2008/02/adding-gd-library-for-mac-os-x-leopard.html>

Só irá funcionar se você tiver a versão certa do OS X (Leopard) e a versão certa do PHP (5). Se não tiver, ou se as instruções não funcionarem para você, é possível consultar os comentários desse site e do site original do GD, <http://www.libgd.org/>, para instruções de instalação mais detalhadas e específicas para a sua própria combinação de OS X e PHP.

Para ajuda na adição do mysqli à sua versão Mac do PHP, que também exige recompilação do PHP, recomendamos as instruções do seguinte site:

<http://dev.mysql.com/downloads/connector/php-mysqlnd/>

Tenha em mente que esta complicação na instalação dos módulos GD e mysqli só se aplica se estiver tentando rodar um servidor web em um computador Mac, por exemplo um servidor de desenvolvimento local. Se estiver apenas usando um Mac para escrever código PHP, que será enviado e testado em algum outro servidor, isso não é problema.



# Índice Remissivo

## Símbolos

- `^` 572
- `_` 26
- `;`
- instruções SQL 111
- MySQL 64
- PHP 25
- `!`
- operador NOT 174
- `. 40`
- `* 70`
- `\ 46`
- `&& 179`
- `% 505`
- `< 168`
- `<> 168`
- `== 167`
- `> 168`
- `>= 168`
- `|| 179`
- `$`
  - cifrão 25
  - `$_COOKIE 376`
  - `$_FILES`
  - erro no upload 269
- `$_GET` 276
- `$i` 264
- `$_POST`
  - array 34
  - fornecer os dados do formulário 91
  - superglobal 33
  - Verifique se o formulário foi submetido 202
- `$result` 135
- `$_SERVER` 342
- `$_SERVER['PHP_SELF']` 200
- `$_SESSION` 391
- `\d` 572
- `/* e */` 335
- `\n` 47
- `<?php>`
  - espaços dentro 305
- `\s` 572
- `\w` 572

## A

- abdução alienígena
- Desconstruindo o script PHP 24
- formulário HTML 5
- receber emails 53
- replicação de vídeos do YouTube 686

## *índice remissivo*

- requisição REST 686
- Test Drive 7
  - Adicione o script YouTube 708
  - cláusula WHERE 97
  - instrução SQL INSERT 69
  - script do Newsfeed 675
- Variáveis 31
- acidentalmente, deletado 149
- ADD COLUMN 232
- addemail.php 126
- Adicionar Pontuação 226
- A função explode() 510
- alias 477
- ALTER TABLE 209
- Anatomia de um Cabeçalho 303
- Aplicação Segura
  - autenticação HTTP 299
    - Autenticando com cabeçalhos 306
    - basic realm 311
    - cabeçalhos 302
    - cabeçalho de localização 309
  - CAPTCHA 611
    - GD (Graphics Draw) 612
    - texto da senha 618
  - Enganando o MySQL com comentários 334
  - injeção de SQL 335
  - INSERT mais seguro (com parâmetros) 337
  - Lado do servidor 57
  - moderação humana 320
    - Passo 1 322
    - Passo 2 324
- Passo 3 326
- Passo 4 327
- PHP 723
- script Autorizar 314
- spam bots 606
- aplicações
  - definidas 424
- Aplicações Web Personalizadas
  - armazenando dados do usuário no servidor (ver sessões) 348
  - comunidade de usuários 347
    - segurança 348
    - cookies (ver cookies) 348
    - logins dos usuários (ver logins dos usuários) (ver também aplicação Mismatch) 348
  - aplicativo Mismatch
    - Armazenando dados do gráfico de barras 632
    - cabeçalho das páginas 421
    - comunidade de usuários 347
    - “desencontrabilidade” 483
      - Compare usuários 487
      - loop FOR 488
      - Prepare-se para a busca 486
    - Faça o Download! 350
    - Formulando um plano para execução dos gráficos de barras 639
    - Fundamentos dos gráficos de barras 644
    - Joins 478
    - logins para os usuários 348
      - Nome do Usuário 348
      - plano 349
      - Preparando o banco de dados para

## *índice remissivo*

- 351  
Senha 348  
menu de navegação 421  
questionário 445  
gerando o formulário 446  
Passo 1 449  
Passo 2 449  
Passo 3 457  
Passo 4 457  
rodapé das páginas 421  
sessões 392  
cadastro de novos usuários 369  
templates 422  
Test Drive 422  
Crie o script My Mismatch 648  
script de login 362  
script do questionário 458  
script mymismatch 494  
Um login baseado em cookies 380  
apóstrofo 47  
armazenamento inicial dos arquivos enviados 245  
arquivo php.ini 252  
arquivos include 254  
array 34  
AS nome 499  
aspas 47  
aspas duplas 192  
aspas simples 47  
asterisco (\*) 580  
atributo action 14  
autenticação HTTP  
Autenticando com cabeçalhos 306  
cabeçalhos 307  
criptografia da senha 360  
Autenticando com cabeçalhos 306  
authorize.php 314  
AUTO\_INCREMENT 209  
aviso 171
- B**
- banco de dados  
integridade referencial 437  
localização 239  
modificações estruturais 471  
Nome do usuário 308  
normalização 462  
benefícios 464  
Três passos 465  
relação de um-para-muitos 439  
relação de um-para-um 439, 441  
Senha 63  
tabela de junção 440  
barra invertida () 46  
biblioteca de gráficos 612
- C**
- cabeçalho de recarregamento 309  
cabeçalho de tipo de conteúdo 309  
cabeçalhos  
Anatomia de um Cabeçalho 303  
Autenticando com cabeçalhos 306  
cabeçalho de recarregamento 309

## *índice remissivo*

caixa baixa 118  
caixa de ferramentas PHP & MySQL  
\$\_COOKIE 414  
\$\_FILES 293  
\$\_POST 57  
\$\_SERVER 342  
\$\_SESSION 414  
ALTER TABLE 221  
array 57  
AS nome 499  
Autenticação HTTP 343  
Biblioteca GD 654  
CAPTCHA 654  
Caracter de escape 57  
checkdnsrr() 604  
Classe de caracteres 604  
cláusula else 184  
comando DELETE 147  
comando DESCRIBE 122  
comando DROP TABLE 124  
declaração if 221  
echo 57  
esquema 499  
Expressão regular 604  
função empty() 221  
função exit() 342  
função explode() 560  
função header() 618  
Função personalizada 560  
função simplexml\_load\_file() 711  
imagecreatetruecolor() 654  
imagedestroy() 654  
imageline() 654  
imagepng() 654  
imagerectangle() 654  
imagestring() 654  
imagestringup() 654  
imagettftext() 654  
implode() 560  
Injeção de SQL 343  
INNER JOIN 499  
instrução DELETE FROM 291  
is\_numeric() 342  
isset() 221  
lado do cliente 165  
Lado do cliente 57  
Lado do servidor 57  
LIKE 560  
LIMIT 293  
loop FOR 488  
loop while 157  
mail() 57  
metacaracteres 572  
Moderação Humana 343  
MySQL 57  
mysqli\_fetch\_array() 157  
mysqli\_real\_escape\_string() 342  
Normalização 499  
operadores lógicos 221  
operador NOT 221  
operador ternário 499  
ORDER BY 293  
pasta images 247  
PHP 57  
preg\_match() 604  
preg\_replace() 604

## **índice remissivo**

require 293  
require\_once 293  
REST 711  
RSS 711  
Script PHP 57  
session\_destroy() 414  
session\_start() 414  
setcookie() 414  
SimpleXMLElement 711  
SQL 57  
str\_replace() 560  
substr() 560  
switch-case 560  
trim() 342  
Validação de Formulário 343  
Variável 57  
WHERE 157  
XML 711  
caixas de verificação 212  
CAPTCHA  
  funções 616  
  Guitar Wars 625  
  Test Drive 613  
  texto da senha 613  
Caracteres de escape 46  
caracteres newline (\n) 47  
caracteres reservados 580  
chaves 436  
chaves primárias  
  cinco regras 210  
  normalização 465  
cláusula else 184  
cláusula WHERE 96  
código duplicado 194  
coleção de objetos 697  
coluna  
  alias 636  
  colunas que não são chaves 465  
comando CREATE DATABASE 111  
comando DELETE  
  apagamentos acidentais 149  
  cláusula WHERE 149  
comando DROP TABLE 124  
comando echo 24  
comando SQL 111  
comando SQL DESCRIBE 123  
comentários  
  Enganando o MySQL com 334  
  multilinhas 335  
condição de teste 166  
conectando-se ao MySQL  
  comando CREATE TABLE 65  
  comando USE 120  
  CREATE DATABASE 64  
  funções PHP 78  
  die() 83  
  mysqli\_close() 78  
  mysqli\_connect() 78  
  mysqli\_query() 78  
  mysqli\_select\_db() 82  
  instrução SELECT 211  
  asterisco (\*) 580  
  cláusula WHERE 504  
  nome do banco de dados 76

## *índice remissivo*

- nome do usuário 39
- query 78
  - executar 86
- requerimentos 311
- script PHP 76
- senha 63
- servidor de banco de dados 76
- tabela 75
- Configure um Ambiente de Desenvolvimento
  - MySQL 731
  - PHP 731
    - instalação 736
- constante GW\_UPLOADPATH 253
- Controle Seus Dados
  - alias 477
  - junções 477
  - normalização 495
- cookies 375
  - apagar 385
  - Definir 375
  - sessões sem 403
  - tamanho dos dados 410
  - tempo de vida 390
- Coppermine Gallery 730
- crie e preencha um banco de dados
  - colunas 71
  - comando CREATE DATABASE 111
  - comando CREATE TABLE 65
  - comando DELETE 147
    - apagamentos acidentais 149
    - cláusula WHERE 149
  - comando DROP TABLE 124
- comando SQL DESCRIBE 123
- comando USE 120
- consulta 147
- Crie um banco de dados e uma tabela 64
- função mail() 144
- instrução ALTER 125
- linha 149
- nomes das colunas 66
- ponto e vírgula (;) 25
- primeiros passos 110
- prompt-> 119
- script addemail.php 127, 133
  - array \$\_POST 134
  - função mysqli\_fetch\_array() 136
  - função mysqli\_query() 79
  - variável \$result 135
- SELECT \* FROM 135
- Tabelas 79
  - definidas 411
  - estrutura 56
  - tipo de dados 113
- Cruzadas PHP & MySQL 155

## D

- Dados Armazenados em Arquivos
  - arquivos externos 240
  - constante GW\_UPLOADPATH 251
  - pasta images 252
    - programa de FTP 248
  - pasta temporária 246
  - Planejando os uploads de arquivos gráficos 231

## *índice remissivo*

- Passo 1 234
  - Passo 2 237
  - Passo 3 238
  - Passo 4 242
  - Passo 5 250
  - Passo 6 257
  - validação 266
    - mensagem de erro 268
  - Dados atômicos
    - normalização 465
  - declaração switch 542
  - Desenhando Gráficos Dinâmicos 605
  - design da aplicação 410
  - display\_errors 724
  - Drupal 730
- 
- ## E
- emails
    - código PHP 43, 46
      - <br /> 46
      - caracteres newline (\n) 47
    - Crie 48
    - função mail() 51
    - obter 54
    - perder 54
    - vazios 171
  - endereço IP 63
  - endereços de email
    - Sufixos de domínios 598
  - Erros PHP com tratamento de exceções 719
  - escapando caracteres reservados 580
- espaços dentro das tags <?php ?> 305
  - expose\_php 724
  - Expressões Regulares
    - classes de caracteres 578
    - definindo 385
    - endereços de email válidos 595
    - Sufixos de domínios 598
    - função checkdnsrr() 599
    - função preg\_match() 584
    - função preg\_replace() 588
    - metacaracteres 572
    - quantificador 577
  - extensão .php 25
- 
- ## F
- Faça o Download!
    - aplicação Mismatch 359
    - aplicação Risky Jobs 506
    - Guitar Wars 357
  - feed RSS
    - gerar dinamicamente 672
    - imagens 670
    - link para 676
  - formulário
    - \$\_POST 32
    - \$\_SERVER['PHP\_SELF'] 200
    - atributo action 14
    - atributo type 6
    - botão Submit 202
    - data-drive 450
    - função isset() 202
    - mailto 6

## **Índice remissivo**

- método GET 278
- método POST 202
- script PHP 11
  - consultas SQL 82
- tags <br /> 41
- tags <form> 6
- tags input 6
- validação 339
- formulário que referencia a si mesmo 199
- formulários permanentes 199
- função array\_push() 449
- função checkdnsrr() 599
- função die() 83
- função empty() 172
- função header() 305
- função now() 238
- função personalizada 537
- função PHP exit() 307
- função preg\_match() 584
- função rand() 613
- função setcookie()
  - Fazer o logout de um usuário 385
- funções personalizadas
  - Construa uma consulta com 537
  - paginação 548
    - cláusula LIMIT 551
    - variáveis 552
- Funções Personalizadas Expostas 538
- funções PHP 78
- G**
- GD (Graphics Draw)
- CAPTCHA 614
- funções 616
- imagecolorallocate() 616
- imagecreatetruecolor() 616
- imagedestroy() 619
- imageellipse() 618
- imagefilledellipse() 618
- imagefilledrectangle() 616, 617
- imageline() 617
- imagepng() 618
- imagesetpixel() 617
- imagestring() 619
- imagestringup() 619
- imagettfttext() 620
- GET e POST 279
- gráfico de barras
  - Fundamentos 644
- Gráfico de Barras Exposto 635
- gráficos dinâmicos 276
- Gráficos Dinâmicos 605
- Guitar Wars
  - Baixe isto! 228
  - CAPTCHA 625
  - Homenageando o Guitar Warrior 261
  - página Admin 272
    - Protegendo 300
  - pasta images 247
  - Planejando os uploads de arquivos gráficos 231
    - passo 1 234
    - passo 2 237
    - passo 3 238
    - passo 4 242

## *índice remissivo*

passo 5 250  
passo 6 257  
pontuações não-verificadas 265  
segurança para a sua aplicação 297  
autenticação HTTP 299  
moderação humana 299  
Proteja seus dados contra injeções de SQL 336  
Test Drive 336  
arquivos include 257  
Crie o script Aprovar 328  
pasta images 247  
removescore.php e admin.php 290  
script Autorizar 314  
validação dos arquivos gráficos 270  
ver a maior pontuação 265

## H

### HTML

cliente 10  
código PHP 27  
linguagem sem vida 2  
Páginas HTML dinâmicas 3

## I

### ícones RSS

banco de dados para o news reader 666  
imagecolorallocate() 616  
imagecreatetruecolor() 616  
imagedestroy() 619  
imageellipse() 618

imagefilledellipse() 618  
imagefilledrectangle() 617  
imageline() 617  
imagens  
feed RSS 670  
imagepng() 618  
imagerectangle() 617  
imagesetpixel() 617  
imagestring() 619  
imagestringup() 619  
imagefttext() 620  
Implode() 513  
injeção de SQL 335  
INSERT INTO  
Test Drive 69  
instrução ALTER 125  
instrução if  
aninhadas 178  
cláusula else 184  
condição de teste 168  
operador ternário 455  
instrução ORDER BY 258  
instrução require\_once 255  
instrução SELECT  
\* 70  
cláusula WHERE 96  
FROM 70  
instrução SQL DELETE FROM 283  
instruções include\_once 255  
integridade referencial 437  
is\_numeric() 342  
isset() 172

*índice remissivo*

# J

join  
    notação com pontos 474

# L

lado do cliente 165  
Lado do servidor 57  
LIMIT  
    consulta DELETE FROM 285  
linguagem markup 661  
linhas 109  
log\_errors 724  
logins 349  
logins dos usuários (ver logins dos usuários)  
    349  
loja Elvis  
    PLANEJE ANTES 108  
    projeto da aplicação 106  
    script addemail 133  
        feedback 183  
        validação 164  
Test Drive 201  
    coluna de chave 211  
    comando DELETE 150  
    comando USE 120  
    Crie o banco de dados e a tabela 118  
    Envie um email para a lista usando o  
    formulário “Enviar Email” 145  
    operadores lógicos 181  
    script addemail.php 129  
    validação 164

# M

mail() 50  
mailto 6  
Maior ou igual a (>=) 168  
maior que (>) 168  
manipulando strings  
    botão “submeter” 6  
    caracter coringa 505  
    classificação à busca dos resultados 532  
    cláusulas LIKE 505  
    cláusula WHERE 535  
    concatenar strings e variáveis 41  
    consulta precisa de termos de busca  
    legítimos 524  
    função substr() 528  
    préprocessando a string 518  
    Substitua caracteres de busca  
    indesejados 520  
    substring 528  
MD5() 355  
Menor ou igual a (<=) 168  
menor que (<) 168  
mensagem de erro 268  
metacaracteres  
    quantificador 577  
metacaracter ponto final 572  
método children() 699  
método GET 278  
modelos 422  
moderação humana  
instrução ALTER TABLE 322  
passo 1 322

## *índice remissivo*

- passo 2 324
- passo 3 326
- passo 4 327
- MODIFY COLUMN 232
- move\_uploaded\_file 245
- MySQL
  - Permissões do usuário 716
  - Relato de erros 718
  - mysqli\_close() 78
  - mysqli\_connect() 78
  - mysqli\_fetch\_array()
    - loop while 140
  - mysqli\_query() 78
  - mysqli\_real\_escape\_string() 336
- N
  - não existem perguntas idiotas
  - \$\_POST 92
  - arquivos de scripts compartilhados 255
  - arrays e os objetos 700
  - aspas simples 47
  - ataque de injeção SQL 335
  - autenticação HTTP 301
  - CAPTCHA 612
  - Caracteres de escape 46
  - classes de caracteres 579
  - cláusula else 184
  - comando SELECT 147
  - comentários em SQL 335
  - condição de teste 170
  - cookie 375
  - apagados 387
- dados atômicos 463
- empty() 566
- endereços de email válidos 595
- formatação HTML em emails que você envia a partir de um script PHP 45
- formulario foi submetido 202
- função array\_push() 449
- função die() 88
- função exit() 311
- função isset() 173
- função mysqli\_query() 88
- função SHA() 355
- GETs e POSTs 278
- gráfico de barras 633
- GW\_MAXFILESIZE 270
- imagens geradas dinamicamente 605
- instrução ALTER 125
- instrução DESCRIBE 123
- instrução INSERT 85
- JOIN 476
- leitor de RSS 662
- loop 489
- namespaces 695
- normalização 467
- objeto SimpleXMLElement 700
- operador -> 700
- operador ternário 459
- O que significa PHP? 15
- paginação 558
- pasta temporária 246
- phpMyAdmin 68
- ponto-e-vírgula 125
- pontuações sem verificação 261

## *índice remissivo*

- prompt -> 119  
require\_once 255  
REST 682  
RSS 662  
session\_start() 397  
sessões 397  
string de busca 518  
substr() 530  
sufixo de domínio válido 597  
tabela mismatch\_category 468  
tag <? 27  
template 423  
terminal MySQL 68  
tipo de dados ENUM 325  
user\_id 351  
validação 165  
VARCHAR 115  
variáveis 27  
XML 670  
navegador 42  
newsfeed 660  
nome do host 63  
Nome do meu banco de dados 76  
nomes das colunas 66  
nomes de variáveis 25  
non-equijoins 480  
normalização  
    benefícios 397  
    chave primária 465  
    dados atômicos 462  
    Três passos 465  
notação com pontos 474  
Nota Geek  
    CAPTCHA 620  
    SUBSTRING() 530  
NOT NULL 209
- O**
- objeto  
    Acesse os dados XML com 696  
    coleção de 697  
    objeto SimpleXMLElement 700  
    open\_basedir 724  
operador lógico  
    ordem 181  
operador ternário 455
- P**
- página Admin  
    links para remoção de pontuações 275  
    Protegendo 300  
paginação  
    cláusula LIMIT 549  
    links de navegação da página 554  
    variáveis 552  
Páginas HTML dinâmicas 3  
pasta images 247  
pasta temporária 246  
Persistência  
    Sessões + Cookies 409  
    temporária 452  
    usuário 383
- PHP

## *índice remissivo*

- Erros PHP com tratamento de exceções 719  
Estendendo 750  
Instalação 736  
navegador 42  
O que significa PHP? 15  
orientado a objetos 721  
regras 25  
servidor 11  
PHP4 714  
phpBB 730  
phpMyAdmin 62  
PHP orientado a objetos 721  
ponto-e-vírgula
  - instruções SQL 111
  - MySQL 64
  - PHP 25ponto final (.) 580  
Pontos Importantes
  - \$\_FILES 239
  - \$\_SERVER 315
  - autenticação HTTP 342
  - cabeçalhos 309
  - CAPTCHA 611
  - caracteres 580
  - comando CREATE DATABASE 111
  - comando CREATE TABLE 65
  - comando DROP TABLE 124
  - conexões a bancos de dados 88
  - cookies 376
  - função checkdnsrr() 598
  - função die() 88
  - função exit() 315funcão header() 305  
funcão mysqli\_connect() 78  
GD (Graphics Draw) 612  
instrução ALTER 252  
loop while 139  
metacaracteres 572  
mysqli\_connect() 78  
mysqli\_query() 79  
pasta images 247  
preg-match() 584  
preg\_replace() 593  
Quantificadores 602  
sessões 317  
tag <input> 453  
variáveis da sessão 390  
post 6  
Precedência 727  
Precedência de operadores 727  
preg\_replace() 588  
PRIMARY KEY 209  
prompt -> 119  
pseudocódigo 641  
**Q**  
quantificador 577  
query
  - executar 86
  - múltiplas tabelas 436
  - query SQL 86
  - termos de busca legítimos 524query SQL 86

## *índice remissivo*

# R

referências a phpinfo() 723  
regex 570  
relação de muitos-para-muitos 440  
Relato de erros para o MySQL 718  
Removendo dados  
    apagamentos acidentais 149  
replicação de vídeos do YouTube  
    função simplexml\_load\_file() 688  
    requisição REST 682  
    XML 690  
replicações e serviços web  
    RSS 661  
require\_once 293  
requisição GET 276  
requisição REST  
    construindo 686  
requisições POST 277  
Risky Jobs  
    build\_query() 537  
    classificar 545  
    links de navegação da página 554  
    paginação 548  
    Revise a consulta para ter resultados 553  
    variáveis da paginação 552  
Faça o Download! 559  
o script Search completo 557  
Test Drive 559  
    Ajuste os números de telefone no script de registro 594  
    formulário de busca 515

    função generate\_sort\_links() 546  
    funções explode() e implode() 526  
    validação de emails 603  
rótulos CASE 542  
RSS 711  
RSS Revelado 671

# S

safe\_mode 724  
schema  
    setas 436  
    direção 438  
script Autorizar 314  
scripts  
    arquivos include 254  
    comunicarem-se uns com os outros 276  
    engenharia reversa 316  
    instrução require\_once 255  
scripts cross-site 725  
script sendemail.php  
    array \$\_POST 134  
    feedback 183  
    função mail() 134  
    lógica por trás 171  
    mysqli\_fetch\_array() 135  
    mysqli\_query() 135  
    validação 164  
    variável \$result 135  
Scripts PHP  
    atributo action 14  
    servidor 12  
segurança visual 353

## *índice remissivo*

- Senha  
autenticação HTTP 357  
criptografia da senha 360
- Comparando 355
- criptografadas 352  
função SHA() 354
- segurança visual 353
- servidor  
identificando 682  
instalação do Apache no Windows 735
- PHP 11
- scripts PHP 12
- rodando 18
- servidor de bancos de dados 61
- session\_destroy() 390
- session\_start() 390
- sessões 395  
cookies 406  
sem cookies 403  
tamanho dos dados 410  
tempo de vida 406
- setas  
direção 438
- SHA()  
Comparando senhas 355
- sinais de igual (==) 167
- sites de comunidades 372
- spam bots  
CAPTCHA 611
- SQL 57
- strings e funções  
função explode() 510
- função implode() 513  
função str\_replace() 520  
função substr() 528
- substr() 528
- Sufixos de domínios 598
- superglobal 33
- superglobal SID 403
- supressão de erros 288
- T**
- tabela de junção 440
- tabela principal 439
- tabelas  
alias 477  
chaves primárias 209  
child 438  
comando CREATE TABLE 65  
definir 113  
estrutura 123  
junção 440  
parent 438
- tags <? 27
- tags <br /> 41
- telas capturadas  
armazenamento inicial dos arquivos enviados 245  
pasta temporária 244
- templates 422
- terminal MySQL  
ponto-e-vírgula 111
- prompt > 119

## **índice remissivo**

Teste múltiplas condições 179  
tipo de dados  
  eficiente 453  
  tipo de dados ENUM 325  
tipos MIME diferentes para imagens JPEG  
  270  
tipo TINYINT 322  
trim() 336

## **U**

“underscore” (\_) 26  
underscore (\_) 505  
unlink() 269  
UPDATE SQL 235  
USE 64  
user\_id 351  
Username 348  
USING 476  
usuários façam logout  
  sessões 393

## **V**

validação  
  código duplicado 194  
  declaração if 339  
  cláusula else 184  
  é 194  
  nes PHP para verificar 172  
  atrás 165  
  é erro 189

769

operador lógico 179  
sendemail.php 175  
tamanho para imagens 266  
Teste múltiplas condições 179  
VALUES 66  
VARCHAR 114  
variáveis  
  \$\_ 264  
  armazenam as partes do email 49  
  concatenar strings e variáveis 41  
  sessões 389  
  superglobal 33  
  variáveis de sessão 389

## **W**

WordPress 730

## **X**

XML  
  Acesse os dados 696  
  coleção (collection) de objetos 697  
  entidades 693  
  hierarquia de elementos 695  
  namespaces 693  
  O YouTube fala 690  
  Penetre nos dados 698  
  RSS 661

## **Y**

YouTube fala XML 690