Desenvolvimento Web

Cap. 8 - Introdução ao PHP

Universidade Estácio de Sá
Análise de Sistemas
Desenvolvimento Web em HTML5, CSS3, Javascript e PHP
Professor: Marlan Külberg
marlan.kulberg@estacio.br

Introdução

- Apesar da evolução do HTML e navegadores, a Web ainda é um ambiente bastante restrito

- Browser não executa qualquer tipo de código
- Códigos simples em outras linguagens podem ser complexos em HTML ou JavaScript

- Qualquer projeto Web sério não usa apenas HTML, CSS e JavaScript
- Também envolve uma infraestrutura no servidor

- Há muitas linguagens e servidores possíveis de serem usados

- Usa-se uma linguagem no servidor para executar tarefas como
    - Gerar páginas dinamicamente com dados de um banco
    - Enviar emails para usuários
    - Processar tarefas complexas
    - Garantir validações de segurança da aplicação
    - Outros

Funcionamento de um Servidor HTTP

- Basta abrir um arquivo .html no navegador para visualizar a página. Processo de serviço é mais complexo

- Protocolo HTTP é utilizado para servir páginas na Web
- Quando um endereço HTTP é acessado, realiza-se uma requisição ao servidor

- Exemplo: o acesso a https://estacio.br/candidato conecta o servidor www.estacio.br e requisita a URL /candidato via HTTP

- Existe um servidor HTTP esperando novas requisições responsável por servir as solicitações do usuário

- Servidor é um programa instalado responsável por processar as requisições

- Servidor não precisa simplesmente ler arquivo HTML e envia o conteúdo
- Também pode executar código e gerar HTML dinamicamente

Funcionamento do PHP no Servidor

- Ao usar PHP, pode-se escolher diversos servidores compatíveis. O mais famoso de todos é o Apache
- Versões mais recentes do PHP já vem com um servidor embutido que dispensa a instalação de um servidor adicional
- Ideal para testes e desenvolvimento.

- Abrir o terminal, entrar na pasta onde está o projeto e usar o comando

```
php -S localhost:8000
```

- Opção -S indica o uso do servidor embutido
- localhost indica o acesso ao servidor no endereço local
- 8000 é a porta que o servidor vai rodar

- Navegando na URL http://localhost:8000/ fará o servidor responder

- Como o projeto só tem arquivos .html, as páginas no navegador serão exibidas sem novidades

- Para executar código do lado do servidor com PHP, basta renomear o arquivo de .html para .php.

- Com essa extensão, é possível usar código dinâmico PHP no HTML

Funcionamento do PHP no Servidor

- Todo código PHP fica dentro de uma tag especial, para diferenciar do código HTML:

```
<?php
    // código PHP
?>
```

- Pode ser misturada com HTML.
- Tags do PHP serão executadas pelo servidor PHP.

```
<h1> Título HTML aqui</h1>
<?php
    // código PHP
?>
<p> Mais HTML</p>
```

- Exemplo: exibir o ano atual no rodapé da página sobre. Para acessar o ano atual:

```
<?php
    print date('Y');
?>
```

- Chama-se a função date do PHP passando como argumento o formato de saída
- Y indica o ano apenas. Outros formatos podem ser vistos em: http://www.php.net/date

- A função date devolve a data mas não mostra na tela. O comando print pega o valor e mostra na tela

Atividade - Executando o PHP

1. O primeiro passo é executar o servidor local do PHP. Para isso, abra o terminal e entre na pasta do projeto:

cd <caminho_da_pasta>/mirror-fashion/

Em seguida, execute o servidor PHP:

php -S 0.0.0.0:8000

2. Abra o navegador e acesse http://localhost:8000
Deve-se ver a página da Mirror Fashion sendo servida pelo PHP agora.

Para testarmos a instalação do PHP, vamos implementar uma funcionalidade. Queremos saber o ano atual dinamicamente e inserir esse valor na página.
Com PHP, basta usar a função date.

3. Primeiro renomeie a extensão do arquivo sobre.html para sobre.php. Lembre que um arquivo PHP nada mais é que um HTML com instruções especiais pra rodar no servidor.

4. O texto que descreve a Mirror Fashion fala de sua fundação em 1932. Acrescente uma frase dinâmica no texto que indica há quantos anos a empresa foi fundada.

```
Fundada há <?php print date("Y"); ?> anos,…
```

5. Repare que no lugar de sair a quantidade de anos, estará exibindo o ano atual. Para resolver isso, vamos subtrair o ano atual menos a data de fundação da Mirror Fashion:

```
Fundada há <?php print date("Y") - 1932; ?> anos,…
```

Reaproveitamento de Código

- Recurso muito comum é a criação de cabeçalho e rodapé únicos que são aproveitados em todas as páginas
- Com HTML puro, a única solução copiar e colar o código em todas as páginas
- O HTML não tem recursos pra que deixar esse código centralizado em um só lugar
- Usando uma tecnologia no servidor como PHP, pode-se fazer o "copiar e colar” dinamicamente.
- Cria um arquivo .php separado que encapsula o código do cabeçalho.
- Depois esse código é incluído em todas as páginas usando o comando include

```
<?php
    include("cabecalho.php");
?>
```

- HTML final que o browser recebe é o mesmo de antes. A inclusão do cabeçalho acontece no servidor

Atividades - Comando Include

1. O primeiro passo é criar a estrutura básica do arquivo. Isto inclui o doctype, tag html, head, body, title. Crie o arquivo produto.php com uma base parecida com essa: (copie e cole da index.html)

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width">
        <title>Mirror Fashion</title>
        <link rel="stylesheet" href="css/reset.css">
        <link rel="stylesheet" href="css/estilos.css">
    </head>
    <body>
    </body>
</html>
```

Como todas as páginas fazem parte da Mirror Fashion, é muito comum que tenham o mesmo cabeçalho.
O HTML sozinho não tem recursos interessantes para reaproveitar pedaços de código entre páginas. Mas no servidor PHP, basta usar a função include .

2. Crie um arquivo chamado cabecalho.php e coloque o conteúdo do cabeçalho existente na home com a tag <header>
.

```
<header class="container">
    <h1>
        <img src="img/logo.png" alt="Mirror Fashion">
    </h1>
    <p class="sacola">
        Nenhum item na sacola de compras
    </p>
    <nav class="menu-opcoes">
        <ul>
            <li><a href="#">Sua Conta</a></li>
            <li><a href="#">Lista de Desejos</a></li>
            <li><a href="#">Cartão Fidelidade</a></li>
            <li><a href="sobre.php">Sobre</a></li>
            <li><a href="#">Ajuda</a></li>
        </ul>
    </nav>
</header>
```

Atividades - Comando Include

3. Na página produto.php, inclua cabecalho.php logo no início do body:

```
<body>
    <?php include("cabecalho.php"); ?>
</body>
```

4. Teste a nova página acessando http://localhost:8000/produto.php. O cabeçalho deve aparecer incluído. Verifique o HTML da página pelo navegador.

5. Crie o arquivo rodape.php e copie o conteúdo do <footer> criado anteriormente na Home:

```
<footer>
    <div class="container">
        <img src="img/logo-rodape.png" alt="Logo Mirror Fashion">
        <ul class="social">
            <li><a href="http://facebook.com/mirrorfashion">Facebook</a></li>
            <li><a href="http://twitter.com/mirrorfashion">Twitter</a></li>
            <li><a href="http://plus.google.com/mirrorfashion">Google+</a></li>
        </ul>
    </div>
</footer>
```

6. Na página produto.php, inclua rodape.php logo antes de fechar o body usando o include do PHP:

```
<body>
    <?php include("cabecalho.php"); ?>
    <!-- Aqui vai vir o miolo da página depois! -->
    <?php include("rodape.php"); ?>
</body>
```

8. Aplique o cabeçalho e o rodapé criados também na Home e na página Sobre. Para isso, transforme esses arquivos em PHP renomeando suas extensões e use o include.

Flexibilidade com Variáveis

- É possível passar variáveis de um arquivo para o outro durante o include
- Exemplo:
- A página do cabeçalho pode receber um título para imprimir no <title> ao invés de deixar um valor fixo

```
<title>
    <?php print $cabecalho_title; ?>
</title>
```

- Na página produto.php, define-se a variável antes de dar o include:

```
<?php
    $cabecalho_title = "Produto da Mirror Fashion";
    include("cabecalho.php");
?>
```

Atividade - Variáveis em PHP

1. Edite o arquivo cabecalho.php para incluir toda a estrutura inicial do arquivo HTML, não só o topo da página. Coloque desde o doctype, abertura da tag html, head, body até o header em si.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Mirror Fashion</title>
        <link rel="stylesheet" href="css/reset.css">
        <link rel="stylesheet" href="css/estilos.css">
        <link rel="stylesheet" href="css/mobile.css" media="(max-width: 939px)">
        <meta name="viewport" content="width=device-width">
    </head>
    <body>
        <header class="container">
        <!-- conteúdo do header aqui -->
        </header>
```

Agora remova o pedaço do doctype, html, head e body de dentro das páginas que fazem include. A ideia é que todo esse pedaço agora é reaproveitável no include e não apenas o header em si.

2. Algumas partes do HTML incluídas no cabecalho.php devem ser dinâmicas.
O <title> não deveria ficar fixo no include, deve ser diferente para cada página.
Uma forma de resolver isso é passando variáveis entre as páginas.
Na produto.php, antes de incluir o cabeçalho, defina uma variável com o título:

```
<?php
    $cabecalho_title = "Produto da Mirror Fashion";
    include("cabecalho.php");
?>
```

E no cabecalho.php, imprimir essa variável dentro do <title> :

```
<title><?php print $cabecalho_title; ?></title>
```

Repare como agora o título é parametrizável. Defina um título nas outras páginas que fazem uso do include também (Home, Sobre etc).

3. Que outros elementos desse cabeçalho podem mudar entre páginas diferentes além do título?
Implemente soluções parecidas usando variáveis pra resolver outros casos de include dinâmico.

Exemplo: cada página pode incluir um arquivo CSS próprio, com seu estilo. Não precisa querer listar todos os arquivos CSS no cabecalho.php. O melhor é cada página declarar qual arquivo .css adicional quer incluir além dos básicos.
Na produto.php, podemos querer incluir um produto.css. Vamos usar uma variável pra isso:

```
<?php $cabecalho_css = '<link rel="stylesheet" href="css/produto.css">'; ?>
```

E no cabecalho.php pode-se incluir essa variável no meio do head:

```
<?php print $cabecalho_css; ?>
```

4. Nem todas as páginas precisam de um arquivo .css extra. Do jeito que foi feito, se a variável $cabecalho_css não for definida, um erro acontecerá.

Uma forma de evitar é tentar imprimir a variável apenas se ela existir, e ignorar isso caso ela não esteja definida. Ou seja, a variável é opcional. Pode-se fazer isso no PHP indicando que o erro de variável não encontrada pode ser ignorado. Para isso, usa-se o @ na frente da variável:

```
<?php print @$cabecalho_css; ?>
```

Submissão de Formulário

- Todo formulário criado no HTML tem seus dados enviados para o servidor quando submetido

- Cada campo do formulário é enviado como parâmetro na requisição feita ao servidor.

- No formulário, indica-se qual página (URL) vai receber os dados preenchidos.
- Deve-se especificar o atributo action

- Exemplo: existe um formulário na página produto.php e será criada uma próxima página, checkout.php, que vai receber o produto escolhido e deixar o usuário proceder com a compra. No formulário de produto faz-se:

```
<form action=“checkout.php">
```

- Há ainda outro atributo do form que indica a maneira como os dados são enviados
- Atributo method que pode receber dois valores: GET ou POST

- Ambos os métodos enviam os dados do formulário ao servidor
- GET faz via parâmetros na URL
- POST envia os dados no corpo da requisição (não é visível na barra de endereços)

```
<form action="checkout.php" method="POST">
```

Parâmetros com PHP

- Os dados enviados no formulário são recebidos no PHP
- É possível acessá-los através de variáveis do próprio PHP:
  - $_GET
  - $_POST

- Para acessar o valor um campo preenchido, é necessário saber seu nome

- No HTML do formulário, sempre que um componente é criado, recebe um nome

```
<input type="text" name="mensagem">
```

- No PHP, pode-se acessar cada parâmetro individualmente, usando sintaxe de arrays:

```
<?= $_GET['mensagem'] ?>
```

ou

```
<?= $_POST['mensagem'] ?>
```

Imprimindo variável com PHP
Quando se quer imprimir algo na tela usa-se a construção:

```
<?php print $dado; ?>
```

PHP também permite uma sintaxe mais curta para declarar bloco de código quando a única ação é imprimir algo e não há várias linhas de código.

```
<?= $dado ?>
```

Listas de Definição em HTML

- Listas em HTML, costumam ser representadas pelas tags <ol> e <ul>

- Uma é ordenada e a outra não

- Existe uma lista que semanticamente serve para definir itens
- Lista de termos e suas respectivas definições.

  - Exemplo: lista das siglas de disciplinas do curso de ADS e respectivos nomes

```
<dl>
   <dt>ARA0017</dt>
   <dd>Introdução a Programação Estruturada em C</dd>

   <dt>ARA0062</dt>
   <dd>Desenvolvimento Web em HTML5, CSS3, Javascript e PHP/dd>
</dl>
```

- A lista é a DL, e cada termo é representado por um DT seguindo por sua definição em um DD

Atividade - Checkout da Compra

1. Crie o arquivo checkout.php com uma estrutura básica.

```
<!doctype html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Checkout Mirror Fashion</title>
        <meta name="viewport" content="width=device-width">
    </head>
    <body>
        <h1>Ótima escolha!</h1>
        <p>Obrigado por comprar na Mirror Fashion!
        Preencha seus dados para efetivar a compra.</p>
    </body>
</html>
```

Não será usado o menu da Mirror Fashion nessa página, para criar uma experiência de checkout mais imersiva

2. O formulário na página de produto precisa enviar os dados escolhidos para a página de checkout.
Para isso, o formulário deve indicar para onde será submetido.
Altere a tag <form> no arquivo produto.php para apontar para a nova página adicionando o atributo action:

```
<form action="checkout.php">
```

Abra a página de produto no navegador e teste o submit. Repare como a página de checkout é chamada e os parâmetros escolhidos na página anterior vão junto na URL.

3. Os parâmetros enviados no capítulo anterior aparecem na URL. porque o formulário, por padrão, é do tipo GET. Pode-se trocar para o tipo POST. Assim, os parâmetros serão enviados mas não estarão visíveis na URL.
Altere a tag <form> no arquivo produto.php para enviar via POST adicionando o atributo method:

```
<form action="checkout.php" method="POST">
```

Na página checkout.php pode-se pegar os valores submetidos através de variáveis do PHP e exibir esses valores na tela. Adicione na página de checkout as informações de cor e tamanho escolhidos. Use uma lista de definições DL com DT/DD pra isso:

```
<h2>Sua compra</h2>
<dl>
    <dt>Cor</dt>
    <dd><?= $_POST['cor'] ?></dd>

    <dt>Tamanho</dt>
    <dd><?= $_POST['tamanho'] ?></dd>
</dl>
```

Teste acessando um produto e clicando em comprar. Os valores escolhidos devem aparecer na página de checkout, através dos parâmetros escolhidos.
Teste várias vezes, com escolhas diferentes, para notar a página de checkout dinamicamente construída com os parâmetros enviados.

4. Imagine que vamos ter vários produtos diferentes na loja, todas enviando as compras pra a página de checkout. E lá queremos saber qual o nome e preço do produto que está sendo comprado (além da cor e do tamanho).
Podemos passar mais dois parâmetros para a página de checkout com o nome e o preço do produto.
Use dois input hidden pra passar essas informações. Na página produto.php, adicione dentro do form:

```
<input type="hidden" name="nome" value="Fuzzy Cardigan">
<input type="hidden" name="preco" value="129,00">
```

Por fim, na página checkout.php, adicione a impressão dos parâmetros nome e do preço dentro da lista de definições que foi feita antes:

```
<dt>Produto</dt>
<dd><?= $_POST['nome'] ?></dd>
<dt>Preço</dt>
<dd>R$ <?= $_POST['preco'] ?></dd>
```

Teste o funcionamento do hidden fazendo uma nova compra de produto.

PHP: Banco de Dados e SQL

- Na maioria dos sistemas Web, dados do negócio ficam separados em um banco de dados

- Em uma loja virtual, os produtos seriam cadastrados nesse banco de dados externo e depois seus dados são exibidos na página produto.php

- Um banco de dados é como uma grande planilha do Excel que possui várias tabelas dentro

- Cada tabela tem colunas e linhas com os dados cadastrados

- Para manipular os dados dessa tabela, usa-se uma linguagem como SQL
- Seu objetivo é permitir buscas nas tabelas por certos tipos de dados
- Inserção, remoção e atualização de dados

- Exemplo: acessar todas as informações da tabela com nome produtos:

  SELECT * FROM produtos

- O comando SELECT indica a seleção de dados. O asterisco indica que se deseja todos os dados. O FROM produtos aponta para o nome da tabela.

- É possível selecionar apenas um dado específico.

- Exemplo: apenas os nomes dos produtos:

  SELECT nome FROM produtos

id	nome	preco	descricao	data	vendas
1	Fuzzy Cardigan	R$ 129,00	Esse á o melhor casaco de Cardigã qu...	2013-09-01	5
2	Camiseta Echo Caveira Bad to The Bone	R$ 48,95	Camiseta confortável ideal para eventos cas...	2013-10-01	90
3	Cardigan Theture Basic	R$ 296,00	Cardigan em ótimo caimento. Ideal fazer sob...	2013-09-29	12
4	Casaco Winter	R$ 699,00	Jaqueta confeccionada em couro ecológico, e...	2013-09-20	3
5	Sport Top	R$ 89,00	Camisa em Algodão estampada ideal para uma ...	2013-09-27	45
6	Top Basic	R$ 45,00	Blusa multicolorida com decote em V, a manga longa...	2013-09-01	178
7	Camiseta Gwol	R$ 125,00	Camiseta em modelagem reta, mangas curtas e decote...	2013-09-01	84
8	Camiseta Tiup	R$ 129,00	Regata básica em seda. Combinaç&atil...	2013-09-03	123
9	Camisa Squares	R$ 199,00	Camisa em tecido mega confortável. Gola fec...	2013-09-18	78
10	Top In Slub	R$ 118,00	Esse á o melhor casaco de Cardigã qu...	2013-09-03	146
11	Shorts Lez a Lez Towel	R$ 77,00	Sport deluxe á a tendência quent&iacu...	2013-09-02	345
12	Camisa Richards Sam	R$ 420,00	Em 1974 a Richards bançou no Brasil um novo...	2013-09-07	256
13	Blusa Lez a Lez Feel Sft I	R$ 107,00	Blusa manga longa, decote redondo com fechamento e...	2013-09-03	42

MySQL e PHPMyAdmin

- Existem muitos bancos de dados no mercado. Todos aceitam comandos SQL, com pequenas variações

- Será usado MySQL que é um dos bancos mais usados no mundo e bastante usado por programadores PHP

- Para ajudar a visualizar e administrar o MySQL pode-se instalar interface gráfica compatível

- Opção muito comum é o phpMyAdmin
- Administrador de MySQL escrito em PHP para Web que roda em um servidor normal e pode ser acessado direto no navegador

Será utilizado o Aplicativo Xampp, que já contém o MySQL, o servidor apache para comunicação e a interface phpMyAdmin

- https://downloadsapachefriends.global.ssl.fastly.net/8.0.19/xampp-windows-x64-8.0.19-0-VS16-installer.exe (Windows)
- https://downloadsapachefriends.global.ssl.fastly.net/8.0.19/xampp-linux-x64-8.0.19-0-installer.run (Linux)
- https://downloadsapachefriends.global.ssl.fastly.net/8.0.19/xampp-osx-8.0.19-0-vm.dmg (Mac OS)

Buscas no MySQL com PHP

- É necessário usar um banco de dados de dentro da página PHP
- Pegar dados no banco e imprimir na tela usando PHP

Conexão

- A conexão com o banco de dados pode ser feita com a função mysqli_connect :

```
$conexao = mysqli_connect("localhost", "root", "<senha>", “mirrorfashion");
```

- Função recebe onde se conectar (localhost), o usuário (root), a senha (em branco) e o nome do banco de dados disponível (mirrorfashion).
- Se tudo der certo, abre a conexão e devolve uma variável $conexao que representa a conexão aberta

Busca

- Próximo passo é mandar o MySQL processar um comando SQL, como o SELECT.
- Usa-se a função mysqli_query que recebe a conexão (aberta anteriormente) e o SQL da busca:

```
$dados = mysqli_query($conexao, "SELECT * FROM produtos”);
```

- Função devolve uma variável $dados com o resultado da busca

Dados no PHP

- Último passo é transformar os dados da tabela em algo usável no PHP
- Maneira comum é transformar os dados em um array com mysqli_fetch_array:

```
$produto = mysqli_fetch_array($dados);
```

- A variável $produto é um array PHP com os dados do primeiro produto da busca indexados pelo nome da coluna no banco de dados.
- Por exemplo, pode-se acessar o preço do produto através de $produto['preco'] e assim por diante para cada coluna

Refinando Buscas com Where

- Há muitas opções no SQL para refino da busca

- SELECT * retorna todos os dados da tabela inteira, o que pode ser muita coisa

- É comum o interesse em apenas um dado específico, de uma certa linha

- Exemplo: Pode-se indicar ao SELECT o desejo pelos dados do produto de um certo ID
- ID identifica o código individual de cada produto

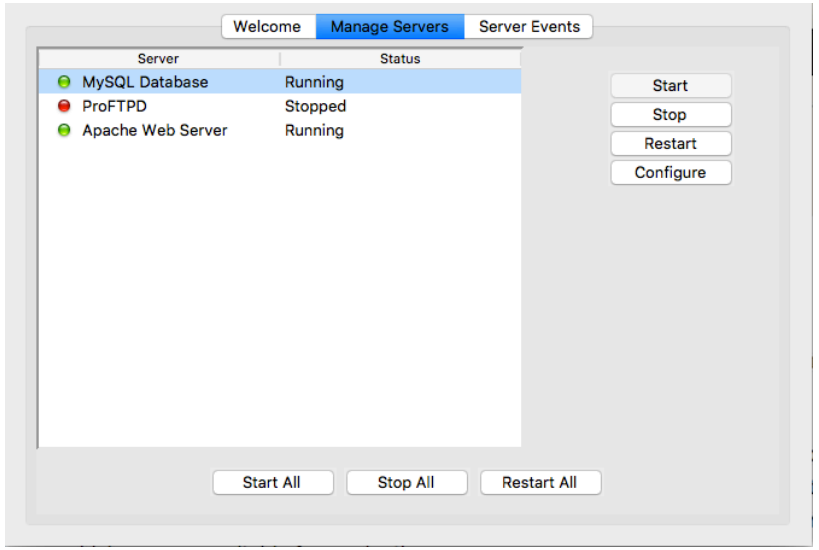
```
SELECT * FROM produtos WHERE id = 4
```

- Esse código apenas devolve todas as colunas do produto cujo id for 4

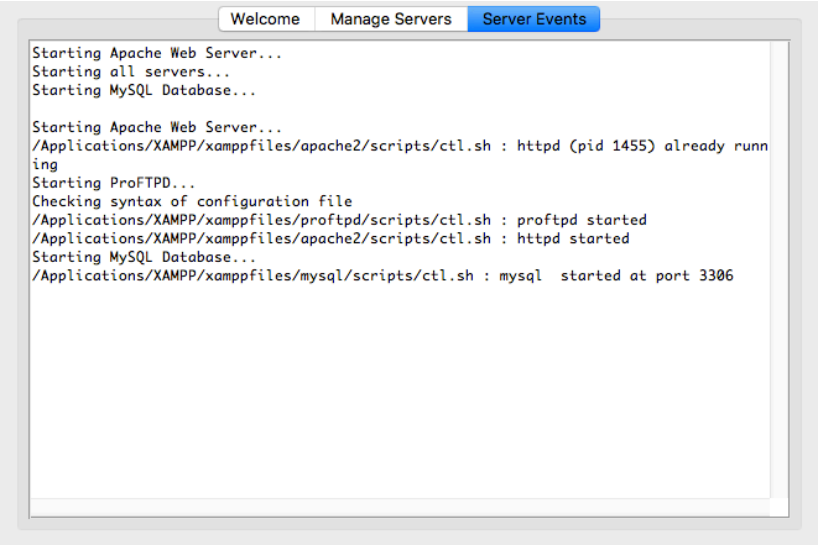
Atividade - PHPMyAdmin

- Quando abrir o Xampp, irá abrir a janela de boas vindas

- Em Manage Servers, dar start no MySQL e no Apache

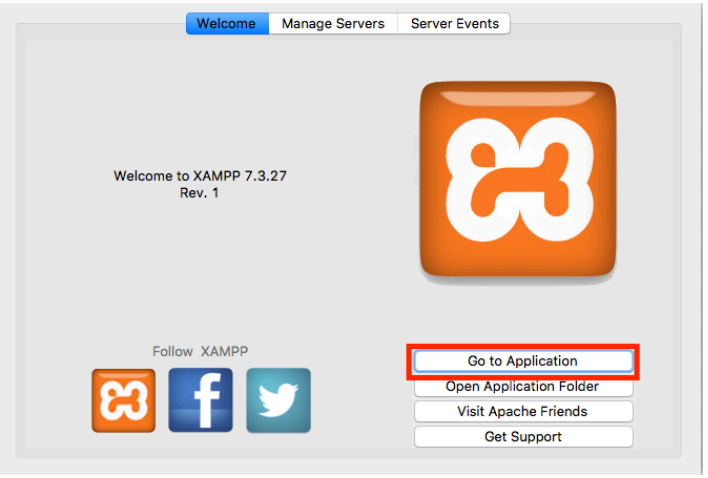


- Em Server Events é possível visualizar como os servidores estão se comportando

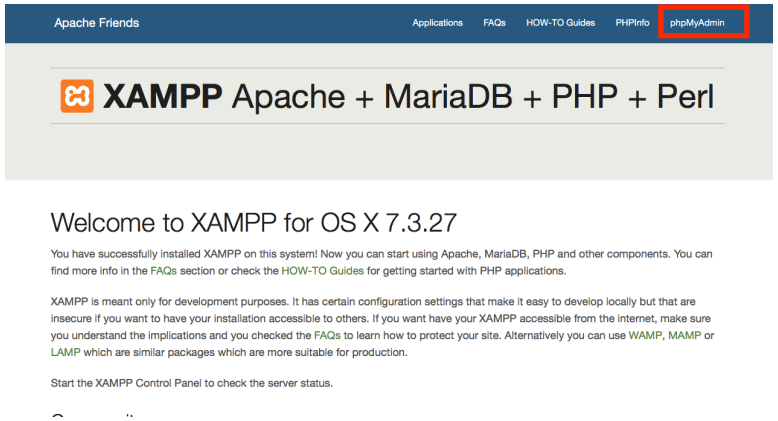


Atividade - PHPMyAdmin

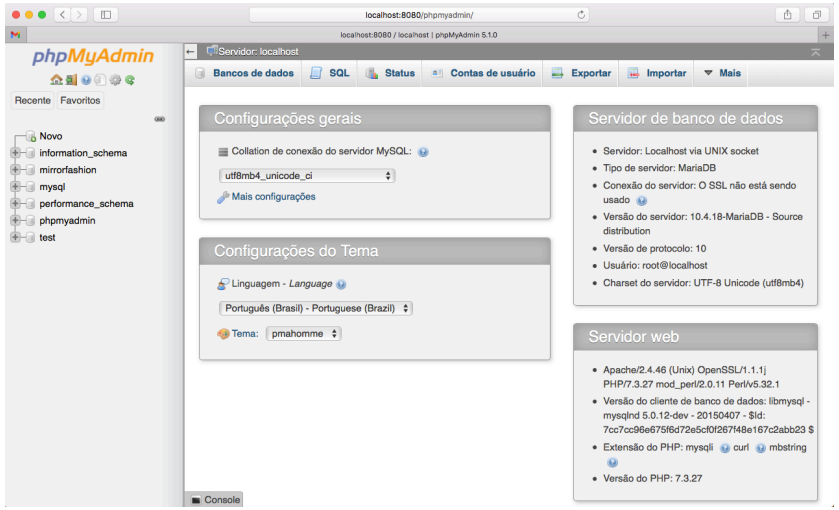
- Para acessar o PHPMyAdmin, na parte de boas vindas, escolher a opção Go to Application



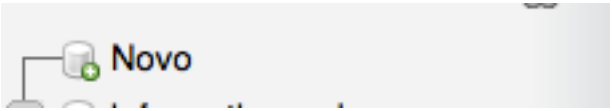
- Será carregada uma página no Navegador. No topo, acessar o link PHPMyAdmin



- Será carregada a página com a aplicação do PHPMyAdmin



Atividade - PHPMyAdmin



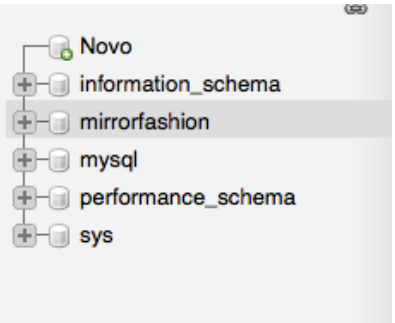
1

2. A primeira tela do phpMyAdmin pode ser assustadora de tantas opções. Existem muitos recursos.

Na parte esquerda da tela, escolha “novo” (1), e escolha Criar Banco de Dados, na parte da direita. Coloque o nome “mirrorfashion no campo e clique em criar (2)

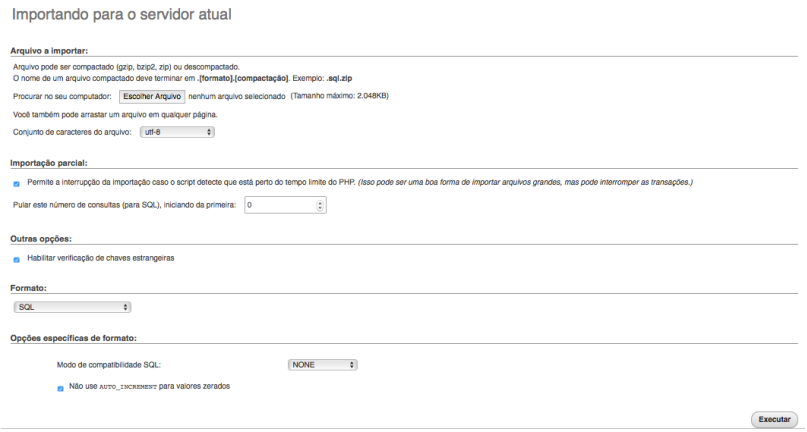
2

Na parte esquerda ficam os bancos de dados disponíveis na máquina. Localize e clique no banco mirrorfashion que corresponde aos dados do curso e que acabou de ser criado. (3)



3

Pode-se importar dados do curso pra esse banco. Eles estão no arquivo dados.sql. Para importar, vá no menu Import no topo do phpMyAdmin. Clique em "Escolher Arquivo" para selecionar o arquivo no seu computador e depois clique em "Executar". (4)



4

Após a importação, selecione a tabela produtos dentro de mirrorfashion no menu da esquerda. Ele deve mostrar os dados que estão dentro da tabela, suas colunas e linhas. Use a aba SQL, copie o comando abaixo e escolha “Executar"

```
SELECT *
FROM `produtos`
LIMIT 0 , 30
```

Atividade - PHPMyAdmin

Bancos de dados | SQL | Status | Contas de usuário | Exportar | Importar | Configurações | M

Rodar consulta(s) SQL no servidor "127.0.0.1":

```
1 SELECT *
2 FROM `produtos`
3 LIMIT 0 , 30
```

Limpar | Formato | Obter consulta auto-salva

Unir parâmetros

Delimitadores ; | Mostrar esta consulta SQL aqui novamente | Manter caixa de consulta | Executar

Console | uando finalizado | Habilitar verificação de chaves estrangeiras

Mostrando registros 0 - 14 (15 no total, Consulta levou 0,0005 segundos.)

```
SELECT * FROM `produtos` LIMIT 0 , 30
```

Perfil [Editar em linha] [Editar] [Demonstrar SQL] [Criar código PHP] [Atualizar]

+ Opções

←T→				id	nome	preco	descricao	data	vendas
Editar	Copiar	Remover		1	Fuzzy Cardigan	R$ 129,00	Esse é o melhor casaco de Cardigã qu...	2013-09-01	5
Editar	Copiar	Remover		2	Camiseta Ecko Caveira Bad to The Bone	R$ 48,95	Camiseta confortável ideal para eventos cas...	2013-10-01	90
Editar	Copiar	Remover		3	Cardigan Thelure Basic	R$ 296,00	Cardigan em ótimo caimento. Ideal fazer sob...	2013-09-29	12
Editar	Copiar	Remover		4	Casaco Winter	R$ 699,00	Jaqueta confeccionada em couro ecológico, e...	2013-09-20	3
Editar	Copiar	Remover		5	Sport Top	R$ 89,00	Camisa em Algodão estampada ideal para uma ...	2013-09-27	45
Editar	Copiar	Remover		6	Top Basic	R$ 45,00	Blusa multicolorida com decote em V, a manga longa...	2013-09-01	178
Editar	Copiar	Remover		7	Camiseta Gwol	R$ 125,00	Camiseta em modelagem reta, mangas curtas e decote...	2013-09-01	84
Editar	Copiar	Remover		8	Camiseta Tiup	R$ 129,00	Regata básica em seda. Combinaç&atil...	2013-09-03	123
Editar	Copiar	Remover		9	Camisa Squares	R$ 199,00	Camisa em tecido mega confortável. Gola fec...	2013-09-18	78
Console						R$	Esse é o melhor casaco de		

Atividade - PHP com MySQL

1. No topo do arquivo produto.php abra a conexão com o banco de dados e selecione os dados do produto:

```
<?php
    $conexao = mysqli_connect("127.0.0.1", “root","”, "mirrorfashion");
    $dados = mysqli_query($conexao, "SELECT * FROM produtos");
    $produto = mysqli_fetch_array($dados);
?>
```

Repare que foi criada uma variável $produto no PHP que contém os dados do produto. Ela é um array e é possível acessar as diferentes colunas através do nome.

Altere os títulos na página para usar os dados dinâmicos do banco:

```
<h1><?= $produto['nome'] ?></h1>
<p>por apenas <?= $produto['preco'] ?></p>
```

Altere também, no final da página, o local onde é exibido o texto da descrição do produto:

```
<p><?= $produto['descricao'] ?></p>
```

Teste a página de produto no navegador e repare que os dados vêm dinamicamente do banco.

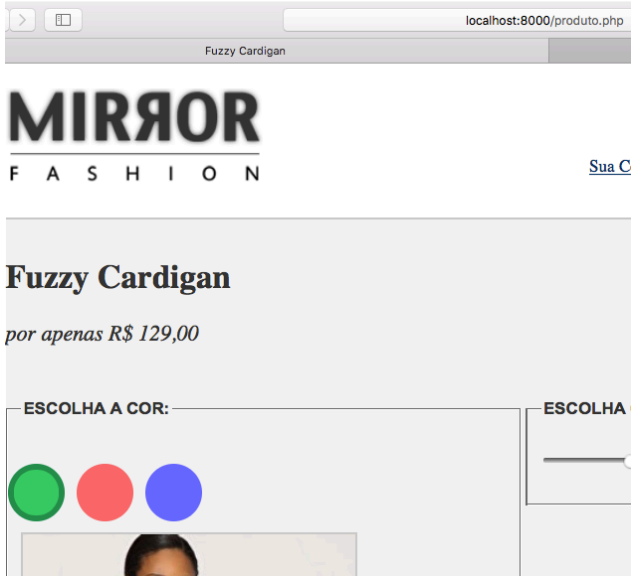
Observe o código fonte HTML final gerado, como é idêntico ao que se tinha antes.

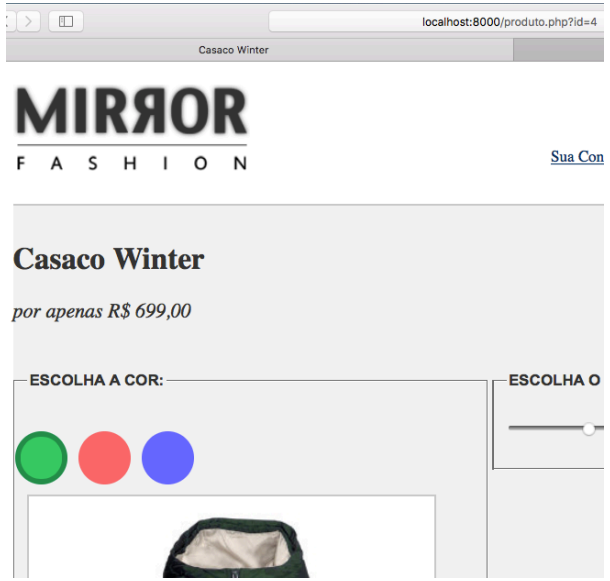
2. Deseja-se que a página seja capaz de exibir os dados de qualquer produto do banco. Para escolher qual produto mostrar, usa-se um parâmetro na URL com o código do produto, o ID.
Altere o código da busca realizada antes para incluir a cláusula WHERE no final baseada no id do produto passado como parâmetro:

"SELECT * FROM produtos WHERE id = $_GET[id]”

Teste a página no navegador passando ids diferentes com parâmetro na URL:

produto.php?id=2.





Atividade - PHP cm MySQL

3. A imagem do produto também é diferente para cada produto. Abra a pasta img/produtos/ e repare nas várias imagens que estão lá. Elas seguem um padrão. O nome contém o código do produto e as três variações de cor. Podemos gerar o endereço das imagens no HTML usando o ID que vem do PHP.
Altere os caminhos das imagens na página de produto pra passar o ID dinamicamente no endereço da foto.
Cuidado com a mistura de código HTML e PHP no meio da tag:

```
<img src="img/produtos/foto<?= $produto['id'] ?>-verde.png”>
```

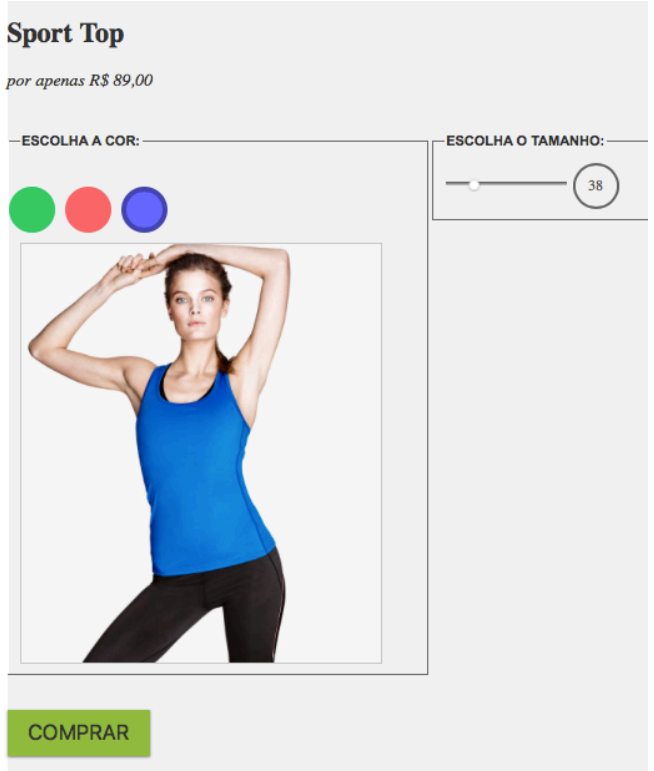
Teste a página no navegador passando ids diferentes com parâmetro na URL: produto.php?id=4.

4. Adicione mais um input hidden dentro do formulário passando o id.

```
<input type="hidden" name="id" value="<?= $produto['id'] ?>”>
```

5. (opcional) O <title> da página é um fator muito importante para motores de busca (SEO). O ideal é ter títulos descritivos e únicos em cada página. Numa loja virtual, o nome do produto deve fazer parte do título da página.
Altere o título da página pra puxar o nome do produto do banco de dados dinamicamente.

6. (opcional) Mude os input hidden feitos anteriormente com nome/preço pra pegar valores dinâmicos do banco de dados usando PHP.



Busca e Ordenação de Resultados

- Pode-se restringir o número de resultados.
- Exemplo: uma tabela pode possuir milhares de registros, mas deseja-se apenas os primeiros 10
- Usa-se o comando LIMIT na busca:

```
SELECT * FROM produtos LIMIT 0, 10
```

- Devolve 10 resultados contando a partir do primeiro (0)
- É possível trocar o primeiro número pra acessar informações em outras partes do banco

- Outro recurso é devolver as informações ordenadas de acordo com algum critério.
- Exemplo: produtos em ordem alfabética por seu nome:

```
SELECT * FROM produtos ORDER BY nome
```

- Também é possível escolher o sentido da ordenação
- ASC (ascendente)
- DESC (decrescente)
- Exemplo, para ordenar produtos no sentido do mais recente para o mais antigo:

```
SELECT * FROM produtos ORDER BY data DESC
```

- Ainda, pode-se misturar todos os comando em um SQL complexo:

```
SELECT * FROM produtos ORDER BY data DESC LIMIT
0, 10
```

Atividade - Mais Buscas com PHP

1. A home page lista os produtos mas, do jeito atual, está tudo estático com dados de mentira. Vamos alterar para fazer uma busca no banco de dados e retornar os produtos a serem exibidos. É muito parecido com o que foi feito na página de produto; a diferença é que vamos listar vários produtos de uma vez ao invés de um só.

Em primeiro lugar, é preciso transformar a página num arquivo PHP, para poder usar o
banco de dados nela. Renomeie o arquivo index.html para index.php se ainda não fez.
No index.php, localize o painel novidades e apague todos os <li> que listam produtos com dados estáticos.
No lugar, escreva um código PHP que faz a busca dos produtos no banco de dados e percorra essa lista com um laço while:

```
<ol>
    <?php
        $conexao = mysqli_connect("127.0.0.1", "root", “", "mirrorfashion");
        $dados = mysqli_query($conexao, "SELECT * FROM produtos");
        while ($produto = mysqli_fetch_array($dados)):
    ?>
    <li>
        <a href="produto.php?id=<?= $produto['id'] ?>">
            <figure>
                <img src="img/produtos/miniatura<?= $produto['id'] ?>.png"
                    alt="<?= $produto['nome'] ?>">
                <figcaption><?= $produto['nome'] ?> por <?= $produto['preco'] ?></
                figcaption>
            </figure>
        </a>
    </li>
    <?php endwhile; ?>
</ol>
```

Repare como foi refeito o <li> de antes mas usando todos os dados dinâmicos do banco de dados.

Teste a home no navegador e veja a busca dinâmica acontecendo. Confira o código fonte HTML gerado, igual ao que existia antes.

Atividade - Mais Buscas com PHP

2. A busca feita antes com o SELECT traz todos os dados da tabela. Isso pode ser bem grande numa loja de verdade. O ideal é restringir a busca apenas pelos dados necessários. Para isso, no MySQL, pode-se usar o comando LIMIT passando o máximo de resultados que estamos interessados.
Altere o código anterior para incluir o LIMIT no SQL da busca:

```
SELECT * FROM produtos LIMIT 0, 12
```

Teste novamente a home.

3. Repare que a ordem que os produtos vêm não é a ordem desejada. No painel novidades, se queria que viessem ordenados pelo produto mais recente ao mais antigo.
Pode-se fazer isso adicionando uma cláusula de ordem no SQL com ORDER BY.
Altere a busca anterior para incluir uma ordenação com base no campo data de maneira decrescente:

```
SELECT * FROM produtos ORDER BY data DESC LIMIT 0, 12
```

Teste novamente a home e veja o resultado.

4. (opcional) Implemente o mesmo recurso de busca dinâmica no outro painel, o de produtos mais vendidos. A única diferença é que queremos ordenar os elementos a partir da quantidade de vendas. Use o campo vendas no ORDER BY pra isso.

5. (opcional) Teste outros valores para o LIMIT nas buscas. O primeiro número indica offset, ou seja a partir de qual item estamos interessados. Usar 3,6 indica que queremos os itens do terceiro ao nono.

Teste também outras ordenações. Além do DESC, temos o ASC.