



Fundação Presidente Antônio Carlos
Faculdade Presidente Antônio Carlos de Teófilo Otoni
Sistemas de informação – Programação para Internet

Prof. Giovanni Camargo Silva.

Trabalho Final de complemento ao aprendizado

Programação para Internet.

O objetivo desse trabalho é aplicar e documentar os conhecimentos adquiridos em sala, ou em pesquisa extraclasse, tem também como finalidade se constituir como a primeira base de consultas para futuros desenvolvimentos na área web na linguagem PHP procedural.

O desenvolvimento de um sistema por mais simples que seja, requer uma significativa demanda de recursos da linguagem, pela qual, ele é desenvolvido, por isso se dá à riqueza desse trabalho.

Os requisitos

Os requisitos aqui apresentados são os mínimos possíveis para o desenvolvimento:

Trata-se de um sistema de gerenciamento de turma de uma determinada instituição de cursos técnicos. O que se sabe que esta instituição disponibiliza três cursos técnicos com a possibilidade de aumentar a demanda. Dependendo da demanda de matrículas pode-se ocorrer a formação de mais de uma turma, para cada curso no mesmo semestre e ano, que serão identificados por letras do alfabeto brasileiro. A conclusão do curso se dá por três períodos sendo estes semestrais. Logo a identificação da turma deve conter tais dados, como curso, letra, período, semestre 1 ou 2 e o ano da turma.

O curso será identificado pelo seu nome e pela sigla.

Os alunos a princípio seriam registrados com matrícula, nome e telefone.

Um aluno não pode fazer parte de duas turmas do mesmo semestre e mesmo ano.

Com isso o sistema deve apresentar meios para cadastrar os cursos, a turma e os alunos. Meio também para alterar, caso precise, esses dados. Apresentar também ferramentas para associar os alunos nas devidas turmas.

Apenas administradores do sistema terão acesso às ferramentas de cadastro e alteração de dados. O sistema deve gerar relatórios de alunos por turma, alunos por curso e relatórios de turmas.

Modelagem

Começando pela modelagem do banco de dados. É notável observar que existem três entidades bem distintas: aluno, turma e curso. Sendo que cada turma pertence a um curso logo existe esse relacionamento. Vários alunos podem pertencer a várias turmas isso gera uma entidade de relacionamento chamado aqui de aluno_turma onde terão os códigos de cada aluno e da turma ao qual ele pertence como relacionado abaixo. Como esse sistema terá administração restrita isso gera uma entidade aqui chamada de administrar onde será registrado o nome, o login, a senha e o status desse usuário, esse último identifica se ele é um usuário ativo ou não.



Fundação Presidente Antônio Carlos

Faculdade Presidente Antônio Carlos de Teófilo Otoni

Sistemas de informação – Programação para Internet

Prof. Giovanni Camargo Silva.

Turma
Cod_turma
Letra
Período int
Cod_curso
Ano
Semestre

Curso
Cod_curso
Nome
Sigla

Aluno
Cod_aluno
Matricula
Nome
Telefone

Aluno_turma
Cod_aluno
Cod_turma

Administrar
Cod_admin
Nome
Login
Senha
status

Turma

Cod_turma	Letra	periodo	Cod_curso	ano	semestre
1	A	3	2	2009	2
2	A	4	3	2009	2
3	A	6	2	2009	1
5	A	7	2	2009	2
6	A	3	1	2008	2
7	A	4	1	2009	2
8	A	6	3	2009	1
9	A	7	2	2009	2

Cod_aluno	Matricula	Nome	Telefone
1	123	Índia Cristina	334545
2	124	Jamilson Pires	345454
3	125	Keila Alves	661232
4	128	Jucilene Alves	098324
5	121	Luiz Gonzaga	879609
6	122	Tainan Moreira	435710
7	129	Tamara Silva	456712
8	130	Thiago do Val	098732
9	131	Maycom Keller	546541
10	132	Gustavo Alves	897867

Cod_aluno	Cod_turma
1	2
2	2
5	1
6	2
3	3
7	3
4	1

Cod_curso	nome	Sigla
1	Informática	INFO
2	Agropecuária	AGRO
3	Mecânica	MEC

Modelando o sistema

Cadastros : Para ser desenvolvida se deve criar um arquivo **principal.php**, para dar opção de menu, o arquivo conecta.php, para estabelecer a conexão com o banco de dados e então ser incluído em todos os arquivos que precisar utilizar bancos de dados, e mais os arquivos: cad.php e cad2.php para cadastrar alunos, curso.php e curso2.php para cadastrar os cursos, turma.php e turma2.php para cadastrar as turmas.

Conecta.php

```
$host = 'localhost';
$banco = 'gestao';
$usuario = 'gestao';
$senha = 'abc123';
$con = mysqli_connect($host, $usuario, $senha, $banco);
```



Fundação Presidente Antônio Carlos
Faculdade Presidente Antônio Carlos de Teófilo Otoni
Sistemas de informação – Programação para Internet

Prof. Giovanni Camargo Silva.

Cad.php

Cadastro de aluno	
Aluno	<input type="text"/>
Matricula	<input type="text"/>
Telefone	<input type="text"/>
<input type="button" value="Submit"/>	

Cad2.php <? ?>

Curso.php

Cadastro de Curso	
Curso	<input type="text"/>
Sigla	<input type="text"/>
<input type="button" value="Ir"/>	

Curso2.php <? ?>

Cadastro de turma	
Letra	<input type="text"/>
Período	<input type="text"/>
Curso	Sistemas de Informação ▼
Ano	<input type="text"/>
Semestre	1 ▼
<input type="button" value="Ir"/>	

Turma.php

Turma2.php <? ?>

Criando o banco de dados no phpmyadmin:

- 1) Instale o appserv.
- 2) Abra o seu browser, navegador da web, e entre no endereço: <http://localhost/phpMyAdmin>. não precisa estar conectado a internet é um servidor local.

Fundação Presidente Antônio Carlos

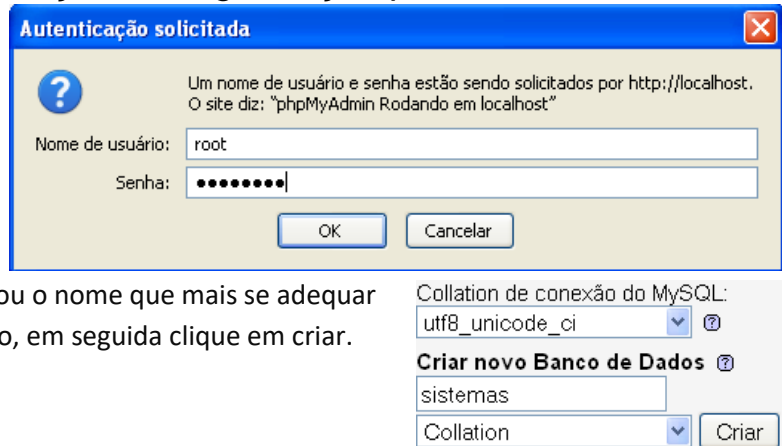
Faculdade Presidente Antônio Carlos de Teófilo Otoni

Sistemas de informação – Programação para Internet

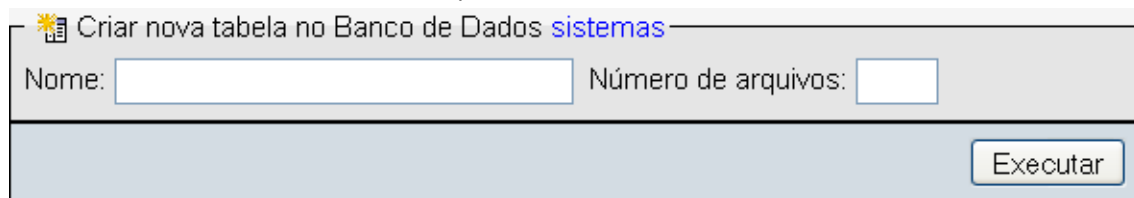
Prof. Giovanni Camargo Silva.

- 3) Digite o usuário root e a senha definida por você na hora de instalar o appserv e clique em OK

- 4) Crie um novo banco de dados com o nome sistemas ou ciências ou o nome que mais se adequar ao seu projeto de desenvolvimento, em seguida clique em criar.



- 5) Pode-se criar as tabelas: aluno, aluno_turma, turma e curso, uma a uma, insira o nome da tabela e quantos campos tem em números de arquivos. Ex: a tabela aluno tem na nossa modelagem 4 campos são eles: cod_aluno, matricula, nome, telefone. Logo em nome colocaria: aluno e em Números de arquivos colocaria: 4.



Logo deverá definir os campos da tabela:

Campo	Tipo	Tamanho/Definir	Extra							Comentários
cod_aluno	INT		auto_increment							
matricula	VARCHAR	10								
nome	VARCHAR	50								
telefone	VARCHAR	14								

Note na imagem, que para os campos do tipo varchar é obrigatório colocar o tamanho do campo, para o inteiro não é necessário, como cod_aluno é a chave primária, deve ser marcado isso no final, e em extra, define-se também que a chave será auto incrementada em seguida clique no botão **salvar**. Isso deverá ser feito para as demais tabelas descritas na modelagem.

Há outras maneiras de se criar o banco como importar um arquivo ou manualmente na aba SQL: basta ir na aba e digitar a seguinte instrução depois de criar o banco:



--

-- Estrutura da tabela `administrar`

--

```
CREATE TABLE `administrar` (
  `cod_admin` int(11) NOT NULL auto_increment,
  `nome` varchar(50) NOT NULL,
  `login` varchar(20) NOT NULL,
  `senha` varchar(40) NOT NULL,
  `status` enum('0','1') NOT NULL default '1',
  PRIMARY KEY (`cod_admin`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```



Fundação Presidente Antônio Carlos
Faculdade Presidente Antônio Carlos de Teófilo Otoni
Sistemas de informação – Programação para Internet

Prof. Giovanni Camargo Silva.

```
--  
-- Estrutura da tabela `aluno`  
-- CREATE TABLE `aluno` (  
  `cod_aluno` int(11) NOT NULL auto_increment,  
  `matricula` varchar(10) NOT NULL,  
  `nome` varchar(50) NOT NULL,  
  `telefone` varchar(14) NOT NULL,  
  PRIMARY KEY (`cod_aluno`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;  
--  
-- Estrutura da tabela `aluno_turma`  
--  
CREATE TABLE `aluno_turma` (  
  `cod_aluno` int(11) NOT NULL,  
  `cod_turma` int(11) NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=utf8;  
--  
-- Estrutura da tabela `curso`  
--  
CREATE TABLE `curso` (  
  `cod_curso` int(11) NOT NULL auto_increment,  
  `nome` varchar(30) NOT NULL,  
  `sigla` varchar(4) NOT NULL,  
  PRIMARY KEY (`cod_curso`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;  
--  
-- Estrutura da tabela `turma`  
--  
CREATE TABLE `turma` (  
  `cod_turma` int(11) NOT NULL auto_increment,  
  `letra` varchar(1) NOT NULL,  
  `periodo` int(11) NOT NULL,  
  `cod_curso` int(11) NOT NULL,  
  `ano` int(11) NOT NULL,  
  `semestre` int(11) NOT NULL,  
  PRIMARY KEY (`cod_turma`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

Gerenciamento

Terminando de se criar as páginas de cadastro mostradas no início é hora de elaborar a administração do sistema. Começando por adicionar alunos nas turmas. Para isso deverá escolher uma turma em uma lista, logo direcionar para uma lista de alunos para escolher os possíveis alunos da turma, então serão três páginas pelo menos. **aturma.php**, que listará as turmas cadastradas, **aturma2.php** que listará os alunos com um checkbox para escolher os alunos da turma escolhida na tela anterior e **aturma3.php** que irá inserir na tabela aluno_turma o código do aluno e da turma.

Prof. Giovanni Camargo Silva.

Cadastrar Aluno na turma
Turmas e Cursos
1º período Ciência da Computação
2º período Ciência da Computação
3º período Ciência da Computação
1º período Sistemas de Informação
2º período Sistemas de Informação
4º período Sistemas de Informação

aturma.php: opção para escolher a turma com um clique e enviar essa informação para a página aturma2.php, consulte do banco todas as turmas disponíveis e liste na página, e envie através hiperlink o código da turma

```
<a href="aturma2.php?cod_turma=
<?=$cod_turma?>">
```

Cadastrar Aluno na turma	
Opções	Aluno
<input type="checkbox"/>	Artur Marsaddi
<input type="checkbox"/>	Bárbara Êmylle
<input type="checkbox"/>	Eduardo César
<input type="checkbox"/>	Eloiza Leal
<input type="checkbox"/>	Emanoel Ferreira
<input type="checkbox"/>	Flávia Gomes
<input type="checkbox"/>	João Gabriel
<input type="checkbox"/>	Pedro Luiz
<input type="checkbox"/>	Wellington Oliveira
<input type="button" value="Confirmar"/>	

aturma2.php: uma página para escolher os alunos da turma escolhida anteriormente: sugestão envie o código da turma através do campo hidden para a página aturma3.php, conte também o número de alunos e dê nome concatenado com um contador iniciado de 1. cad1, cad2, cad3... o contador também deve ser enviado para o destino com um campo hidden, isso para resgatar os valores dos campos checks.

Aturma3.php

Primeiro recupere o código da turma, em seguida, recupere a quantidade de alunos para testar quais alunos foram selecionados. Veja o código:












```
1 <?php
2 include("conecta.php");
3 $cod_turma = $_POST['cod_turma']; //campo hidden
4 $qtd = $_POST['qtd_aluno']; //campo hidden
5 for($i=1;$i<=$qtd;$i++){
6     $cod_aluno = $_POST["cod".$i];
7     if($cod_aluno){
8         $sql = "insert into aluno_turma values($cod_aluno, $cod_turma)";
9         mysql_query($sql);
10    }
11 }
12 ?>
13
```

Prof. Giovanni Camargo Silva.

Manter os dados atualizados

Para manter os dados é necessário apresentar para o responsável uma lista de opções para atualizações, como exemplo os alunos, interessante também fazer um filtro para facilitar a busca se tratando de aluno que seja por nome

Ter-se-ia os arquivos :

Aluno	Excluir	Alterar
Artur Marsadi		
Bárbara Êmylle		
Eduardo César		
Eloiza Leal		
Emanoel Ferreira		
Flávia Gomes		
João Gabriel		
Pedro Luiz		
Wellington Oliveira		

consulta.php, que listaria os alunos em ordem alfabética.

Ao clicar na opção de alterar do lado de cada aluno deverá ser direcionada para a página

Nome	<input type="text" value="Eduardo César"/>
Matrícula	<input type="text" value="3900001"/>
Telefone	<input type="text" value="35264349"/>
<input type="button" value="ALTERAR"/>	

alt2.php que apresentará todos os dados do aluno, que podem ser alterados, organizados em um formulário semelhante ao de cadastro.

Esses dados serão enviados para página altera.php que irá atualizar os dados novos no banco de dados.

Todos os passos devem ser repetidos para alterar turma e curso.

Segurança

Há alguns meios de fazer segurança dos dados, como já mencionado anteriormente temos as sessões de usuários. Em php para fazer uso das sessões é necessário habilitar a página com a função `session_start()` na primeira linha da página. Através do array `$_SESSION[]`, pode-se criar uma sessão e atribuir um valor a ela ou resgatar o valor atribuído. **index.php**

Login	<input type="text"/>
Senha	<input type="password"/>
<input type="button" value="Logar"/>	

Figura 1 tela login

A página inicial do sistema deve ser uma página de login, ou seja o seu index vai ter a configuração ao lado, esses dados vão ser enviados para uma página chamada de logar.php cujo objetivo é verificar junto a base de dados se esse login existe com tal

usuário e senha. Se existir carregará as funções e permitirá acesso a página principal de menu se não existir avisa o erro e volta para página de login.

Prof. Giovanni Camargo Silva.

Logar.php

```

1  <?php session_start();
2  include("conecta.php");
3  $log = $_POST['login'];
4  $sen = sha1($_POST['senha']);
5  $sql = "select cod_admin, nome from administrar
6         where login = '$log'
7         and senha = '$sen'
8         and status = '1'";
9  $exe = mysql_query($sql);
10 if($linha = mysql_fetch_array($exe)){
11     $_SESSION['usu_nome'] = $linha['nome'];
12     $_SESSION['usu_cod'] = $linha['cod_admin'];
13     echo "<meta content='0;url=principal.php' http-equiv='refresh' />";
14     exit;
15 }
16 else{
17     session_destroy();
18     echo "<hl>Login ou senha não confere!</hl>";
19     echo "<meta content='4;url=login.php' http-equiv='refresh' />";
20     exit;
21 }
22 ?>

```

Feito isso, é necessário em cada página de acesso restrito, verificar se as sessões dos usuários estão ativas, como é feito como um padrão pode então fazer um arquivo em separado e incluí-lo em cada página, pois fica mais fácil para se fazer manutenção e diminui a quantidade de código a ser escrito.

IMPORTANTE: Esse arquivo deve ser incluído na primeira linha de cada página, ele tem a função `session_start()` que deve ser colocada como o primeiro comando quando utilizada, esse arquivo vai se chamar `testar.php`.

testar.php

```

1  <?php session_start();
2  $_SESSION['usu_nome'] = $_SESSION['usu_cod'] = 1;
3  if(empty($_SESSION['usu_nome']) && empty($_SESSION['usu_cod'])){
4      echo "<hl>Acesso restrito por favor faça o seu login</hl>";
5      echo "<meta content='3;url=index.php' http-equiv='refresh' />";
6      exit;
7  }
8  ?>

```

Para sair do sistema é preciso desativar as sessões do usuário, por isso deve ter um arquivo que chamará `sair.php`, deverá também ter no menu uma opção de sair do sistema e essa saída deve esgotar as sessões ativas.

sair.php

```

1  <?php session_start();
2      echo "<hl>Obrigado " . $_SESSION['usu_login'] . " você está saindo do sistema</hl>";
3      session_destroy();
4      echo "<meta content='3;url=index.php' http-equiv='refresh' />";
5      exit;
6  ?>

```


Fundação Presidente Antônio Carlos

Faculdade Presidente Antônio Carlos de Teófilo Otoni

Sistemas de informação – Programação para Internet

Prof. Giovanni Camargo Silva.

Para se criar estas opções, é necessário uma nova tabela na base de dados que será a tabela administrar com os campos cod_admin, nome, login, senha, status.

O campo status é para desativar o usuário que não mais poder ter acesso ao sistema pois não é recomendado removê-los. Então, defini-se que valor 1 ele é um usuário ativo, valor 0 ele é inativo.

Esse é o comando que deve ser inserido na aba SQL do phpmyadmin caso não tenha ainda

```
CREATE TABLE `administrar` (
  `cod_admin` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `nome` VARCHAR( 50 ) NOT NULL ,
  `login` VARCHAR( 20 ) NOT NULL ,
  `senha` VARCHAR( 40 ) NOT NULL ,
  `status` ENUM( '0', '1' ) NOT NULL DEFAULT '1'
) ENGINE = MYISAM ;
```

É bom que se use um administrador padrão para poder acessar seu sistema, pelo menos na primeira vez que acessá-lo.

para isso basta executar o comando: seu usuário será adm e sua senha adm.

```
INSERT INTO `ciencias`.`administrar` (
  `cod_admin`, `nome`, `login`, `senha`, `status` )
```

```
VALUES ( NULL , 'Administrador', 'adm',
  SHA1( 'adm' ) , '1' );
```

A sua página principal.php deve ter uma apresentação do sistema e um menu de opções de acesso pra aprendizado é bom que se use o iframe, em uma fase mais avançada busque por novos recursos que mantenha um padrão de elegância em suas páginas. O modelo já usado anteriormente pode ser usado aqui também, onde cada menu representa um comportamento do sistema. Um ponto importante que deve ser elaborado nesse sistema é um recurso para cadastrar novos usuários e inabilitar usuários. Recuperação de senhas e alteração de senhas. A recuperação de senhas deve ser feita a partir da geração de uma nova senha, para manter a política de sigilo do usuário

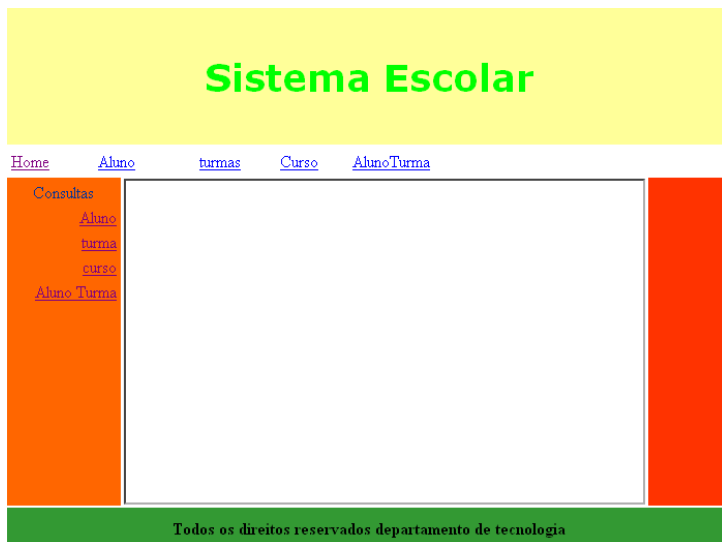


Figura 2 Isto é apenas uma amostra - façam design bonito

imprescindível em qualquer sistema computacional.

Nota: O appServ é só uma sugestão, se quiser utilizar outra ferramenta fique a vontade.

Nota para avaliação do Trabalho;

Alguns recursos estão desatualizados, será quesito de avaliação também estas atualizações, como usar as funções *mysqli_* ao invés de *mysql_*.

A criatividade também será apreciada como critério avaliativo, organize o seu trabalho.

O esforço de cada um é imprescindível, deve se usar recursos de apresentação e comportamento com css e javascript.

O que será enviado para o portal?

Para o portal deve ser enviado um relatório de como foi feito o trabalho e como, este, contribuiu no seu processo de aprendizado. Cada trabalho será avaliado nos laboratórios com as apresentações e também deve-se enviar toda a estrutura elaborada compactada para o e-mail: vannipassos@gmail.com. Com o assunto PHP-Final, e no corpo deve conter os dois integrantes da dupla que desenvolveu o trabalho.

As imagens são meramente intuitivas, por favor, façam um design bonito... será avaliado.