

*Identifique sua apostila*

Nome: \_\_\_\_\_

Turma: \_\_\_\_\_

# LPII – Ap 2 PHP

Apostila 2 PHP – Disciplina LPII (PHP).

Conteúdo: Banco de Dados. Curso:

Informática – Turmas: 72A, 72B, 72C, 72D

Prof<sup>a</sup> Ariane Scarelli e Prof<sup>a</sup> Kátia Zambon

2017

## SUMÁRIO

1	Introdução .....	3
2	Tipos de Banco de Dados (BD) .....	3
3	Conexão e manipulação do BD PostgreSQL .....	3
3.1	Conectar.....	3
3.2	Inserir .....	4
3.3	Alterar.....	13
3.4	Excluir .....	18
4	REFERÊNCIAS BIBLIOGRÁFICAS.....	26
5	SITES ÚTEIS.....	27
6	ANEXOS.....	28
6.1	FUNÇÕES PostgreSQL .....	28

Todos os direitos reservados e protegidos por Lei de Direito Autoral, de 19/02/1998, artigo 7º. Nenhuma parte desse material poderá ser reproduzida ou transmitida sem citação de autoria.

## 1 Introdução

A partir de agora estudaremos o uso de banco de dados em PHP.

## 2 Tipos de Banco de Dados (BD)

O PHP suporta conexões com diferentes bancos de dados, como: MySQL, PostgreSQL, Firebird/Interbase, DB2, Oracle, SQLServer etc. Para desenvolvermos os nossos scripts PHP com banco de dados, utilizaremos o PostgreSQL.

## 3 Conexão e manipulação do BD PostgreSQL

Lembre-se que o endereço para a conexão remota com o servidor do CTI é **200.145.153.175**. Para acessar os BDs e visualizar as tabelas utilizaremos o endereço <http://200.145.153.175/phppgadmin>.

### 3.1 Conectar

A função para conexão do PostgreSQL com o PHP é **pg\_connect("parâmetros")**.

Devem ser informados como parâmetros o host (127.0.0.1 ou localhost), a porta de comunicação com o BD (a padrão é 5432), o nome do banco de dados, assim como, o usuário e senha do banco de dados. A string de conexão deve estar entre aspas duplas ("" ). Observe abaixo a string de conexão:

```
$conecta = pg_connect("host=127.0.0.1 port=5432 dbname=nome_do_banco_de_dados  
user=nome_do_usuario_do_banco password=senha_do_banco");
```

Exemplo – script de conexão do PHP com o PostgreSQL

Nome para salvar: **conexao\_explo.php**

```
1 <!DOCTYPE html>  
2 <html lang="pt-br">  
3   <head>  
4     <meta charset="utf-8" />  
5     <title>Conecta com BD PostgreSQL</title>  
6   </head>  
7   <body>  
8     <?php  
9       //Conecta com o PostgreSQL  
10      $conecta = pg_connect("host=localhost port=5432 dbname=testeDB1  
11                          user=x password=y");  
12      if (!$conecta)  
13      {  
14        echo "Não foi possível estabelecer conexão com o banco de dados!<br><br>";  
15        exit;  
16      }
```

```

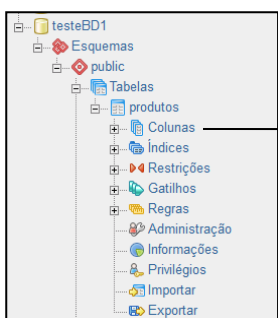
17     else
18         echo "Conexão estabelecida com o banco de dados!<br><br>";
19     ?>
20 </body>
21 </html>

```

**Observação:** em nosso caso, os bancos de dados utilizados serão criados pelos professores responsáveis da disciplina e “doados” para que vocês possam fazer uso com o login e a senha pré-estabelecidos, pois o acesso é realizado somente no servidor e não local.

### 3.2 Inserir

Para inserir dados numa tabela podemos utilizar a instrução *insert* e preparar a string SQL no PHP. Após a execução da string utilizaremos as funções **pg\_query(string\_de\_conexão, string\_sql)**, que faz uma consulta no banco de dados, e **pg\_affected\_rows(resultado\_da\_consulta)**, que retorna o número de registros afetados. O exemplo abaixo faz uma inserção no BD de nome testeBD1 que está no nosso servidor e contém a tabela PRODUTOS, descrita na sequência.



Coluna	Tipo	Não nulo	Padrão	Restrições
cod_produto	integer	NOT NULL	nextval('produtos_cod_produto_seq'::regclass)	KEY
descricao	character varying(80)	NOT NULL		
qtde	integer			
preco	money			
cod_fornecedor	integer			

Antes da execução do código abaixo, a tabela pode conter alguns registros já inseridos ou estar vazia:

cod_produto	descricao	qtde	preco	cod_fornecedor
358	Café Azul	10	R\$ 578,00	10
359	Chocolate Bis	50	R\$ 310,00	12
360	Molho de tomate	100	R\$ 375,00	10

Alguns detalhes das funções do PHP que serão utilizadas no exemplo de inserção:

**pg\_query(string\_de\_conexão, string\_sql)**<sup>1</sup>:

- Retorna um recurso (resource) de resultado da consulta (query) como TRUE se a consulta pôde ser executada. Retorna FALSE em caso de falha ou se a conexão não é uma conexão válida;
- Detalhes sobre os erros podem ser recuperados usando a função `pg_last_error()` se a conexão é válida;

<sup>1</sup> The PHP Group. Disponível em: <[http://php.net/manual/pt\\_BR/function.pg-query.php](http://php.net/manual/pt_BR/function.pg-query.php)>. Acesso em: 09 ago. 2014.

- `pg_query()` envia uma declaração SQL para o banco de dados PostgreSQL especificado pelo recurso de conexão `connection`.

**`pg_affected_rows(resultado_da_consulta)`<sup>2</sup>:**

- Retorna o número de linhas (instâncias/registros/linhas/tuplas) afetadas por consultas (queries) `INSERT`, `UPDATE` e `DELETE` executados por `pg_query()`. Se nenhuma linha foi afetada, ela retornará 0.

Exemplo – script de inserção de dados pelo código PHP, sem formulário HTML

Nome para salvar: `insere_exemplo.php`

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta charset="utf-8" />
5     <title>Insere no BD do PostgreSQL</title>
6   </head>
7   <body>
8     <?php
9       include "conexao_explo.php"; //O include anexa o script de conexão aqui!
10
11       /*como exemplo fizemos abaixo uma atribuição de valores, mas, obviamente,
12         esses dados serão enviados de um formulário */
13       $descricao='Açucar';
14       $qtde=33;
15       $preco='2,50'; //utilize vírgula, formato brasileiro, e entre apóstrofes
16       $cod_fornecedor=1;
17       $excluido='n';
18
19       // abaixo, string sql:
20       $sql="insert into produtos
21         values(nextval('produtos_cod_produto_seq'::regclass),
22               '$descricao',
23               $qtde,
24               '$preco',
25               $cod_fornecedor,
26               '$excluido')";
27       /* A expressão: nextval('produtos_cod_produto_seq'::regclass)
28          foi utilizada porque o campo é autoincremento. */
29
30       $resultado=pg_query($conecta,$sql);
31       $linhas=pg_affected_rows($resultado);
32       if ($linhas > 0)
33         echo "Produto gravado !!!<br><br>";
34       else
35         echo "Erro na gravação do produto !!!<br><br>";
36       // Fecha a conexão com o PostgreSQL
37       pg_close($conecta);
38     ?>
39   </body>
```

<sup>2</sup> The PHP Group. Disponível em: <[http://php.net/manual/pt\\_BR/function.pg-affected-rows.php](http://php.net/manual/pt_BR/function.pg-affected-rows.php)>. Acesso em: 09 ago. 2014.

40 </html>

Após a execução, o registro da cláusula SQL foi inserido na tabela:

Registro Gravado !!!

cod_produto	descricao	qtde	preco	cod_fornecedor
358	Café Azul	10	R\$ 578,00	10
359	Chocolate Bis	50	R\$ 310,00	12
360	Molho de tomate	100	R\$ 375,00	10
368	Açúcar	33	R\$ 2,50	1

A entrada de dados através de um formulário HTML deixa a aplicação mais elegante. Então, os exemplos a seguir mostram um formulário HTML que chama um script PHP para executar a cláusula SQL. O resultado da execução é apresentado a seguir:

**Cadastro Tabela Produtos - BD testeDB1**

Descrição:

Quantidade:

Preço:

Código do Fornecedor:

Registro Gravado !!!

cod_produto	descricao	qtde	preco	cod_fornecedor
358	Café Azul	10	R\$ 578,00	10
359	Chocolate Bis	50	R\$ 310,00	12
360	Molho de tomate	100	R\$ 375,00	10
368	Açúcar	33	R\$ 2,50	1
370	Chá Mate Leão	22	R\$ 3,15	1

Exemplo – formulário HTML para inserção de dados tabela Produtos – banco testeBD1

Nome para salvar: **form\_inserere\_prod.html**

```

1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta charset="utf-8" />
5     <title>Cadastro Tabela Produtos - banco testeDB1</title>
6   </head>
7   <body>
8     <h1>Cadastro Tabela Produtos - DB testeBD1</h1>
9     <form action="insere_prod.php" method="post">
10       <label>
11         <strong>Descrição:</strong><br />
12         <input type="text" name="descricao" /><br />
13       </label>

```

```

14     </label>
15
16     <label>
17         <strong>Quantidade:</strong><br />
18         <input type="text" name="qtde" /><br />
19         <br />
20     </label>
21
22     <label>
23         <strong>Preço:</strong><br />
24         <input type="text" name="preco" /><br />
25         <br />
26     </label>
27
28     <label>
29         <strong>Código do Fornecedor:</strong><br />
30         <input type="text" name="cod_fornecedor" /><br />
31         <br />
32     </label>
33
34     <input type="submit" name="button" id="button" value="Enviar" />
35 </form>
36 </body>
37 </html>

```

Exemplo - script de inserção chamado pelo formulário para inserir dados na tabela Produtos

Nome para salvar: **insere\_prod.php**

```

1 <!DOCTYPE html>
2 <html lang="pt-br">
3     <head>
4         <meta charset="utf-8" />
5         <title>Insere dados no banco testeDB1 do PostgreSQL</title>
6     </head>
7     <body>
8         <?php
9             include "conexao_explo.php"; //O include anexa o script de conexão aqui!
10
11             /*Dados enviados do formulário form_insere_prod.html*/
12             $descricao=$_POST['descricao'];
13             $qtde=$_POST['qtde'];
14             $preco=$_POST['preco'];
15             $cod_fornecedor=$_POST['cod_fornecedor'];
16             $excluido='n';
17
18             // abaixo, string sql:
19             $sql="insert into produtos
20                 values(nextval('produtos_cod_produto_seq'::regclass),
21                     '$descricao',
22                     $qtde,
23                     '$preco',
24                     $cod_fornecedor,
25                     '$excluido')";
26             /* A expressão: nextval('produtos_cod_produto_seq'::regclass)
27             foi utilizada porque o campo é autoincremento. */

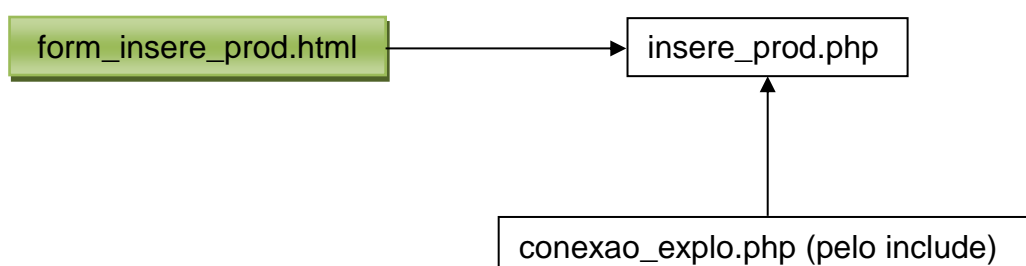
```

```

28
29     $resultado=pg_query($conecta,$sql);
30     $linhas=pg_affected_rows($resultado);
31     if ($linhas > 0)
32         echo "Produto gravado !!!<br><br>";
33     else
34         echo "Erro na gravação do produto!!!<br><br>";
35     // Fecha a conexão com o PostgreSQL
36     pg_close($conecta);
37     ?>
38 </body>
39 </html>

```

Esquema gráfico dos arquivos utilizados:



**Exercícios** (continua a numeração da Apostila 1 de PHP)

66) Considere o banco de dados identificado pela sua turma que está criado no servidor. Ele possui uma única tabela descrita abaixo. Faça uma aplicação que receba os dados do filme e insira um registro na tabela.

**Bancos criados e doados:** cinema\_72a, cinema\_72b, cinema\_72c, cinema\_72d

**Nome da tabela:** filme

**Colunas (campos):**

Coluna	Tipo	Não nulo	Padrão	Restrições
id_filme	integer	NOT NULL	nextval('filme_id_filme_seq'::regclass)	
nome_filme	character varying(80)			
pais_origem	character varying(30)			
genero	character varying(15)			
sinopse	character varying(300)			
classificacao	character varying(10)			
duracao	integer			
exclusao	character varying(1)			

**Scripts =>** conecta\_cinema.php  
insere\_filmes.php

**Form =>** form\_inserere\_filmes.html



67) Considere o banco de dados identificado pela sua turma que está criado no servidor. Ele possui uma única tabela descrita abaixo. Faça uma aplicação que receba os dados do país e insira um registro na tabela.

**Bancos criados e doados:** mundo\_72a, mundo\_72b, mundo\_72c, mundo\_72d

**Nome da tabela:** pais

**Colunas (campos):**

Coluna	Tipo	Não nulo	Padrão	Restrições
id_pais	integer	NOT NULL	nextval('pais_id_pais_seq'::regclass)	
nome_pais	character varying(50)			
capital_pais	character varying(40)			
continente	character varying(25)			
area	integer			
populacao_pais	numeric			
caracteristica	character varying(20)			
descricao	character varying(70)			
exclusao	character varying(1)			

**Scripts =>** conecta\_mundo.php  
insere\_mundo.php

**Form =>** form\_insere\_mundo.html

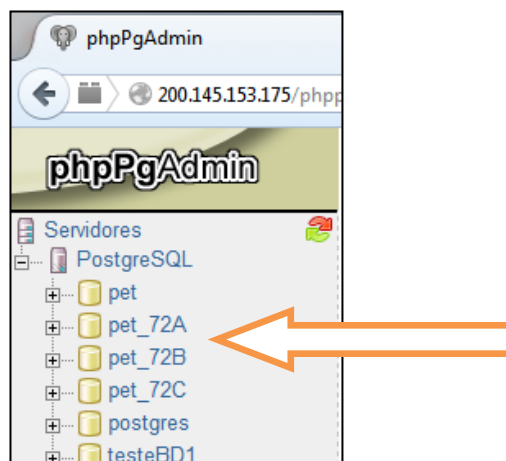
O campo **caracteristica** da tabela deve aparecer no form HTML em um select, com as seguintes opções: Comida típica, Ponto turístico, Idioma, Esporte, Curiosidade, para que apenas uma opção seja selecionada.

O campo **descricao** da tabela deve ser um texto que descreverá a característica escolhida no select. Por exemplo: característica "Comida típica", descrição "Feijoada".

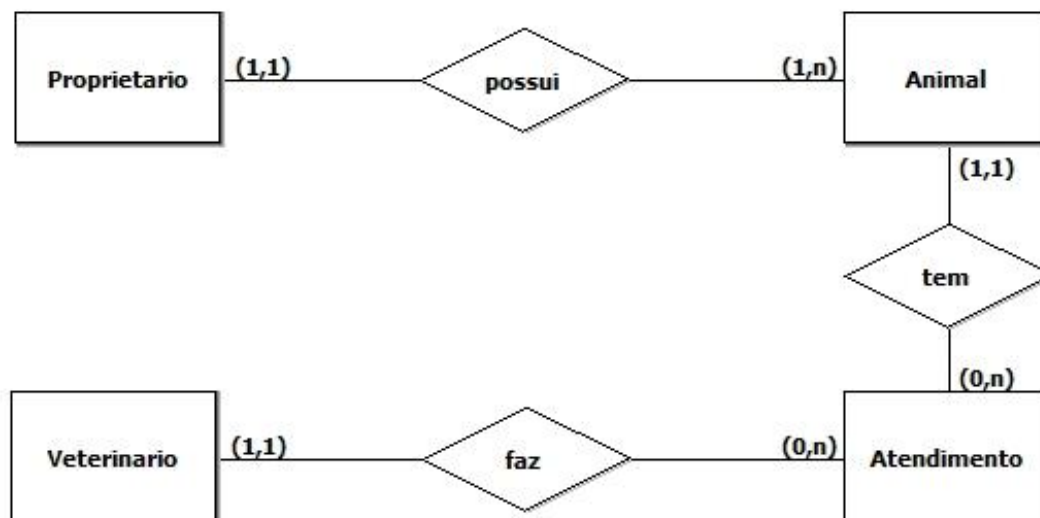
**Atividades prática de um banco com várias tabelas:**

Criação do banco de dados para as atividades propostas.

Os BDs foram criados pelos administradores e vocês poderão acessá-los. Trabalharemos com um para cada turma, da forma como aparece na figura abaixo. Referenciaremos somente "pet" e cada um deverá fazer uso do correspondente da sua turma.



O modelo conceitual de dados do banco pet (pet\_72A, pet\_72B, pet\_72C, pet72D) está representado no DER descrito abaixo:



Os *layouts* das tabelas são apresentados abaixo:



*Layout da Tabela Proprietario:*

Coluna	Tipo	Não nulo	Padrão	Restrições
id_prop	integer	NOT NULL	nextval('proprietario_id_prop_seq'::regclass)	
nome_prop	character varying(50)	NOT NULL		
endereco_prop	character varying(60)			
telefone_prop	character varying(15)			
cidade_prop	character varying(20)			
estado_prop	character varying(2)			
excluido	character varying(1)			
data_exclusao	date			

*Layout da Tabela Veterinario:*





Coluna	Tipo	Não nulo	Padrão	Restrições
id_vet	integer	NOT NULL	nextval('veterinario_id_vet_seq'::regclass)	
nome_vet	character varying(50)	NOT NULL		
telefone_vet	character varying(15)			
datanasc_vet	date			
crmvet_vet	character varying(20)			
excluido	character varying(1)			
data_exclusao	date			

### Layout da Tabela Animal:

Coluna	Tipo	Não nulo	Padrão	Restrições
id_animal	integer	NOT NULL	nextval('animal_id_animal_seq'::regclass)	
nome_animal	character varying(50)	NOT NULL		
datanasc_animal	date			
raca_animal	character varying(30)			
id_prop	integer			
tipo_animal	character varying(10)			
valor_animal	money			
excluido	character varying(1)			
data_exclusao	date			

Chave estrangeira

### Layout da Tabela Atendimento:

Coluna	Tipo	Não nulo	Padrão	Restrições
id_animal	integer	NOT NULL		 
data_atend	date	NOT NULL		
id_vet	integer			
descricao_atend	character varying(200)			
preco_atend	money			
excluido	character varying(1)			
data_exclusao	date			

68) Faça uma função para conectar remotamente o BD pet.

Nome para salvar: **conexao\_pet72x.php** (onde x é a letra da sua turma)

69) Crie um formulário HTML5 para cadastrar os proprietários dos animais. Faça teste de consistência de dados para não causar nenhum erro na gravação, como nos dados que não devem ser nulos. Envie os dados para um programa PHP que fará a inserção dos mesmos no banco de dados. Retorne ao formulário HTML para inserir novo registro. Mantenha um botão (ou link) para encerrar a aplicação. O formulário abaixo é só para representar a solicitação. Faça o seu modelo.

Form: **form\_inserir\_prop.html**

Script: **insere\_prop.php**

### Cadastro Tabela Proprietario - BD pet

Nome:

Endereço:

Telefone:

Cidade:

Estado:

70) Crie um formulário HTML5 para cadastrar os animais. Faça teste de consistência de dados para não causar nenhum erro na gravação, como nos dados que não devem ser nulos. Atente para o fato de que existe uma chave estrangeira nesta tabela, que é o proprietário do animal.

Form: **form\_inserir\_animal.html**

Script: **insere\_animal.php**

**Cadastro Tabela Animal - BD pet**

Nome do seu pet:

Data de nascimento:

Raça:

Proprietário:

Tipo do animal:  
☐ Doador ☒ Comprado

Se comprado, valor:

**Data de nascimento:**  
 dd/mm/aaaa  
 setembro de 2014  

dom	seg	ter	qua	qui	sex	sáb
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

**Raça:**  
 Outro  
 Cães  
 Labrador  
 Poodle  
 Yorkshire Terrier  
 Beagle  
 Gatos  
 Persa  
 Siamês  
 Maine coon  
 Angorá  
 Outro

71) Crie um formulário HTML5 para cadastrar os veterinários. Faça teste de consistência de dados para não causar nenhum erro na gravação, como nos dados que não devem ser nulos. Mostre calendário para a data de nascimento. Envie os dados para um programa PHP que fará a inserção dos mesmos no banco de dados. Retorne ao formulário HTML para inserir novo registro. Mantenha um botão (ou link) para encerrar a aplicação.

Form: **form\_inserir\_vet.html**

Script: **insere\_vet.php**

72) Crie um formulário HTML5 para cadastrar atendimentos. Faça teste de consistência de dados para não causar nenhum erro na gravação, como nos dados que não devem ser nulos. Observe que a chave primária é composta. Atente também para as duas chaves estrangeiras. Mostre calendário para a data de atendimento. Envie os dados para um programa PHP que fará a inserção dos mesmos no banco de dados. Retorne ao formulário HTML para inserir novo registro. Mantenha um botão (ou link) para encerrar a aplicação.

Form: **form\_inserir\_atend.html**

Script: **insere\_atend.php**

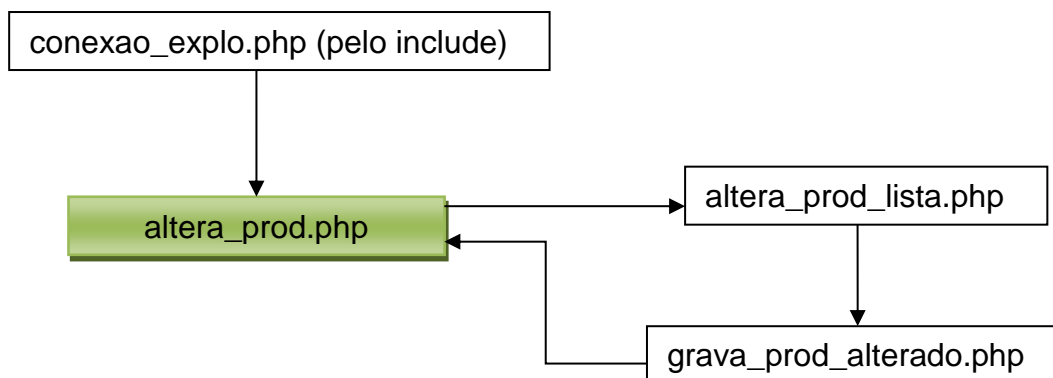
### 3.3 Alterar

Para alterar dados numa tabela podemos utilizar a instrução *update* e preparar a string SQL no PHP. No exemplo apresentado abaixo, uma lista dos registros armazenados na tabela é mostrada no navegador para que o usuário possa escolher que dados serão alterados.

O usuário escolhe quais dados deseja alterar clicando no link "Alterar" abaixo de cada registro. A tela de execução do exemplo vem a seguir:

Registros Encontrados	
Código do produto....:	358
Descrição.....:	Cafê Azul
Quantidade.....:	10
Preço.....:	R\$ 578,00
Código do fornecedor:	10
	<a href="#">Alterar</a>
Código do produto....:	359
Descrição.....:	Chocolate Bis
Quantidade.....:	50
Preço.....:	R\$ 310,00
Código do fornecedor:	12
	<a href="#">Alterar</a>
Código do produto....:	360
Descrição.....:	Molho de tomate

Esquema gráfico dos arquivos utilizados:



Exemplo – script de alteração de dados; uso de *include* "conexao\_explo.php"; link para *altera\_prod\_lista.php*

Nome para salvar: **altera\_prod.php**

```

1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta charset="utf-8" />
5     <title>Altera no BD exemplo do PostgreSQL</title>
6   </head>
7   <body>
8     <?php
9       include "conexao_explo.php";
10      $sql="select * from produtos where excluido='n' order by cod_produto";
  
```

```

11 $resultado= pg_query($conecta, $sql);
12 $qtde=pg_num_rows($resultado);
13 if ($qtde > 0)
14 {
15     echo "Produtos Encontrados<br><br>";
16     for ($cont=0; $cont < $qtde; $cont++)
17     {
18         $linha=pg_fetch_array($resultado);
19         echo "Código do produto...: ".$linha['cod_produto']."<br>";
20         echo "Descrição.....: ".$linha['descricao']."<br>";
21         echo "Quantidade.....: ".$linha['qtde']."<br>";
22         echo "Preço.....: ".$linha['preco']."<br>";
23         echo "Código do fornecedor: ".$linha['cod_fornecedor']."<br>";
24
25         echo "<a href='altera_prod_lista.php?cod_produto=".$linha['cod_produto']."'>
26 Alterar</a><br><br>";
27     }
28 }
29 else
30     echo "Não foi encontrado nenhum produto !!!<br><br>";
31 ?>
32 </body>
33 </html>

```

As funções utilizadas foram:

- **pg\_query(string\_de\_conexão, string\_sql)**: já foi descrita anteriormente;
- **pg\_num\_rows(resultado\_da\_consulta)**: retorna o número de linhas do resultado\_da\_consulta;
- **pg\_fetch\_array(resultado\_da\_consulta)**: retorna uma linha como um vetor (*array*), chamado de matriz associativa, com os nomes dos campos da tabela.

O array \$linha que foi criado possui a seguinte representação para dois registros do exemplo:

	cod_produto	descricao	qtde	preco	cod_fornecedor
cont =0	358	Café Azul	10	R\$ 578,00	10
cont=1	359	Chocolate Bis	50	R\$ 310,00	12

Após clicar, o script *altera\_prod\_lista.php* será chamado. O código é mostrado abaixo:

Exemplo – script com a lista de produtos disponíveis para alteração; link para *grava\_prod\_alterado.php*

Nome para salvar: **altera\_prod\_lista.php**

```

1 <?php
2     include "conexao_explo.php";
3     //dados enviados do script altera_prod.php
4     $cod_produto = $_GET["cod_produto"];

```

```

5  $sql="select * from produtos where cod_produto = $cod_produto";
6  $resultado=pg_query($conecta,$sql);
7  $qtde=pg_num_rows($resultado);
8  if ( $qtde == 0 )
9  {
10     echo "Produto nao encontrado !!!<br><br>";
11     exit;
12 }
13 $linha = pg_fetch_array($resultado);
14 ?>
15
16 <form action="grava_prod_alterado.php" method="post">
17     Código do produto <input type="text" name="cod_produto"
18         value="<?php echo $linha[cod_produto]; ?>" readonly><br>
19     Descrição <input type="text" name="descricao"
20         value="<?php echo $linha[descricao]; ?>" ><br>
21     Quantidade <input type="text" name="qtde_"
22         value="<?php echo $linha[qtde]; ?>" ><br>
23     Preço <input type="text" name="preco"
24         value="<?php echo $linha[preco]; ?>" ><br>
25     Código do fornecedor <input type="text" name="cod_fornecedor"
26         value="<?php echo $linha[cod_fornecedor]; ?>" ><br>
27     <input type="submit" value="Gravar">
28     <input type="reset" value="Voltar">
29 </form>

```

Código do produto	358
Descrição	Café Azul
Quantidade	10
Preço	R\$ 578,00
Código do fornecedor	10
<input type="button" value="Gravar"/> <input type="button" value="Voltar"/>	

Acima, o usuário escolheu o registro de código 358 para alterar. Todos os campos da tabela poderão sofrer alterações, exceto o campo Código do produto por ser chave primária. Para garantir que o Código do produto não seja alterado, foi adicionado o comando “*readonly*” na linha 18 do script. Após alterar o que for necessário, clicará no botão “Gravar” para enviar os novos dados para o servidor. O botão aciona o script *gravar\_prod\_alterado.php*. Se quiser restaurar os valores originais dos campos poderá clicar em “Limpar”. Abaixo, tela de execução com alteração dos dados:

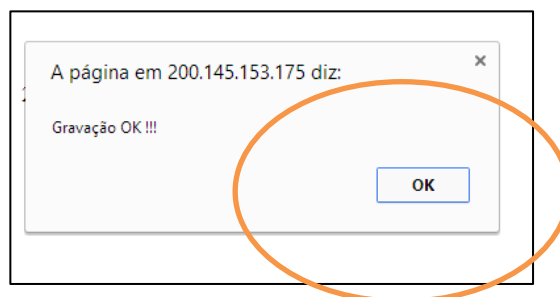
Código do produto	358
Descrição	Café Prata
Quantidade	8
Preço	R\$ 78,78
Código do fornecedor	10
<input type="button" value="Gravar"/> <input type="button" value="Voltar"/>	

Exemplo – script que salva na tabela as alterações efetuadas no registro; link para *altera\_prod.php*

Nome para salvar: **grava\_prod\_alterado.php**

```
1 <?php
2 include "conexao_explo.php";
3 //dados enviados do script altera_prod_lista.php
4 $cod_produto=$_POST["cod_produto"];
5 $descricao=$_POST["descricao"];
6 $qtde=$_POST["qtde_"];
7 $preco=$_POST["preco"];
8 $cod_fornecedor=$_POST["cod_fornecedor"];
9 $sql="update produtos
10     set
11         descricao = '$descricao',
12         qtde = $qtde_,
13         preco = '$preco',
14         cod_fornecedor = $cod_fornecedor
15     where cod_produto = $cod_produto";
16 $resultado=pg_query($conecta,$sql);
17 $qtde=pg_affected_rows($resultado);
18 if ($qtde > 0)
19 {
20     echo "<script type='text/javascript'>alert('Gravação OK !!!')</script>";
21     echo "<meta HTTP-EQUIV='refresh' CONTENT='0';URL='altera_prod.php'>";
22 }
23 else
24     echo "Erro na gravacao !!!<br><br>";
25 ?>
```

Em seguida, a tela de execução de confirmação da gravação bem sucedida, com javascript.



Ao clicar o botão “OK” é chamado novamente o script inicial de alteração, *altera\_prod.php*. Observe o registro alterado.



Código do produto....: 358
Descrição.....: Café Prata
Quantidade.....: 8
Preço.....: R\$ 78,78
Código do fornecedor: 10
<a href="#">Alterar</a>
Código do produto....: 359
Descrição.....: Chocolate Bis
Quantidade.....: 50
Preço.....: R\$ 310,00
Código do fornecedor: 12
<a href="#">Alterar</a>
Código do produto....: 360
Descrição.....: Molho de tomate
Quantidade.....: 100
Preço.....: R\$ 375,00
Código do fornecedor: 10
<a href="#">Alterar</a>

## Exercícios

- 73) Crie um formulário HTML5 que mostre todos os dados cadastrados na tabela animal. Abaixo de cada registro apresentado mostre na forma de link a opção Alterar. Ao clicar em alterar, remeta para outra tela onde o usuário poderá modificar todos os dados do registro, exceto a chave primária. Mostre na forma de link ou botão as opções Salvar e Limpar. Depois de executada quaisquer opções, volte para o script inicial e apresente todos os registros, incluindo os alterados, e reinicie o processo.
- 74) Crie um formulário HTML5 que mostre todos os dados cadastrados na tabela proprietario. Abaixo de cada registro apresentado mostre na forma de link a opção Alterar. Ao clicar em alterar, remeta para outra tela onde o usuário poderá modificar todos os dados do registro, exceto a chave primária. Mostre na forma de link ou botão as opções Salvar e Limpar. Depois de executada quaisquer opções, volte para o script inicial e apresente todos os registros, incluindo os alterados, e reinicie o processo.
- 75) Crie um formulário HTML5 que mostre todos os dados cadastrados na tabela veterinário. Abaixo de cada registro apresentado mostre na forma de link a opção Alterar. Ao clicar em alterar, remeta para outra tela onde o usuário poderá modificar todos os dados do registro, exceto a chave primária. Mostre na forma de link ou botão as opções Salvar e Limpar. Depois de executada quaisquer opções, volte para o script inicial e apresente todos os registros, incluindo os alterados, e reinicie o processo.
- 76) Desenvolva um formulário HTML5 que mostre todos os dados cadastrados na tabela atendimento. Abaixo de cada registro apresentado mostre na forma de link a opção Alterar. Ao clicar em alterar, remeta para outra tela onde o usuário poderá modificar todos os dados do registro, exceto a chave primária. Mostre na forma de link ou botão as opções Salvar e Limpar. Depois de executada quaisquer opções, volte para o script inicial e apresente todos os registros, incluindo os alterados, e reinicie o processo.

### 3.4 Excluir

Os scripts de exclusão de dados podem ser desenvolvidos a partir de dois métodos diferentes. O primeiro método trabalha com a exclusão física do registro da tabela, já o segundo, com a exclusão lógica.

#### ✓ Exclusão física

Esse tipo de exclusão faz a eliminação física de um registro previamente gravado na tabela do banco de dados. É necessário confirmar a exclusão com o usuário antes de efetivá-la a fim de evitar exclusões acidentais. Deve-se destacar que o registro a ser eliminado pode ter um ou mais campos da tabela vinculados a outros campos de outras tabelas (chave estrangeira). Quando isso ocorre, o procedimento de exclusão ocorrerá em cascata, que significa excluir também os registros associados das outras tabelas. Abaixo, trecho com exclusão física de registro:

```
...
include "conexao_explo.php";
$cod_produto = $_GET['cod_produto'];
$sql="delete from produtos where cod_produto = $cod_produto";
$resultado=pg_query($conecta,$sql);
$qtde=pg_affected_rows($resultado);
if ($qtde > 0 )
    echo "Produto excluído!!!";
else
    echo "Erro na exclusão do produto!!!";
...
```

#### ✓ Exclusão lógica

Na exclusão lógica o registro permanece fisicamente na tabela, mas um campo que representa a exclusão é setado como verdadeiro o que caracteriza a eliminação lógica do registro. Por isso, é preciso que cada tabela contenha um campo criado para essa finalidade, para representar se o registro “existe” naquele momento ou se já foi excluído. Pode ser um campo caractere que conterá o valor ‘S’, caso tenha sido excluído, ou o valor ‘N’, caso contrário. Pode ser um campo inteiro, seguindo a mesma lógica do campo caractere, com os valores 1 ou 0, por exemplo.

Utilizaremos o método de exclusão lógica em nossas atividades. Abaixo, trecho com exemplo:

```
...
include "conexao_explo.php";
$cod_produto = $_GET['cod_produto'];
$sql="update produtos
        set excluido = 's'
        where cod_produto = $cod_produto";
$resultado=pg_query($conecta,$sql);
$qtde=pg_affected_rows($resultado);
if ($qtde > 0 )
    echo "Produto excluído!!!";
else
    echo "Erro na exclusão do produto!!!";
...
```

A seguir, há uma sequência de scripts com as opções de alterar e excluir logicamente um produto, conforme imagem:

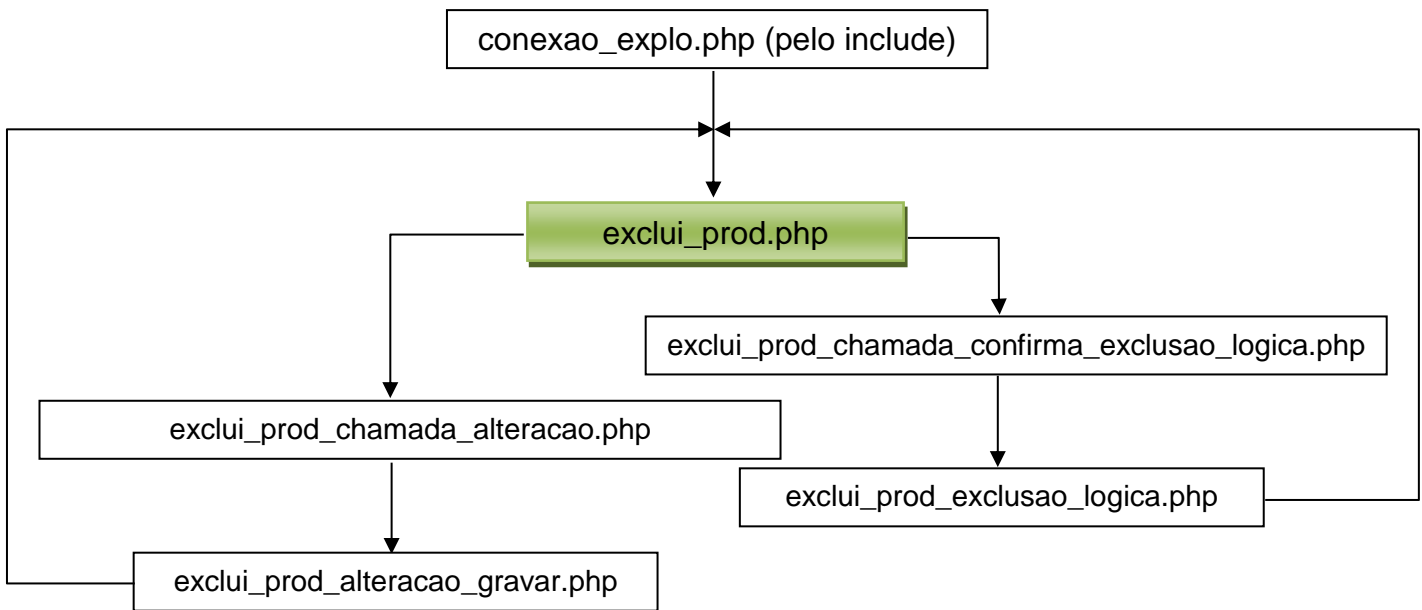
Produtos encontrados

Código do produto...: 5838  
 Descrição.....: Chocolate Bis  
 Quantidade.....: 30  
 Preço.....: R\$ 1,45  
 Código do fornecedor: 50  
[Alterar](#) [Excluir](#)

Código do produto...: 5839  
 Descrição.....: Produto Certo  
 Quantidade.....: 50  
 Preço.....: R\$ 12,50  
 Código do fornecedor: 10  
[Alterar](#) [Excluir](#)

Código do produto...: 5840  
 Descrição.....: Vinagre branco  
 Quantidade.....: 12  
 Preço.....: R\$ 3,77  
 Código do fornecedor: 2  
[Alterar](#) [Excluir](#)

Esquema gráfico dos arquivos utilizados:



### Exemplo – script de exclusão lógica e de alteração de dados

Nome para salvar: **exclui\_prod.php**

```

1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4 <meta charset="utf-8" />
5 <title>Exclusão lógica no BD exemplo</title>
6 </head>
7 <body>
8 <?php
9     include "conexao_explo.php";

10
11     $sql="select * from produtos
12           where excluido != 's'
13           order by cod_produto";

14
15     $resultado= pg_query($conecta, $sql);
16     $qtde=pg_num_rows($resultado);
17     if ($qtde > 0)
18     {
19         echo "Produtos encontrados<br><br>";
20         for ($cont=0; $cont < $qtde; $cont++)
21         {
22             $linha=pg_fetch_array($resultado);
23             echo "Código do produto...: ".$linha[cod_produto]."<br>";//ou $linha[0]
24             echo "Descrição.....: ".$linha[descricao]."<br>";
25             echo "Quantidade.....: ".$linha[qtde]."<br>";
26             echo "Preço.....: ".$linha[preco]."<br>";
27             echo "Código do fornecedor: ".$linha[cod_fornecedor]."<br>";

28
29             echo "<a href='exclui_prod_chamada_alteracao.php?cod_produto=$linha[0]'">
30             Alterar</a>&nbsp;&nbsp; ";
31

```

```

32         echo "<a
33 href='exclui_prod_chamada_confirma_exclusao_logica.php?cod_produto=$linha[0]'">
34         Excluir</a><br><br>";
35     }
36 }
37 else
38     echo "Não foi encontrado nenhum produto !!!<br><br>";
39 ?>
40 </body>
41 </html>

```

Após clicar o link *Excluir* para o produto “Chocolate Bis” é chamado o script para confirmação da exclusão. Todos os campos são apresentados com o atributo *readonly*.

Código do produto	5838
Descrição	Chocolate Bis
Quantidade	30
Preço	R\$ 1,45
Código do fornecedor	50
<input type="button" value="Confirma exclusão"/>	

Exemplo – script apresenta os dados na tela para confirmação da exclusão

Nome para salvar: **exclui\_prod\_chamada\_confirma\_exclusao\_logica.php**

```

1 <?php
2     include "conexao_explo.php";
3
4     //dados enviados do script exclui_prod.php
5     $cod_produto = $_GET["cod_produto"];
6
7     $sql="select * from produtos
8         where cod_produto = $cod_produto";
9
10    $resultado=pg_query($conecta,$sql);
11    $qtde=pg_num_rows($resultado);
12    if ( $qtde == 0 )
13    {
14        echo "Registro nao encontrado !!!<br><br>";
15        exit;
16    }
17    $linha = pg_fetch_row($resultado);
18 ?>
19 <form action="exclui_prod_exclusao_logica.php" method="post">
20     Código do produto <input type="text" name="cod_produto"
21         value="<?php echo $linha[0]; ?>" readonly><br>
22     Descrição <input type="text" name="descricao"
23         value="<?php echo $linha[1]; ?>" readonly><br>
24     Quantidade <input type="text" name="qtde_"
25         value="<?php echo $linha[2]; ?>" readonly><br>
26     Preço <input type="text" name="preco"
27         value="<?php echo $linha[3]; ?>" readonly><br>
28     Código do fornecedor <input type="text" name="cod_fornecedor"
29         value="<?php echo $linha[4]; ?>" readonly><br>
30     <input type="submit" value="Confirma exclusão">

```

31 &lt;/form&gt;

Após confirmação, é chamado novamente o script *exclui\_prod.php*. Observe que o produto excluído não aparece.

```
Produtos encontrados

Código do produto...: 5839
Descrição.....: Produto Certo
Quantidade.....: 50
Preço.....: R$ 12,50
Código do fornecedor: 10
Alterar Excluir

Código do produto...: 5840
Descrição.....: Vinagre branco
Quantidade.....: 12
Preço.....: R$ 3,77
Código do fornecedor: 2
Alterar Excluir
```

Exemplo – script que faz a exclusão lógica após confirmação

Nome para salvar: **exclui\_prod\_exclusao\_logica.php**

```
1 <?php
2     include "conexao_explo.php";
3
4     //dados enviados do script exclui_prod_chamada_confirma_exclusao_logica.php
5     $cod_produto = $_POST['cod_produto'];
6     $data=date('d/m/Y'); //Y' (maiusculo) para ano com 4 dígitos
7
8     $sql="update produtos
9         set excluido = 's', data_exclusao = '$data'
10        where cod_produto = $cod_produto";
11        //inserida a data de exclusao do produto para documentação
12
13    $resultado=pg_query($conecta,$sql);
14    $qtde=pg_affected_rows($resultado);
15    if ($qtde > 0 )
16    {
17        echo "<script type='text/javascript'>alert('Exclusão OK !!!')</script>";
18        echo "<meta HTTP-EQUIV='refresh' CONTENT='0';URL=exclui_prod.php>";
19    }
20    else
21    {
22        echo "Erro na exclusão !!!<br>";
23        echo "<a href='exclui_prod.php'>Voltar</a>";
24    }
25 ?>
```

Se o usuário optar por clicar o link *Alterar* para o produto “Vinagre branco”, é chamado o script para alteração dos dados. Somente o campo código do produto é *readonly*.

Exemplo – script apresenta os dados na tela para alteração e envio dos dados

Nome para salvar: **exclui\_prod\_chamada\_alteracao.php**

```

1 <?php
2     include "conexao_explo.php";
3
4     //dados enviados do script exclui_prod.php
5     $cod_produto = $_GET["cod_produto"];
6
7     $sql="select * from produtos
8         where cod_produto = $cod_produto";
9
10    $resultado=pg_query($conecta,$sql);
11    $qtde=pg_num_rows($resultado);
12    if ( $qtde == 0 )
13    {
14        echo "Produto não encontrado !!!<br><br>";
15        exit;
16    }
17    $linha = pg_fetch_row($resultado);
18 ?>
19 <form action="exclui_prod_chamada_alteracao_gravar.php" method="post">
20     Código do produto <input type="text" name="cod_produto"
21         value="<?php echo $linha[0]; ?>" readonly><br>
22     Descrição <input type="text" name="descricao"
23         value="<?php echo $linha[1]; ?>" ><br>
24     Quantidade <input type="text" name="qtde_"
25         value="<?php echo $linha[2]; ?>" ><br>
26     Preço <input type="text" name="preco"
27         value="<?php echo $linha[3]; ?>" ><br>
28     Código do fornecedor <input type="text" name="cod_fornecedor"
29         value="<?php echo $linha[4]; ?>" ><br>
30     <input type="submit" value="Salvar">
31 </form>

```

## Produtos encontrados

Código do produto...: 5839  
Descrição.....: Produto Certo  
Quantidade.....: 50  
Preço.....: R\$ 12,50  
Código do fornecedor: 10  
[Alterar](#) [Excluir](#)

Código do produto...: 5840  
Descrição.....: Vinagre branco de álcool  
Quantidade.....: 15  
Preço.....: R\$ 3,85  
Código do fornecedor: 2  
[Alterar](#) [Excluir](#)

Exemplo – script armazena no BD os dados alterados

Nome para salvar: **exclui\_prod\_chamada\_alteracao\_gravar.php**

```
1 <?php
2     include "conexao_explo.php";
3
4     //dados enviados do script exclui_prod_chamada_alteracao.php
5     $cod_produto=$_POST["cod_produto"];
6     $descricao=$_POST["descricao"];
7     $qtde=$_POST["qtde_"];
8     $preco=$_POST["preco"];
9     $cod_fornecedor=$_POST["cod_fornecedor"];
10
11     $sql="update produtos
12
13         set
14         descricao = '$descricao',
15         qtde = $qtde_,
16         preco = '$preco',
17         cod_fornecedor = $cod_fornecedor
18         where cod_produto = $cod_produto";
19
20     $resultado=pg_query($conecta,$sql);
21     $qtde=pg_affected_rows($resultado);
22     if ($qtde > 0)
23     {
24         echo "<script type='text/javascript'>alert('Gravação OK !!!')</script>";
25         echo "<meta HTTP-EQUIV='refresh' CONTENT='0;URL=exclui_prod.php'>";
26     }
27     else
28     {
29         echo "Erro na exclusao !!!<br>";
30         echo "<a href='exclui_prod.php'>Voltar</a>";
31     }
32 ?>
```



### Exercícios (somente exclusão lógica)

- 77) Crie um formulário HTML5 que mostre todos os dados cadastrados na tabela animal. Abaixo de cada registro apresentado mostre na forma de link as opções Alterar e Excluir. Ao clicar em alterar, remeta para outra tela onde o usuário poderá modificar todos os dados do registro, exceto a chave primária. Ao clicar em excluir, remeta para outra tela onde o usuário visualiza todos os dados do registro e confirmará a exclusão. Depois de executada quaisquer opções, volte para o script inicial e apresente todos os registros, incluindo os alterados e eliminados os excluídos, e reinicie o processo.
- 78) Crie um formulário HTML5 que mostre todos os dados cadastrados na tabela proprietario. Abaixo de cada registro apresentado mostre na forma de link as opções Alterar e Excluir. Ao clicar em alterar, remeta para outra tela onde o usuário poderá modificar todos os dados do registro, exceto a chave primária. Ao clicar em excluir, remeta para outra tela onde o usuário visualiza todos os dados do registro e confirmará a exclusão. Depois de executada quaisquer opções, volte para o script inicial e apresente todos os registros, incluindo os alterados e eliminados os excluídos, e reinicie o processo.
- 79) Crie um formulário HTML5 que mostre todos os dados cadastrados na tabela veterinario. Abaixo de cada registro apresentado mostre na forma de link as opções Alterar e Excluir. Ao clicar em alterar, remeta para outra tela onde o usuário poderá modificar todos os dados do registro, exceto a chave primária. Ao clicar em excluir, remeta para outra tela onde o usuário visualiza todos os dados do registro e confirmará a exclusão. Depois de executada quaisquer opções, volte para o script inicial e apresente todos os registros, incluindo os alterados e eliminados os excluídos, e reinicie o processo.
- 80) Crie um formulário HTML5 que mostre todos os dados cadastrados na tabela atendimento. Abaixo de cada registro apresentado mostre na forma de link as opções Alterar e Excluir. Ao clicar em alterar, remeta para outra tela onde o usuário poderá modificar todos os dados do registro, exceto a chave primária. Ao clicar em excluir, remeta para outra tela onde o usuário visualiza todos os dados do registro e confirmará a exclusão. Depois de executada quaisquer opções, volte para o script inicial e apresente todos os registros, incluindo os alterados e eliminados os excluídos, e reinicie o processo.

#### 4 REFERÊNCIAS BIBLIOGRÁFICAS

BENTO, E. J. **Desenvolvimento web com PHP e MySQL**, Ed. Casa do Código. Disponível em: <[http://issuu.com/hebertcoelho/docs/desenvolvimento\\_web\\_com\\_php\\_e\\_mysql](http://issuu.com/hebertcoelho/docs/desenvolvimento_web_com_php_e_mysql)>. Acesso em: 08 jul. 2015.

BRUNO, O. M.; ESTROZI, L. F.; BATISTA NETO, J. E. S. **Programando para a internet com PHP**, Rio de Janeiro: Brasport, 2010.

FLATSCHART, F. **HTML5: embarque imediato**, Rio de Janeiro: Brasport, 2011.

SAVOIA, H. R. **HTML e CSS + PHP e MySQL**, Ribeirão Preto, SP: IELD, 2013.

**The PHP Group**. Disponível em <<http://www.php.net/>>. Acesso em: 18 dez. 2013.

**W3C**. Disponível em <<http://www.w3.org/>>. Acesso em: 18 dez. 2013.

## 5 SITES ÚTEIS

<http://www.w3c.br/Home/WebHome>

<http://www.w3schools.com/>

[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

[http://validator.w3.org/#validate\\_by\\_input](http://validator.w3.org/#validate_by_input)

<http://jigsaw.w3.org/css-validator/>

<http://www.php.net/>

[http://www.php.net/manual/pt\\_BR/](http://www.php.net/manual/pt_BR/)

<http://codex.wiki.br/Html/Html>

<http://tableless.com.br/html5/>

<http://pt.wikipedia.org/wiki/HTML5>

<http://www.postgresql.org/docs/current/interactive/>

## 6 ANEXOS

### 6.1 FUNÇÕES PostgreSQL

[http://www.php.net/manual/pt\\_BR/funcref.php](http://www.php.net/manual/pt_BR/funcref.php)

#### CXXII. PostgreSQL

[pg\\_affected\\_rows](#) -- Retorna o número de registros afetados (linhas)  
[pg\\_cancel\\_query](#) -- Cancela uma consulta (query) assíncrona  
[pg\\_client\\_encoding](#) -- Retorna a codificação do cliente  
[pg\\_close](#) -- Fecha uma conexão PostgreSQL  
[pg\\_connect](#) -- Abre uma conexão PostgreSQL  
[pg\\_connection\\_busy](#) -- Verifica se a conexão está ocupada ou não  
[pg\\_connection\\_reset](#) -- Reinicia uma conexão (reconecta)  
[pg\\_connection\\_status](#) -- Retorna o status da conexão  
[pg\\_convert](#) -- Converte os valores de um array associativo em uma declaração SQL apropriada.  
[pg\\_copy\\_from](#) -- Insere registros em uma tabela a partir de um array  
[pg\\_copy\\_to](#) -- Copia uma tabela em um array  
[pg\\_dbname](#) -- Retorna o nome do banco de dados  
[pg\\_delete](#) -- Remove registros.  
[pg\\_end\\_copy](#) -- Sincroniza com o backend PostgreSQL  
[pg\\_escape\\_bytea](#) -- Gera binários para o tipo bytea  
[pg\\_escape\\_string](#) -- Gera string para o tipo text/char  
[pg\\_execute](#) -- Sends a request to execute a prepared statement with given parameters, and waits for the result.  
[pg\\_fetch\\_all\\_columns](#) -- Fetches all rows in a particular result column as an array  
[pg\\_fetch\\_all](#) -- Retorna todas as linhas (registros) como um array  
[pg\\_fetch\\_array](#) -- Retorna uma linha (registro) como um array  
[pg\\_fetch\\_assoc](#) -- Retorna uma linha (registro) como uma matriz associativa  
[pg\\_fetch\\_object](#) -- Retorna uma linha (registro) como um objeto  
[pg\\_fetch\\_result](#) -- Retorna valores a partir de um recurso (resource) de resultado  
[pg\\_fetch\\_row](#) -- Retorna uma linha como um array enumerado  
[pg\\_field\\_is\\_null](#) -- Testa se o campo é **NULL**  
[pg\\_field\\_name](#) -- Retorna o nome de um campo  
[pg\\_field\\_num](#) -- Retorna o número do campo referido  
[pg\\_field\\_prtlen](#) -- Retorna o comprimento impresso  
[pg\\_field\\_size](#) -- Retorna o tamanho de armazenamento interno de um campo  
[pg\\_field\\_table](#) -- Returns the name or oid of the tables field  
[pg\\_field\\_type\\_oid](#) -- Returns the type ID (OID) for the corresponding field number  
[pg\\_field\\_type](#) -- Retorna o nome do tipo para o campo de nome correspondente  
[pg\\_free\\_result](#) -- Libera a memória usada pelo resultado  
[pg\\_get\\_notify](#) -- Gets SQL NOTIFY message  
[pg\\_get\\_pid](#) -- Gets the backend's process ID  
[pg\\_get\\_result](#) -- Retorna o resultado de uma consulta (query) assíncrona  
[pg\\_host](#) -- Retorna o nome da máquina associada a uma conexão  
[pg\\_insert](#) -- Insere um array na tabela  
[pg\\_last\\_error](#) -- Retorna a string da última mensagem de erro de uma conexão  
[pg\\_last\\_notice](#) -- Retorna a última notificação do servidor PostgreSQL  
[pg\\_last\\_oid](#) -- Retorna o último oid do objeto  
[pg\\_lo\\_close](#) -- Fecha um objeto grande (large object)  
[pg\\_lo\\_create](#) -- Cria um objeto grande (large object)  
[pg\\_lo\\_export](#) -- Exporta um objeto grande (large object) para um arquivo  
[pg\\_lo\\_import](#) -- Importa um objeto grande (large object) a partir de um arquivo  
[pg\\_lo\\_open](#) -- Abre um objeto grande (large object)  
[pg\\_lo\\_read\\_all](#) -- Lê um objeto grande (large object) inteiro e o envia diretamente para o navegador  
[pg\\_lo\\_read](#) -- Lê um objeto grande (large object)

[pg\\_lo\\_seek](#) -- Procura uma posição em um objeto grande (large object)  
[pg\\_lo\\_tell](#) -- Retorna a posição atual de um objeto grande (large object)  
[pg\\_lo\\_unlink](#) -- Remove um objeto grande (large object)  
[pg\\_lo\\_write](#) -- Escreve em um objeto grande (large object)  
[pg\\_meta\\_data](#) -- Retorna os metadados da tabela.  
[pg\\_num\\_fields](#) -- Retorna o número de campos  
[pg\\_num\\_rows](#) -- Retorna o número de linhas  
[pg\\_options](#) -- Retorna as opções associadas com a conexão  
[pg\\_parameter\\_status](#) -- Looks up a current parameter setting of the server.  
[pg\\_pconnect](#) -- Abre uma conexão PostgreSQL persistente  
[pg\\_ping](#) -- Faz um ping na conexão com o banco de dados  
[pg\\_port](#) -- Retorna o número da porta associada com a conexão.  
[pg\\_prepare](#) -- Submits a request to create a prepared statement with the given parameters, and waits for completion.  
[pg\\_put\\_line](#) -- Envia uma string terminada em NULL para o backend PostgreSQL  
[pg\\_query\\_params](#) -- Submits a command to the server and waits for the result, with the ability to pass parameters separately from the SQL command text.  
[pg\\_query](#) -- Executa uma consulta (query)  
[pg\\_result\\_error\\_field](#) -- Returns an individual field of an error report.  
[pg\\_result\\_error](#) -- Retorna a mensagem de erro associada a um resultado  
[pg\\_result\\_seek](#) -- Altera a posição do ponteiro interno de um recurso (resource) de resultado  
[pg\\_result\\_status](#) -- Retorna o status do resultado da consulta (query)  
[pg\\_select](#) -- Seleciona registros.  
[pg\\_send\\_execute](#) -- Sends a request to execute a prepared statement with given parameters, without waiting for the result(s).  
[pg\\_send\\_prepare](#) -- Sends a request to create a prepared statement with the given parameters, without waiting for completion.  
[pg\\_send\\_query\\_params](#) -- Submits a command and separate parameters to the server without waiting for the result(s).  
[pg\\_send\\_query](#) -- Envia uma consulta (query) assíncrona  
[pg\\_set\\_client\\_encoding](#) -- Define a codificação do cliente  
[pg\\_set\\_error\\_verbosity](#) -- Determines the verbosity of messages returned by [pg\\_last\\_error\(\)](#) and [pg\\_result\\_error\(\)](#).  
[pg\\_trace](#) -- Habilita rastreamento em uma conexão PostgreSQL  
[pg\\_transaction\\_status](#) -- Returns the current in-transaction status of the server.  
[pg\\_tty](#) -- Retorna o nome da tty associada com a conexão  
[pg\\_unescape\\_bytea](#) -- Faz escape binário para o tipo bytea  
[pg\\_untrace](#) -- Desabilita o rastreamento de uma conexão PostgreSQL  
[pg\\_update](#) -- Atualiza uma tabela.  
[pg\\_version](#) -- Returns an array with client, protocol and server version (when available)