

# **DESENVOLVIMENTO DE SOFTWARE PARA WEB**

## **CAPÍTULO 4 - PODEMOS DEIXAR NOSSAS PÁGINAS MAIS SEGURAS?**

Fernando Skackauskas Dias

# Introdução

Quando a internet se tornou popular, ninguém imaginava o impacto que esta tecnologia iria trazer para a humanidade. Os sistemas que são acessados pela web abrangem quase todas as tarefas da sociedade atual, por exemplo, a maioria dos serviços bancários, acesso aos serviços públicos e privados, inclusive fazer um curso superior inteiramente pela internet. Podemos perceber, portanto, a responsabilidade que carrega o desenvolvedor de um *software* para a *web*. Este *software* deve ser plenamente confiável, de fácil usabilidade, ser acessível a todos e com alto nível de segurança. Informações sigilosas transitam entre sistemas e são armazenados em bancos de dados espalhados pelo mundo. Informações de todas as naturezas são gravadas, consultadas, alteradas e excluídas pelos sistemas disponíveis na *web*.

Com isto, a questão da segurança se torna estratégica. Como tornar os *softwares* para *web* mais seguros? Como excluir registros com segurança? Como tornar os sistemas mais confiáveis? Como implementar regras de segurança e autenticação de usuários? Estas questões são de fundamental importância ao se desenvolver e implantar sistemas para a internet, visto a alta vulnerabilidade do tráfego dos dados na rede mundial.

Após a leitura deste capítulo você será capaz de responder estas questões e criar páginas *web* implantando os processos fundamentais de inclusão, exclusão, alteração e consulta. Além disto, o capítulo irá abordar os tópicos de sessão, validação e segurança das informações em um *software* para *web* e a utilização do CSS.

Bons estudos!

## 4.1 Desenvolvimento *web*: apagar registro no banco de dados

A grande maioria dos sistemas, sejam em plataforma *web* ou *desktop*, apresenta as quatro funcionalidades principais: incluir, excluir, alterar e consultar os registros em um banco de dados. Todas estas funções têm sua sintaxe própria e devem ser submetidos a um tratamento de validação. Por exemplo, somente se pode incluir um registro, caso seja verificado previamente que ele ainda não exista no banco de dados, caso contrário haverá uma redundância de informações, gerando inconsistência de informações. Neste tópico será abordada a exclusão de registros em um banco de dados de um sistema *web*. A exclusão é uma função bastante crítica, pois uma exclusão indevida pode acarretar erros graves de consistência e comprometer toda a base de dados. Portanto, como é executado o comando de exclusão de um registro? Qual a sua sintaxe? Estas questões serão apresentadas, discutidas e aprofundadas a seguir.

### 4.1.1 Delete

Um dos processos mais críticos na implantação de um *software* para a *web* é a exclusão de um registro. Caso uma exclusão seja realizada de forma não intencional, e não seja possível recuperar esta informação, pode acarretar graves problemas para o usuário. Como excluir registros com segurança? Quais são os comandos necessários para excluir um registro de um sistema *web*? Inicialmente, iremos estudar quais os comandos fundamentais para a exclusão de um registro, para compreender melhor as questões de segurança. Clique para acompanhar e conhecer esses comandos.

A exclusão de um registro em um banco de dados é realizada por meio de um comando SQL, que é uma linguagem de programação utilizada para manipular dados, tendo sua sintaxe própria.

SQL é sigla inglesa de “*Structured Query Language*” que significa, em português, Linguagem de Consulta Estruturada, uma linguagem padrão de gerenciamento de dados que interage com os principais bancos de dados baseados no modelo relacional (MYSQL, 2019).

O comando que exclui os dados é o **DELETE**. Este comando tem a seguinte sintaxe:

**DELETE FROM** <nome da tabela> **WHERE** (<condições>);

Neste comando deve-se indicar qual a tabela que será realizada a exclusão e - mas não necessariamente - o comando **WHERE** que é a cláusula que impõe uma condição sobre a execução do comando.

Vamos usar como exemplo a tabela abaixo (Tabela\_Estoque), para exercitar este comando. A tabela se refere a uma rede de lojas com filiais em quatro capitais. Ao final do ano, foi enviada uma tabela com o balanço final de estoque, contendo o nome da capital, quantidade de produtos e data de levantamento. Segue abaixo:

CIDADE	QUANTIDADE	DATA
Belo Horizonte	1.850	30/12/2018
Curitiba	1.950	31/12/2018
São Paulo	1.990	31/12/2018
Rio de Janeiro	1.750	29/12/2018

Quadro 1 - Balanço de estoque de quatro capitais de uma empresa fictícia.

Fonte: Elaborado pelo autor, 2019.

Vamos supor as seguintes situações, nas quais se pode usar o comando **DELETE** para manipular exclusões do arquivo. Clique para conferir.

- 

Excluir todos os registros do arquivo

**DELETE FROM** Tabela\_Estoque;

É importante ressaltar que a ausência de condição **WHERE** compreende que todos os registros serão excluídos.

- 

Excluir um registro específico, por exemplo a cidade de “Belo Horizonte”

**DELETE FROM** Tabela\_Estoque **WHERE** Cidade = “Belo horizonte”;

Após este comando, a tabela conterá os registros de Curitiba, São Paulo e Rio de Janeiro.

- 

Excluir mais de um registro, por exemplo a cidade de Belo Horizonte e Rio de Janeiro

**DELETE FROM** Tabela\_Estoque **WHERE** Cidade = “Belo horizonte” **AND** Cidade = “Rio de Janeiro”;

Neste caso, incluímos o comando **AND** como forma de executar mais de uma condição. Após este comando a tabela conterá os registros de Curitiba e São Paulo.

- 

Excluir uma faixa de registros, por exemplo todos os registros, nos quais a data de envio seja maior que 30/12/2018

**DELETE FROM** Tabela\_Estoque **WHERE** Data > “30/12/2018”;

Após este comando a tabela conterá os registros de Belo Horizonte e Rio de Janeiro. Isto porque o comando exclui os registros onde a data é maior que 30/12/2018 (Curitiba e São Paulo), mas não ela mesma. Para que sejam executados comandos de faixa onde o próprio limite da faixa seja considerado, usa-se os símbolos ">=" ou "<=" (maior ou igual, menor ou igual). No caso do comando acima, se consideramos a data de 30/12/2018, inclusive, teremos:

**DELETE FROM** Tabela\_Estoque **WHERE** Data >= "30/12/2018";

Após este comando a tabela conterá somente o registro do Rio de Janeiro. Isto porque o comando exclui os registros onde a data é **maior ou igual** a 30/12/2018.

- 

Excluir mais de uma faixa de registros, por exemplo todos os registros, nos quais a data de envio seja maior ou igual a 30/12/2018 e o estoque seja menor que 1900

**DELETE FROM** Tabela\_Estoque **WHERE** Data >= "30/12/2018" **AND** Quantidade < 1900;

Após este comando a tabela conterá os registros de Curitiba, São Paulo e Rio de Janeiro. Belo Horizonte foi excluído porque é o único registro onde a data é maior (ou igual) a 30/12/2018 e o estoque é menor que 1900.

Na sequência, vamos conhecer uma curiosidade.

## VOCÊ SABIA?



Na grande maioria dos sistemas, quase sempre é necessário realizar a declaração de uma variável, para inserir uma informação e recuperar o seu conteúdo, quando for necessário. Quando fazemos uma referência a uma variável *string*, devemos colocá-la entre aspas. Se a referência é para um numeral, não se coloca aspas.

Agora, iremos compreender como é realizado o comando SQL em uma página *web*. Para isso, precisamos conhecer os comandos fundamentais do PHP.

PHP (um acrônimo recursivo para PHP: *Hypertext Preprocessor*) é uma linguagem de *script* de código aberto e de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento *web* e que pode ser embutida dentro do HTML (PHP, 2019).

O uso dos comandos SQL por meio da linguagem PHP é necessário, porque as páginas HTML (linguagem de desenvolvimento dos *sites*), não interpreta os comandos SQL diretamente, precisando do PHP como forma de interpretação. Vamos estudar a sintaxe PHP, como se dá o acesso ao banco de dados e executar o comando DELETE.

Para o acesso ao banco de dados devem ser declaradas as variáveis. Serão declaradas as três variáveis para se acessar o banco de dados: o servidor (\$host), o usuário do banco de dados (\$user), a senha para acessar o banco (\$pass). O código, então, fica da seguinte maneira:

```
<?php
$host = "servidor_banco_de_dados.com.br";
$user = "usuario_banco_de_dados";
$pass = "senha_banco_de_dados";
?>
```

O próximo comando é o acesso efetivo ao banco de dados. Para isto, o PHP disponibiliza o comando “mysql\_connect”. O código fica da seguinte maneira.

```
<?php
$host = "servidor_banco_de_dados.com.br";
$user = "usuario_banco_de_dados";
$pass = "senha_banco_de_dados";
$conecta = mysql_connect($host, $user, $pass) or print (mysql_error());
?>
```

Analisando o código, vemos que o retorno da função “mysql\_connect()” é armazenado na variável \$conecta. O comando mysql\_connect usa os parâmetros necessários para o acesso ao banco. Na mesma linha de comando é introduzido o comando “or print” que exibe algum possível erro retornado pelo comando “mysql\_error()”.

## VOCÊ SABIA?



PHP (*Hypertext Preprocessor*) é uma das linguagens de programação mais usadas pelos desenvolvedores de sistemas e no desenvolvimento de páginas *web*. Você sabia que é possível escrever aplicações desktop com o PHP? É possível utilizar alguns dos seus recursos avançados nas aplicações cliente, com a versão para desktop PHP-GTK.

A seguir, vamos realizar o comando de acesso à base de dados. O código fica da seguinte maneira:

```
<?php
$host = "servidor_banco_de_dados.com.br";
$user = "usuario_banco_de_dados";
$pass = "senha_banco_de_dados";
$conecta = mysql_connect($host, $user, $pass) or print (mysql_error());
$db = "base_de_dados";
mysql_select_db($db, $conecta);
?>
```

Continuando a construção do código PHP para exclusão de um registro, vamos simular o exemplo usado no tópico anterior.

- Excluir mais de uma faixa de registros, por exemplo todos os registros, nos quais a data de envio seja maior ou igual 30/12/2018 e o estoque seja menor que 1900:

**DELETE FROM** Tabela\_Estoque **WHERE** Data >= “30/12/2018” **AND** Quantidade < 1900;

Neste caso, é necessário declarar uma variável que receberá o comando SQL. Após esta declaração, é utilizado o comando “mysql\_query()” para, efetivamente, realizar o processamento. O código fica da seguinte maneira:

```
<?php
$host = "servidor_banco_de_dados.com.br";
$user = "usuario_banco_de_dados";
$pass = "senha_banco_de_dados";
$conecta = mysql_connect($host, $user, $pass) or print (mysql_error());
$db = "base_de_dados";
mysql_select_db($db, $conecta);
$sql = "DELETE FROM Tabela_Estoque WHERE Data >= '30/12/2018' AND Quantidade < 1900;";
$query = mysql_query($sql);
?>
```

Portanto, vimos a construção de um código PHP e a sua interação com o comando **DELETE** do SQL.

No próximo tópico, iremos desenvolver uma página *web* com as quatro funções fundamentais: incluir, consultar, alterar e excluir registros, também conhecido como CRUD (*Create, Read, Update, Delete*). Iremos aprender os comandos fundamentais das linguagens HTML, PHP, SQL e JavaScript que trabalham de maneira interativa para que o sistema *web* funcione de forma eficiente.

## 4.2 Desenvolvimento de um CRUD utilizando PHP e Banco de Dados

Quando estamos construindo aplicativos, é necessário que os modelos forneçam quatro tipos básicos de funcionalidade. O modelo deve ser capaz de criar, ler, atualizar e excluir recursos. Este modelo é denominado CRUD (*Create, Read, Update, Delete* – Incluir, Consultar, Alterar e Excluir).

A implantação dos comandos CRUD pode ser realizada na construção de um site por meio da linguagem PHP, em interação com os comandos JavaScript. Isto se faz necessário pelo fato de uma página *web* ser síncrona e os comandos CRUD são assíncronos. Ou seja, quando estamos acessando uma página na *web*, os seus recursos são carregados automaticamente. Porém, quando realizamos uma operação em um banco de dados, não há como recarregar a página novamente, sendo, portanto necessário a utilização do PHP e JavaScript, de forma assíncrona.

### VOCÊ QUER VER?



A linguagem PHP é largamente utilizada para desenvolver, tanto sistemas *web*, quanto sistemas desktop. A palestra Programação funcional em PHP – saia da zona de conforto (SANTOS, 2016) demonstra o quanto o PHP é uma linguagem altamente funcional e que se adapta a qualquer plataforma, tendo recursos avançados que permitem desenvolver sistemas de alta qualidade e grande usabilidade. Disponível em <[https://www.youtube.com/watch?v=cts\\_NStWJ5Q](https://www.youtube.com/watch?v=cts_NStWJ5Q)>.

O paradigma CRUD é utilizado em todos os tipos de sistemas. Vamos supor uma estrutura de um registro de uma biblioteca com os seguintes componentes:

```
"book": {  
  "id": <Integer>,  
  "titulo": <String>,  
  "autor": <String>,  
  "codigo": <Integer>  
}
```

Para tornar esse sistema de biblioteca implantável, o sistema deve oferecer todos os recursos do CRUD. Clique nas abas e acompanhe a seguir.

Inclusão

Isso consistiria em uma função que chamaríamos quando um novo livro da biblioteca fosse adicionado ao catálogo. O programa chama a função iria fornecer os valores para “titulo”, “autor”, e “codigo”. Após esta função ser chamada, deve haver uma nova inclusão no banco de dados correspondente a este novo livro. Além disso, a nova entrada recebe um único código, que pode ser usado para acessar esse recurso posteriormente.

Consulta

Isso considera uma função que deve ser chamada para recuperar todos os livros, atualmente gravados no banco de dados. Essa chamada de função não alteraria os livros no catálogo - ela simplesmente recuperaria o recurso e exibiria os resultados.

#### Alteração

Esta função tem como objetivo atualizar todas as informações sobre um livro que foi alterado. O programa chama a função forneceria os novos valores para “titulo”, “autor”, e “codigo”.

#### Exclusão

Esta função serve para remover um registro da biblioteca do catálogo. O programa chama a função iria fornecer um ou mais valores do “id”, para identificar o livro, e em seguida, este livro seria retirado do banco de dados.

Veja, a seguir, uma ilustração do sistema na etapa de inclusão.



Figura 1 - O desenvolvimento de uma página web dinâmica compreende a interação entre os códigos HTML, JavaScript e PHP.

Fonte: Elaborada pelo autor, 2019.

## VOCÊ O CONHECE?



O cientista da computação John Backus (1924-2007) foi o criador da linguagem FORTRAN (uma das primeiras linguagens de programação) e da Programação Funcional. Ele fez a seguinte famosa declaração “na ciência e em todo trabalho de criação, nós falhamos repetidas vezes. Normalmente para cada ideia bem-sucedida há dúzias de outras que não funcionaram” (FONSECA FILHO, 2007, p. 162).

Os comandos do CRUD correspondem aos comandos **INSERT**, **SELECT**, **UPDATE** e **DELETE** do SQL, que são chamadas pelas instruções PHP. As páginas HTML interagem de forma assíncrona com as páginas PHP, nas quais são implementados os comandos CRUD a partir dos comandos SQL.

A seguir, iremos estudar as formas de segurança para implantar *softwares* na *web*.

## 4.3 Desenvolvimento *web*: autenticação de sistemas utilizando sessão

É cada vez mais fundamental desenvolver sistemas com segurança, principalmente quando estão disponíveis na *web*, devido à alta vulnerabilidade dos dados. É necessário assegurar que estão sendo utilizadas as melhores práticas de segurança para aplicações web é ter um plano traçado. Segundo Sommerville (2011, p. 6),

Segurança e confiança. Pelo fato de o *software* estar presente em todos os aspectos de nossas vidas, é essencial que possamos confiar nele. Isso se torna verdade especialmente para sistemas remotos acessados através de uma página *Web* ou uma interface de *web service*. Precisamos ter certeza de

que os usuários maliciosos não possam atacar nosso *software* e de que a proteção da informação seja mantida.

Uma boa prática é classificar as práticas de segurança em ordem de prioridade (crítico, importante, normal). Sem priorizar em quais aplicativos focar primeiro, será difícil fazer qualquer análise de segurança significativa.



Figura 2 - O desenvolvimento de sistemas para web é composto pela integração de diversas linguagens e componentes que interagem para compor o sistema como um todo.

Fonte: Shutterstock, 2019.

Segundo Sommerville (2011), uma prática importante é executar aplicativos usando o mínimo de privilégios. É útil sempre usar as configurações menos permissivas para todas as aplicações *web*. Isso significa que as aplicações devem ser “amarradas”. Somente pessoas com autorizações de alto nível devem poder fazer alterações no nível de sistema.

Vamos analisar, a seguir, as variáveis de sessão.

#### **4.3.1 Variáveis de sessão**

Em um site ou sistema *web*, a sessão é fundamental, quando se deseja mais segurança na página ou quando é necessário ter um controle de usuário. Alguns programadores utilizam este recurso para guardar informações para uso posterior no *software*. Veremos, a seguir, como fazer, passo-a-passo, a utilização das variáveis de sessão para um aplicativo *web*, em um código, usado como referência (MELHORWEB, s/d).

- **Formulário para autenticação**

Este formulário será usado para que um usuário se autentique no sistema. Se tentar acessar uma página do sistema sem estar autenticado será redirecionado para ela.

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



```

<label>Nome de usuário</label>
<input type="text" name="login" id="login" /><br />
<label>Senha</label>
<input type="password" name="senha" id="senha" /><br />
<input type="submit" value="Login" />
</form>

```

- **Página inicial**

Esta página analisará os dados enviados pelo formulário. Se estiver correto, irá criar uma sessão, senão será redirecionado para a página de autenticação login.php.

```

<?PHP
$login = $_POST['login'];
$senha = $_POST['senha'];
$conexao = mysql_connect($hostname, $username, $password) or die ("Erro");
$selecionabd = mysql_select_db($database,$conexao) or die ("Banco de dados inexistente.");
$sql = "SELECT * FROM usuário WHERE usuario = '$login' AND usuario = '$senha'";
$resultado = mysql_query($sql,$conexao) or die ("Erro na seleção da tabela.");
if (mysql_num_rows ($resultado) > 0) {
// session_start inicia a sessão
session_start();
$_SESSION['login'] = $login;
$_SESSION['senha'] = $senha;
}
else {
session_destroy();
unset ($_SESSION['login']);
unset ($_SESSION['senha']);
header('location:login.php');
}
?>

```

- **Código de verificação de sessão**

```

<?PHP
session_start();
if ( !isset($_SESSION['login']) and !isset($_SESSION['senha']) ) {
session_destroy();
unset ($_SESSION['login']);
unset ($_SESSION['senha']);
header('location:login.php');
}
?>

```

A seguir será realizada a sumarização com a utilização de CSS, validação JavaScript e Banco de Dados, por meio de um estudo de caso.

## 4.4 Sistematização final para a criação de um sistema web completo utilizando PHP e MySQL

Inicialmente, é necessário compreender o componente *Cascading Style Sheets* (CSS). O CSS determina como serão exibidos os elementos visuais e comportamentais de uma página HTML. A grande vantagem do ponto de vista da arquitetura de *software*, é a separação entre o formato e o conteúdo de um documento. Segundo Lemay ( *et al.*, 2002, p. 45):

embora o HTML não diga muito sobre como uma página se parece quando ela é visualizada, as CSS ( *Cascading Style Sheets*, folhas de estilo em cascata) permitem que você aplique uma formatação avançada às *tags* HTML.

Existe uma questão também relevante com relação ao CSS que é a aparência da página. Clique a seguir e acompanhe os pontos que levantamos sobre essa questão.

Aparência

Novos estilos de aparência mais elegantes de cores, formatos de fontes e *layout* com melhor visualização.

Programa recomendado

É mais recomendado utilizar CSS, em vez de terminar o estilo dentro da própria *tag* HTML. Isto evita repetição de comando, tornando o código mais coeso.

Exemplo

Por exemplo, tem-se três campos de um código HTML, nos quais são definidos os estilos na própria *tag*, sendo também é chamado de CSS *inline*.

Veja abaixo, um exemplo de código HTML com CSS *inline*.

```
<html>
<head>
<title>CSS – Disciplina de Desenvolvimento de Software para Web</title>
</head>
<body>
<p style="color:red; font-style: italic ">Texto 1 /p>
<p style="color:red; font-style: italic ">Texto 2 /p>
<p style="color:red; font-style: italic ">Texto 3 /p>
</body>
</html>
```

Agora, utilizando o CSS na mesma página, se faz a referência aos componentes somente uma vez, e não diversas vezes. Portanto, eliminamos a declaração "style" no corpo da *tag* do código HTML. Elas são declaradas de forma global, evitando, assim, inserir o estilo em cada *tag* específica. A seguir, vemos o mesmo exemplo dado acima, mas com o CSS, na mesma página.

```
<html>
<head>
<title> CSS – Disciplina de Desenvolvimento de Software para Web </title>
<style type="text/css">
p{
color:red; font-style: italic;
}
</style>
</head>
```

```

<body>
<p>Texto 1</p>
<p>Texto 2</p>
<p>Texto 3</p>
</body>
</html>

```

A declaração `p{}` dentro da cláusula `<style>`, irá tornar todas as tags “p” com o estilo determinado. Portanto, é possível perceber como o código ficou mais “limpo” sem a repetição do “style” em cada *tag*. Além disso, caso seja necessário alterar a cor ou a *font*, altera-se somente no CSS uma vez, e não três vezes no modelo CSS *inline*. Este procedimento faz parte do desenvolvimento de *software* com critérios de qualidade, sendo fundamental para o desenvolver *web* ter pleno conhecimento desses princípios.

#### **4.4.1. Sumarizar a utilização de CSS, validação JavaScript e Banco de Dados por meio de um estudo de caso**

Para estruturar nosso estudo de caso, vamos criar uma página HTML de Cadastro inicial de aluno em uma faculdade. Neste cadastro haverá a inclusão do nome do aluno, endereço, idade e curso. O primeiro passo a ser dado é compreender a estrutura de uma página HTML, vamos compreender, então, que

HTML, a *Hyper Text Markup Language* é uma linguagem de programação baseada em *tags* ou marcações que teve sua criação em 1990 por Tim Berners-Lee. Essas páginas ou textos são interpretadas por um navegador ou *browser* também chamado de cliente (GIROLDO; FRESSATI, 2015, p. 3).

O HTML é uma linguagem, pois é composto por um código e tem uma sintaxe como qualquer outra linguagem. A seguir é apresentado um exemplo de um código HTML de cadastro de aluno.

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
<title>Faculdade Virtual</title>
<style type="text/css">
h{
color:black; font-style: bold;
}
p{
color:black; font-style: bold;
}
font{
color:black; font-style: bold;
}
input{
color:black; font-style: bold;
}
</style>
</head>
<body>
<h1>Cadastro de Aluno</h1><p></p>
<p>Inclua as informações do aluno abaixo</p>
<font> Nome: </font><p></p>

```

```

<input type="text" id="nome"></input><p></p>
<font> Endereço: </font><p></p>
<input type="text" id="endereco"></input><p></p>
<font> Idade: </font><p></p>
<input type="text" id="idade"></input><p></p>
<font> Curso: </font><p></p>
<input type="text" id="idade"></input><p></p>
<input type="button" value="Incluir" id="botao_incluir">
<input type="button" value="Excluir" id="botao_excluir">
<input type="button" value="Alterar" id="botao_alterar">
<input type="button" value="Consultar" id="botao_consultar">
</body>
</html>

```

Observamos na figura a seguir como seria a exibição desta página em um navegador.

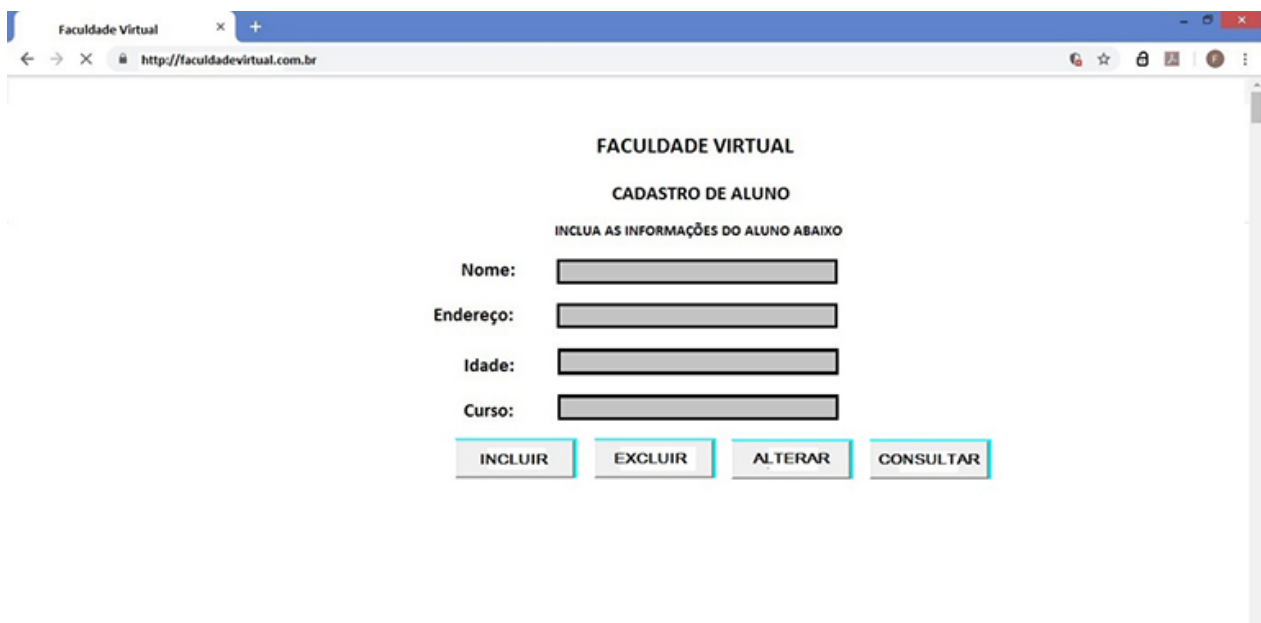


Figura 3 - Simulação da apresentação em um navegador de um sistema para web de uma faculdade virtual.

Fonte: Elaborada pelo autor, 2019.

Clique a seguir para continuar a elaboração do nosso cadastro de aluno.

Após o usuário efetuar a entrada dos dados, é necessária a inclusão de um comando, para que o programa “leia” as informações cadastradas e faça execute o comando (Incluir, Excluir, Alterar ou Consultar) no banco de dados. Para isso, para cada *tag* de *type*="button" deve ser incluído o atributo “onClick” direcionando para uma função que recupere as informações digitadas.

Ocorre que, para executar a função “onClick” é necessário que a página HTML interaja com os comandos JavaScript na *tag* <script>. Então, compreender como uma página HTML pode interagir com o <script>.

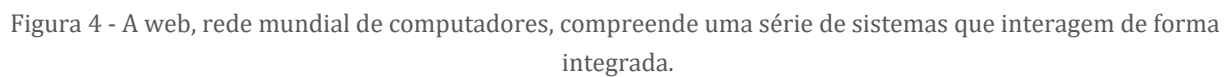
A <script> *tag* é usada para definir um *script* do lado do cliente (JavaScript). O <script> elemento contém instruções de *script* ou aponta para um arquivo de script externo, por meio do *src* atributo.

Usos comuns para JavaScript são manipulação de imagem, validação de formulário e alterações dinâmicas de conteúdo (W3SCHOOLS, 2019).

O script é escrito pela sintaxe <script> comandos JavaScript </script> e deverá residir na área do <head> do código HTML. As funções que são chamadas pelos comandos “onClick” do HTML deverão estar dentro do declaração function() {...}. A seguir o código com os comandos “onClick” de cada “button” e as respectivas declarações function() de cada comando.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<title>Faculdade Virtual</title>
<style type="text/css">
h{
color:black; font-style: bold;
}
p{
color:black; font-style: bold;
}
font{
color:black; font-style: bold;
}
input{
color:black; font-style: bold;
}
</style>
<script>
function Inclusao() {
}
function Exclusao() {
}
function Alteracao() {
}
function Consulta() {
}
</script>
</head>
<body>
<h1>Cadastro de Aluno</h1><p></p>
<p>Inclua as informações do aluno abaixo</p>
<font> Nome: </font><p></p>
<input type="text" id="nome"></input><p></p>
<font> Endereço: </font><p></p>
<input type="text" id="endereco"></input><p></p>
<font> Idade: </font><p></p>
<input type="text" id="idade"></input><p></p>
<font> Curso: </font><p></p>
<input type="text" id="idade"></input><p></p>
<input type="button" value="Incluir" id="botao_incluir" onClick="Inclusao()">
<input type="button" value="Excluir" id="botao_excluir" onClick="Exclusao()">
<input type="button" value="Alterar" id="botao_alterar" onClick="Alteracao()">
<input type="button" value="Consultar" id="botao_consultar" onClick="Consulta()">
</body>
```

Percebam que as `functions()` estão vazias. O próximo passo é executar os comandos SQL relativos ao CRUD. Neste momento devemos saber que os comandos SQL são interpretados por meio da linguagem PHP. Mas a página *web* é síncrona, ou seja, os processos são executados no momento de carregamento da página.



Então, no código HTML, usamos os comandos AJAX que fazem parte da linguagem JavaScript, que usaremos para carregar o código PHP e, efetivamente, executar os comandos CRUD no banco de dados. Toda esta interação entre linguagens se faz necessário porque, em uma aplicação *web*, o navegador busca as informações no servidor e retorna ao cliente, mas no Ajax ocorre de forma diferente. No carregamento da página, a lógica de processamento de dados é passada ao cliente. Quando o usuário emite uma requisição (`function()`), quem busca e traz essas informações é o JavaScript, de forma assíncrona, não causando o “*reload*” na tela.

## VOCÊ QUER VER?



O livro “HTML e CSS projete e construa *websites*” (DUCKETT, 2016) ensina de forma clara e concisa como utilizar CSS e HTML no desenvolvimento de páginas *web*. O uso de HTML e CSS torna o desenvolvimento de páginas web mais profissional e coesa, tornando o código com alto nível de qualidade. O livro aborda os temas de HTML e CSS com clareza e fácil de aplicar em sistemas *web*.

Outra informação importante neste ponto de desenvolvimento da nossa página é a leitura dos campos na tela (nome, endereço, etc.). Esta recuperação da informação é realizada por meio do comando `document.getElementById("id do campo").value`. Vamos mostrar como fica todo o código completo, incluindo as chamadas Ajax e JQuery, e o encaminhamento para o código PHP. Observe com detalhes todo o código.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<title>Faculdade Virtual</title>
<style type="text/css">
h{
color:black; font-style: bold;
}
p{
color:black; font-style: bold;
}
font{
color:black; font-style: bold;
}
input{
color:black; font-style: bold;
}
</style>
<script>
function Inclusao() {
var dadosajax = {
'nome_aluno': document.getElementById("nome").value;
'endereco_aluno': document.getElementById("endereco").value;
'idade_aluno': document.getElementById("idade").value;
'curso_aluno': document.getElementById("curso").value;
};
pageurl = 'incluir_aluno.php';
$.ajax({
url: pageurl,
data: dadosajax,
type: 'POST',
cache: false,
```

```

error: function(XMLHttpRequest, textStatus, errorThrown)
{
alert('Erro: Acessar Função PHP!!');
},
success: function(result)
{
alert('Inclusão realizada com sucesso!');
}
})
}
function Exclusao() {
var dadosajax = {
'nome_aluno': document.getElementById("nome").value;
};
pageurl = 'excluir_aluno.php';
$.ajax({
url: pageurl,
data: dadosajax,
type: 'POST',
cache: false,
error: function(XMLHttpRequest, textStatus, errorThrown)
{
alert('Erro: Acessar Função PHP!!');
},
success: function(result)
{
alert('Exclusão realizada com sucesso!');
}
})
}
function Alteracao() {
var dadosajax = {
'nome_aluno': document.getElementById("nome").value;
'endereco_aluno': document.getElementById("endereco").value;
'idade_aluno': document.getElementById("idade").value;
'curso_aluno': document.getElementById("curso").value;
};
pageurl = 'alterar_aluno.php';
$.ajax({
url: pageurl,
data: dadosajax,
type: 'POST',
cache: false,
error: function(XMLHttpRequest, textStatus, errorThrown)
{
alert('Erro: Acessar Função PHP!!');
},
success: function(result)
{

```



```

alert('Alteração realizada com sucesso!');
}
})
}
function Consulta() {
var dadosajax = {
'nome_aluno': document.getElementById("nome").value;
};
pageurl = 'consultar_aluno.php';
$.ajax({
url: pageurl,
data: dadosajax,
type: 'POST',
cache: false,
error: function(XMLHttpRequest, textStatus, errorThrown)
{
alert('Erro: Acessar Função PHP!!');
},
success: function(result)
{
alert('Consulta realizada com sucesso!!');
}
})
}
</script>
</head>
<body>
<h1>Cadastro de Aluno</h1><p></p>
<p>Inclua as informações do aluno abaixo</p>
<font> Nome: </font><p></p>
<input type="text" id="nome"></input><p></p>
<font> Endereço: </font><p></p>
<input type="text" id="endereço"></input><p></p>
<font> Idade: </font><p></p>
<input type="text" id="idade"></input><p></p>
<font> Curso: </font><p></p>
<input type="text" id="idade"></input><p></p>
<input type="button" value="Incluir" id="botao_incluir" onClick="Inclusao()">
<input type="button" value="Excluir" id="botao_excluir" onClick="Exclusao()">
<input type="button" value="Alterar" id="botao_alterar" onClick="Alteracao()">
<input type="button" value="Consultar" id="botao_consultar" onClick="Consulta()">
</body>
</html>

```

Agora, vamos analisar o código. Clique nas abas a seguir para acompanhar.

- Em primeiro lugar, logo após o início da cláusula function(), é declarada uma variável “dadosajax”. Ela irá conter os campos necessários para executar a função.
-

Nas functions de Inclusao() e Alteracao() são passados por esta variável todos os campos do aluno (nome, endereço, idade e curso), no caso das functions Exclusao().

- Consulta() é agregado à variável “dadosajax” somente o campos nome, pois é somente dele que precisamos para efetuar uma exclusão ou consulta no banco de dados.

As linguagens de programação evoluem de forma a se tornarem mais robustas e atender as demandas tecnológicas, conforme ilustração a seguir.



Figura 5 - As linguagens de programação e sua evolução.

Fonte: Oleksiy Mark, Shutterstock, 2019.

Logo após esta declaração declaramos uma variável “pageurl” que vai indicar o nome do código que irá efetuar a gravação dos dados. Neste caso, iremos criar um código chamado “incluir\_aluno.php”, “excluir\_aluno.php”, “alterar\_aluno.php” e “consutra\_aluno.php”.

Em seguida, é feita a declaração da ação Ajax, colocando o símbolo “\$.”, que indica ser um comando JQuery.

## VOCÊ QUER LER?



Para os desenvolvedores que já possuem experiência em desenvolvimento de páginas web, é fundamental aprofundar nos comandos mais avançados do HTML. O livro “Programação profissional em HTML 5” (ALBERS; LUBBERS; SALIM, 2013), ensina de forma clara como desenvolver páginas web com o HTML 5, utilizando os recursos mais básicos e aprofundando nas funções mais avançadas.

O jQuery é uma biblioteca JavaScript rápida, pequena e rica em recursos. Ele torna as coisas, como passagem e manipulação de documentos HTML, manipulação de eventos, animação e Ajax, muito mais simples, com uma API fácil de usar que funciona em vários navegadores (JQUERY, 2019).

O código PHP é inserido para gravação dos dados, indicando o tipo de processamento, que é um tipo “post” e demonstrando que não há um “cache” para o processamento.

Depois de realizar estes comandos, temos as duas condições de sucesso ou de fracasso da execução dos programas “incluir\_aluno.php”, “excluir\_aluno.php”, “alterar\_aluno.php” e “consultar\_aluno.php”.

## CASO

Uma indústria de grande porte estava reorganizando seu portal web, que já se encontrava em um formato obsoleto. Foram propostas duas metodologias diferentes a serem aplicadas: uma metodologia de *design* centrada no usuário e outra metodologia de design participativo. Esse tipo de abordagem iria impactar no modelo de arquitetura de implantação do novo portal. O objetivo era propor uma metodologia para a reestruturação, priorizando a organização e facilidade de uso do portal. A primeira etapa foi a realização de uma entrevista com os usuários para obter as necessidades fundamentais. O problema é se seria adotada a metodologia orientada no perfil do usuário ou na tarefa. A metodologia orientada ao perfil do usuário leva a uma arquitetura de um modelo no qual o foco está centrado na interface. Por outro lado, uma metodologia orientada na tarefa faz com que a arquitetura esteja mais centrada no modelo cliente/servidor.

Assista ao vídeo abaixo e aprenda mais sobre o tema.

[https://cdnapisec.kaltura.com/p/1972831/sp/197283100/embedIframeJs/uiconf\\_id/30443981/partner\\_id/1972831?iframeembed=true&playerId=kaltura\\_player\\_1549478683&entry\\_id=1\\_pvkj9mw0](https://cdnapisec.kaltura.com/p/1972831/sp/197283100/embedIframeJs/uiconf_id/30443981/partner_id/1972831?iframeembed=true&playerId=kaltura_player_1549478683&entry_id=1_pvkj9mw0)

Agora, vamos criar o nosso código em php que irá executar a gravação do cliente.

O navegador recebe os resultados da execução desse script. Assim, podemos criar os códigos PHP “incluir\_aluno.php”, “excluir\_aluno.php”, “alterar\_aluno.php” e “consultar\_aluno.php”, e analisar a lógica e sintaxe deles.

**incluir\_aluno.php**

```
<?php
```

```
$host = "servidor_banco_de_dados.com.br";
```

```
$user = "usuario_banco_de_dados";
```

```

$pass = "senha_banco_de_dados";
$db = "banco_de_dados";
$conecta = mysql_connect($host, $user, $pass) or print (mysql_error());
mysql_select_db($db, $conecta);
$nome_aluno = $_REQUEST['nome_aluno'];
$endereco_aluno = $_REQUEST['endereco_aluno'];
$idade_aluno = $_REQUEST['idade_aluno'];
$curso_aluno = $_REQUEST['curso_aluno'];
$sql = "INSERT INTO Aluno
(NomeAluno,
EnderecoAluno,
IdadeAluno,
CursoAluno)
VALUES
('$nome_aluno',
'$endereco_aluno',
'$idade_aluno',
'$curso_aluno');";
$query = mysql_query($sql);
?>

```

#### **excluir\_aluno.php**

```

<?php
$host = "servidor_banco_de_dados.com.br";
$user = "usuario_banco_de_dados";
$pass = "senha_banco_de_dados";
$db = "banco_de_dados";
$conecta = mysql_connect($host, $user, $pass) or print (mysql_error());
mysql_select_db($db, $conecta);
$nome_aluno = $_REQUEST['nome_aluno'];
$sql = "DELETE Aluno WHERE NomeAluno = '$nome_aluno';";
$query = mysql_query($sql);
?>

```

#### **alterar\_aluno.php**

```

<?php
$host = "servidor_banco_de_dados.com.br";
$user = "usuario_banco_de_dados";
$pass = "senha_banco_de_dados";
$db = "banco_de_dados";
$conecta = mysql_connect($host, $user, $pass) or print (mysql_error());
mysql_select_db($db, $conecta);
$nome_aluno = $_REQUEST['nome_aluno'];
$endereco_aluno = $_REQUEST['endereco_aluno'];
$idade_aluno = $_REQUEST['idade_aluno'];
$curso_aluno = $_REQUEST['curso_aluno'];
$sql = "UPDATE Cliente SET
EnderecoAluno = '$endereco_aluno',
IdadeAluno = '$idade_aluno',
CursoAluno = '$sexo_aluno',
WHERE NomeAluno= '$nome_aluno';";

```

```

$query = mysql_query($sql);
?>
consultar_aluno.php
<?php
$host = "servidor_banco_de_dados.com.br";
$user = "usuario_banco_de_dados";
$pass = "senha_banco_de_dados";
$db = "banco_de_dados";
$conecta = mysql_connect($host, $user, $pass) or print (mysql_error());
mysql_select_db($db, $conecta);
$nome_aluno = $_REQUEST['nome_aluno'];
$sql = "SELECT * FROM Aluno WHERE NomeAluno = '$nome_aluno'";
$query = mysql_query($sql);
?>

```

Agora, vamos analisar os códigos PHP dos comandos CRUD de um *site* de uma faculdade virtual. Inicialmente, são criadas as variáveis para conectar com o banco de dados (\$host, \$user, \$pass e \$db). Logo após é inserida em uma variável (\$conecta) os parâmetros para a função mysql\_connect(), inclusive retornando o sucesso – ou não – da execução do comando.



Figura 6 - A linguagem PHP pode ser utilizada, tanto em sistemas web, quanto em sistemas desktop.

Fonte: Dzianis\_Rakhuba, Shutterstock, 2019.

Depois disso, são recebidos os parâmetros necessários para executar os comandos CRUD, dependendo de cada comando (excluir, incluir, alterar ou consultar).

Após o recebimento destas informações, é direcionado para uma variável (\$sql) o comando a ser executado pelo SQL (INSERT, DELETE, UPDATE ou SELECT). O processo é finalizado com a instrução `mysql_query`, que, efetivamente, executa o comando necessário.

Conforme demonstrado, o comando de exclusão de registros merece um tratamento especial, devido a problemas de inconsistência que podem acarretar pelo uso devido do comando. Portanto, a aplicação correta da sintaxe, os tratamentos de erro são fundamentais para a construção de um sistema confiável e de alta qualidade. Também ficou clara a importância do uso das sessões e dos comandos CSS para o desenvolvimento e implantação de *softwares* confiáveis.

## Síntese

Finalizamos o conteúdo! Aqui, foi possível aprender como fazer as exclusões no banco de dados, recuperar informações e alterar por meio de uma página *web*. Vimos todas as funções que fazem a interação com o banco de dados e que são realizadas por meio dos comandos JavaScript e Ajax do JQuery, sendo este um processo assíncrono. Por outro lado, o Ajax chama um programa PHP, que consegue, este sim, interagir com o banco de dados por meio dos comandos SQL.

Vimos como desenvolver um sistema web com segurança por meio das sessões, além de práticas seguranças de desenvolvimento de sistemas. Foi possível aprender, também, como utilizar o recurso CSS para construção de páginas mais eficientes e profissionais. Foi possível aprender, também como desenvolver os comandos fundamentais CRUS (Incluir, Excluir, Alterar e Excluir).

Podemos verificar que, quando acessamos uma página na internet, na realidade, existe uma série de linguagens que trabalham de forma coordenada e assíncrona para que possa realizar todas as transações que necessitamos, como consultas, recuperar informações e alterar dados.

Neste capítulo, você teve a oportunidade de:

- conhecer os comandos fundamentais de exclusão das informações em banco de dados por meio de uma página *web*;
- entender como criar páginas em PHP e executar o comando de exclusão de registros em um banco de dados;
- implementar regras de segurança em sistemas *web*;
- compreender como são utilizadas as variáveis de sessão para realizar uma autenticação no sistema;
- analisar a estrutura de CSS e validações em JavaScript;
- entender os fundamentos da linguagem PHP e a sua aplicabilidade nos mais diversos tipos de sistemas.

## Bibliografia

ALBERS, B.; LUBBERS, P.; SALIM, F. **Programação Profissional em HTML 5**. São Paulo: Alta Books, 2013.

DUCKETT, J. **Html e Css - Projeto e Construa Websites**. São Paulo: Alta Books, 2016.

FONSECA FILHO, C. **História da Computação: o caminho do pensamento e da tecnologia**. Porto Alegre: EDIPUCRS, 2007. Disponível em: <<http://www.pucrs.br/edipucrs/online/historiadacomputacao.pdf>>. Acesso em: 3/2/2019.

GIROLDO, B. C.; FRESSATI, W. Evolução no Desenvolvimento com o HTML 5. In: XVII Seinpar – in... **Anais.. Semana de Informática e XIV Mostra de Trabalhos de Iniciação Científica de Paranaíba**. Paranaíba, 2015.

JQUERY. **jQuery API**. Portal jQuery, The jQuery Foundation, 2019. Disponível em: <<http://api.jquery.com/>>. Acesso em: 3/2/2019.

LEMAY, L.; COLBURN, R.; TYLER, D. **Aprenda a criar páginas web com HTML e XHTML em 21 dias**. São Paulo: Pearson Education do Brasil, 2002. Disponível em: <<http://aulaaberta.bv3.digitalpages.com.br/users/publications/9788534614283>>. Acesso em: 13/1/2019.

MELHORWEB. **Sessão(session) para login em PHP**. Portal MelhorWeb, s/d. Disponível em: <<https://www.melhorweb.com.br/artigo/18082-Sessao-session--para-login-em-PHP.htm>>. Acesso em: 3/2/2019.

MYSQL. **MySQL Documentation**. Portal MySQL, Oracle Corporation, 2019. Disponível em: <<https://dev.mysql.com/doc/>>. Acesso em: 3/2/2019.

PHP. **Documentation**. Portal PHP, 2019. Disponível em: <<http://php.net/docs.php>>. Acesso em: 3/2/2019.

SANTOS, M. **Programação funcional em PHP – saia da zona de conforto**. PHP Experience 2016 – São Paulo. Canal iMasters, YouTube, publicada em 23/05/2016. Disponível em: <[https://www.youtube.com/watch?v=cts\\_NStWj5Q](https://www.youtube.com/watch?v=cts_NStWj5Q)>. Acesso em: 3/2/2019.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Addison Wesley, 2011.

W3SCHOOLS. **HTML 5 Tutorial**. Portal W3Schools, 2019. Disponível em: <<https://www.w3schools.com/html/>>. Acesso em: 3/2/2019.