

# 5

## Outras linguagens de programação

Quando o assunto são linguagens voltadas para o desenvolvimento web, geralmente três linguagens, ou notações, vêm à tona: a de marcação HTML, as folhas de estilo CSS e a de programação Javascript. Tais linguagens são voltadas, prioritariamente, para o desenvolvimento front-end ou client-side (ou seja, sob a perspectiva do cliente/navegador – essas linguagens são orientadas para o desenvolvimento de interfaces para o usuário).

Porém, para o desenvolvimento de aplicações web, torna-se necessário, também, o uso de linguagens de programação voltadas para administrar informações sob a perspectiva do server-side (isto é, do servidor), ou seja, para a gestão de dados e de informações necessários para que a página cumpra seus propósitos.

Consideremos as páginas de comércio eletrônico. Além da interface em que os produtos são mostrados ao cliente, faz-se necessária a gestão das bases de dados em que esses produtos estão localizados, juntamente às informações como o preço, a quantidade em estoque e tudo mais que compõe as informações necessárias para cada produto.

Neste capítulo, trataremos a respeito de linguagens de programação voltadas para a administração de uma página sob a perspectiva do servidor. Primeiramente, da clássica linguagem de programação Preprocessed Hypertext Pages (PHP), ainda utilizada para a gestão de serviços na web (webservices). Dentre várias atribuições, essa linguagem permite efetuar a comunicação entre a interface do usuário e as bases de dados que fornecem as informações necessárias para a gestão da página de internet, conforme já exemplificamos com o comércio eletrônico.

Em seguida, abordaremos tópicos relacionados à gestão de dados e de informações – mais precisamente da linguagem SQL e do banco de dados MySQL. Esse banco de dados pode ser administrado a partir da linguagem PHP, com base em parâmetros passados para arquivos desenvolvidos com essa linguagem de programação.

Finalmente, falaremos a respeito de modelos de referência (frameworks) para o desenvolvimento de aplicações web. Alguns desses modelos são fundamentados na linguagem de programação Javascript, outros na Python. Conhecer esses modelos é essencial, pois um deles pode ser útil quando formos desenvolver e viabilizar um projeto de tecnologia web.

Esperamos que esse capítulo proporcione os conhecimentos não somente a respeito de cada linguagem de programação ou framework, mas também de como aplicá-los, de modo eficaz, no desenvolvimento de páginas web. Tenha uma boa leitura!



#### Objetivo de aprendizagem

Com o estudo deste capítulo, você será capaz de:

- conhecer outras linguagens de programação utilizadas para o desenvolvimento de aplicativos web.

## 5.1 A linguagem PHP

### Vídeo



### Biografia

William Stedweld Demchick/Wikimedia Commons



Bacharel em Ciências Aplicadas em Engenharia de Projeto de Sistemas, Rasmus Lerdof atuou em grandes empresas, como Yahoo! e Etsy. Também é conhecido pela criação das primeiras versões da linguagem PHP.

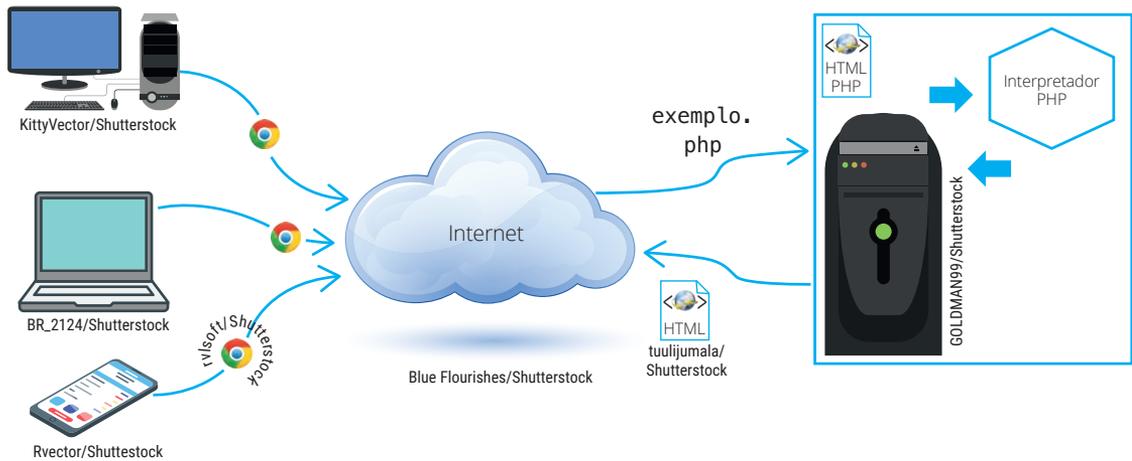
A linguagem PHP foi uma das primeiras utilizadas para o desenvolvimento web. Surgiu em 1994, desenvolvida por Rasmus Lerdof, sendo o seu código-fonte liberado para o público em geral em 1995. Com essa atitude, o criador da linguagem encorajou que os usuários da PHP buscassem erros no código e, dessa forma, melhorassem sua estrutura geral.

Atualmente em sua sétima versão (PHP 7), essa linguagem continua sendo amplamente usada para o desenvolvimento de aplicações web voltadas ao gerenciamento de dados em servidores. Em outras palavras, a PHP é uma linguagem de programação do tipo server-side, ou seja, tudo é processado do lado do servidor – ao contrário das linguagens client-side (como o Javascript), em que o processamento de dados ocorre no navegador do usuário. Tal fato proporciona à PHP uma relevante característica: ninguém, portando apenas o navegador de internet, consegue obter uma cópia do código-fonte da linguagem ou do que está sendo processado. Na Figura 1 vemos o papel da PHP em uma arquitetura no modelo cliente-servidor.



Figura 1

Interpretação do PHP no modelo cliente-servidor



Fonte: Adaptada de Basso; Miletto, 2014, p. 163

A linguagem PHP apresenta, ainda, uma série de vantagens para sua utilização. Em primeiro lugar, pode ser executada em servidores de diferentes sistemas operacionais, como Windows, Linux e iOS, bastando que tenham o suporte dessa linguagem instalado. Além disso, a linguagem tem código-fonte aberto, sendo assim, é gratuita para download. Também, suporta a conexão a uma variedade de bancos de dados – em especial, aos bancos de dados MySQL e similares –, apresentando funções específicas para essa conexão.

Desenvolver um programa em PHP pode ser uma tarefa simples, dependendo das funcionalidades desejadas. No exemplo a seguir, há um código HTML em que o código PHP se encontra inserido no próprio arquivo exibindo uma mensagem de texto de abertura.

```
<html>
<body>

<h1>Um cabeçalho em HTML apresentando a página!</h1>

<?php
    echo "Esse texto foi escrito em PHP!";
?>

</body>
</html>
```

O código pode ser, também, escrito em um arquivo separado .PHP – nesse caso, sendo necessário instalar o interpretador PHP no computador ou servidor para que funcione. No exemplo a seguir temos quatro variáveis declaradas – a declaração de variáveis se dá por meio do uso do caractere especial \$. As variáveis x e y contêm, respectivamente, os números 10 e 20. Obtemos a variável z por meio do resultado da soma das variáveis x e y. E a variável \$txt consiste em um texto em que, concatenada (ou unida) a este, encontramos a variável z pelo uso de pontos ('.'), exibindo na tela do usuário a soma dos números por meio do comando echo.

```
<?php

$x = 10;
$y = 20;

$z = $x + $y;
$txt = "A soma dos números é ".$z."!";

echo $txt;

?>
```

Assim como ocorre com outras linguagens de programação, a PHP também adota conceitos, tais como estruturas de repetição e condicionais. No exemplo a seguir, encontramos descrito um código para exibição de uma estrutura condicional. No caso, a partir de determinada temperatura (armazenada em uma variável), o código verifica, com base em parâmetros, se essa temperatura está adequada, quente ou fria demais.

```
<?php

$temperatura = 36;

if ($temperatura > 35) {
    echo "A temperatura está alta demais!";
} else if ($temperatura > 20) {
    echo "A temperatura está amena!";
} else {
    echo "A temperatura está fria demais!";
}

?>
```

Também, a linguagem PHP adota o conceito de funções ou sub-rotinas, ou seja, uma sequência de códigos agrupada sob um nome específico e chamada quando necessário no programa. No exemplo a seguir, construímos uma função denominada *soma*, que recebe dois parâmetros de entrada, no caso, dois números a serem somados (**n1** e **n2**). A soma é realizada e armazenada na variável `$resultado`, que é exibida na tela pelo comando `echo`. A função é chamada pelo seu nome com os parâmetros de entrada escolhidos: `soma(5,15);`.

```
<?php

function soma($n1,$n2) {

$resultado = $n1 + $n2;
echo "A soma entre os dois números é ".$resultado."!";

}

soma(5,15);

?>
```

Também podemos utilizar a linguagem PHP para a administração de formulários em HTML. Nesse caso, no preenchimento e envio destes, um arquivo em PHP é chamado para fazer o gerenciamento dos dados e retornar o que foi preenchido para o usuário. Vejamos o exemplo a seguir.

```
<html>
<body>

<form action="form.php" method="get">

Nome completo: <input type="text" name="nome"><br>
E-mail: <input type="text" name="email"><br>
Endereço: <input type="text" name="endereco"><br>

<input type="submit">

</form>

</body>
</html>
```

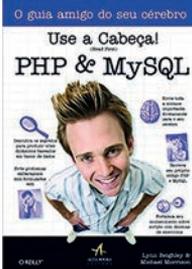
Ao ser enviado, o formulário armazenará os dados do usuário em variáveis e retornará as mesmas informações ao usuário, conforme o código a seguir.

```
form.php

<html>
<body>

Seu nome completo é <?php echo $_GET["nome"]; ?>!<br>
Seu endereço de e-mail é: <?php echo $_GET["email"]; ?>!<br>
Seu endereço é: <?php echo $_GET["endereco"]; ?>!<br>

</body>
</html>
```



No livro *Use a cabeça! PHP & MySQL*, os autores ensinam, de maneira didática e com o uso de linguagem cotidiana, os princípios básicos do desenvolvimento de websites utilizando a linguagem de programação PHP. Também abordam o uso de MySQL aliado ao PHP, para a construção de websites contendo bancos de dados.

BEIGHLEY, L.; MORRISON, M. Rio de Janeiro: Alta Books, 2013.

Esses exemplos referem-se ao envio e à exibição do formulário utilizando o método – ou protocolo – GET. Nesse método, as informações do formulário são enviadas na URL da própria requisição, ou seja, com o endereço digitado no navegador. Já em outro método, denominado *POST*, as informações são anexadas no corpo da requisição, não sendo exibidas na URL da mensagem. Dessa forma, o método *POST* é mais adequado no caso de informações de formulário que, a princípio, não podem ser vistas por outros usuários.

A linguagem PHP também é muito utilizada para operações relacionadas a bancos de dados, como a execução de comandos da linguagem SQL. Por meio da PHP, é possível acessarmos um banco de dados específico utilizando um comando de consulta, ou query (BEIGHLEY; MORRISON, 2013). Na seção a seguir mostramos como estruturar um banco de dados com o uso da linguagem SQL.

## 5.2 O banco de dados e a linguagem SQL



As linguagens do lado do servidor, como a PHP, além de executarem comandos em nível de servidor, têm o intuito de gerenciar bases (ou bancos) de dados. Tal gerenciamento se dá por meio de comandos fornecidos para uma outra linguagem, esta responsável por operações de gerenciamento de dados.

Um banco de dados consiste em uma coleção de dados armazenados em um computador de maneira ordenada. Tomemos como exemplo o cadastro de clientes de uma loja, contendo informações como nome da pessoa, CPF e e-mail. Essas informações, nos bancos de dados, geralmente encontram-se organizadas no formato de uma tabela, sendo relacionadas tanto de modo horizontal como vertical. O exemplo a seguir ilustra uma tabela de banco de dados típica (os dados inseridos são fictícios).

Nome	E-mail	CPF
Bruno Henrique	brunoh@emailficticio.com	000.999.222-12
Maria das Dores	mariadores@emailficticio.com	888.222.333-11

No caso desse banco de dados, as informações relativas a nome, e-mail e CPF estão relacionadas. Por exemplo, o CPF de número 000.999.222-12 remete diretamente ao nome Bruno Henrique de maneira horizontal. Já verticalmente, os nomes Bruno Henrique e Maria das Dores se relacionam por pertencerem ao mesmo grupo, ou variável, denominada *Nome*. A esse modelo de informações em bancos de dados, em que estes se relacionam de modo horizontal e vertical com base em conjuntos, denominamos de *modelo relacional*.

Segundo Rob e Coronel (2014, p. 67), “a tabela é vista como uma estrutura bidimensional composta de linhas e colunas [...]. É possível pensar em uma tabela como uma representação permanente de uma relação lógica, ou seja, de uma relação cujo conteúdo possa ser salvo definitivamente para uso futuro”.

A linguagem de programação utilizada para a gestão de bancos de dados é a Structured Query Language (SQL). Ela é amplamente usada por diferentes bancos de dados, como o SQL, o MySQL, o MariaDB, entre outros, sendo, dessa forma, considerada uma linguagem padrão para bancos de dados.

A linguagem SQL se divide em sublinguagens conforme sua utilização para a gestão do banco de dados. Atualmente, as sublinguagens da SQL consistem nas seguintes:

- **Data Manipulation Language (DML):** linguagem utilizada para a manipulação de dados em uma base – no caso, inserção (pelo

comando `INSERT`), atualização (pelo comando `UPDATE`) ou eliminação dos dados (pelo comando `DELETE`).

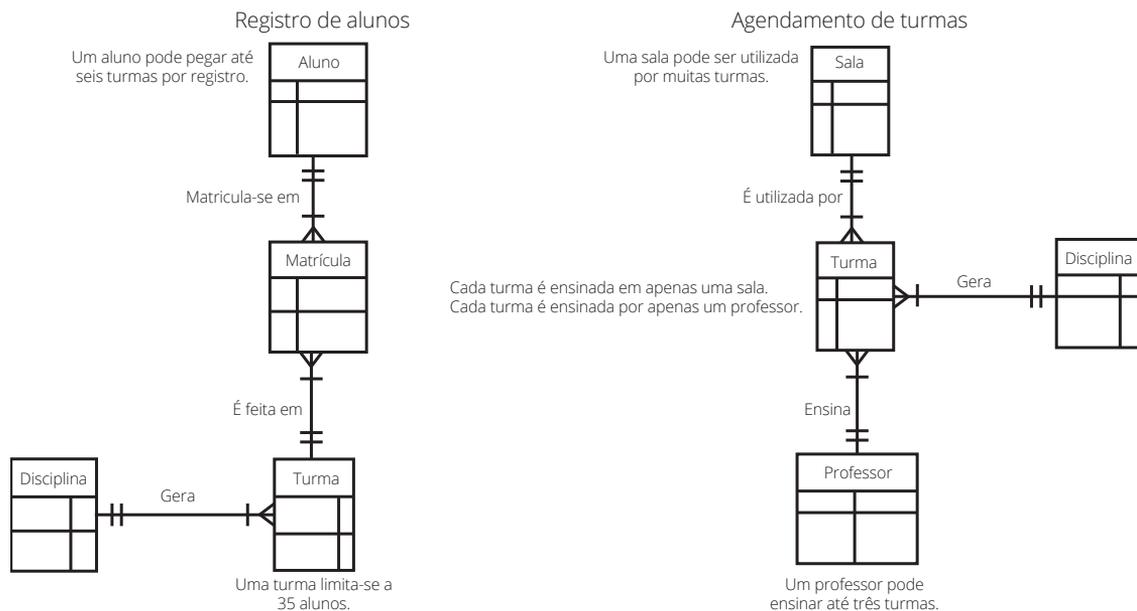
- **Data Definition Language (DDL):** linguagem para a definição de elementos essenciais de dados, como variáveis, tabelas, dentre outros – no caso, a linguagem tem comandos essenciais, por exemplo, criar (pelo comando `CREATE`) e eliminar (pelo comando `DROP`) objetos no banco de dados.
- **Data Control Language (DCL):** linguagem para o controle de acesso aos dados apresenta comandos importantes, tais como garantir (pelo comando `GRANT`) e revogar (pelo comando `REVOKE`) acessos de usuários a comandos essenciais de outras sublinguagens, como os das DML e DDL.
- **Data Query Language (DQL):** linguagem utilizada para processos de consulta de dados (também denominada *query*). Tal consulta pode ser feita por meio do comando `SELECT`, especificando, nesse comando, o que desejamos encontrar.

As informações, em um banco de dados, podem chegar a quantidades que se tornam difíceis de gerenciar se não for com o apoio de um sistema de gerenciamento de banco de dados (SGBD): “um SGBD executa várias funções que garantem a integridade e a consistência dos dados no banco” (ROB; CORONEL, 2014, p. 23). Ainda, de acordo com esses autores, um SGBD inclui as seguintes funções essenciais:

- gerenciamento de dicionário e armazenamento de dados;
- transformação e apresentação de dados;
- gerenciamento de segurança;
- controle de acesso de multiusuário;
- gerenciamento de backup e recuperação;
- gerenciamento de integridade de dados;
- linguagens de acesso ao banco de dados;
- interfaces de programação de aplicações;
- interfaces de comunicação do banco de dados.

Para organizar os dados em um SGBD, fazemos o uso de diagramas, como o diagrama entidade-relacionamento (DER). Um DER “utiliza representações gráficas para modelar os componentes do banco de dados” (ROB; CORONEL, 2014, p. 43). A Figura 2 ilustra um modelo de DER.

 **Figura 2**  
Exemplo de DER



Fonte: Rob; Coronel, 2014, p. 53.

Nesse exemplo de DER existem dois processos, em uma instituição de ensino, que demandam bases de dados: o registro de alunos e o agendamento de turmas. Cada caixa representa um conjunto de dados: alunos, matrícula, turma, disciplina, sala e professor. No caso do processo de registro de alunos, o aluno se relaciona com os dados referentes à sua matrícula, que, no caso, irá se relacionar com os dados referentes à sua turma e à disciplina que cursará. Já com relação ao agendamento de turmas, os conjuntos de dados relevantes são os relacionados ao professor, à sala, à turma e à disciplina.

A sintaxe da linguagem SQL adota uma lógica similar às das demais linguagens de programação. Os comandos são executados segundo parâmetros específicos, considerando os objetos ou dados a serem criados, alterados ou eliminados.

Tomemos como exemplo o comando CREATE, da linguagem DDL. Por meio dele, efetuamos a criação tanto do banco de dados quanto de uma tabela dentro desse banco. O código a seguir exemplifica como criamos uma base de dados e uma tabela no SQL.

```

CREATE DATABASE Escola; /* Cria uma base de dados denominada Escola */

USE Escola;

CREATE TABLE Aluno (
    id INT AUTO_INCREMENT PRIMARY KEY,
    PrimeiroNome VARCHAR (30) NOT NULL,
    UltimoNome VARCHAR (30) NOT NULL,
    Email VARCHAR (30),
    Idade INT
);

```

No código anterior, primeiramente, criamos uma base de dados denominada *Escola*. Em seguida, pelo comando `USE`, selecionamos essa base de dados para o próximo comando, o de criar uma tabela. A tabela é, então, criada contendo as seguintes colunas: **id**, **PrimeiroNome**, **UltimoNome**, **Email** e **Idade**. E, para cada coluna, atribuímos parâmetros específicos, conforme descrito na tabela a seguir.

Parâmetro	Significado
INT	Variável (ou coluna) do tipo inteira (numérica)
VARCHAR (nº. caracteres)	Variável do tipo sequência de caracteres (entre parênteses: máximo de caracteres aceito)
NOT NULL	A variável não aceita valores nulos – é necessário inserir conteúdo quando uma nova linha na base de dados for criada.
AUTO_INCREMENT	A cada linha que for criada na base de dados, o valor da variável é inserido automaticamente e incrementa de 1 em 1.
PRIMARY KEY	A variável é a chave primária da tabela, isto é, o identificador único, ou seja, os dados não se repetem nesta coluna.

Com base na criação do banco de dados, podemos alimentá-lo com dados conforme as colunas definidas. Vejamos um exemplo disso.

```

INSERT INTO Aluno (PrimeiroNome, UltimoNome, Email, Idade) VALUES
("Maria", "Das Dores", "mariadores@emailficticio.com", "25");

INSERT INTO Aluno (PrimeiroNome, UltimoNome, Email, Idade) VALUES
("Bruno", "Henrique", "brunoh@emailficticio.com", "21");

INSERT INTO Aluno (PrimeiroNome, UltimoNome, Email, Idade) VALUES
("Luan", "Da Silva", "luansilva@emailficticio.com", "18");

INSERT INTO Aluno (PrimeiroNome, UltimoNome, Email, Idade) VALUES
("Mariana", "Nunes", "marinunes@emailficticio.com", "19");

INSERT INTO Aluno (PrimeiroNome, UltimoNome, Email, Idade) VALUES
("André", "Luiz", "andreluiz@emailficticio.com", "40");

```

Nesse comando, percebemos que, primeiro, selecionamos a tabela em que os dados serão inseridos e, entre parênteses, as variáveis que serão afetadas na tabela. A seguir, pelo parâmetro VALUES, determinamos quais valores serão inseridos em quais variáveis, em sequência. O resultado da execução desse comando encontramos na tabela a seguir.

Id	PrimeiroNome	UltimoNome	Email	Idade
1	Maria	Das Dores	mariadores@emailficticio.com	25
2	Bruno	Henrique	brunoh@emailficticio.com	21
3	Luan	Da Silva	luansilva@emailficticio.com	18
4	Mariana	Nunes	marinunes@emailficticio.com	19
5	André	Luiz	andreluiz@emailficticio.com	40

A tabela Alunos foi alimentada com cinco inserções (ou linhas) de dados – a primeira contendo os dados pessoais de Maria, e as demais contendo os de Bruno, Luan, Mariana e André (nomes fictícios). A coluna 'Id', em ambas as tabelas, foi alimentada automaticamente com os valores sequenciais '1', '2', '3', '4' e '5', tendo em vista o parâmetro AUTO\_INCREMENT passado a essa coluna no momento de sua criação.

Outro exemplo de operação que podemos realizar em um banco de dados é a busca, ou 'query'. Por meio dessa operação, é possível encontrarmos uma ou mais linhas de dados que correspondam a um critério específico. Para isso, utilizamos o comando SELECT, conforme exemplificamos a seguir.

```
SELECT * FROM Aluno WHERE UltimoNome = 'Das Dores';
```

Nesse caso, todas as linhas da base de dados que atendam ao critério estabelecido (a variável UltimoNome = 'Das Dores') serão exibidas. Como na nossa base de dados há somente uma aluna com esse último nome registrado, o resultado será conforme a tabela a seguir.

Id	PrimeiroNome	UltimoNome	Email	Idade
1	Maria	Das Dores	mariadores@emailficticio.com	25

Outra possibilidade é a seleção por meio de critério numérico – nesse caso, podemos selecionar um usuário de acordo com sua idade, bastando escrevermos a seguinte linha de código:

```
SELECT * FROM Aluno WHERE Idade > 20;
```

Diante disso, selecionaremos apenas as linhas referentes às pessoas que têm idade maior de 30 anos, exibindo a tabela a seguir.

#### Livro



O livro *Sistemas de banco de dados: projeto, implementação e gerenciamento* apresenta, de maneira didática e abrangente, conceitos relativos a bancos de dados, englobando desde a estruturação de diagramas (como de entidade-relacionamento), até aspectos de estruturação e programação. Leitura sugerida para quem deseja se aprofundar na estruturação de bancos de dados de modo avançado.

ROB, P.; CORONEL, C. São Paulo: Cengage Learning, 2014.

Id	PrimeiroNome	UltimoNome	Email	Idade
1	Maria	Das Dores	mariadores@emailficticio.com	25
2	Bruno	Henrique	brunoh@emailficticio.com	21
5	André	Luiz	andreluiz@emailficticio.com	40

As bases de dados são um elemento marcante em um aplicativo ou página web por abrigarem informações relevantes que podem ser consultadas pela pessoa que navega. Nesta seção, abordamos conceitos importantes relativos a bases de dados, bem como demonstramos o processo de criação de uma base de dados e uma tabela. Depois, apresentamos o processo de busca por um conteúdo nessa base fundamentando-se em um critério específico. A seção a seguir mostra boas práticas utilizadas para o desenvolvimento web, também essenciais para a eficiência e eficácia de um website.

## 5.3 Frameworks para desenvolvimento web

📺 Vídeo



Nas seções anteriores tratamos a respeito de duas das principais linguagens utilizadas para a gestão de dados do lado do servidor web: a PHP e a SQL. Tais linguagens são importantes para que as funcionalidades definidas do lado do cliente (interfaces) sejam executadas com o intuito esperado (o preenchimento de formulários, por exemplo).

Porém, quando o assunto é desenvolvimento web, os desenvolvedores fazem o uso de estruturas – ou modelos – de referência para que o código tenha o resultado esperado. Esses modelos, também denominados *frameworks*, são desenvolvidos em diferentes linguagens de programação, sendo utilizados conforme a necessidade e a compatibilidade destes com os demais elementos do sistema que está sendo desenvolvido.

Existem vários frameworks disponíveis no mercado, boa parte gratuitos e de fácil aprendizado. Dentre as linguagens utilizadas para o seu desenvolvimento, as principais são a Javascript e a Python. As tecnologias que descreveremos, logo, fazem o uso de uma dessas duas linguagens de programação, sendo essas: AJAX, Angular, React, Django e Flask.

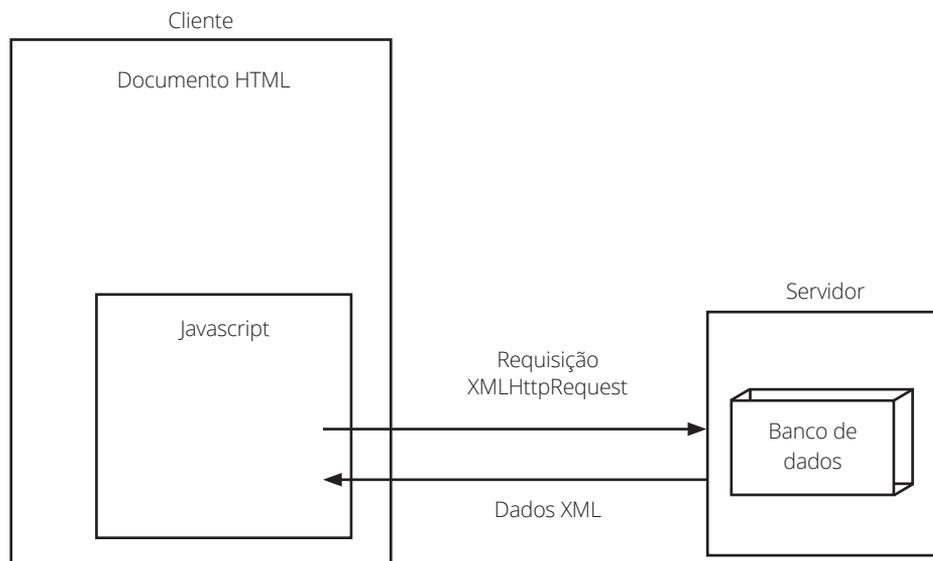
O framework Asynchronous JavaScript and XML (AJAX) consiste em um conjunto de códigos em diferentes notações juntamente com o Javascript, os quais resultam em mensagens que vão sendo trocadas com o servidor para que as informações inseridas pelo usuário tenham respostas rápi-

das, sem a necessidade de que a página de internet seja carregada toda vez que uma informação nova for inserida. Isso é possível por meio do envio de “uma requisição *XMLHttpRequest* para o servidor, que retornará informações em formato XML, as quais, por sua vez, serão tratadas no lado cliente. Tudo isso é feito enquanto o usuário permanece visualizando sua página” (NIEDERAUER, 2014, p. 100, grifo do original). Na Figura 3 vemos como ocorre a interação entre o lado do cliente e o servidor no AJAX.



**Figura 3**

Interação das tecnologias no AJAX



Fonte: Niederauer, 2014, p. 100.

O framework React consiste em uma biblioteca desenvolvida em Javascript, voltada principalmente para o desenvolvimento de interfaces de usuário. Criada em código aberto, pode ser utilizada abertamente, sendo inclusive adotada por websites reconhecidos, como Netflix, Airbnb, eBay, dentre outros. Podemos utilizar o React com outros frameworks, tais como o AngularJS, dada sua compatibilidade.

Também voltado para o desenvolvimento de interfaces de usuário, o framework AngularJS utiliza como base a linguagem Javascript. Faz uso do conceito de arquitetura Model-View-Controller (SOMMERVILLE, 2011), composto dos seguintes elementos:

- **Model (modelo):** estrutura responsável pelo processamento das informações recebidas da interface do usuário.
- **View (visão):** a informação ao alcance do usuário (o que ele vê).

- **Controller (controle):** estrutura que responde aos comandos fornecidos pela visão e que repassa essas informações ao modelo, interagindo com este.

Agora, trataremos a respeito dos frameworks web desenvolvidos para a linguagem de programação Python. Essa linguagem tem sido, nos últimos anos, a mais utilizada para diferentes finalidades, em especial para aprendizado de máquina, gestão de bancos de dados, automação, dentre outras. Para finalidades relacionadas à web não é diferente: existem vários frameworks nessa linguagem de programação que podemos utilizar para o desenvolvimento de páginas de internet dinâmicas, conforme a seguir.

#### www Site

A página consiste no tutorial do Django, um framework utilizado para o desenvolvimento web fundamentado na linguagem de programação Python. De modo didático, com o uso de passo a passo, é possível aprendermos como desenvolver e implantar websites interativos baseados em Django.

Disponível em: <https://www.djangoproject.com/>. Acesso em: 22 fev. 2022.

O framework Django é um dos mais versáteis desenvolvidos na linguagem Python. Sendo de código aberto, é utilizado e aperfeiçoado pela própria comunidade de desenvolvimento. O código das páginas pode ser associado a outros frameworks do lado do cliente, como HTML, XML, JSON etc. (INTRODUÇÃO..., 2022). Dentre os exemplos de websites desenvolvidos com esse framework, encontramos o Instagram, o Disqus, o Dropbox e o Spotify.

O framework Flask, por sua vez, consiste em uma ferramenta para a construção rápida de páginas web em Python. Propõe-se a ser um framework simples, ou ‘micro’, em que não há, por exemplo, camadas de abstração para bancos de dados – embora haja a possibilidade de adição de extensões que providenciem tal integração com bancos de dados (WHAT..., 2010).



## CONSIDERAÇÕES FINAIS

Neste capítulo, aprendemos como estruturar e implantar aplicações web do lado do servidor com o uso de linguagens de programação próprias para isso, como PHP e MySQL. Também, estudamos alguns modelos, ou frameworks, próprios para o desenvolvimento web.

Desenvolver páginas web vai além do projeto e do desenvolvimento de interfaces para o usuário, também abrange o desenvolvimento de aplicativos para a gestão de dados e de informações do servidor. Devemos considerar elementos importantes como bancos de dados, uma vez que a página, quando acessada pelo usuário, exibirá esse conteúdo.

Logo, convém ao desenvolvedor web dominar as ferramentas tanto de desenvolvimento de interfaces quanto de desenvolvimento via servidor.