

04

CLASSIFICAÇÃO POR LINGUAGEM



Introdução a Programação para backend

02



As tecnologias de back-end são indispensáveis na criação, execução e gerenciamento de aplicativos modernos. Na verdade, selecionar o back-end certo pode fazer a diferença entre o sucesso ou o fracasso dos projetos de desenvolvimento de aplicativos. É por isso que empresas de todos os tamanhos e tomadores de decisão de TI precisam pesquisar o melhor back-end para seus projetos desde a fase de planejamento.





Introdução a Programação para backend

02



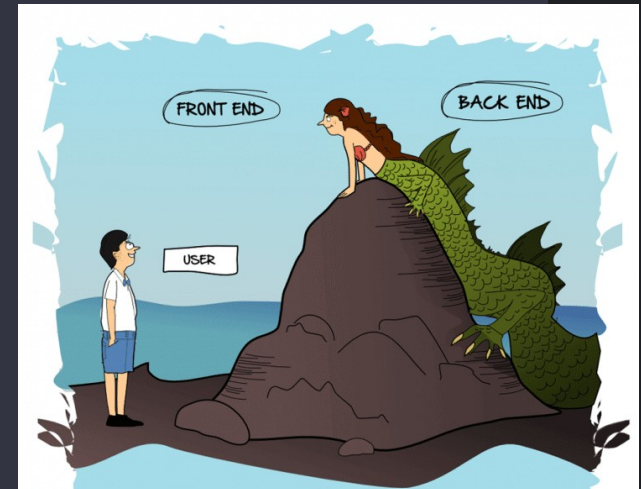
A tecnologia de back-end certa aprimora as operações de aplicativos sem complicações, melhorando a velocidade, escalabilidade e capacidade de resposta do aplicativo. No entanto, filtrar as opções para selecionar o melhor back-end para um projeto de aplicativo pode ser um desafio.



O que é um back-end de aplicativo?

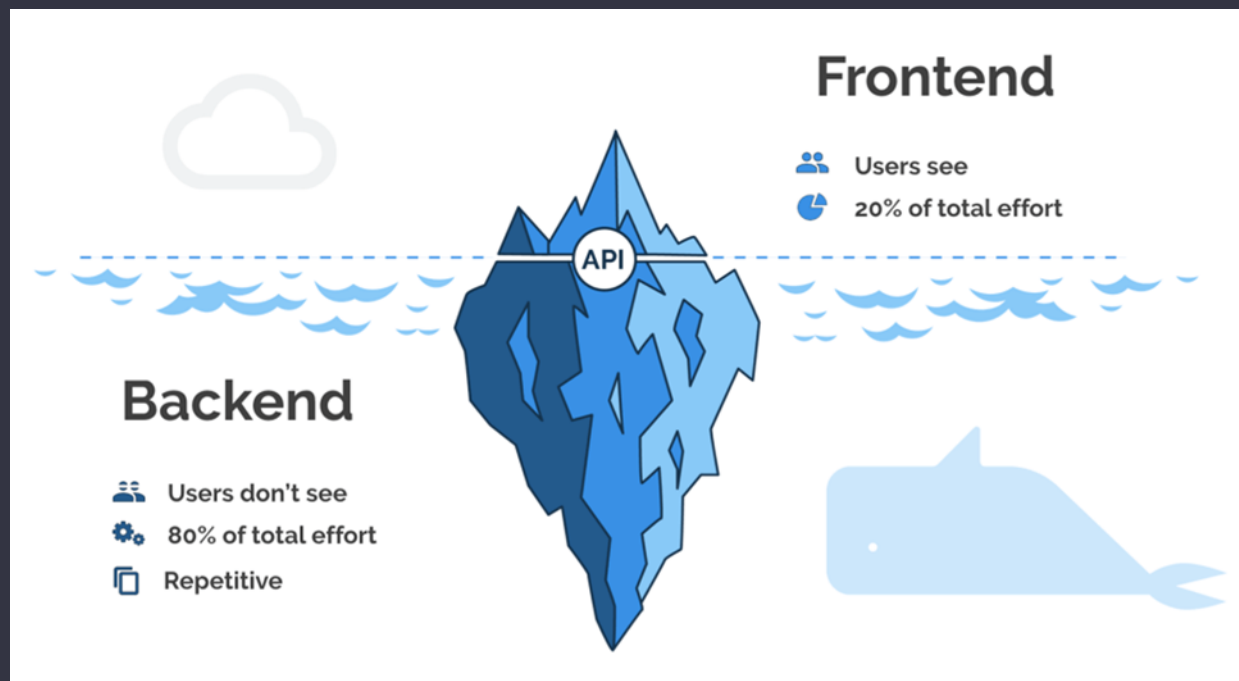
02

No mundo da programação, o back-end se refere aos códigos de computador que lidam com operações do lado do servidor, como lógica do servidor, funções de banco de dados e muito mais. Quando você usa um aplicativo, a grande maioria dos dados que você envia e recebe são gerenciados pelo back-end do aplicativo. No entanto, as funções do back-end são totalmente invisíveis para o usuário do aplicativo.



O que é um back-end de aplicativo?

02



FRONT-END

- Usuários veem
- 20% de esforço total

BACK-END

- Usuários não veem
- 80% de esforço total
- Repetitivo

Informalmente, dizemos que a parte visual é representada pelo *front-end* e a “nãovisual”, pelo *back-end*.



O que é uma linguagem de programação?

02



Em termos leigos, a linguagem de programação é um conjunto de comandos e instruções digitais que usam sintaxes específicas para criar aplicativos de computador. Essas linguagens de programação são categorizadas da seguinte forma:

- Linguagem de programação de alto nível
- Linguagem de programação de baixo nível



O que é uma linguagem de programação?

02



Linguagem de programação de alto nível

Este tipo de linguagem escreve códigos com palavras e símbolos comumente usados em conversas diárias. É por isso que os códigos de linguagem de programação de alto nível são fáceis de ler e examinar.

- JavaScript
- Python
- PHP
- Node.js



O que é uma linguagem de programação?

02

Linguagem de programação de alto nível



Vantagens

As principais vantagens desta linguagem é a facilidade no aprendizado e a produtividade que o programador ganha ao usar esse tipo de linguagem, até porque, ele consegue escrever mais códigos em menos tempo.

O processo de manutenção também é simplificado, caso outra pessoa necessite verificar seu código, ela terá mais facilidade em entender o que foi escrito.



O que é uma linguagem de programação?

02

Linguagem de programação de alto nível



Desvantagens

Essa linguagem também apresenta desvantagens. E uma delas é que, tem situações que o desempenho do programa se prejudica, pois a linguagem exige um tempo maior de processamento.

Além de ocupar mais memória se fizer uma comparação em relação às linguagens de baixo nível.



O que é uma linguagem de programação?

02



Linguagem de programação de baixo nível

Este tipo de linguagem de programação consiste em linguagens de máquina e linguagens assembly com instruções peculiares. Ao contrário da linguagem de programação de alto nível, eles escrevem códigos complicados que são difíceis de ler. Um programa chamado assembler traduz o código do assembly em código de máquina que a CPU entende, mas ininteligível para os humanos.

- A principal linguagem de baixo nível é o Assembly



O que é uma linguagem de programação?

02

Linguagem de programação de baixo nível



Vantagens

As vantagens desta linguagem é o tempo de processamento, que é mais rápido e a arquitetura do dispositivo que é mais aproveitada com esse tipo de linguagem.

Desvantagens

As desvantagens dessa linguagem é o tempo de compreensão, bem como domínio da sintaxe que será maior devido à complexidade, além de ser necessário também ter o conhecimento do hardware da máquina, e isso demanda investimentos em estudo e treinamento.



As cinco principais linguagens de programação de back-end em todo o mundo

02

JavaScript

- O JavaScript emergiu como a linguagem de programação mais popular na Pesquisa Anual StackOverflow.
- JavaScript não é mais apenas uma linguagem de script do lado do cliente, porque tecnologias como Node.js permitem que você execute operações do lado do servidor.
- O Node.js depende das estruturas do lado do servidor chamadas Express.js para criar uma plataforma que permite aos desenvolvedores escrever códigos que são executados no servidor.

✕ As cinco principais linguagens de programação de back-end em todo o mundo

02

JavaScript

○ O Express.js instalado e funcionando com o Node.js, os desenvolvedores poderão usar JavaScript como linguagem de desenvolvimento de front-end e back-end. Ele também oferece uma interface de programação de aplicativos (API) para a criação de vários aplicativos, incluindo aplicativos móveis, híbridos, da web, de uma e várias páginas.

express



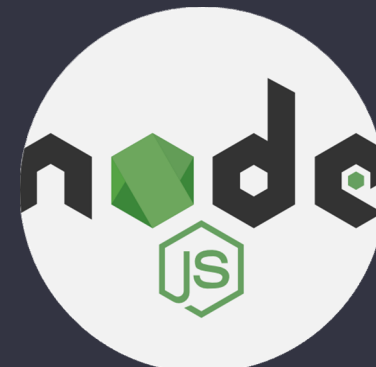
As cinco principais linguagens de programação de back-end em todo o mundo

02

JavaScript

Características:

- Desenvolvimento rápido
- Menos sobrecarga de script
- Tecnologia Lean Backend
- Corte de custos
- Vantagem de código aberto
- Recursos de I / O



express

✕ As cinco principais linguagens de programação de back-end em todo o mundo

02

○ Python

Desde que o Python foi criado em 1991 por Guido van Rossum, ele cresceu e se tornou uma das linguagens de programação multifuncionais líderes no mundo hoje. Os desenvolvedores de back-end utilizam seus códigos organizados e altamente legíveis para criar scripts funcionais para lidar com atribuições de back-end.

Uma pesquisa de 2020 feita por Stack Overflow revelou que é a principal linguagem de programação de back-end que os desenvolvedores esperam aprender. Python liderou essa categoria por quatro anos consecutivos. Ele também foi classificado em terceiro lugar entre as linguagens de programação mais amadas existentes.



As cinco principais linguagens de programação de back-end em todo o mundo

02

Python



4 frameworks de Python para diferentes usos

1. **BotCity** é uma opção de framework para automação de tarefas em Python, com o apoio de robôs;
2. **Django** é uma opção mais robusta para desenvolvimento web;
3. **Bottle** É um micro framework para criação de APIs e para interconectar servidores e aplicações;
4. **Kivy** é o framework especializado em desenvolvimento mobile.



As cinco principais linguagens de programação de back-end em todo o mundo

02

Python



Características

- Relativamente fácil de aprender
- Bibliotecas enormes
- Eficiente em termos de custos
- Recursos de IoT (Raspberry Pi)
- Códigos incorporáveis





As cinco principais linguagens de programação de back-end em todo o mundo

02

PHP



PHP foi desenvolvido por Rasmus Lerdorf em 1994; desde então, o PHP evoluiu para uma das principais linguagens de programação do lado do servidor hoje. Na verdade, uma pesquisa da W3Tech mostra que mais de 79% dos sites globais são alimentados por PHP.

PHP é uma linguagem de programação multifuncional fácil de usar. Funciona perfeitamente com uma ampla variedade de bancos de dados e sistemas operacionais. Estruturas modernas, uma base de código enorme e a comunidade PHP ativa são fatores que impulsionam a evolução contínua do PHP.



As cinco principais linguagens de programação de back-end em todo o mundo

02

PHP



4 frameworks de PHP para diferentes usos

1. **Laravel** é um framework de aplicações web com sintaxe expressiva e elegante.
2. **Symfony** é tanto uma framework PHP quanto uma coleção de componentes PHP para a construção de sites.
3. **CodeIgniter** ajuda a construir aplicativos web rapidamente, já que tem configuração mínima.
4. **CakePHP** serve para o desenvolvimento rápido e limpo do PHP.



As cinco principais linguagens de programação de back-end em todo o mundo

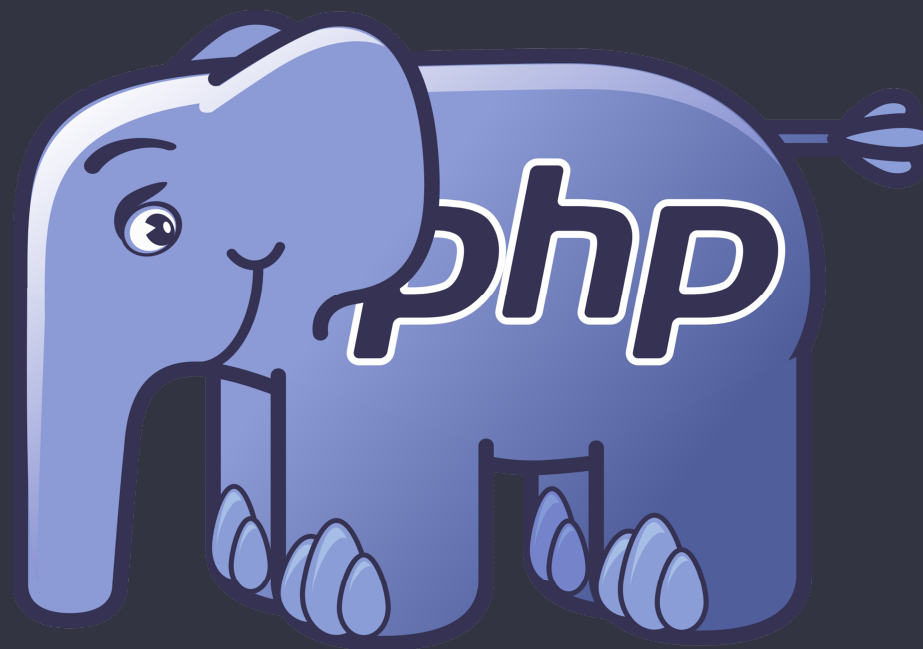
02

PHP



Características

- Código aberto e versátil
- Econômico
- Fácil de usar
- Excelente para iniciantes
- Segurança embutida





As cinco principais linguagens de programação de back-end em todo o mundo

02

JAVA



O índice 2021 TIOBE classifica Java em segundo lugar entre as tecnologias de desenvolvimento de backend hoje. Essa classificação indica que Java é uma das linguagens de programação mais influentes hoje. James Gosling inventou a linguagem de programação em 1991, mas não foi até 1995 que a Sun Microsystems a publicou.

Ao longo dos anos, o Java surgiu como a plataforma de escolha para desenvolvedores que preferem criar aplicativos da web com recursos personalizados e inovadores. Java também é útil para desenvolvimento móvel e aplicativos em dispositivos incorporados.



As cinco principais linguagens de programação de back-end em todo o mundo

02

JAVA



4 frameworks de JAVA para diferentes usos

1. **Spring** é uma alternativa leve mas poderosa para interagir com Enterprise Java (JEE) em diferentes cenários.

2. **Play.** é outra alternativa leve mas poderosa para interagir com Java e permite a criação de aplicações web modernas com foco em dispositivos móveis.



As cinco principais linguagens de programação de back-end em todo o mundo

02

JAVA



4 frameworks de JAVA para diferentes usos

3. **Struts** é uma solução robusta de código aberto para aplicações web, que segue o modelo MVC4.

4. **Hibernate** foi criado para oferecer mapeamento relacionado a objetos e trazer uma comunicação aperfeiçoado entre o Java e sistemas de gerenciamento de bancos de dados relacionais (RBDMS).



As cinco principais linguagens de programação de back-end em todo o mundo

02

JAVA



Características

- Escalável e direto
- Multi-Threading
- Bibliotecas de código aberto massivas
- Segurança aprimorada



✕ As cinco principais linguagens de programação de back-end em todo o mundo

02

RUBY

○ Ruby foi desenvolvido em 1990 por Yukihiro Matsumoto, um especialista japonês em computadores. A linguagem de programação que compartilha sintaxe semelhante com Python, Java e Perl também surgiu como uma das tecnologias de back-end líderes hoje. É uma plataforma popular para o desenvolvimento de aplicativos da web e é preferida pela maioria dos desenvolvedores para prototipagem.

Algumas das principais plataformas online, como Shopify, Airbnb e Etsy, usam Ruby para automatizar tarefas de back-end. Os recursos orientados a objetos, flexíveis e escaláveis o tornam a plataforma preferida.



As cinco principais linguagens de programação de back-end em todo o mundo

02

RUBY



4 frameworks de RUBY para diferentes usos

1. **Ruby on Rails** é um framework web de backend de Ruby que foi originalmente fundado em 2004.
2. **Roda** é outra estrutura Ruby confiável que fornece uma estrutura de desenvolvimento de aplicativo escalonável e uma boa variedade de APIs.



As cinco principais linguagens de programação de back-end em todo o mundo

02

RUBY



4 frameworks de RUBY para diferentes usos

3. **Camping** é uma das estruturas que menos ocupam espaço, e tem um design admiravelmente notável. Cria aplicativo completo – os projetos da Web através do Camping são perfeitamente capazes de ser aplicativos autônomos.

4. **Hanami** (anteriormente chamado de Lotus) é um framework notável que visa velocidade, eficiência e segurança.



As cinco principais linguagens de programação de back-end em todo o mundo

02

RUBY



Características

- Altamente produtivo
- Metaprogramação
- Biblioteca massiva
- Recursos de teste
- Rápido e confiável:





Exercício: CRUD



Objetivo: Criar um pequeno sistema em PHP + MySQL para cadastrar usuários em um banco de dados e exibir a lista de usuários cadastrados.

Passos:

1. Criar o banco de dados e a tabela:
 - Nome do banco: **exercicio_php_sql**.
 - Nome da tabela: **usuarios**.
2. Criar um formulário em PHP para cadastrar usuários.
3. Exibir a lista de usuários cadastrados.



Exercício: CRUD



Cadastro de Usuários

```
<h2>Cadastro de Usuários</h2>
```

```
<form method="POST">
```

```
  Nome: <input type="text" name="nome" required><br><br>
```

```
  Email: <input type="email" name="email" required><br><br>
```

```
  Senha: <input type="password" name="senha" required><br><br>
```

```
  <button type="submit">Cadastrar</button>
```

```
</form>
```



Exercício: CRUD



```
CREATE DATABASE exercicio_php_sql;
```

```
CREATE TABLE usuarios (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100),  
    email VARCHAR(100),  
    senha VARCHAR(255)  
);
```



Exercício: CRUD



Conexão com o banco de dados

```
<?php
// Conexão com o banco de dados
$conn = new mysqli("localhost", "root", "", "exercicio_php_sql");

// Verifica se houve erro na conexão
if ($conn->connect_error) {
    die("Falha na conexão: " . $conn->connect_error);
}
```




Exercício: CRUD



Verifica se o formulário foi enviado

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $nome = $_POST["nome"];  
    $email = $_POST["email"];  
    $senha = password_hash($_POST["senha"], PASSWORD_DEFAULT);
```

```
// Criptografa a senha
```



Exercício: CRUD



```
$sql = "INSERT INTO usuarios (nome, email, senha) VALUES ('$nome', '$email', '$senha')";
```

```
if ($conn->query($sql) === TRUE) {  
    echo "Usuário cadastrado com sucesso!";  
} else {  
    echo "Erro: " . $conn->error;  
}  
}
```

// Busca todos os usuários cadastrados

```
$buscar = "SELECT * FROM usuarios";  
$res = mysqli_query($conn, $buscar);
```



Exercício: CRUD



Lista de Usuários

```
<h2>Lista de Usuários</h2>
```

```
<table border="1">
```

```
<tr>
```

```
<th>ID</th>
```

```
<th>Nome</th>
```

```
<th>Email</th>
```

```
</tr>
```

```
<?php while ($row = mysqli_fetch_array($res)) {
```

```
    echo "<td>".$row['id']. "</td>
```

```
    <td>".$row['nome']. "</td>
```

```
    <td>".$row['email']. "</td>"; ?>
```



Exercício: CRUD



Lista de Usuários

```
<td><a href="editar.php?id=<?php echo $r['id']; ?>"> Editar</td>
<td><a href="excluir.php?id=<?php echo $r['id']; ?>"> Excluir </td>

</tr>

<?php } ?>

</table>
```



Exercício: CRUD



Passos:

1. Adicionar link para editar e excluir cada usuário.
2. Criar uma página para editar usuários, permitindo modificar nome e e-mail.
3. Implementar a funcionalidade de exclusão com uma confirmação antes de excluir.
4. Garantir que a **senha do usuário** não seja exibida ou alterada na listagem e na edição.



Exercício: CRUD



```
// Busca todos os usuários cadastrados
$buscar = "SELECT * FROM usuarios";
$res = mysqli_query($conn, $buscar);
```

```
<h2>Lista de Usuários</h2>
```

```
<table border="1">
```

```
<tr>
```

```
<th>ID</th>
```

```
<th>NOME</th>
```

```
<th>E-MAIL</th>
```

```
<th>EDITAR</th> <!-- ADD EDITAR E EXCLUIR -->
```

```
<th>EXCLUIR</th>
```

```
</tr>
```



Exercício: CRUD



```
<?php while ($row = mysqli_fetch_array($res)) {  
    echo "<td>".$row['id']. "</td>  
        <td>".$row['nome']. "</td>  
        <td>".$row['email']. "</td>"; ?>  
  
    <td><a href="editar.php?id=<?php echo $r['id']; ?>"> Editar</td>  
    <td><a href="excluir.php?id=<?php echo $r['id']; ?>"> Excluir </td>  
</tr>  
    <?php } ?>  
</table>
```



Exercício: CRUD



```
<?php
include "conexao.php";

$id = $_GET['id'] ?? "";
$usuario = null;

if ($id) {
    $result = $conn->query("SELECT * FROM usuarios WHERE id = $id");
    $usuario = $result->fetch_assoc();
}
```




Exercício: CRUD



```
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST["editar"])) {  
    $nome = htmlspecialchars($_POST["nome"]);  
    $email = htmlspecialchars($_POST["email"]);  
  
    $sql = "UPDATE usuarios SET nome='$nome', email='$email' WHERE id=$id";  
  
    if ($conn->query($sql) === TRUE) {  
        echo "Usuário atualizado com sucesso!";  
        header("Location: cadastro_usuarios.php");  
        exit();  
    } else {  
        echo "Erro: " . $conn->error;  
    }  
}
```



Exercício: CRUD



```
<body>
<h2>Editar Usuário</h2>
  <form method="POST">
    <label>Nome:</label>
    <input type="text" name="nome" value="<?php echo $usuario['nome']; ?>"
required><br><br>
    <label>E-mail:</label>
    <input type="email" name="email" value="<?php echo $usuario['email']; ?>"
required><br><br>
    <button type="submit" name="editar">Salvar Alterações</button>
  </form>
  <br>
  <a href="cadastro_usuarios.php">Voltar</a>
</body>
```



Exercício: CRUD



```
<?php
include "conexao.php";

$id = $_GET['id'] ?? "";

if ($id) {
    $sql = "DELETE FROM usuarios WHERE id = $id";
    if ($conn->query($sql) === TRUE) {
        echo "Usuário excluído com sucesso!";
    } else {
        echo "Erro: " . $conn->error;
    }
}
header("Location: cadastro_usuarios.php");
exit();
?>
```



Exercício: CRUD



//Dentro da tabela

```
<td><a href="excluir.php?id=<?php echo $r['id']; ?>"  
    onclick="return confirmarExclusao(<?php echo $r['id']; ?>)">Excluir</a></td>
```

//Antes do fechamento do Body

```
<script>  
function confirmarExclusao(id) {  
    let confirmacao = confirm("Tem certeza que deseja excluir o usuário de ID " +  
id + "?");  
    return confirmacao; // Retorna true para excluir, false para cancelar  
}  
</script>
```

**Até a próxima
aula!**

