



02

PRINCIPAIS FRAMEWORKS BACK-END



PRINCIPAIS FRAMEWORKS BACK-END

01



COMO OS FRAMEWORKS DE BACKEND DIFEREM DOS OUTROS

Frameworks de backend da Web são projetados especificamente para lidar com a lógica do lado do servidor de aplicativos da Web. Eles facilitam tarefas como interações de banco de dados, configuração de servidor, desenvolvimento de API e autenticação de usuário, enquanto frameworks de frontend focam em renderizar os elementos visuais e gerenciar a experiência do usuário no navegador.



PRINCIPAIS FRAMEWORKS BACK-END

01

VANTAGENS DOS FRAMEWORKS DE BACKEND



Robustez

Eles fornecem um ambiente estruturado e confiável para o desenvolvimento de aplicativos do lado do servidor, incluindo recursos integrados como roteamento, middleware e tratamento de erros, que ajudam a criar aplicativos estáveis e sustentáveis.



PRINCIPAIS FRAMEWORKS BACK-END

01

VANTAGENS DOS FRAMEWORKS DE BACKEND



Integrações

Frameworks de backend da Web são projetados para integrar-se perfeitamente com vários bancos de dados, serviços de terceiros e outras ferramentas. Eles oferecem bibliotecas e plugins que simplificam o processo de conexão com APIs externas, serviços de nuvem e arquiteturas de microsserviços. Essa capacidade de integração acelera o desenvolvimento e permite que os desenvolvedores criem aplicativos complexos de forma eficiente.



PRINCIPAIS FRAMEWORKS BACK-END

01

VANTAGENS DOS FRAMEWORKS DE BACKEND



Escalabilidade

Frameworks de backend da Web são construídos para lidar com cargas crescentes gerenciando recursos de forma eficiente e fornecendo recursos como balanceamento de carga e dimensionamento horizontal. Frameworks como Spring Boot e ASP.NET Core, por exemplo, oferecem suporte robusto para dimensionar aplicativos em vários servidores, garantindo que os aplicativos possam crescer para atender à demanda do usuário sem reengenharia significativa.



PRINCIPAIS FRAMEWORKS BACK-END

01

VANTAGENS DOS FRAMEWORKS DE BACKEND



Suporte à Comunidade

Frameworks de backend populares se beneficiam de comunidades grandes e ativas que contribuem para seu desenvolvimento e fornecem suporte. Esse envolvimento da comunidade garante uma riqueza de recursos, incluindo tutoriais, documentação e bibliotecas de terceiros. Isso também significa que os problemas são rapidamente identificados e resolvidos, e novos recursos são adicionados regularmente para manter os frameworks atualizados com os padrões da indústria.



PRINCIPAIS FRAMEWORKS BACK-END

01

VANTAGENS DOS FRAMEWORKS DE BACKEND



Segurança

A segurança é uma prioridade máxima para as estruturas de backend do `цyи`, que geralmente incluem medidas integradas para proteção contra vulnerabilidades comuns, como injeção de SQL, script entre sites e falsificação de solicitações entre sites.



PRINCIPAIS FRAMEWORKS BACK-END

01

10 Most Popular Backend Frameworks in 2025

Framework	Programming Language	Famous Use Cases
Django	Python	Instagram, Pinterest
Laravel	PHP	Laracasts, Cachet
Ruby on Rails	Ruby	GitHub, Shopify
Express.js	JavaScript (Node.js)	Uber, LinkedIn
Flask	Python	Netflix, Reddit
Spring Boot	Java	Amazon, eBay
Asp.NET Core	C#	Stack Overflow, Microsoft
Koa JS	JavaScript (Node.js)	Bulb, Gofile
Nest JS	TypeScript (Node.js)	Trilon, Adidas
Phoenix	Elixir	Discord, Bleacher Report

INVERITA



PRINCIPAIS FRAMEWORKS BACK-END

01

Django



Django é considerado o framework de backend mais popular por muitos programadores. É um framework web Python de código aberto, que facilita o desenvolvimento rápido com código conciso e um design elegante. Sua sintaxe amigável simplifica o processo de desenvolvimento, e sua arquitetura MVT (Model-View-Template) garante um design de aplicativo web visualmente atraente e altamente eficiente.

django



PRINCIPAIS FRAMEWORKS BACK-END

01

Django



Oferecendo recursos de segurança robustos e escalabilidade, Django é frequentemente chamado de framework de backend mais rápido, versátil e adequado para projetos que variam de sites simples de uma página a sistemas de computação científica complexos.

django



PRINCIPAIS FRAMEWORKS BACK-END

01

Django



Principais características:

- ORM poderoso: Interação com bancos de dados de forma simplificada.
- Painel administrativo pronto: Facilita a gestão de dados.
- Segurança embutida: Protege contra SQL Injection, XSS e CSRF



PRINCIPAIS FRAMEWORKS BACK-END

01

Django



✓ Vantagens:

- Muito rápido para desenvolvimento.
- Segurança avançada.
- Alta escalabilidade.

✗ Desvantagens:

- Pode ser pesado para pequenos projetos.
- Maior curva de aprendizado do que Flask.

The Django logo, featuring the word "django" in a white, lowercase, sans-serif font, set against a dark green rounded rectangular background.



PRINCIPAIS FRAMEWORKS BACK-END

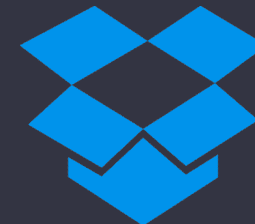
01

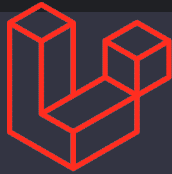


Django

Empresas que usam:

- Instagram – Backend escalável.
- Spotify – Gestão de usuários e dados.
- Dropbox – Infraestrutura inicial baseada em Django.



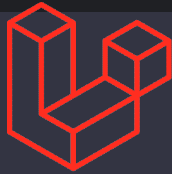


PRINCIPAIS FRAMEWORKS BACK-END

01

Laravel

O Laravel também é um dos frameworks de backend mais usados. É um framework web de código aberto baseado em PHP conhecido por seus recursos avançados como roteamento, middleware, mecanismo de template e sistema Object-Relational Mapping (ORM), simplificando o desenvolvimento web. Com sua sintaxe elegante e ferramentas amigáveis, o Laravel se tornou uma escolha popular entre desenvolvedores PHP para criar sites totalmente funcionais com lógica personalizada.

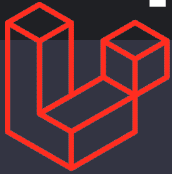


PRINCIPAIS FRAMEWORKS BACK-END

01

Laravel

É uma estrutura da web PHP de código aberto para o desenvolvimento de aplicativos da web baseados em Symfony que seguem a arquitetura model – view – controller (MVC). Ele oferece um sistema de empacotamento modular equipado com um gerenciador de dependências dedicado. O Laravel também fornece a seus usuários várias maneiras de acessar bancos de dados relacionais, juntamente com manutenção de aplicativos e utilitários de implantação. O Laravel possui uma licença MIT e um código-fonte hospedado no GitHub.



PRINCIPAIS FRAMEWORKS BACK-END

01

Laravel

Principais características:

- Blade Template Engine: Facilita criação de interfaces dinâmicas.
- Eloquent ORM: Simplifica interações com bancos de dados.
- Sistema de migração: Controle fácil de versões do banco de dados.



PRINCIPAIS FRAMEWORKS BACK-END

01

Laravel

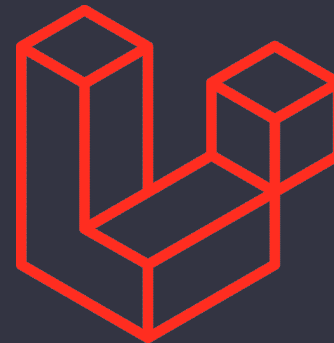


✓ Vantagens:

- Fácil de aprender.
- Rápido para desenvolvimento.
- Código organizado e legível.

✗ Desvantagens:

- Pode ser pesado para aplicações muito grandes.
- Menos performático do que Node.js e Go.





PRINCIPAIS FRAMEWORKS BACK-END

01

Laravel



Empresas que usam:

- BBC – Gestão de conteúdo.
- 9GAG – Backend da plataforma.
- Toyota – Aplicações internas.



TOYOTA



PRINCIPAIS FRAMEWORKS BACK-END

01

Ruby on Rails



Ruby on Rails é um framework Model View Controller (MVC) de código aberto escrito em Ruby com uma licença MIT, projetado para desenvolvimento de aplicativos web e integração de servidores web. Ele segue o princípio "não se repita" (DRY) para reduzir a duplicação de código e o princípio "convenção sobre configuração", que fornece definições de configuração padrão que geralmente não precisam de ajustes.



PRINCIPAIS FRAMEWORKS BACK-END

01

Ruby on Rails



🚀 Principais características:

- Active Record ORM: Mapeia automaticamente tabelas do banco de dados para objetos Ruby.
- Gems: Ecossistema rico de bibliotecas para adicionar funcionalidades rapidamente.
- Estrutura de código padronizada: Facilita colaboração entre desenvolvedores.



PRINCIPAIS FRAMEWORKS BACK-END

01

Ruby on Rails



✓ Vantagens:

- Desenvolvimento muito rápido e produtivo.
- Código limpo e fácil de manter.
- Comunidade ativa e muitas bibliotecas disponíveis.



✗ Desvantagens:

- Velocidade de desenvolvimento: Desempenho inferior a frameworks como Django e Express.js.
- Falta de flexibilidade: seu projeto exigir funcionalidades exclusivas, e os módulos padrão podem limitar sua flexibilidade.



PRINCIPAIS FRAMEWORKS BACK-END

01

Ruby on Rails



Empresas que usam:

- Airbnb – Backend da plataforma de hospedagem.
- GitHub – Versão inicial foi desenvolvida em Rails.
- Shopify – Infraestrutura de e-commerce.





PRINCIPAIS FRAMEWORKS BACK-END

01

Express JS



Express.js é um dos frameworks de backend mais populares. É um framework de aplicativo web versátil e leve para Node.js, conhecido por sua simplicidade e flexibilidade. Seguindo a filosofia minimalista, o Express.js fornece aos desenvolvedores um conjunto robusto de recursos para a construção de aplicativos web e APIs. Sua natureza imparcial permite que os desenvolvedores estruturam seus aplicativos de acordo com suas preferências, permitindo desenvolvimento rápido e fácil escalabilidade.



PRINCIPAIS FRAMEWORKS BACK-END

01

Express.js



O Express.js é um Framework rápido e um dos mais utilizados em conjunto com o Node.js, facilitando no desenvolvimento de aplicações back-end e até, em conjunto com sistemas de templates, aplicações full-stack. É utilizado por diversas empresas ao redor do mundo, dentre elas a Fox Sports, PayPal, IBM, Uber, entre outras.

express



PRINCIPAIS FRAMEWORKS BACK-END

01

Express.js

express



✈ Principais características:

- Middleware: Permite manipular requisições antes de enviá-las ao servidor.
- Roteamento flexível: Facilita a criação de endpoints RESTful.
- Integração com bancos de dados: Suporte a MongoDB, PostgreSQL, MySQL e outros via ORM como Sequelize ou Mongoose.



PRINCIPAIS FRAMEWORKS BACK-END

01

Express.js



✓ Vantagens:

- Leve e rápido.
- Fácil aprendizado e adoção.
- Grande suporte da comunidade.

express

✗ Desvantagens:

- Código pode se tornar desorganizado sem um bom padrão.
- Não possui muitas funcionalidades embutidas, exigindo dependências extras.



PRINCIPAIS FRAMEWORKS BACK-END

01

Express.js



Empresas que usam:

- Uber – Backend de tempo real.
- PayPal – Processamento de pagamentos.
- Netflix – Microserviços.

Uber



NETFLIX



PRINCIPAIS FRAMEWORKS BACK-END

01

Flask



Flask é uma boa alternativa ao Django para aplicativos menores que podem ser ampliados se necessário. Ele é leve e fácil de usar, o que o torna ótimo para projetos menores, mas também é poderoso o suficiente para lidar com tarefas maiores, se necessário. Com Flask, você obtém simplicidade e flexibilidade em um pacote, junto com amplo suporte da comunidade e documentação para orientá-lo.



PRINCIPAIS FRAMEWORKS BACK-END

01

Flask



Com o Flask, os desenvolvedores têm acesso a uma ampla gama de extensões e bibliotecas, permitindo que adicionem funcionalidades como autenticação, integração de banco de dados e APIs RESTful com facilidade. Seu servidor de desenvolvimento integrado e depurador embutido facilitam o teste e a depuração de aplicativos durante o desenvolvimento.



PRINCIPAIS FRAMEWORKS BACK-END

01

Flask



Principais características:

- Estrutura Leve e Flexível: É um microframework minimalista, oferecendo apenas o essencial para criar aplicações web.
- Roteamento Simples e Suporte a Templates: Facilita a criação de rotas e o mapeamento de URLs, além de usar o Jinja2 para gerar páginas dinâmicas.
- Extensível e Fácil de Integrar com Bancos de Dados: Oferece suporte fácil para integração com bancos de dados e extensões, como SQLAlchemy.



PRINCIPAIS FRAMEWORKS BACK-END

01

Flask



✓ Vantagens:

- Leve e rápido para APIs pequenas.
- Extremamente flexível.
- Facilidade de aprendizado.

✗ Desvantagens:

- Requer mais configuração do que Django.
- Menos adequado para grandes aplicações.





PRINCIPAIS FRAMEWORKS BACK-END

01

Flask



Empresas que usam:

- Reddit - Para algumas partes de seu backend.
- Pinterest - Para criar e gerenciar seus microsserviços.
- Netflix – Para APIs e Microsserviços.



NETFLIX



PRINCIPAIS FRAMEWORKS BACK-END

01

Spring Boot



Spring Boot é uma estrutura de código aberto que aprimora o desenvolvimento Spring automatizando a configuração, tornando mais rápido e fácil criar e implantar aplicativos Spring. Diferentemente do framework Spring tradicional, que prioriza a flexibilidade, o Spring Boot foca em reduzir a complexidade do código e simplificar o desenvolvimento de aplicativos web. Ele usa anotações e configurações predefinidas para acelerar o processo de desenvolvimento, permitindo a criação de aplicativos autônomos com configuração mínima necessária.



PRINCIPAIS FRAMEWORKS BACK-END

01

Spring Boot



Principais características:

- Arquitetura modular: Suporte a microsserviços.
- Segurança avançada: Spring Security integrado.
- Integração fácil com bancos de dados e mensageria (Kafka, RabbitMQ).



PRINCIPAIS FRAMEWORKS BACK-END

01

Spring Boot



✓ Vantagens:

- Extremamente seguro e confiável.
- Suporte empresarial forte (usado por bancos e fintechs).
- Ideal para aplicações de larga escala.

✗ Desvantagens:

- Curva de aprendizado maior devido à complexidade.
- Pode consumir mais recursos do que frameworks minimalistas.



PRINCIPAIS FRAMEWORKS BACK-END

01

Spring Boot



Empresas que usam:

- eBay – Facilidade na Criação de APIs REST.
- Amazon – Infraestrutura de serviços.
- Alibaba – Backend de e-commerce.





EXERCICIO DE FRONT-END / BACK-END



Estruturas Condicionais:

1. Crie um arquivo chamado exercicio2.php.
2. Defina uma variável idade com um valor inteiro.
3. Use if, else if e else para exibir mensagens como:
 - "Você é menor de idade" (se idade for menor que 18).
 - "Você é adulto" (se idade estiver entre 18 e 60).
 - "Você é idoso" (se idade for maior que 60).



EXERCICIO DE FRONT-END / BACK-END



Estruturas Condicionais:

```
<?php
// Definir a variável idade
$idade = 25; // Você pode mudar esse valor para testar diferentes situações
// Estrutura condicional para verificar a faixa etária
if ($idade < 18) {
    echo "Você é menor de idade.";
} elseif ($idade >= 18 && $idade <= 60) {
    echo "Você é adulto.";
} else {
    echo "Você é idoso.";
}
?>
```



EXERCICIO DE FRONT-END / BACK-END



Receber os dados do usuário:

```
<?php
// Inicializa a variável da mensagem como vazia
$mensagem = "";

// Verifica se o formulário foi enviado
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Obtém e converte a idade para inteiro
    $idade = intval($_POST["idade"]);
```



EXERCICIO DE FRONT-END / BACK-END



Receber os dados do usuário:

```
// Verifica a faixa etária
if ($idade < 18) {
    $mensagem = "Você é menor de idade.";
} elseif ($idade >= 18 && $idade <= 60) {
    $mensagem = "Você é adulto.";
} else {
    $mensagem = "Você é idoso.";
}
}
?>
```




EXERCICIO DE FRONT-END / BACK-END



🚀 Criando uma página WEB para receber os dados:

```
<h2>Verificação de Idade</h2>
<!-- Formulário -->
<form action="" method="POST">
  <label for="idade">Digite sua idade:</label>
  <input type="number" name="idade" id="idade" required>
  <button type="submit">Enviar</button>
</form>
```



EXERCICIO DE FRONT-END / BACK-END



 **Exibindo a mensagem:**

```
<!-- Exibe a mensagem se houver -->
<?php if (!empty($mensagem)) : ?>
    <div class="resultado">
        <p><?php echo $mensagem; ?></p>
    </div>
<?php endif; ?>
```



EXERCICIO DE FRONT-END / BACK-END



 O código fica assim:

teste.php / num / head

```
1  <?php
2  // Inicializa a variável da mensagem como vazia
3  $mensagem = "";
4
5  // Verifica se o formulário foi enviado
6  if ($_SERVER["REQUEST_METHOD"] == "POST") {
7      // Obtém e converte a idade para inteiro
8      $idade = intval($_POST["idade"]);
9
10     // Verifica a faixa etária
11     if ($idade < 18) {
12         $mensagem = "Você é menor de idade.";
13     } elseif ($idade >= 18 && $idade <= 60) {
14         $mensagem = "Você é adulto.";
15     } else {
16         $mensagem = "Você é idoso.";
17     }
18 }
19 ?>
```

```
20
21 <!DOCTYPE html>
22 <html lang="pt-BR">
23 <head>
24     <meta charset="UTF-8">
25     <meta name="viewport" content="width=device-width, initial-scale=1.0">
26     <title>Verificação de Idade</title>
27 </head>
28 <body>
29
30     <h2>Verificação de Idade</h2>
31
32     <!-- Formulário -->
33     <form action="" method="POST">
34         <label for="idade">Digite sua idade:</label>
35         <input type="number" name="idade" id="idade" required>
36         <button type="submit">Enviar</button>
37     </form>
38
39     <!-- Exibe a mensagem se houver -->
40     <?php if (!empty($mensagem)) : ?>
41         <div class="resultado">
42             <p><?php echo $mensagem; ?></p>
43         </div>
44     <?php endif; ?>
45
46 </body>
47 </html>
```



EXERCICIO DE FRONT-END / BACK-END



 **RESULTADO:**

Verificação de Idade

Digite sua idade:

Enviar

Você é menor de idade.

**Até a próxima
aula!**

