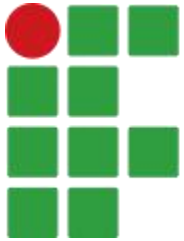
Frameworks para desenvolvimento back-end

Curso: Análise e Desenvolvimento de Sistemas

Componente: Programação para a Internet II 5º Período

Professor: Lucas Sampaio Leite



INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Pernambuco

Objetivos e competências

- Apresentar o papel dos frameworks back-end no desenvolvimento web.
- Analisar o funcionamento de uma aplicação simples desenvolvida em Flask.
- Ao final da aula, o aluno deverá ser capaz de:
  - Diferenciar front-end e back-end em uma aplicação web;
  - Entender a motivação para o uso de frameworks back-end;
  - Ser capaz de comparar frameworks de acordo com a necessidade da aplicação;
  - Compreender a filosofia do Flask como microframework;
  - Identificar os conceitos básicos do Flask (aplicação, rotas e views).
  - Analisar o funcionamento de uma aplicação Flask simples.

Revisando conceitos

- O que é front-end?

Fonte: elemento Canva

Revisando conceitos

- O front-end é a parte do desenvolvimento web que **trata da construção da interface e interação com o usuário**, ou seja: a camada que roda no navegador e que efetivamente é vista e utilizada pelo usuário (QUEIRÓS; PORTELA, 2020).

Fonte: elemento Canva

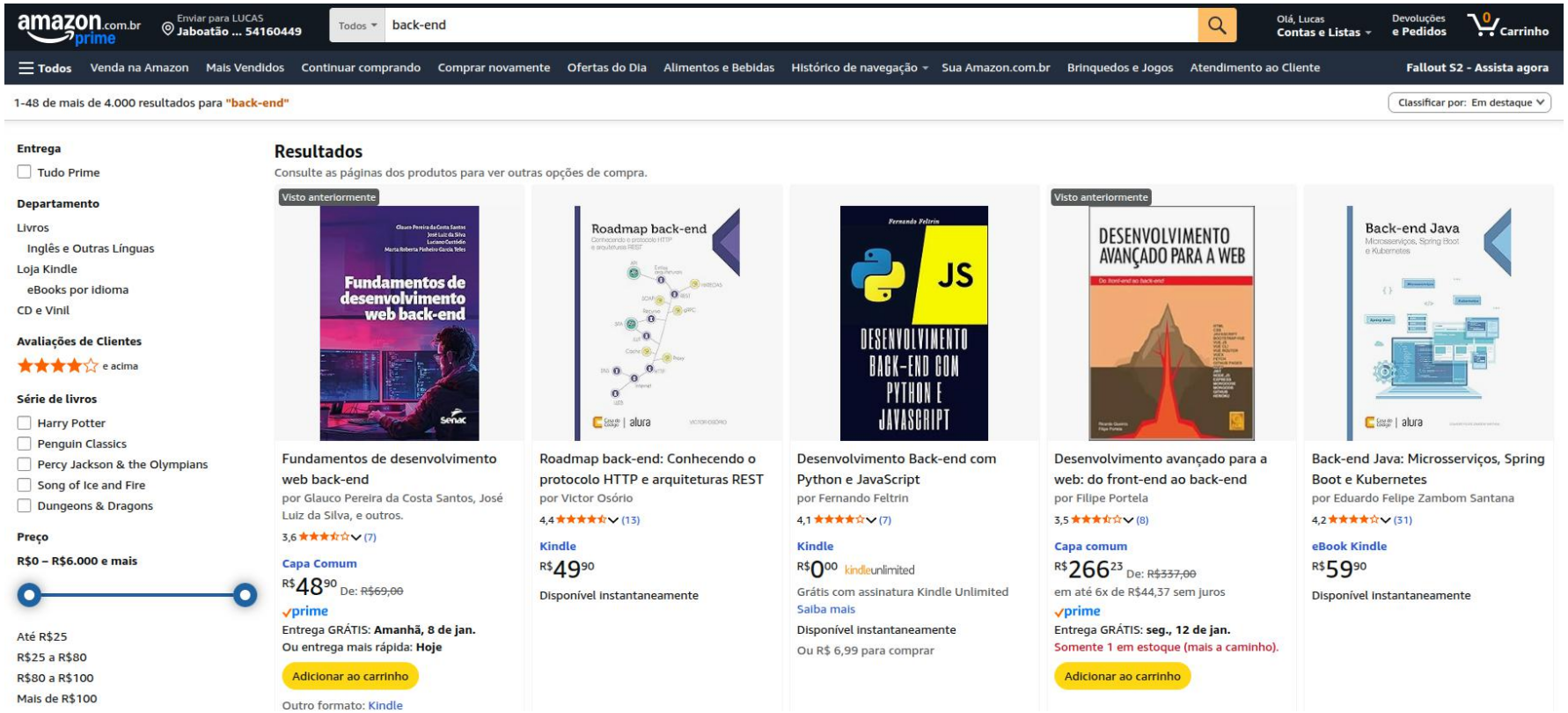
Revisando conceitos

- O back-end corresponde à parte não visível ao usuário final, sendo a camada da aplicação web **responsável pelo processamento no lado do servidor** (QUEIRÓS; PORTELA, 2020).

Fonte: elemento Canva

Revisando conceitos

- Quando um usuário **adiciona um produto ao carrinho** e **realiza um pagamento** em um site de vendas, onde termina o front-end e onde começa o back-end da aplicação?



Fonte: https://www.amazon.com.br/

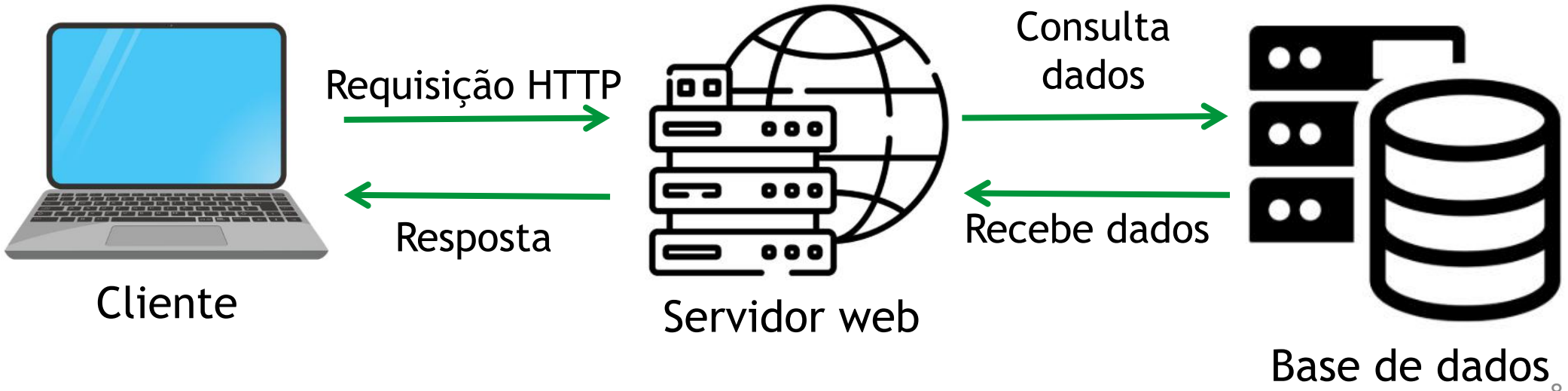
Revisando conceitos

- Se o front-end executa no navegador e o back-end no servidor, separados por toda a internet...
  - Como essas duas partes se comunicam em tempo real?
  - O que, afinal, conecta esses dois mundos?

Fonte: elemento Canva

Revisando conceitos

- Se o front-end executa no navegador e o back-end no servidor, separados por toda a internet...
  - Como essas duas partes se comunicam em tempo real?
  - O que, afinal, conecta esses dois mundos?



Fonte: elaborado pelo autor com ementos Canva

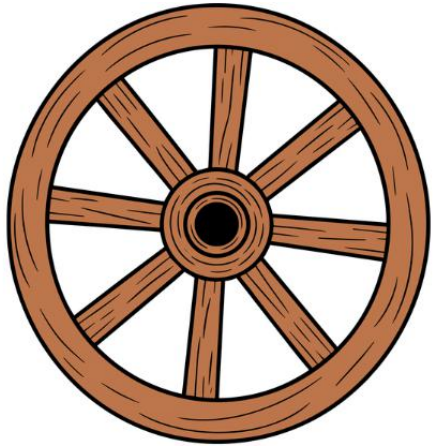
Motivação

- Considerando uma aplicação web que deve responder a múltiplas URLs, como evitar a implementação manual de todo o processamento de requisições HTTP?

Fonte: elemento Canva

Motivação

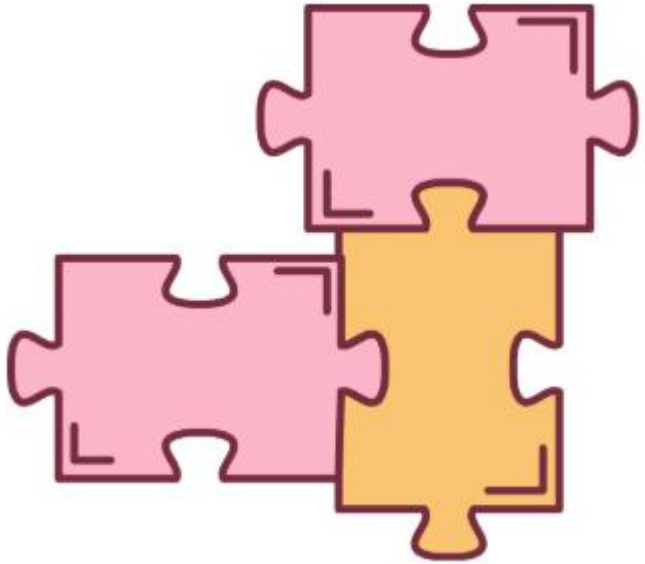
- Considerando uma aplicação web que deve responder a múltiplas URLs, como evitar a implementação manual de todo o processamento de requisições HTTP?
  - Sistemas web estão em todo lugar;
  - Desenvolver tudo do zero é custoso;
  - Necessidade de produtividade e organização.


Fonte: elemento Canva

Conceito de Framework

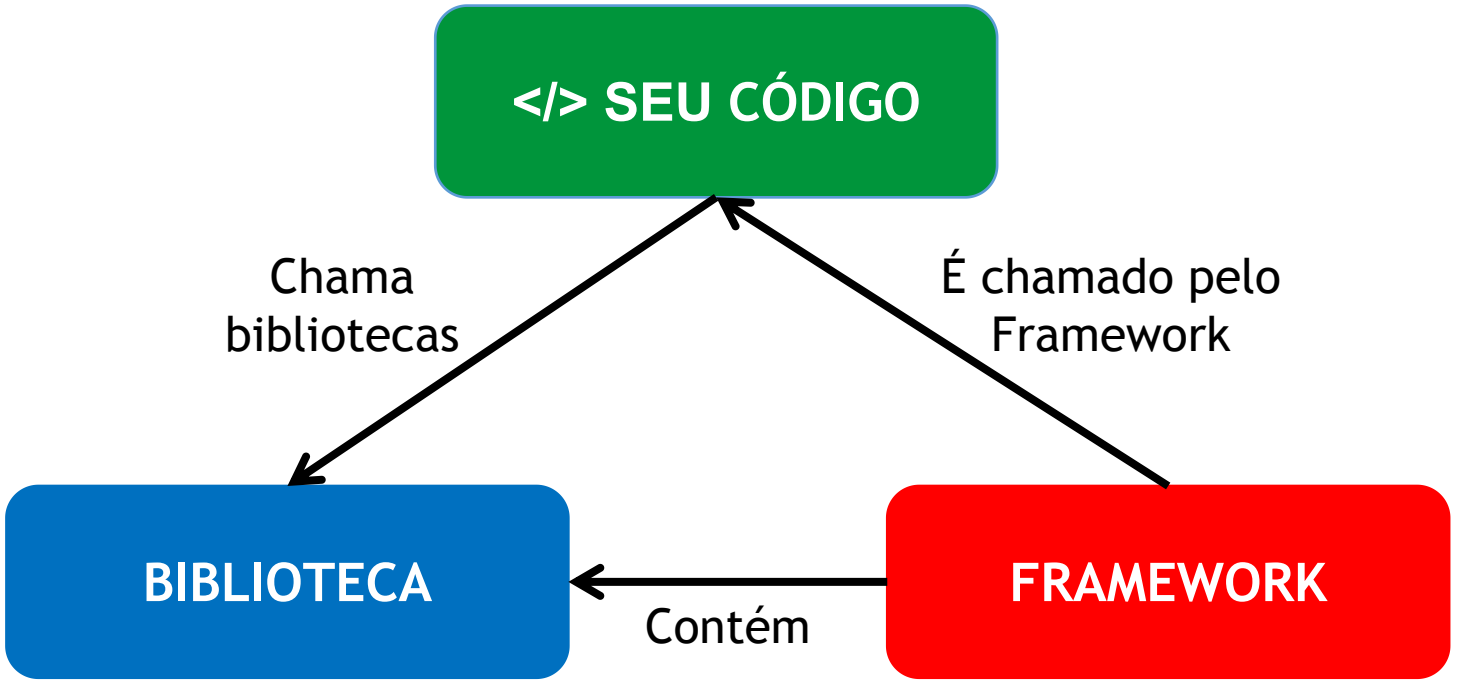
- Um framework é uma coleção de componentes **reutilizáveis de software** que fornecem **funcionalidades genéricas**, podendo ser **adaptadas** para o desenvolvimento de aplicações específicas (SOMMERVILLE, 2019).



Fonte: elemento Canva

Frameworks x bibliotecas

- Biblioteca: o programador controla o fluxo;
- Framework: o framework controla o fluxo;
- Inversão de controle (IoC).

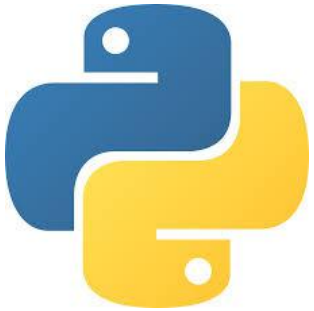


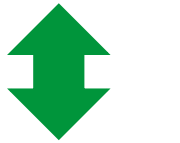
Fonte: elaborado pelo autor

Frameworks back-end

- Frameworks back-end:
  - Controlam o fluxo da aplicação no servidor -> (IoC);
  - Tratam requisições HTTP e respostas;
  - Facilitam acesso a banco de dados;
  - Oferecem padrões e estruturas de desenvolvimento.
- Variam em:
  - nível de abstração;
  - flexibilidade;
  - complexidade;
  - recursos prontos.

Frameworks back-end





Flask

Fonte: adaptado pelo autor a partir de logotipos disponíveis nos sites oficiais

Frameworks back-end

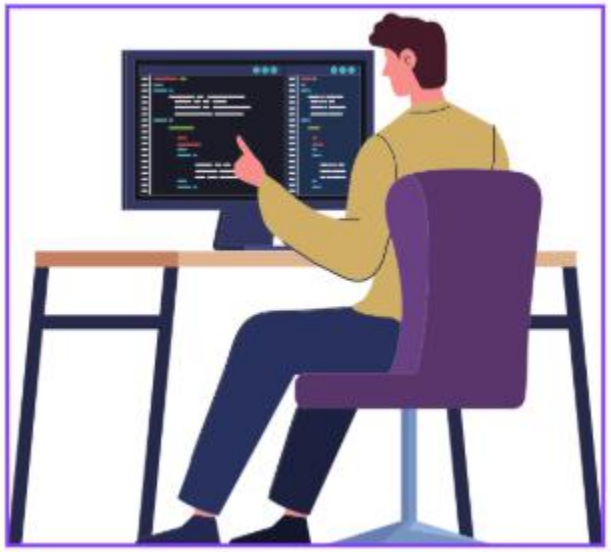
- Leve e simples: ideal para aprender back-end e entender cada camada;
- Flexível: você escolhe as bibliotecas conforme a necessidade;
- Curva de aprendizado menor para projetos pequenos;
- Pouca “mágica”: código mais explícito e fácil de entender.

- Tudo pronto: ORM, autenticação, painel admin e segurança
- Alta produtividade em sistemas médios e grandes
- Admin automático poderoso
- Padrões bem definidos, com menos decisões arquiteturais

Fonte: adaptado pelo autor a partir de logotipos disponíveis nos sites oficiais

Frameworks back-end

- Imaginem que vocês foram contratados para desenvolver um sistema web para uma empresa, com cadastro de usuários, autenticação, permissões de acesso e um painel administrativo para gerenciamento dos dados.
  - O sistema deve ser seguro, escalável e desenvolvido em um prazo curto.



Fonte: elemento Canva

Frameworks back-end

- Imaginem agora que o objetivo seja criar uma API simples, que será consumida por um aplicativo mobile ou por um front-end em React, com poucas regras de negócio e foco em desempenho.

Fonte: elemento Canva

Frameworks back-end

- Maior produtividade;
- Código organizado;
- Segurança embutida;
- Facilidade de manutenção.



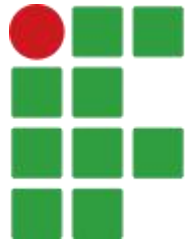

Fonte: elemento Canva

Introdução ao Flask

- O Flask é um microframework web em Python baseado no padrão WSGI (PALLETS PROJECTS, 2026).
- Principais características:
  - Microframework web em Python para criação de aplicações web;
  - Leve e minimalista: inicia simples, sem estruturas complexas;
  - Extensível: adiciona apenas os componentes necessários;
  - Fácil e rápido de aprender, ideal para projetos pequenos e médios;
  - Baseado em WSGI, padrão de comunicação web em Python.

Fonte: (PALLETS PROJECTS, 2026)

Introdução ao Flask



Project Links

Donate
PyPI Releases
Source Code
Issue Tracker
Chat

Contents

Welcome to Flask
- User's Guide
- API Reference
- Additional Notes

Quick search

Go

Flask

Welcome to Flask's documentation. Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.

Get started with Installation and then get an overview with the Quickstart. There is also a more detailed Tutorial that shows how to create a small but complete application with Flask. Common patterns are described in the Patterns for Flask section. The rest of the docs describe each component of Flask in detail, with a full reference in the API section.

Flask depends on the Werkzeug WSGI toolkit, the Jinja template engine, and the Click CLI toolkit. Be sure to check their documentation as well as Flask's when looking for information.

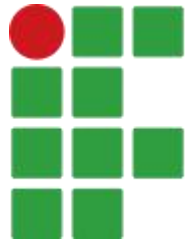
User's Guide

Flask provides configuration and conventions, with sensible defaults, to get started. This section of the documentation explains the different parts of the Flask framework and how they can be used, customized, and extended. Beyond Flask itself, look for community-maintained extensions to add even more functionality.

- Installation
  - Python Version
  - Dependencies
  - Virtual environments
  - Install Flask
- Quickstart
  - A Minimal Application

Documentação: https://flask.palletsprojects.com/en/stable/

Introdução ao Flask



- Verificar a versão do python: Python: >= 3.8
  - python --version
  - Exemplo de retorno: Python 3.10.12
- Verificar se o pipenv está instalado:
  - pipenv --version
  - Exemplo de retorno: pipenv, version 2024.4.1
- Se o pipenv não estiver instalado:
  - pip install pipenv
- Instalando o Flask:
  - pipenv install flask

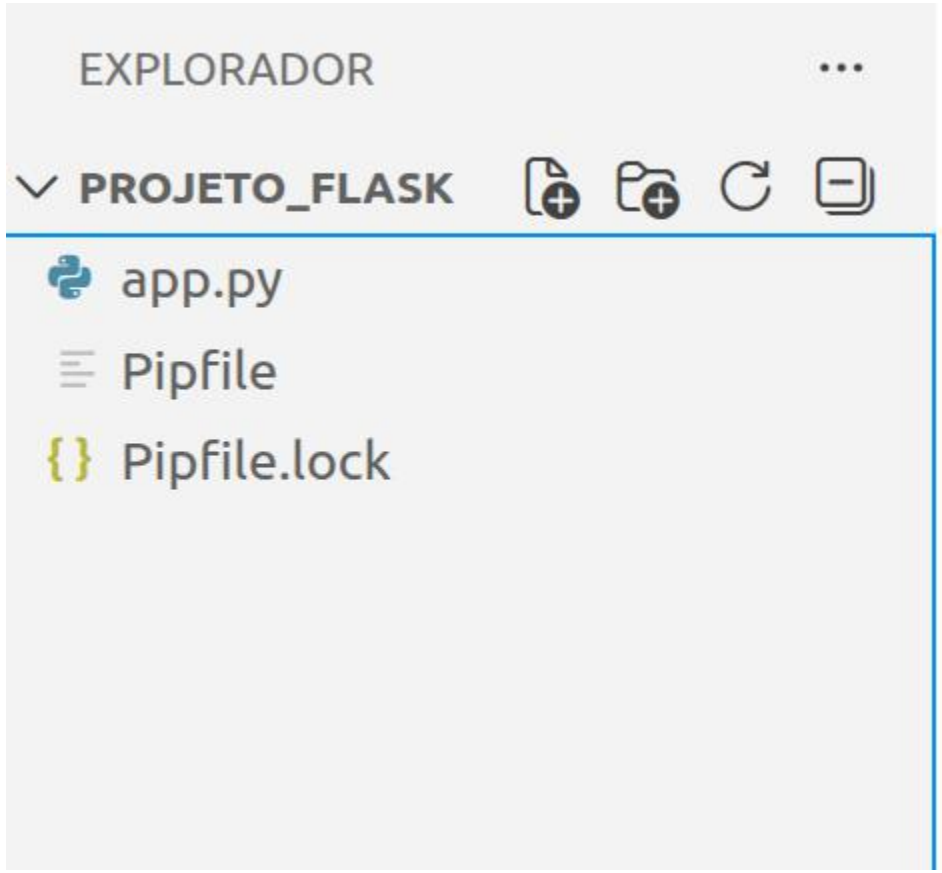
Introdução ao Flask

- Dependências do Flask:
  - pipenv graph

Introdução ao Flask

- Componentes principais do Flask (GRINBERG, 2018):
  - Aplicação Flask:
    - É o objeto principal da aplicação;
    - Responsável por configurar e iniciar o servidor, centralizar rotas, configurações e extensões.
  - Rotas (Routes):
    - Associam uma URL a uma funcionalidade;
    - Suportam diferentes métodos HTTP (GET, POST, etc.).
  - Funções de Visão (Views):
    - Processam a requisição;
    - Retornam uma resposta HTTP (HTML, JSON, texto, etc.).

Introdução ao Flask

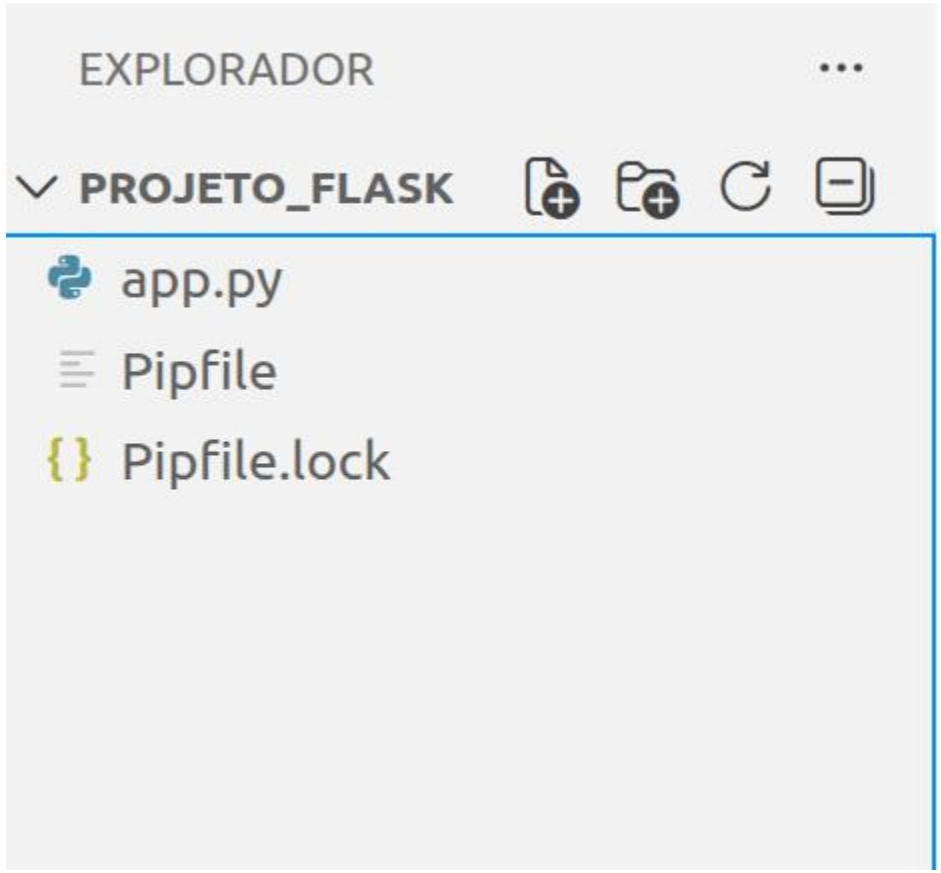


app.py

app.py > ...

```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

```

Introdução ao Flask



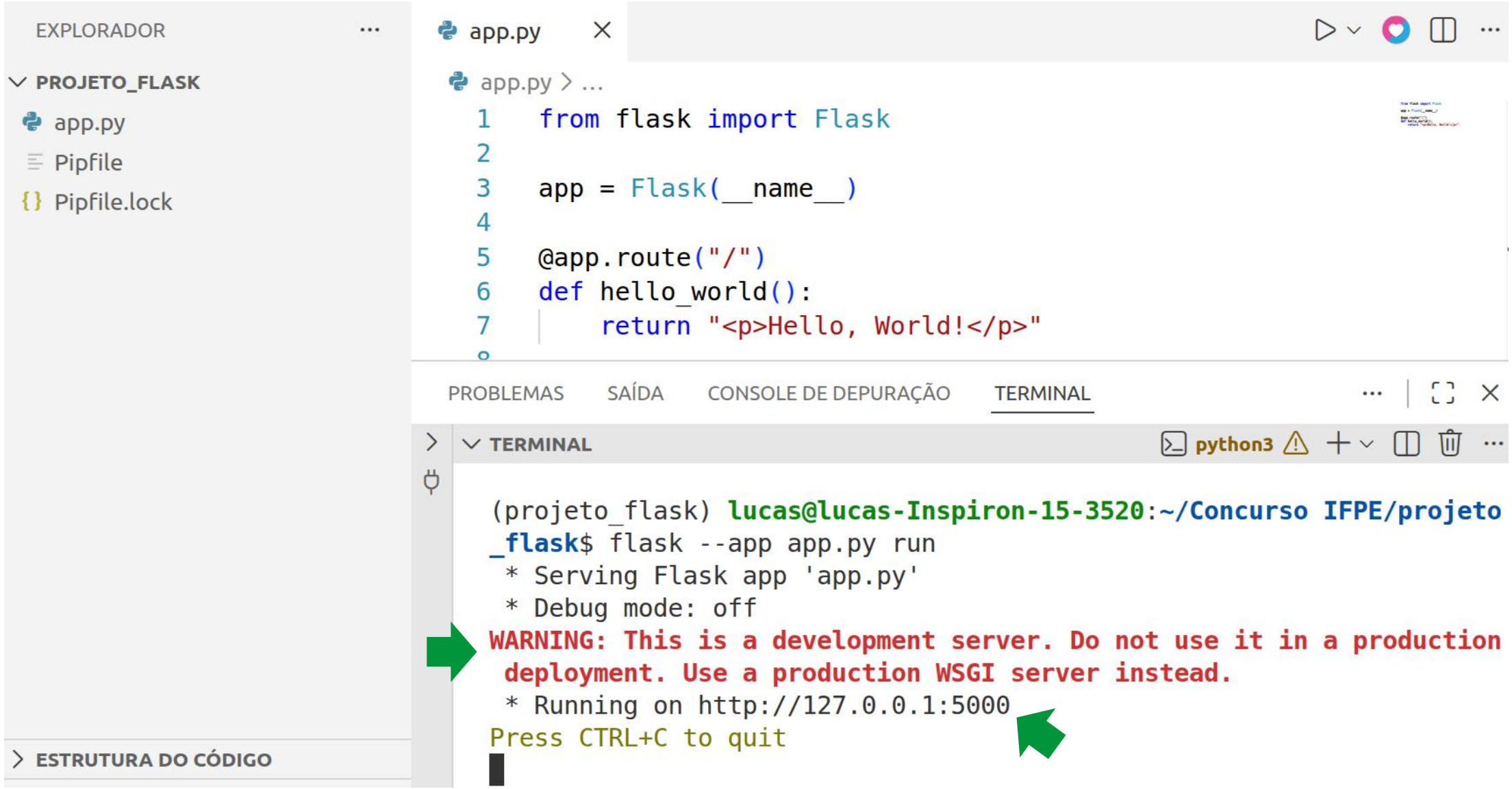
app.py

app.py > ...

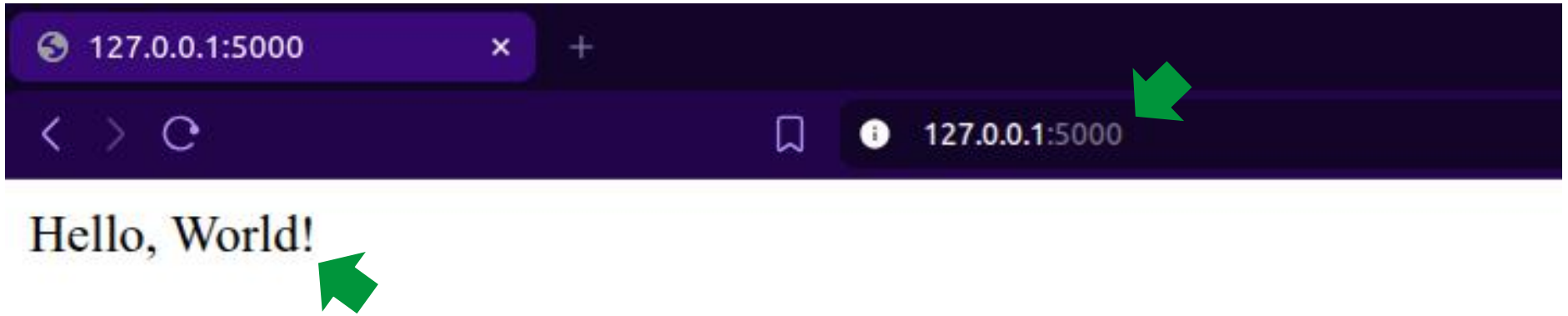
```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

```

- Executando a aplicação: flask --app app.py run

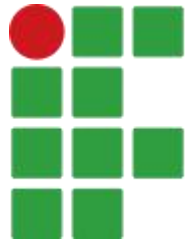
Introdução ao Flask



Introdução ao Flask



Introdução ao Flask



- Fluxo de uma requisição no Flask:
  - Recebimento da requisição HTTP → GET /
  - Roteamento para a função de visão:

```
@app.route("/")
def hello_world():
```

  - Processamento e resposta:

```
def hello_world():
    return "<p>Hello, World!</p>"
```

Introdução ao Flask

app.py > ...

```python
from flask import Flask

app = Flask(__name__)

@app.route("/hello")
def hello_world():
    return "<p>Hello, World!</p>"

```

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

Introdução ao Flask

app.py > ...

```python
from flask import Flask

app = Flask(__name__)

@app.route("/hello")
def hello_world():
    return "<p>Hello, World!</p>"

```

Introdução ao Flask

app.py > ...

```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

@app.route("/<user>")
def hello_user(user):
    return f"<p>Hello, {user}!</p>"

```

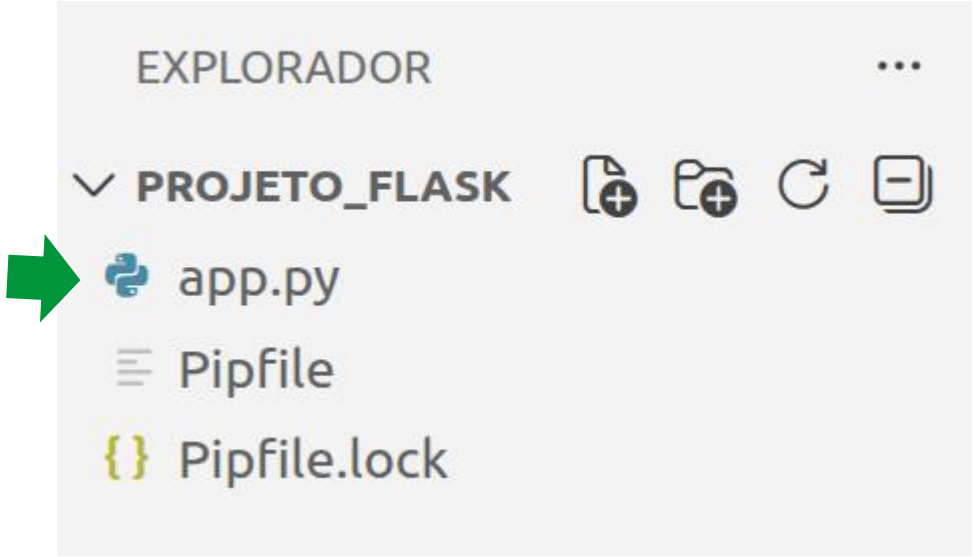
Introdução ao Flask

app.py > ...

```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

@app.route("/<user>")
def hello_user(user):
    return f"<p>Hello, {user}!</p>"
```

Introdução ao Flask

- Não há uma estrutura obrigatória de pastas;
- Basta começar com um único arquivo (app.py);
- A organização do projeto é flexível;
- A estrutura cresce conforme a necessidade.



Introdução ao Flask

- Não há uma estrutura obrigatória de pastas;
- Basta começar com um único arquivo (app.py);
- A organização do projeto é flexível;
- A estrutura cresce conforme a necessidade.


```
meu_projeto/
├── manage.py
├── meu_projeto/
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
└── teste/
    ├── __init__.py
    ├── apps.py
    ├── models.py
    └── views.py
```


Fonte: adaptado pelo autor a partir de logotipos disponíveis nos sites oficiais

Introdução ao Flask

- Banco de Dados:
  - Flask-SQLAlchemy – ORM mais usado com Flask
  - Flask-Migrate – Controle de versões do banco
  - Flask-MongoEngine – Integração com MongoDB
  - Flask-MySQLdb – Integração com MySQL
- Autenticação e Autorização:
  - Flask-Login – Gerenciamento de sessão de usuários
  - Flask-Security – Autenticação, autorização e roles (usa Flask-Login)
  - Flask-JWT-Extended – Autenticação via JWT (APIs REST)
  - Flask-Bcrypt – Hash seguro de senhas

Fonte: (PALLETS PROJECTS, 2026)

Síntese

- Back-end é responsável pela lógica da aplicação e pelo processamento no servidor.
- Frameworks surgem para evitar retrabalho e padronizar soluções comuns no desenvolvimento back-end.
- O Flask se destaca por ser um microframework leve, flexível e extensível, permitindo que o desenvolvedor construa a aplicação de forma gradual.

Próxima aula

- Compreender e manipular códigos de status HTTP em respostas do Flask.
- Explorar o Postman como ferramenta para testar APIs.
- Aprender a usar os verbos HTTP (GET, POST, PUT, DELETE) em rotas do Flask.

Conclusão

- Frameworks back-end facilitam o desenvolvimento e manutenção de aplicações web.
- Flask é um microframework leve e flexível, ideal para projetos rápidos ou aprendizagem de conceitos básicos.
- Conhecer frameworks permite escolher a ferramenta certa para cada projeto.

Atividade avaliativa

- Duas questões objetivas e 3 subjetivas abordando o conteúdo visto na aula.
- Enviar para lucas.sampaio.leite@gmail.com.

QR-CODE Atividade

Referências

- QUEIRÓS, Ricardo; PORTELA, Filipe. Desenvolvimento avançado para a web: do front-end ao back-end. São Paulo: FCA, 2020. ISBN 978-9727229154.
- PALLETS PROJECTS. Flask Documentation. Disponível em: <https://flask.palletsprojects.com/en/stable/\>. Acesso em: 6 jan. 2026.
- SOMMERVILLE, Ian. Engenharia de Software. 10. ed. São Paulo: Pearson, 2019.
- GRINBERG, Miguel. Desenvolvimento web com Flask: desenvolvendo aplicações web com Python. 1. ed. Português. São Paulo: Novatec Editora, 2018. 312 p. ISBN 978-85-7522-681-0.

Dúvidas?

Fonte: elemento Canva

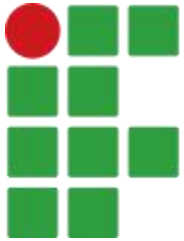
QR-CODE
Slides de Aula

Frameworks para desenvolvimento back-end

Curso: Análise e Desenvolvimento de Sistemas

Componente: Programação para a Internet II 5º Período

Professor: Lucas Sampaio Leite



INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Pernambuco