



# **Minicurso: Desenvolvimento Prático de Sistemas com JavaScript e Flask**

Tutor: Bernardo de Castro Monteiro Franco Gomes

## Sobre mim

Me chamo Bernardo de Castro Monteiro Franco Gomes, sou estudante de Ciências da Computação na Universidade Federal de São João Del Rei.

Atualmente trabalho como professor particular de desenvolvimento de software full-stack, já trabalhei na escola de cursos InfinitySchool como Monitor dos cursos de Desenvolvimento Full-Stack e Data Science, Professor Particular, Palestrante e tutor de minicursos e workshops.



---

**A pergunta do nosso minicurso é:  
QUAL É A GRANDE PROBLEMÁTICA  
ATUAL NO DESENVOLVIMENTO DE  
SOFTWARE ATUALMENTE?**

---

# DESENVOLVIMENTO DE SOFTWARE ÁGIL

---

# Metodologias ágeis aplicadas erroneamente



## Sobre o projeto desse minicurso

Durante este minicurso iremos realizar o desenvolvimento de um sistema full-stack utilizando principalmente JavaScript e FlaskAPI. Porém dando mais detalhes as stacks que iremos utilizar são:

- FlaskAPI
- SQLite
- HTML/CSS
- Bootstrap
- JavaScript

---

# Sistema CRUD

## Como funciona um CRUD?

O CRUD funciona a partir da ideia da sua sigla: **Create (Criar)**, **Read (Ler)**, **Update (Atualizar)** e **Delete (Deletar)**, o que descreve as quatro operações básicas realizada em dados armazenados em bancos de dados ou sistemas de informação.







## Como criar um CRUD?

1. Escolha a tecnologia
2. Crie o banco de dados
3. Conecte-se ao banco de dados (IMPORTANTE)
4. Implemente as operações CRUD
5. Crie uma interface



FlaskAPI

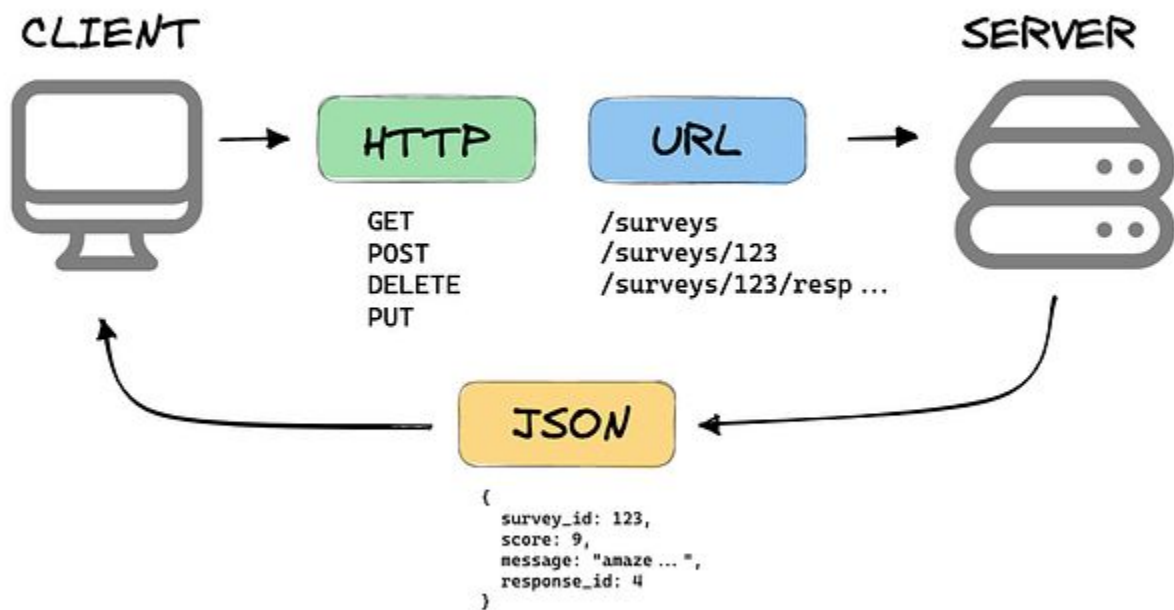


# O que é FlaskAPI?

FlaskAPI é uma framework Flask (Python) que facilita o desenvolvimento de APIs RESTful.

API RESTful: APIs RESTful são usadas para fornecer acesso a recursos (dados, serviços, etc.) através de URLs e métodos HTTP como GET, POST, PUT e DELETE. Elas são projetadas para serem leves, escaláveis e fáceis de usar, tornando-as populares para o desenvolvimento de aplicações web e móveis

# WHAT IS A REST API?





# Como utilizar FlaskAPI

Para utilizar a FlaskAPI é necessário fazer download do Flask.

No site <https://pypi.org/project/Flask/> tem todas instruções de como baixar Flask e rodar seu primeiro “Hello, World!”



# Importância do FlaskAPI

- Permite integrar aplicações diferentes.
- Torna os dados acessíveis via HTTP.
- Facilita a construção de sistemas modernos (front-end separado do back-end).
- É simples e rápido para prototipar.

---

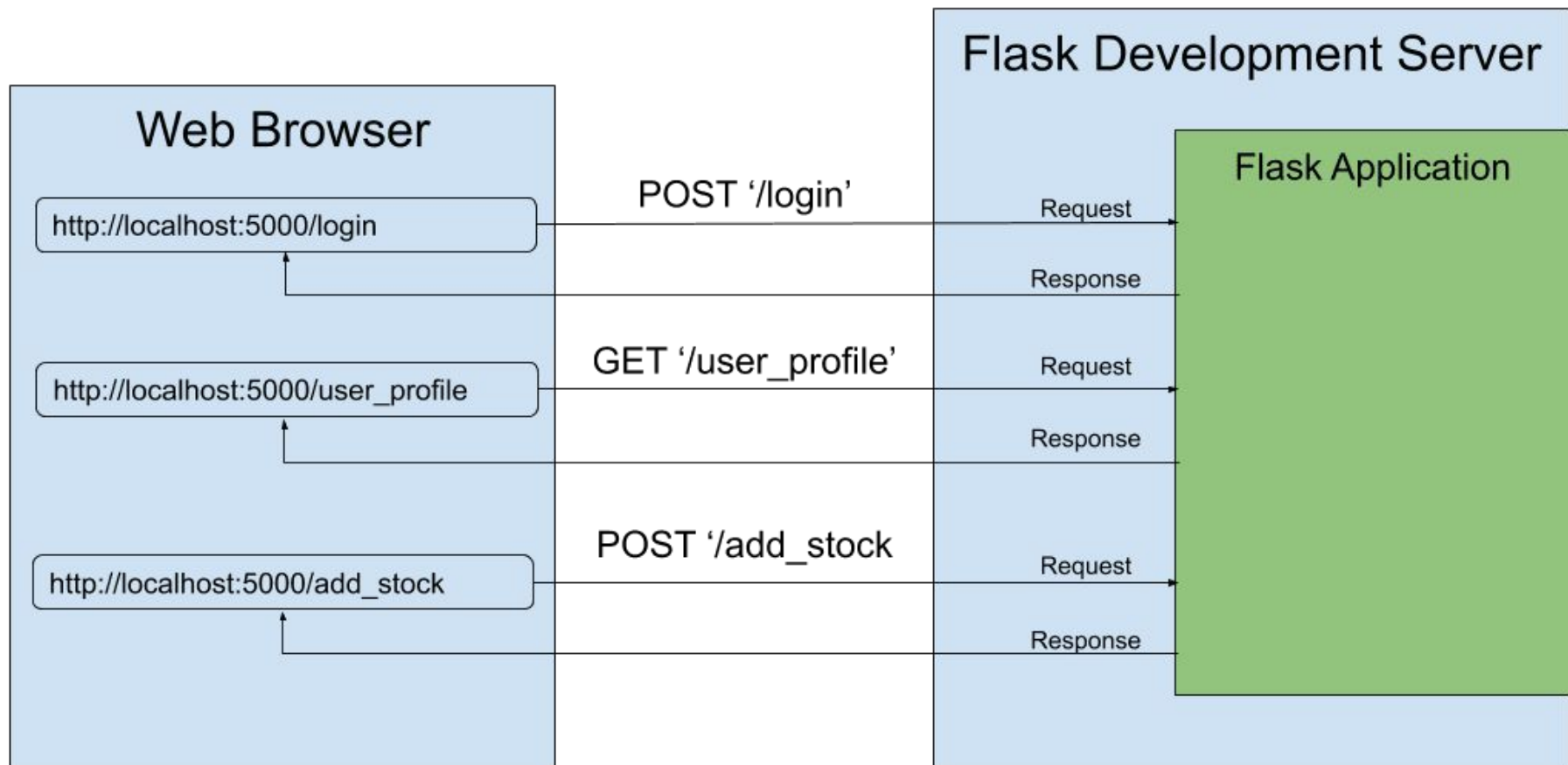
# Tratamento de Dados com FlaskAPI

## Como funciona o tratamento de dados com FlaskAPI?

- Receber dados via requisições HTTP (GET, POST, PUT, DELETE).
- Validar e tratar os dados recebidos.
- Retornar respostas estruturadas (JSON).









# Como funciona a Flask Session?

**Flask Session** é usada para armazenar informações temporárias sobre o usuário (por exemplo, login) durante a navegação.

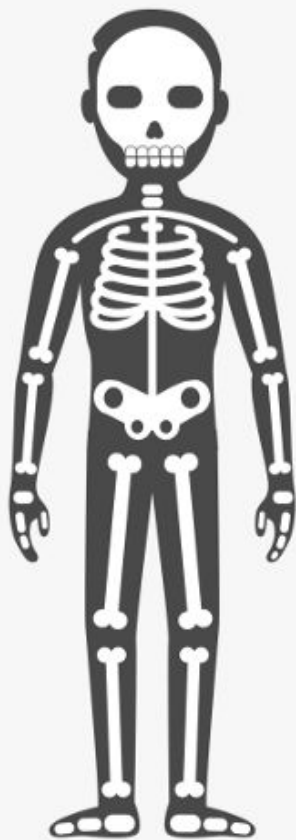
---

# HTML, CSS e Bootstrap

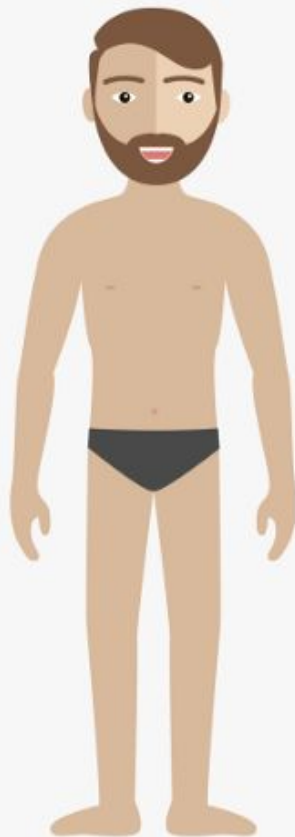


## Básico de sites HTML e CSS

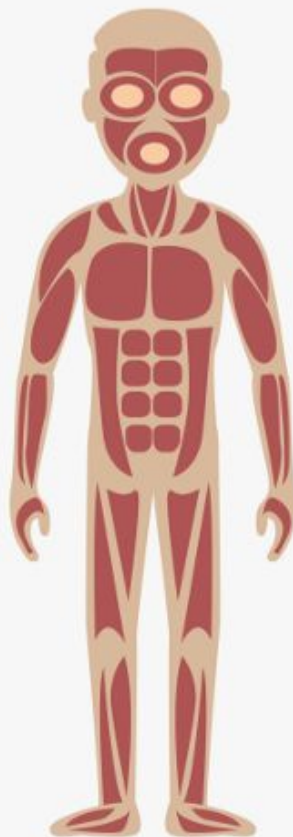
O HTML é utilizado para criar a estrutura da página, uma estrutura básica é gerada automaticamente no visual studio code, a partir dessa estrutura é possível adicionar o CSS e criar uma estilização para essa estrutura básica.



**HTML**



**CSS**



**Javascript**



# Como utilizar Bootstrap

O Bootstrap é uma extensão do CSS e do JavaScript, com ela fica mais fácil de se criar a estilização de um site, ele já possui funções pré-definidas que podem ser utilizadas no código HTML.

---

# JavaScript



# Uso de APIs no JavaScript

O uso de APIs no JavaScript é feito a partir das funções `fetch` e suas sub-funções, onde você utiliza `fetch('link')` e depois os sub-comandos do `fetch`, como `.json()` para pegar os dados.



```
1  fetch('https://api.exemplo.com/dados', {
2    method: 'GET', // ou 'POST', 'PUT', 'DELETE', etc.
3    headers: {
4      'Authorization': 'Bearer sua_token_de_autenticação',
5      'Content-Type': 'application/json' // dependendo da API
6    }
7  })
8    .then(response => response.json())
9    .then(data => {
10      // Faça algo com os dados da resposta da API
11    })
12    .catch(error => {
13      // Lida com erros
14    });
```



# Eventos DOM

DOM (Document Object Model) é uma representação da nossa árvore de elementos do HTML. Ela nos traz todo o modelo do nosso documento HTML, nos permitindo manipulá-lo de diversas formas.



## Como utilizar: Try e Catch

Em JavaScript, utilizamos try e catch porque eles permitem lidar com erros e exceções de maneira controlada, evitando que o código pare de funcionar abruptamente.



## Como utilizar: Async e Await

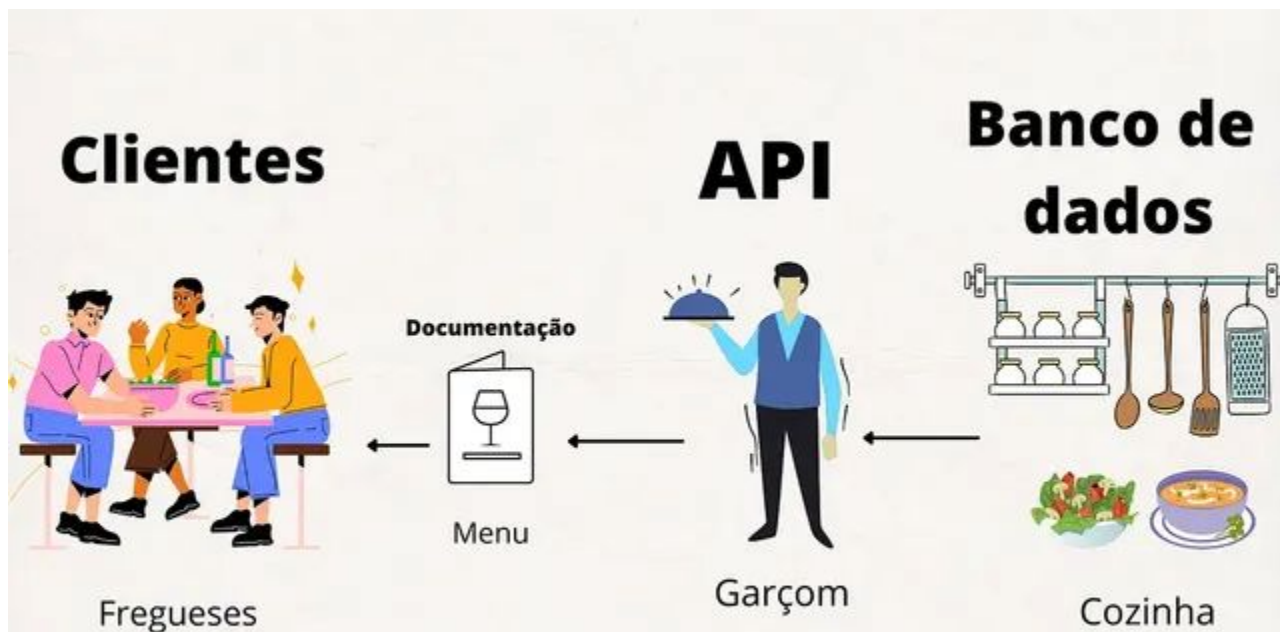
Função Síncrona: Quando falamos ao telefone, as informações chegam e saem em sequência, uma após a outra; fazemos uma pergunta, recebemos logo em seguida a resposta, com os dados dessa resposta fazemos outro comentário, etc.

Função Assíncrona: uma conversa online via algum mensageiro, como o WhatsApp, enviamos uma mensagem e não ficamos olhando para a tela, esperando, até a outra pessoa responder. Afinal de contas, não temos como saber quando, e se, essa resposta vai chegar.

---

# Conhecimentos básicos sobre uma API

## Como funciona uma API?





## Como criar uma API utilizando FlaskAPI?

```
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///mydatabase.db'
db = SQLAlchemy(app)

# Defina os modelos (tabelas)
class User(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(50), nullable=False)

# Rota inicial
@app.route('/')
def index():
    # Exemplo de consulta ao banco de dados
    users = User.query.all()
    return f'Total de usuários: {len(users)}'
```

**FLASK PYTHON**



## Como enviar dados em uma porta da API

É necessário fazer uma requisição HTTP para o endpoint desejado, utilizando o método HTTP apropriado (como POST, PUT, ou PATCH) e incluindo os dados no corpo da requisição.



---

**Diario**

---

# Caso Prático

---

# Introdução Temas Avançados

---

# Aprofundamento SQLite

---

# Aprofundamento FlaskAPI e desenvolvimento de APIs

---

# Tratamento de dados avançado

---

# Dúvidas