

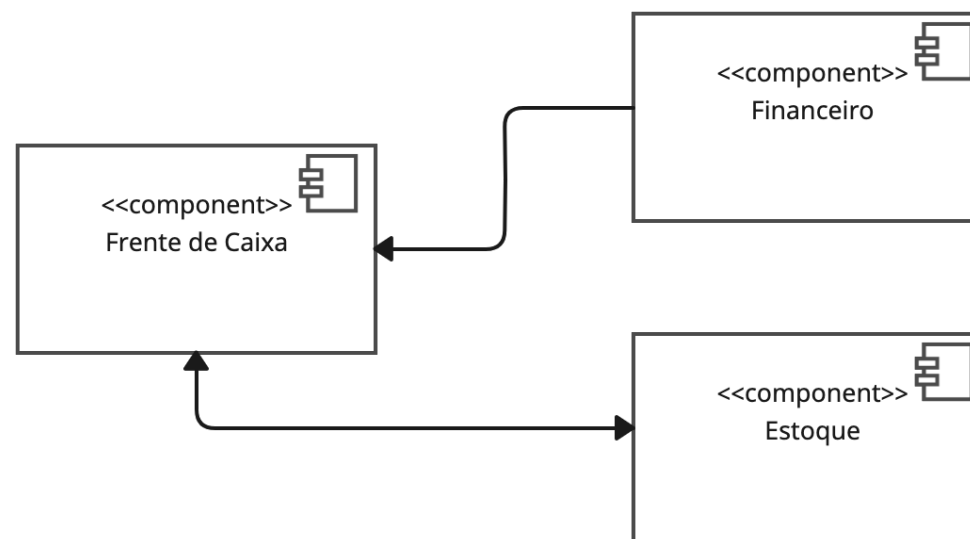
# Desenvolvimento de Serviços e APIs

Prof. Dr. Anderson Rodrigues



# O que são micro serviços de software?

**Arquitetura** para o desenvolvimento de sistemas de software, onde um **aplicativo é dividido em componentes autônomos** e independentes, conhecidos como micro serviços ou serviços.



# Tipos de micro serviços

- **Cliente:**  
Que apenas solicitam serviços.
- **Servidor ou Provedor:**  
Que apenas fornecem serviços.
- **Cliente e Servidor:**  
Que tanto fornecem quando solicitam serviços.

O que são  
*frameworks* de  
desenvolvimento  
de software?

Um *framework* de desenvolvimento de micro serviços é um conjunto de **ferramentas**, **bibliotecas** e **padrões** que facilita a criação, implementação, escalabilidade e gerenciamento de arquiteturas baseadas em micro serviços.



# Características dos *frameworks*?

1. **Comunicação entre Serviços:** Facilita a comunicação entre micro serviços, muitas vezes oferecendo suporte a protocolos como HTTP/REST, gRPC, mensageria assíncrona, etc.
2. **Descoberta de Serviços:** Fornece mecanismos para descoberta dinâmica de serviços, permitindo que os micro serviços encontrem e se comuniquem uns com os outros de maneira eficiente.
3. **Balanceamento de Carga:** Oferece recursos para distribuir o tráfego entre instâncias de micro serviços para melhorar a escalabilidade e a disponibilidade.
4. **Gerenciamento de Configuração:** Ajuda na gestão centralizada de configurações, permitindo alterações dinâmicas sem a necessidade de reinicialização dos serviços.
5. **Segurança:** Integra recursos de segurança, como autenticação e autorização, para proteger os micro serviços contra ameaças externas e garantir o controle de acesso adequado.
6. **Monitoramento e Rastreamento:** Inclui funcionalidades para monitorar o desempenho, rastrear solicitações através dos micro serviços e coletar métricas para análise.
7. **Implementação de Padrões de Arquitetura:** Suporta a implementação de padrões comuns em arquiteturas de micro serviços, como Circuit Breaker, Service Registry, Service Mesh, etc.
8. **Facilidades para Testes:** Oferece ferramentas e práticas recomendadas para testar unidades individuais de micro serviços, bem como testes de integração.

# Principais *frameworks* para desenvolvimento de serviços

- **Express.js (Node.js/JavaScript):**

O Express.js é um framework leve e flexível para Node.js, amplamente utilizado para construir aplicativos web e APIs RESTful. Sua simplicidade o torna uma escolha popular para o desenvolvimento de micro serviços em JavaScript.

- **Django (Python):**

Embora conhecido principalmente como um framework web para Python, o Django também pode ser usado para desenvolvimento de micro serviços. Ele oferece uma estrutura sólida e convenções que facilitam a criação de aplicativos escaláveis.

- **NestJs (Node.js/JavaScript/TypeScript):**

NestJS é um framework para construção de aplicativos Node.js usando TypeScript. Ele é inspirado no Angular e é projetado para facilitar a construção de micro serviços escaláveis e modularizados.

# Principais *frameworks* para desenvolvimento de serviços

- **Spring Boot (Java):**

O Spring Boot é um framework baseado no ecossistema Spring para desenvolvimento rápido de aplicativos Java. Ele facilita a criação de micro serviços Java de maneira eficiente, fornecendo configurações padrão e uma variedade de recursos.

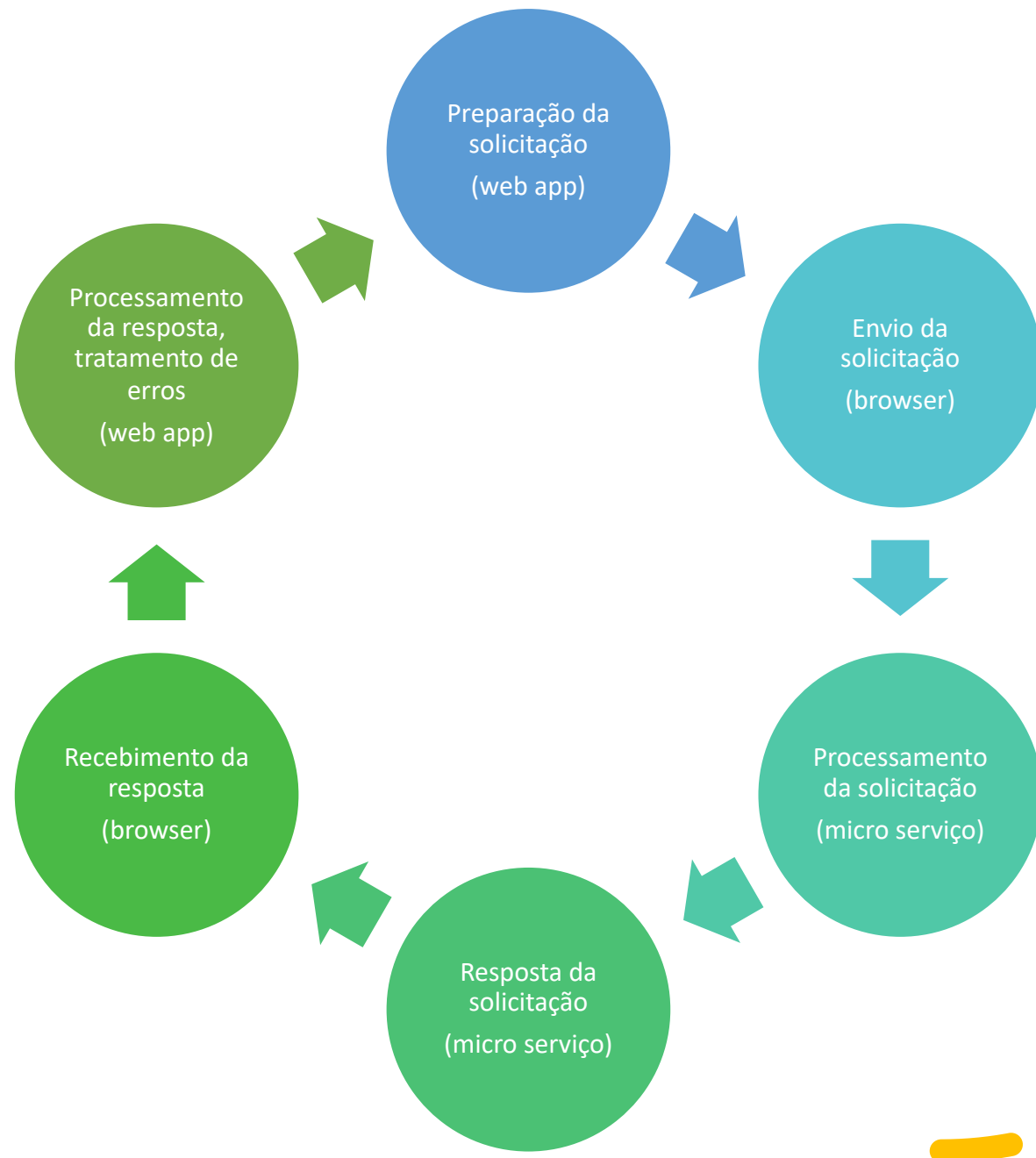
- **Flask (Python):**

Flask é um framework Python leve e fácil de usar, adequado para o desenvolvimento rápido de micro serviços. Ele fornece o essencial para criar APIs e pode ser estendido conforme necessário.

- **ASP.NET Core (C#):**

O ASP.NET Core é um framework da Microsoft para desenvolvimento de aplicativos web e serviços, incluindo microserviços. Ele é multiplataforma e suporta o desenvolvimento em C#.

# Ciclo de chamada de um micro serviço





# Construindo o frontend:

Ciclo de  
chamada de um  
micro serviço:  
(Exemplo  
prático)

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>API Financeira</title>
</head>

<body>

  <h1>Taxa de Câmbio USD para REAL</h1>

  <div id="exchange-rate">
    <button name="btnExchange" onclick="fetchExchangeRate()">Chamar
    API</button>
    <div id="exchange-resp"></div>
  </div>

</body>
</html>
```

# Preparando a solicitação

Ciclo de chamada de um micro serviço:  
(Exemplo prático)

Escolhendo o provedor:

<https://www.exchangerate-api.com/>

Conhecendo a URL da API:

[https://v6.exchangerate-api.com/v6/\*\*YOUR-API-KEY\*\*/latest/\*\*USD\*\*](https://v6.exchangerate-api.com/v6/YOUR-API-KEY/latest/USD)

Onde:

**YOUR-API-KEY** é uma chave fornecida pelo provedor após cadastro  
**USD** é a moeda base usada para a conversão das demais moedas

Codificando: Montando a URL da API

```
const apiKey = `SUA-CHAVE`;  
const apiUrl = `https://v6.exchangerate-api.com/v6/${apiKey}/latest/USD`
```

Ciclo de chamada de um micro serviço:  
(Exemplo prático)

# Enviando a solicitação e recebendo a resposta:

Enviando a solicitação:

```
const response = await fetch(apiUrl);
```

Recebendo e processando a resposta:

```
const data = await response.json();
```

```
// Extraíndo a taxa de câmbio de USD para REAL  
const exchangeRate = data.conversion_rates.BRL;
```

Apresentando na página:

```
// Atualizando o conteúdo da página para o usuário final  
document.getElementById('exchange-resp').innerHTML = `Taxa de câmbio:  
USD 1 = R$ ${exchangeRate}`;
```



# Adicionando o código ao frontend

Ciclo de chamada de um micro serviço:  
(Exemplo prático)

```
<script>

// Substitua 'YOUR_API_KEY' pela chave de API fornecida pelo ExchangeRate-API
const apiKey = `a094f67928d1a2e49003eaf7`;
const apiUrl = `https://v6.exchangerate-api.com/v6/${apiKey}/latest/USD`

// Função para fazer a solicitação à API
async function fetchExchangeRate() {
  try {
    const response = await fetch(apiUrl);
    const data = await response.json();

    // Extraindo a taxa de câmbio de USD para REAL
    const exchangeRate = data.conversion_rates.BRL;
    const number_of_tickets =
      Object.keys(data.conversion_rates).length;

    // Atualizando o conteúdo da página
    document.getElementById('exchange-resp').innerHTML
      = `Taxa de câmbio: USD 1 = R$ ${exchangeRate}`;
  } catch (error) {
    console.error('Erro ao acessar a API:', error);
  }
}

</script>
```

# Ciclo de chamada de um micro serviço

## Exercício

Visite o site do provedor utilizado nesta aula e verifique através da documentação disponível quais as outras moedas que podem ser convertidas para o Real e apresente na página pelo menos mais 3 valores de câmbio.

