Slide 1: Arquitetura e Organização de Serviços Web

Objetivos:

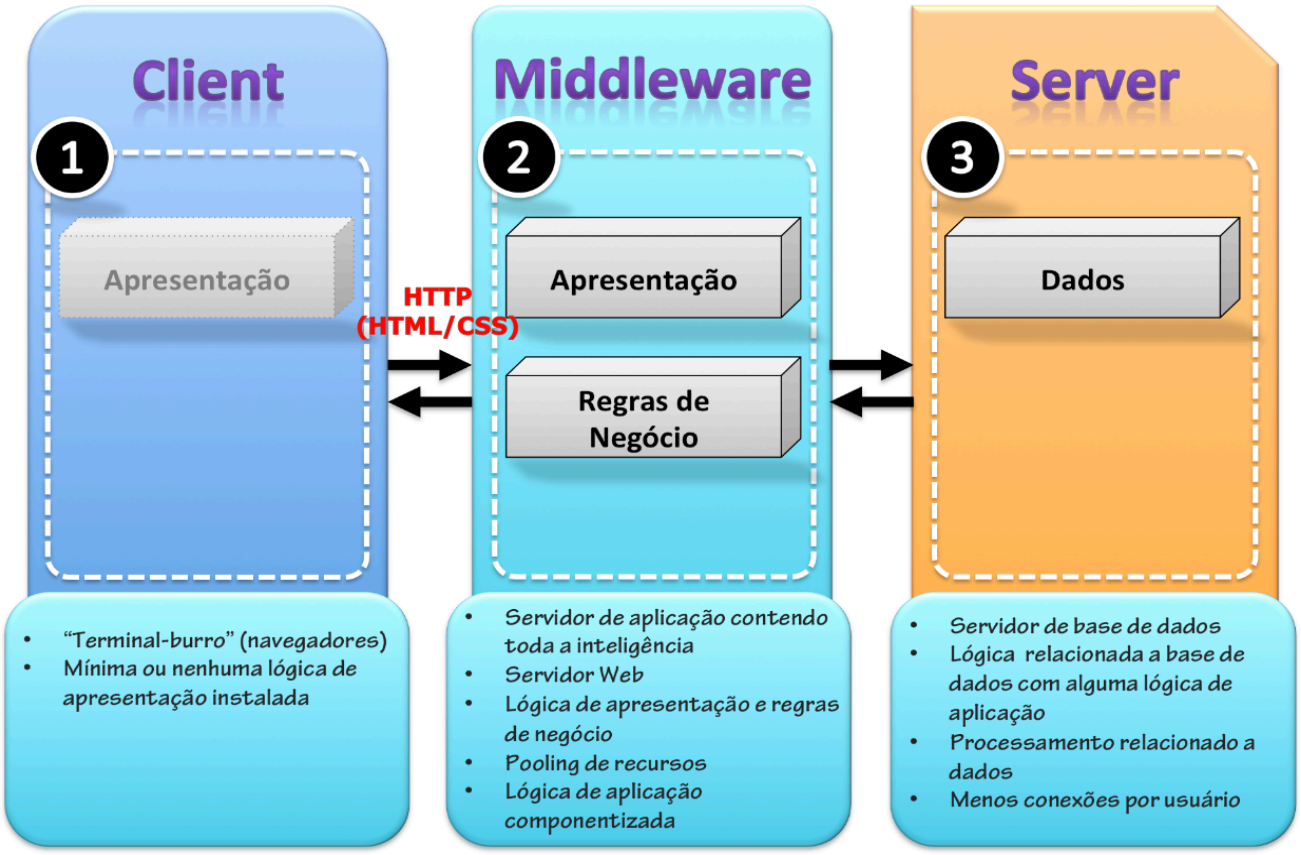
1. Entender o que levou ao surgimento da Arquitetura de Serviços Web;
2. Compreender o que é um Serviço Web;
3. Quais tipos de Serviços Web existem;
4. Diferenciar um serviço SOAP de um serviço REST.

Pre-Requisitos

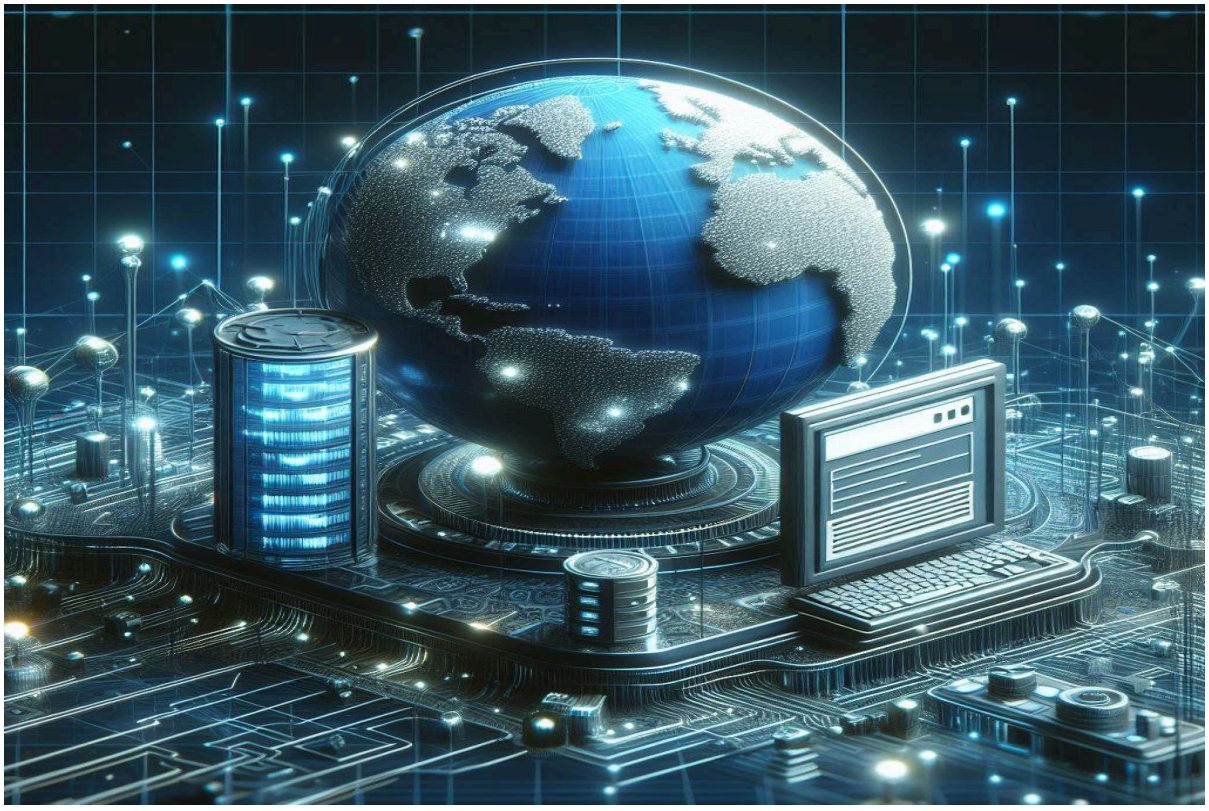
1. Conhecimento sobre a arquitetura Web
2. Javascript e NodeJS
3. Ter instalado o Visual Code e nodeJS

Slide 2: Por Que Surgiu a Arquitetura de Serviços Web?

Arquitetura Web Tradicional



Slide 2: Por Que Surgiu a Arquitetura de Serviços Web?



Motivações para a Arquitetura de Serviços Web

- **Interoperabilidade** : Necessidade de integrar sistemas diferentes e heterogêneos.
    - WEB
- **Complexidade Crescente** : Sistemas e aplicações tornaram-se mais complexos e precisavam de soluções modulares e escaláveis.
    - Comercio Eletronico (Amazon, Mercado Livre)
    - Redes Sociais (Facebook, Instragram, Ticktok)
    - Sistemas Bancários (Mercado Pago, Nuback)
    - Sistemas Governamentais (GOV.BR , Carteira Digital)
- **Evolução Tecnológica** : Adaptação rápida a novas tecnologias e padrões.
    - XML
    - Ajax
    - JSON
- **Economia de Recursos** : Redução da duplicação de esforços e reutilização de componentes existentes.
    - Sistemas Web
    - Sistemas moveis (Andriod e IOS)
- **Conectividade Global** : Demanda crescente por conectividade e comunicação global entre diferentes plataformas e dispositivos.
- **Flexibilidade e Agilidade** : Necessidade de respostas rápidas às mudanças de mercado e requisitos de negócios.

Slide 3: Histórico dos Serviços Web

Histórico dos Serviços Web

- **1990s** : Início da Internet comercial; necessidade de integração de sistemas.
- **CORBA** : Introduzido pela OMG (Object Management Group) em 1991 para permitir a comunicação entre sistemas distribuídos heterogêneos.
- **DCOM** : Desenvolvido pela Microsoft em 1993 para permitir a comunicação entre componentes de software distribuídos.
- **EJB** : Introduzido pela Sun Microsystems em 1997 para simplificar o desenvolvimento de aplicações corporativas em Java.
- **1998** : Lançamento do SOAP (Simple Object Access Protocol) pela Microsoft.
- **2000s** : Adoção crescente do SOAP; desenvolvimento do WSDL (Web Services Description Language).
- **2000s** : Introdução do REST (Representational State Transfer) por Roy Fielding.

- **2010s** : Popularização dos microservices; aumento do uso de APIs RESTful.
- **Presente** : Ferramentas avançadas de gerenciamento de APIs, segurança aprimorada, e uso crescente de service mesh.

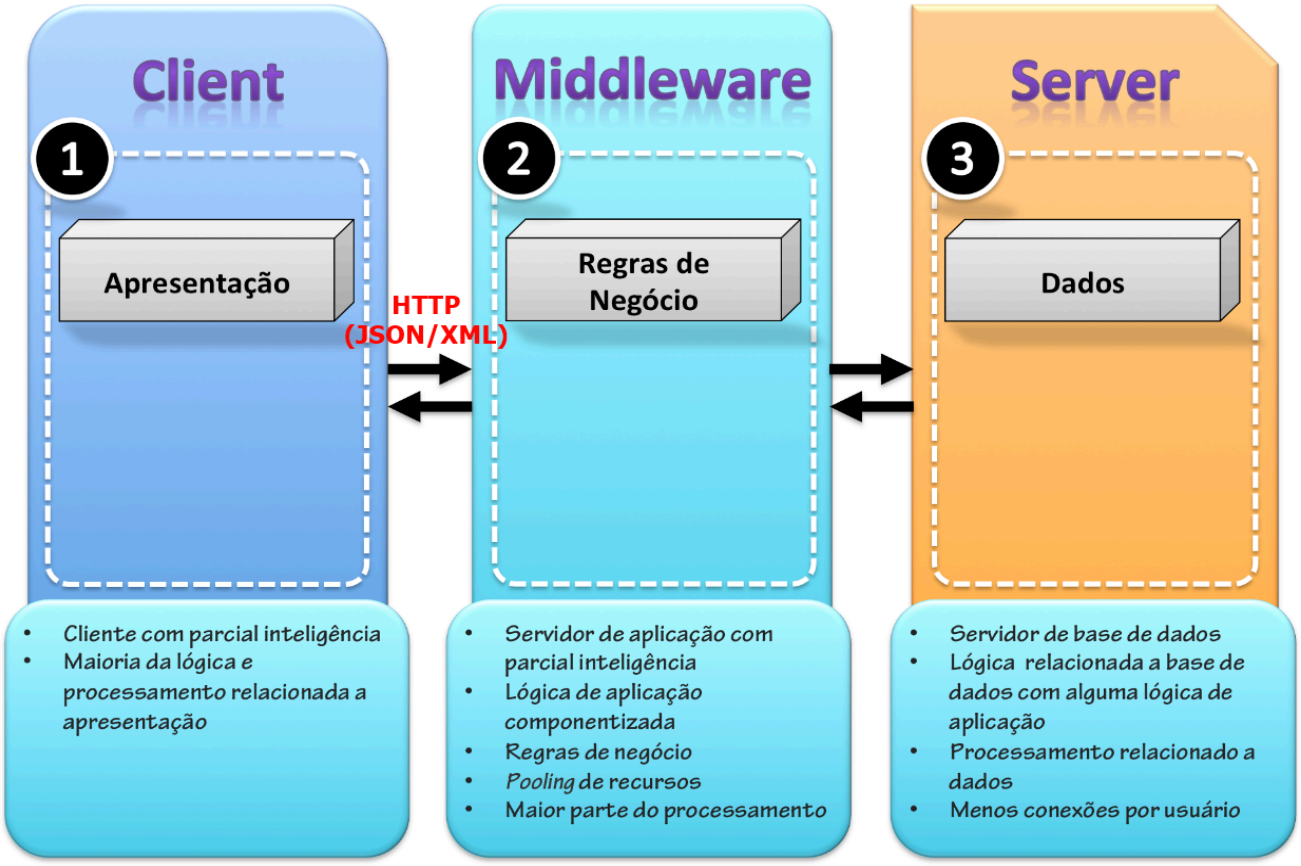
Slide 4: Introdução

O que são Serviços Web?

- Permitem a comunicação entre diferentes aplicações através da web.
- ***"Geralmente, uma interface de serviço Web consiste em um conjunto de operações que podem ser usadas por um cliente na Internet. As operações de um serviço Web podem ser fornecidas por uma variedade de recursos diferentes, por exemplo, programas, objetos ou bancos de dados. Um serviço Web pode ser gerenciado por um servidor Web, junto a páginas Web, ou pode ser um serviço totalmente separado."***
- Utilizam padrões abertos como XML, JSON, SOAP, REST.

Slide 2: Introdução?

Arquitetura SOA



Slide 5: Componentes Principais

Componentes Principais de Serviços Web

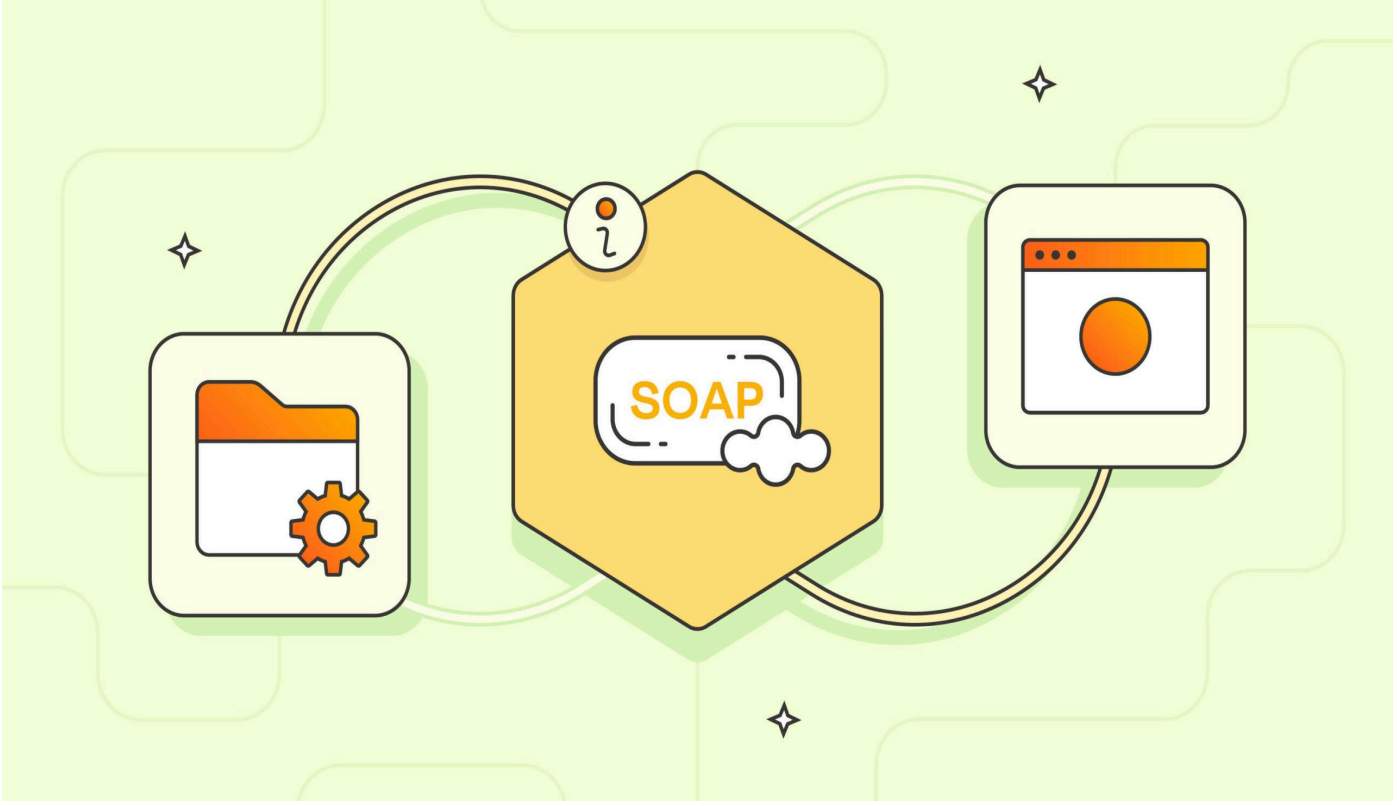
1. **Cliente** : Faz as requisições aos serviços.
2. **Servidor** : Responde às requisições com os dados/processamentos solicitados.
3. **Protocolo** : Define como a comunicação é feita (HTTP, HTTPS, SMTP).

Slide 6: Arquitetura de Serviços Web

Modelos de Arquitetura

1. **SOA (Service-Oriented Architecture) / SOAP ()Simple Object Access Protocol)**
   - Coleção de serviços que se comunicam entre si.
2. **REST (Representational State Transfer)**
   - Baseado em recursos, usa HTTP e operações padrão (GET, POST, PUT, DELETE).
3. **Microservices**
   - Serviços pequenos e independentes, focados em uma única funcionalidade.

Slide 7: SOAP

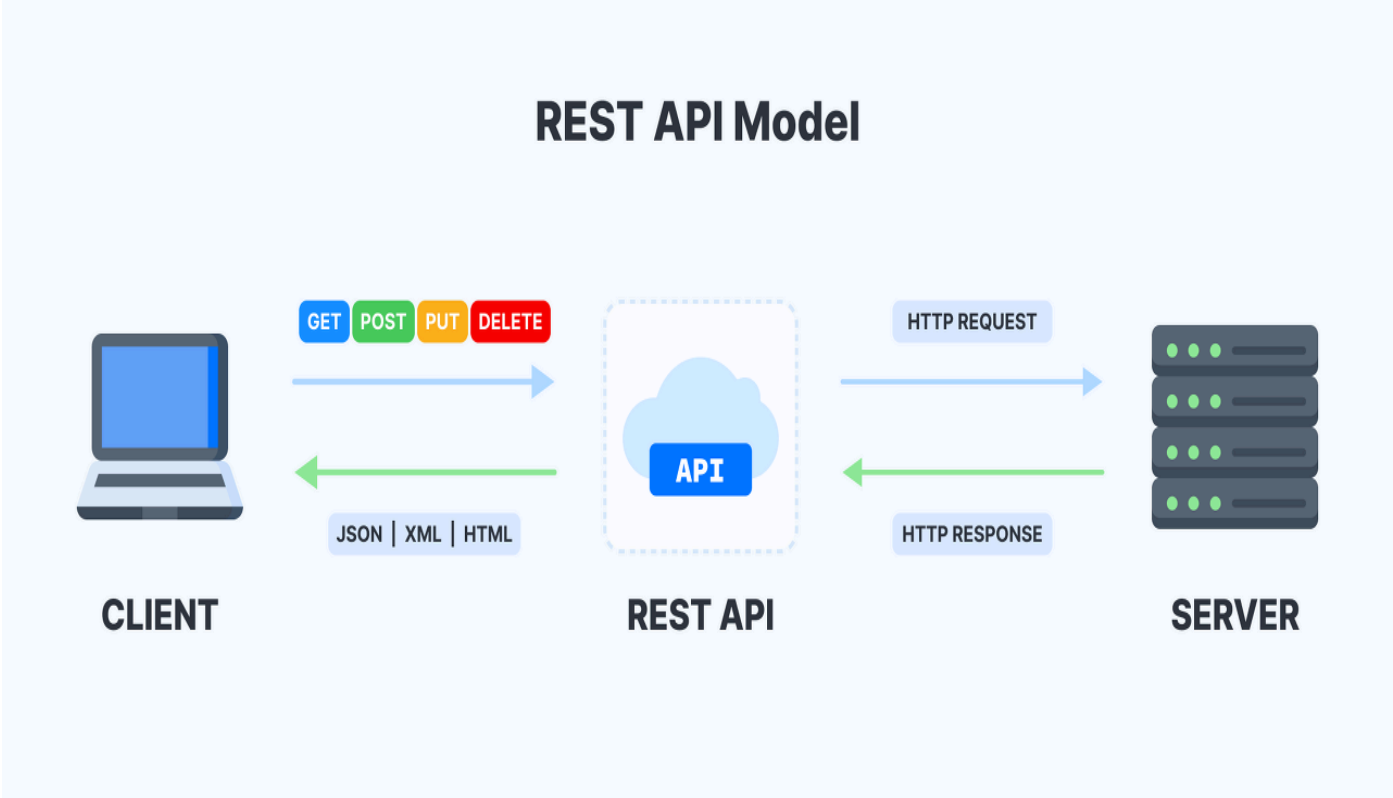


Definição: *" SOAP, ou Simple Object Access Protocol (não confundir com SOA), é um protocolo para criação de **serviços web distribuídos** e descentralizados, os famosos web services, que permitem a troca de informações entre sistemas, usando o formato **XML**."*

Características do SOAP

- **Protocol-Based** : Baseado em protocolo, normalmente sobre HTTP/HTTPS.
- **Extensível** : Suporta WS-Security, WS-ReliableMessaging.
- **Estruturado** : Usa XML para mensagens (WSDL).

Slide 8: RESTful Services



Definição: "REST não é um protocolo ou padrão, mas sim um conjunto de restrições de arquitetura. Os desenvolvedores de API podem implementar a arquitetura REST de maneiras variadas."

Características de RESTful Services

- **Stateless** : Cada requisição do cliente ao servidor deve conter todas as informações necessárias.
- **Cacheable** : Respostas podem ser armazenadas em cache para melhorar a performance.
- **Uniform Interface** : Utilização de métodos HTTP padronizados.
- **Layered System** : Estrutura em camadas para modularidade e segurança.

Slide 9: Organização de Serviços Web

Organização de Serviços Web

- **API Gateway** : Ponto de entrada único para APIs, gerencia chamadas, autenticação, e controle de tráfego.
- **Service Registry** : Catálogo de serviços disponíveis, facilita a descoberta de serviços.
- **Service Mesh** : Gestão de comunicação entre microserviços, focado em resiliência, observabilidade e segurança.

Slide 10: Segurança em Serviços Web

Segurança em Serviços Web

- **Autenticação e Autorização** : Uso de OAuth, JWT.
- **Criptografia** : SSL/TLS para comunicação segura.
- **Firewalls e Gateways** : Proteção contra ataques externos.

Slide 11: Ferramentas e Tecnologias

Ferramentas e Tecnologias

- **REST APIs** : Flask, Django, Express.js, Spring Boot.
- **SOAP** : Apache Axis, JAX-WS.
- **API Management** : Kong, Apigee, AWS API Gateway.
- **Service Mesh** : Istio, Linkerd.

Slide 12: Exemplo de Aplicação SOA com JavaScript

Exemplo de Aplicação SOA com JavaScript

1. **Servidor SOAP** :

javascript
Copiar código

```
const soap = require('soap');
   const express = require('express');
   const app = express();

   const service = {
     MyService: {
       MyPort: {
         MyFunction: function(args) {
           return { result: args.value * 2 };
         }
       }
     }
```

```
  };

  const xml = require('fs').readFileSync('service.wsdl',
'utf8');

  app.listen(8000, function() {
    soap.listen(app, '/wsdl', service, xml);
    console.log('SOAP server listening on port 8000');
  });
```

1. **Cliente SOAP** :

javascript
Copiar código

```
const soap = require('soap');

  const url = 'http://localhost:8000/wsdl?wsdl';
  const args = { value: 5 };

  soap.createClient(url, function(err, client) {
    client.MyFunction(args, function(err, result) {
      console.log(result);
    });
  });
```

Slide 13: Exemplo de Aplicação REST com JavaScript

Exemplo de Aplicação REST com JavaScript

1. **Servidor REST** :

javascript
Copiar código

```
const express = require('express');
  const app = express();
  const port = 3000;

  app.use(express.json());

  app.get('/double/:value', (req, res) => {
```

```
  const value = parseInt(req.params.value);
  res.json({ result: value * 2 });
});

app.listen(port, () => {
  console.log(`REST server listening on port ${port}`);
});
```

1. **Cliente REST** :

javascript
Copiar código

```
const fetch = require('node-fetch');

fetch('http://localhost:3000/double/5')
  .then(response => response.json())
  .then(data => console.log(data));
```

Slide 14: Conclusão

Conclusão

- Serviços Web são essenciais para a integração de sistemas.
- Escolha da arquitetura depende das necessidades do projeto.
- Segurança e gestão são críticas para um funcionamento eficaz.

Slide 15: Perguntas?

Perguntas?

- Espaço para discussão e dúvidas.

Slide 16: Quiz - Perguntas e Respostas

Quiz - Teste seus conhecimentos sobre Serviços Web

1. **O que são serviços web?**
    - A) Aplicações que rodam localmente sem conexão à internet.
    - B) Aplicações que permitem a comunicação entre diferentes sistemas através da web.
    - C) Apenas páginas web estáticas.
2. **Quais são os três componentes principais de um serviço web?**
    - A) Cliente, Servidor e Protocolo.
    - B) Front-end, Back-end e Banco de Dados.
    - C) HTTP, HTTPS e FTP.
3. **O que significa SOAP?**
    - A) Simple Object Access Protocol.
    - B) Secure Open API Protocol.
    - C) Standardized Object Application Platform.
4. **Qual é a principal característica de um serviço RESTful?**
    - A) Stateless.
    - B) Stateful.
    - C) Relacional.
5. **Qual ferramenta não é usada para gerenciar APIs?**
    - A) Kong.
    - B) Apigee.
    - C) GitHub.
6. **Em que ano foi introduzido o CORBA?**
    - A) 1991.
    - B) 1993.
    - C) 1997.
7. **O que é um API Gateway?**
    - A) Um ponto de entrada único para APIs que gerencia chamadas, autenticação e controle de tráfego.
    - B) Um tipo de banco de dados.
    - C) Uma ferramenta de desenvolvimento de front-end.
8. **Qual é a função principal do Service Mesh?**
    - A) Gerenciar a comunicação entre microserviços com foco em resiliência, observabilidade e segurança.
    - B) Armazenar dados de aplicações.
    - C) Criar interfaces de usuário.
9. **Qual dessas características é específica de SOAP?**
    - A) Uso de XML para mensagens.
    - B) Uso de JSON para mensagens.

- C) Utilização exclusiva de GET e POST.

Respostas do Quiz:

1. **B** - Aplicações que permitem a comunicação entre diferentes sistemas através da web.
2. **A** - Cliente, Servidor e Protocolo.
3. **A** - Simple Object Access Protocol.
4. **A** - Stateless.
5. **C** - GitHub.
6. **A** - 1991.
7. **A** - Um ponto de entrada único para APIs que gerencia chamadas, autenticação e controle de tráfego.
8. **A** - Gerenciar a comunicação entre microserviços com foco em resiliência, observabilidade e segurança.
9. **A** - Uso de XML para mensagens.