



{ Introdução às Redes e à Internet }

Aula 04

<Módulo 01 />

Serviços e Aplicações na Internet



Introdução

Neste encontro exploraremos o mundo dos serviços web.

Você compreenderá o papel fundamental que esses serviços desempenham na economia digital e como eles evoluíram desde a concepção da *World Wide Web*.

Discutiremos o panorama atual dos serviços web avançados e sua relevância para empresas, organizações e desenvolvedores.

APIs e Integração de Sistemas

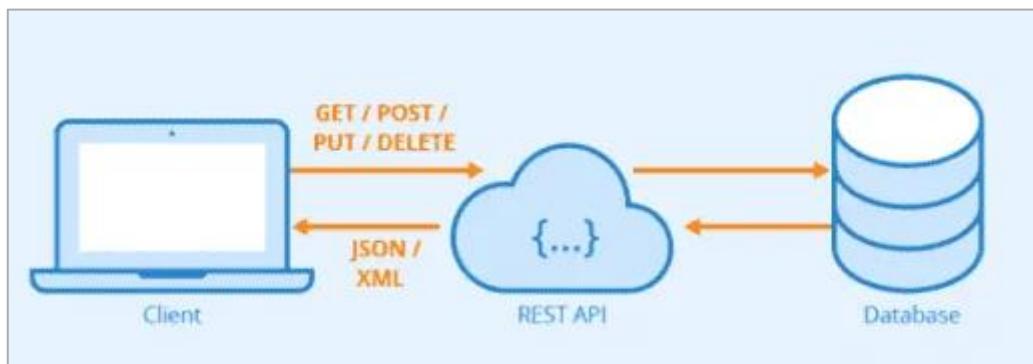
As **APIs** (*Application Programming Interfaces*) são os blocos de construção dos serviços web e desempenham um papel fundamental nos serviços web, permitindo a comunicação e a integração eficiente de sistemas e aplicativos.

O que são APIs?

As APIs são conjuntos de regras e protocolos que permitem que aplicativos e sistemas se comuniquem uns com os outros.

Elas atuam como intermediárias, permitindo que diferentes componentes de software compartilhem dados e funcionalidades.

As APIs definem as mensagens e solicitações que aplicativos podem fazer uns aos outros, facilitando a interoperabilidade.



Como as APIs funcionam?

• Solicitações e Respostas

As APIs operam com base em solicitações (**requests**) e respostas (**responses**). Um aplicativo envia uma solicitação para a API, especificando a ação desejada.

A API processa a solicitação e retorna uma resposta contendo os dados solicitados ou uma confirmação da ação.

• Padrões e Formatos

As APIs seguem padrões e formatos de comunicação, como **JSON** (*JavaScript Object Notation*) ou **XML** (*eXtensible Markup Language*), para estruturar os dados transmitidos.

Isso garante que diferentes aplicativos possam entender e interpretar as informações compartilhadas.

Serviços e Aplicações na Internet

Web Services e Protocolos (SOAP, REST)

Web Services são uma tecnologia essencial para a integração de sistemas distribuídos e a comunicação eficiente entre aplicativos.

Neste tópico, você explorará a arquitetura de **web services**, compreenderá como eles possibilitam a interação entre sistemas e a troca de dados, e examinará dois dos principais protocolos:

- **SOAP** (*Simple Object Access Protocol*) e;
- **REST** (*Representational State Transfer*).

O que são Web Services?

Web Services são serviços baseados na web que permitem que sistemas diferentes comuniquem entre si por meio de padrões e protocolos específicos.

Eles fornecem uma maneira padronizada de disponibilizar funcionalidades e compartilhar dados pela Internet.

Protocolo SOAP (*Simple Object Access Protocol*)

• Características

SOAP é um protocolo baseado em **XML** que define uma estrutura rígida para a comunicação entre sistemas.

Ele usa mensagens XML para transmitir informações, incluindo solicitações e respostas.

• Vantagens do SOAP

SOAP é altamente estruturado e oferece recursos robustos, como segurança e transações.

É ideal para ambientes corporativos e aplicações críticas.

• Aplicações

SOAP é comumente usado em serviços web de empresas, sistemas de gerenciamento de pedidos e aplicações que requerem comunicação segura e confiável.

Protocolo REST (*Representational State Transfer*)

• Características

REST é uma abordagem mais leve e flexível para serviços web. Ele se baseia nos princípios da arquitetura web e usa os métodos **HTTP (GET, POST, PUT, DELETE)** para acessar e manipular recursos.

• Vantagens do REST

REST é simples e escalável, facilitando a criação de serviços web que podem ser acessados de qualquer lugar.

É amplamente utilizado na web, incluindo APIs de mídias sociais e aplicativos móveis.

• Aplicações

REST é comum em serviços web públicos, como redes sociais, aplicativos móveis e integração com a Internet das Coisas (**IoT**).

Serviços e Aplicações na Internet

Comparação entre SOAP e REST

- **Complexidade**

SOAP é mais estruturado e pode ser mais complexo de implementar, enquanto **REST** é mais simples e flexível.

- **Desempenho**

REST tende a ser mais eficiente, pois usa o protocolo HTTP diretamente.

SOAP pode ser mais pesado devido ao **XML**.

- **Segurança**

SOAP oferece recursos de segurança integrados, enquanto a segurança em **REST** pode ser implementada por meio de **HTTPS** e **tokens**.

- **Casos de Uso**

A escolha entre **SOAP** e **REST** depende das necessidades do projeto e do ambiente, com **SOAP** sendo mais adequado para integrações corporativas complexas e **REST** sendo preferível para a web aberta e aplicativos móveis.

Microsserviços

Nos últimos anos, a arquitetura de **microsserviços** emergiu como um paradigma revolucionário no desenvolvimento de software.

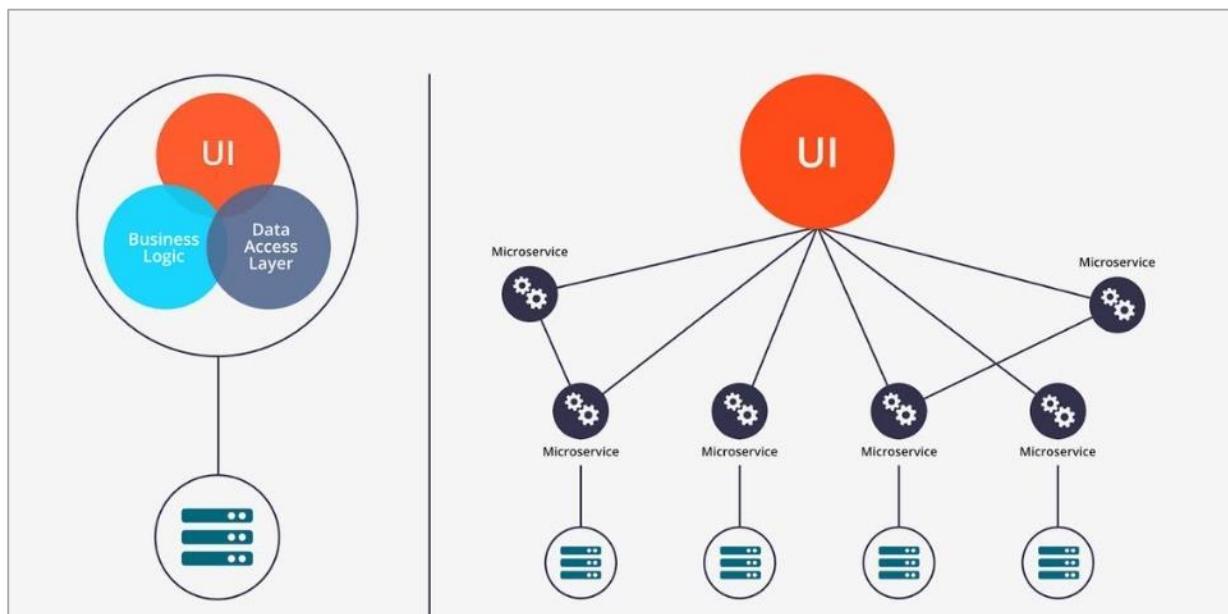
Ela representa uma mudança significativa na maneira como as aplicações são projetadas, construídas e implantadas.

Em vez de seguir a abordagem monolítica tradicional, onde toda a lógica de negócios é integrada em um único aplicativo, os **microsserviços** se baseiam na ideia de dividir uma aplicação em componentes independentes e autônomos.

Microsserviços são uma abordagem arquitetônica na qual uma aplicação é dividida em pequenos serviços independentes, cada um executando uma função específica.

Cada **microsserviço** é uma unidade autônoma que pode ser desenvolvida, implantada e dimensionada independentemente dos outros.

Esses serviços são projetados para serem altamente coesos, o que significa que eles devem se concentrar em realizar uma única tarefa ou função e fazê-la bem.



Serviços e Aplicações na Internet

Principais Características dos Microsserviços

• Desacoplamento

Cada **microsserviço** é desacoplado dos outros, o que significa que eles não têm conhecimento interno uns dos outros. Isso permite que eles evoluam independentemente.

• Independência Tecnológica

Os **microsserviços** podem ser desenvolvidos em diferentes linguagens de programação e tecnologias, desde que possam se comunicar de maneira eficaz.

• Escalabilidade

A escalabilidade é mais simples, pois você pode dimensionar apenas os microsserviços que exigem mais recursos, em vez de escalar a aplicação inteira.

• Manutenção Simplificada

Corrigir, atualizar ou estender um **microsserviço** não afeta os outros serviços, o que simplifica a manutenção.

• Implantação Contínua

Os **microsserviços** podem ser implantados de forma independente, permitindo a implantação contínua e rápida de novas funcionalidades.

• Resistência a Falhas

A falha em um **microsserviço** não afeta necessariamente o funcionamento de toda a aplicação, tornando-a mais resiliente.

Vantagens dos Microsserviços

• Flexibilidade e Agilidade

Os **microsserviços** permitem que as equipes de desenvolvimento sejam ágeis, implantando e atualizando partes específicas da aplicação sem afetar o todo.

• Facilidade de Escalabilidade

É mais fácil dimensionar **microsserviços** individualmente para atender às demandas, economizando recursos.

• Facilidade de Manutenção

A manutenção é simplificada, pois cada **microsserviço** pode ser gerenciado de forma independente.

• Resistência a Falhas

A falha em um serviço não causa uma interrupção completa do aplicativo.

Desafios dos Microsserviços

• Complexidade da Comunicação

A comunicação entre **microsserviços** pode ser desafiadora, exigindo soluções eficazes, como APIs.

• Gerenciamento e Orquestração

É necessário um sistema de gerenciamento e orquestração para manter o controle dos serviços e suas interações.

• Testes e Monitoramento

Testar e monitorar múltiplos **microsserviços** pode ser mais complexo do que em uma aplicação monolítica.



Os microsserviços se comunicam utilizando APIs e, no caso da web, utilizando o padrão REST!

Serviços e Aplicações na Internet

Aplicativos Web Interativos (Web 2.0)

O conceito de **Web 2.0** marcou uma transformação significativa na forma como os aplicativos web são projetados, desenvolvidos e usados.

Investigaremos as tecnologias-chave por trás dos aplicativos **Web 2.0**, como **AJAX** (*Asynchronous JavaScript and XML*).

Tecnologias-Chave da Web 2.0

- **AJAX** (*Asynchronous JavaScript and XML*)

O **AJAX** é uma tecnologia que permite a comunicação assíncrona entre o navegador e o servidor, sem a necessidade de recarregar a página. Isso resulta em uma experiência de usuário mais rápida e interativa.

- **Redes Sociais**

Plataformas de redes sociais, como Facebook, Twitter e LinkedIn, são exemplos notáveis de aplicativos Web 2.0 que permitem que os usuários compartilhem informações, interajam e colaborem em tempo real.

- **Wikis e Colaboração Online**

Wikis, como a **Wikipedia**, demonstram como a colaboração pode criar um recurso incrivelmente valioso, permitindo que milhares de usuários contribuam com conteúdo de forma colaborativa.

- **Ferramentas de Produtividade Online**

Aplicativos de produtividade, como o **Google Docs**, permitem a colaboração em documentos, planilhas e apresentações em tempo real.

Exemplos de Aplicativos Web 2.0

- **Google Maps**

Este serviço revolucionou a forma como as pessoas interagem com mapas online, oferecendo recursos de zoom, direções, visualização de ruas e integração de dados de localização.

- **Gmail**

O Gmail introduziu a ideia de uma caixa de entrada baseada em **AJAX**, fornecendo uma experiência de e-mail mais rápida e interativa.

- **YouTube**

Este serviço de compartilhamento de vídeos possibilitou que usuários façam upload, compartilhem e assistam a vídeos, transformando a maneira como o conteúdo de vídeo é consumido na web.

- **Trello**

Uma ferramenta de gerenciamento de projetos online que permite a colaboração em tempo real em quadros e listas.

Benefícios da Web 2.0

- **Experiência do Usuário Aprimorada**

Os aplicativos oferecem interfaces mais ricas e interativas, tornando a experiência do usuário mais envolvente.

- **Colaboração em Tempo Real**

A capacidade de colaborar instantaneamente em documentos, projetos e mídia social aprimora a eficiência e a produtividade.

- **Compartilhamento de Conteúdo**

Os aplicativos permitem que os usuários compartilhem conteúdo e informações com facilidade, criando comunidades e redes online.

- **Acesso em Dispositivos Móveis**

Muitos aplicativos são acessíveis em dispositivos móveis, permitindo o acesso em qualquer lugar.

Serviços e Aplicações na Internet

Autenticação e Segurança em Serviços Web

A segurança é uma consideração crítica em serviços web avançados.

Abordaremos a importância da autenticação, autorização e segurança de dados em serviços web. Exploraremos técnicas de autenticação, como **OAuth**, e como elas são usadas para proteger serviços web sensíveis. Além disso, discutiremos as melhores práticas de segurança, como criptografia e proteção contra ameaças comuns.

Serviços web frequentemente lidam com informações sensíveis e transações financeiras, tornando a segurança uma prioridade essencial.

Segurança em Serviços Web

• Confidencialidade

Garantir que os dados transmitidos e armazenados estejam protegidos contra acesso não autorizado.

• Integridade

Certificar-se de que os dados não sejam alterados durante a transmissão ou armazenamento.

• Autenticidade

Verificar a identidade dos usuários e sistemas que acessam os serviços.

• Autorização

Garantir que apenas usuários autorizados tenham acesso a recursos específicos.



Técnicas de Autenticação

• Oauth

OAuth é um protocolo de autorização amplamente usado que permite que os usuários concedam permissões a aplicativos de terceiros sem compartilhar suas credenciais.

Ele é comumente usado em serviços web para autenticar usuários por meio de plataformas de terceiros, como a autenticação com contas do Google ou do Facebook.

• Token de Acesso

O uso de tokens de acesso, como **JWT (JSON Web Tokens)**, é uma maneira eficaz de autenticar usuários em serviços web. Esses tokens são emitidos após a autenticação e são usados em solicitações subsequentes.

Melhores Práticas de Segurança

• Criptografia

O uso de criptografia **SSL/TLS** é fundamental para proteger a comunicação entre o cliente e o servidor.

Isso garante que os dados sejam transmitidos de forma segura.

• Validação de Dados

A validação de dados de entrada é crucial para evitar ataques de injeção, como **SQL injection** e **cross-site scripting (XSS)**.

• Proteção contra Ameaças Comuns

Implementar medidas de segurança contra ameaças comuns, como proteção contra **DDoS**, firewall de aplicativos da web e monitoramento de segurança.

• Gerenciamento de Identidade

Implementar soluções de gerenciamento de identidade para garantir que apenas usuários autenticados e autorizados tenham acesso aos recursos.

• Logs e Monitoramento

Manter registros detalhados de atividades e implementar sistemas de monitoramento para detectar atividades suspeitas.

Serviços e Aplicações na Internet

Casos de Uso e Exemplos Práticos

Exploraremos diversos casos de uso e exemplos práticos de serviços web avançados. Você verá como esses serviços são aplicados em diferentes setores e como eles impulsionam a inovação e a eficiência.

• Setor de Saúde – Telemedicina

A telemedicina é um exemplo prático de serviços web avançados. Ela permite que pacientes e médicos se conectem virtualmente, realizem consultas médicas remotas e acessem registros médicos eletrônicos com segurança.

• Comércio Eletrônico - Personalização e Recomendações

Plataformas de comércio eletrônico usam serviços web avançados para personalizar a experiência do cliente. Algoritmos de recomendação analisam o histórico de compras e o comportamento de navegação do usuário para sugerir produtos relevantes.

• Finanças - Pagamentos Online

Serviços de pagamento online, como **PayPal** e **Stripe**, facilitam as transações financeiras pela internet, oferecendo segurança e eficiência nas compras online e transferências de dinheiro.

• Redes Sociais - Integração de Mídias Sociais

A integração de mídias sociais em aplicativos e sites é um exemplo prático de serviços web avançados. Isso permite que os usuários compartilhem conteúdo, façam login usando suas contas de mídias sociais e interajam com outras plataformas.

• Educação - Plataformas de Aprendizado Online

Plataformas de aprendizado online, usam serviços web para fornecer cursos, materiais de ensino e interação entre alunos e instrutores.

• Streaming de Mídia - Transmissão de Vídeo Online

Serviços de streaming de vídeo, como **Netflix** e **YouTube**, aproveitam os serviços web avançados para entregar conteúdo de mídia sob demanda, recomendações personalizadas e recursos de compartilhamento.

• Transporte e Mobilidade - Aplicativos de Transporte Compartilhado

Aplicativos de transporte compartilhado, como **Uber** e **Lyft**, utilizam serviços web para rastreamento de veículos, cálculo de tarifas e conexão entre motoristas e passageiros.

• Agricultura - Monitoramento de Culturas

Na agricultura de precisão, serviços web avançados são usados para coletar dados de sensores em campo e fornecer informações em tempo real sobre o estado das culturas, otimizando o uso de recursos.

• Governo e Serviços Públicos - Portais de Serviços

Governos usam serviços web avançados para fornecer acesso online a serviços públicos, como emissão de documentos, pagamento de impostos e agendamento de serviços.



Esses exemplos práticos ilustram como os serviços web avançados estão transformando uma ampla variedade de setores.

Eles melhoraram a eficiência, a acessibilidade e a experiência do usuário em muitas áreas da vida cotidiana.

À medida que a tecnologia continua a evoluir, novas aplicações e casos de uso emergirão, tornando os serviços web uma parte essencial de nossa sociedade digital.