

**CENTRO PAULA SOUZA  
ETEC PROF MARIA CRISTINA MEDEIROS  
Técnico em Informática para Internet Integrado ao Ensino Médio**

**Giulia Pontelli Filippini**

**Pesquisa  
Pesquisa sobre aplicações web e sistemas**

**Ribeirão Pires  
2024**



## **LISTA DE ILUSTRAÇÕES**

### **TABELAS**

Tabela 1: Comparação de vantagens e desvantagens.....	10
---	----

# SUMÁRIO

1	O que são aplicações Web.....	5
1.1	Características.....	5
1.2	Tipos de aplicações.....	5
1.3	Tecnologias utilizadas.....	5
1.4	Exemplos de aplicações web.....	6
2	O QUE SÃO SISTEMAS DISTRIBUÍDOS .....	7
1.5	Características de sistemas distribuídos .....	7
1.6	Tipos e exemplos .....	7
1.7	Desafios.....	7
3	ARQUITETURAS .....	9
1.8	Arquitetura monolítica .....	9
1.8.1	Objetivos Específicos.....	10
1.9	Arquitetura de microserviços.....	9
1.9.1	Objetivos Específicos.....	10
4	MONOLÍTICO X MICROSERVIÇOS .....	10
5	CONCLUSÃO.....	11
6	REFERENCIAS.....	12

## **OBJETIVOS**

Compreender as funcionalidades e características das aplicações Web, entender o que são sistemas distribuídos e as diferenças entre as arquiteturas.

## 1 O que são aplicações Web

As **aplicações web** são programas ou sistemas que rodam em servidores e são acessados por meio de navegadores da web, sem a necessidade de instalação local no dispositivo do usuário. Elas utilizam tecnologias como **HTML, CSS e JavaScript** para a interface, enquanto o backend pode ser desenvolvido em linguagens como **Python, PHP, Java, JavaScript (Node.js), Ruby, entre outras**.

### 1.1 Características

As aplicações web tem como principais características a **acessibilidade universal**, já que podem ser acessadas de qualquer dispositivo com acesso a internet; as atualizações e correções são feitas diretamente no servidor, o que **facilita a manutenção**. As aplicações web também oferecem **independência de plataforma**, pois podem funcionar em diferentes sistemas sem necessidade de adaptação, além de oferecerem recursos dinâmicos e responsivos que **melhoram a interação em tempo real**.

### 1.2 Tipos de aplicações

**Aplicações estáticas** tem o **conteúdo fixo**, o usuário não consegue uma interação dinâmica, já as **dinâmicas** utilizam de **bancos de dados** e permitem **mais interações**.

**Single Page Applications (SPA)** são aplicações que se atualizam dinamicamente e **carregam apenas uma página**, um exemplo de SPA é o gmail. Existe também as **Progressive Web Apps (PWA)** que funcionam como **aplicativos móveis**, porém são **acessadas pelo navegador**.

### 1.3 Tecnologias utilizadas

Diferentes tecnologias podem ser usadas para o desenvolvimento de aplicações Web, alguns exemplos do que pode ser utilizado são:

Para o **Frontend**: HTML, CSS, JavaScript (React, Angular, Vue.js).

Para o **Backend**: Node.js, PHP, Python (Django, Flask), Java (Spring Boot).

Para a construção de **Bancos de dados**: MySQL, PostgreSQL, MongoDB, Firebase.

#### 1.4 Exemplos de aplicações web

Alguns exemplos de aplicações web são **redes sociais** (Facebook, X, instagram), **E-commerce** (amazon, mercado livre, shoppe), **serviços de streaming** (netflix, spotify, youtube) e **sistemas bancários**.

## 2 O QUE SÃO SISTEMAS DISTRIBUÍDOS

Um **sistema distribuído** é um conjunto de computadores interconectados que compartilham recursos e processam tarefas de maneira coordenada. O objetivo é aumentar a **eficiência, disponibilidade e escalabilidade** das aplicações.

### 1.5 Características de sistemas distribuídos

Os sistemas distribuídos têm componentes independentes que podem executar processos separadamente, isso contribui para que a tolerância de falhas seja maior, pois se algo falhar o sistema continua operando.

Sistemas distribuídos também tem uma escalabilidade melhor, já que novos servidores podem ser adicionados conforme a demanda, além da transparência que faz com que o usuário não perceba que está interagindo com múltiplos servidores.

### 1.6 Tipos e exemplos

Existem diferentes tipos de sistemas distribuídos que podem ser usados para várias funções, como por exemplo, **Computação em Nuvem** que envolve serviços como AWS, Google Cloud e Azure, essas plataformas fornecem infraestrutura e recursos computacionais sob demanda. **Sistemas de Arquivos Distribuídos**, como o Google File System e Hadoop, que permitem o armazenamento e processamento de grandes volumes de dados de forma distribuída e o **Blockchain** que é uma tecnologia de redes descentralizadas utilizada para transações seguras, como o Bitcoin e Ethereum.

O **Google Search** que processa bilhões de buscas diariamente por meio de data centers distribuídos globalmente, a **Netflix** que utiliza servidores distribuídos para garantir o streaming contínuo de conteúdos em diferentes regiões e as **Redes P2P** (peer-to-peer), como BitTorrent, são exemplos de plataformas que utilizam os sistemas distribuídos.

### 1.7 Desafios

Como tudo na área de tecnologia, os sistemas distribuídos também têm seus desafios, esses sendo a **latência de rede** que se refere ao tempo necessário para



que a comunicação entre os diferentes servidores ou computadores ocorra. Se essa comunicação for lenta, pode impactar negativamente o desempenho do sistema; A **segurança**, pois sistemas distribuídos estão sujeitos a ataques cibernéticos, como invasões e roubos de dados, o que pode comprometer a integridade das informações armazenadas e transmitidas, e o **gerenciamento da consistência dos dados**, como as informações são distribuídas entre vários servidores, é essencial garantir que todos os nós tenham acesso às mesmas versões dos dados, o que pode ser um processo complexo e desafiador.

### 3 ARQUITETURAS

#### 1.8 Arquitetura monolítica

A **arquitetura monolítica** é um modelo tradicional onde toda **a aplicação é construída como um único bloco**, incluindo interface, lógica de negócios e banco de dados, **todas as funcionalidades estão no mesmo código-fonte**.

Suas características mais marcantes são o desenvolvimento e implantação unificados, ou seja, qualquer mudança exige a redistribuição do sistema inteiro, além de que se uma parte der problema isso afeta o site inteiro.

#### 1.9 Arquitetura de microsserviços

A **arquitetura de microsserviços** divide a aplicação em serviços menores e independentes, que **se comunicam por meio de APIs**.

Os **serviços são independentes**, o que significa que cada um pode ser desenvolvido e implantado separadamente. Além disso, a escalabilidade é modular, permitindo que apenas partes específicas do sistema sejam ampliadas conforme a necessidade. Esses sites podem utilizar de protocolos como HTTP(REST) ou sistemas de mensageria, como Kafka e RabbitMQ.

## 4 MONOLÍTICO X MICROSERVIÇOS

Tabela 1: Comparação de vantagens e desvantagens

Arquitetura Monolítica		Arquitetura de microsserviços
VANTAGENS	Fácil desenvolvimento inicial	Facilidade de escalabilidade
	Desempenho otimizado para pequenos sistemas	Manutenção simplificada
	Menos custos com infraestrutura	Melhor resiliência
DESVANTAGENS	Dificuldade de escalabilidade	Aumento da complexidade
	Risco de falha única	Maior consumo de recursos
	Dificuldade em atualizações	Gerenciamento de dados mais complexo

## 5 CONCLUSÃO

As aplicações web são essenciais na tecnologia moderna, permitindo acesso remoto a sistemas e serviços. Para tornar essas aplicações mais eficientes, os sistemas distribuídos possibilitam escalabilidade e alta disponibilidade. Na arquitetura de software, a escolha entre monólito e microsserviços depende das necessidades do projeto: enquanto monólitos são mais simples e fáceis de desenvolver inicialmente, microsserviços oferecem mais flexibilidade e escalabilidade para sistemas complexos.

## 6 REFERENCIAS

FOWLER, Martin. **Monolithic vs. Microservices Architecture**. Disponível em: <https://martinfowler.com/articles/microservices.html>. Acesso em: 11/02/2025.

AWS. **What is a Monolithic Architecture?** Disponível em: <https://aws.amazon.com/>. Acesso em: 12/02/2025.

NETFLIX Tech Blog. **Evolution of Microservices at Netflix**. Disponível em: <https://netflixtechblog.com/>. Acesso em: 12/02/2025.