CENTRO PAULA SOUZA

ETEC PROF MARIA CRISTINA MEDEIROS

Técnico em Informática para Internet Integrado ao Ensino Médio

Guilherme Nakamura Carvalho

APLICAÇÕES WEB E SISTEMAS DISTRIBUÍDOS

Ribeirão Pires

2025

Guilherme Nakamura Carvalho

APLICAÇÕES WEB E SISTEMAS DISTRIBUÍDOS

Trabalho de Sistemas WEB II apresentado ao Curso Técnico em Informática para Internet da Etec MCM, orientado pelo Prof. Anderson Vanin, como requisito parcial para obtenção do título de técnico em Fundamentos da Internet

Ribeirão Pires

2025

SUMÁRIO

[1 INTRODUÇÃO 3](#_Toc371608834)

[2 APLICAÇÕES WEB 4](#_Toc2003344961)

[2.1 O QUE SÃO? 5](#_Toc788720314)

[2.2 CARACTERÍSTICAS 5](#_Toc505019347)

[2.3 EXEMPLOS 5](#_Toc2082287435)

[3 SISTEMAS DISTRIBUÍDOS 6](#_Toc1153370625)

[3.1 O QUE SÃO? 7](#_Toc264940062)

[3.2 CARACTERÍSTICAS 7](#_Toc196411148)

[3.3 EXEMPLOS 7](#_Toc597979420)

[4 ARQUITETURA MONOLÍTICA 8](#_Toc1318719141)

[4.1 O QUE É? 9](#_Toc1383318649)

[4.2 CARACTERISTÍCAS 9](#_Toc1968319205)

[4.3 EXEMPLOS 9](#_Toc1807103195)

[5 ARQUITETURA DE MICROSERVIÇOS 10](#_Toc494783578)

[5.1 O QUE É? 11](#_Toc593100954)

[5.2 CARACTERÍSTICAS 11](#_Toc1870100365)

[5.3 EXEMPLOS 12](#_Toc43980472)

[6 ARQUITETURA MONOLÍTICA X MICROSERVIÇO 12](#_Toc1277984806)

[7 CONCLUSÃO 13](#_Toc426786960)

[8 BIBLIOGRAFIA 14](#_Toc736921212)

# INTRODUÇÃO

Com o avanço da tecnologia e a crescente digitalização dos serviços, diferentes arquiteturas e modelos de desenvolvimento de software têm sido utilizados para atender às necessidades do mercado. Entre esses modelos, destacam-se as aplicações web, os sistemas distribuídos, a arquitetura monolítica e a arquitetura de microsserviços, cada um com suas próprias características, vantagens e desafios.

As aplicações web tornaram-se essenciais no dia a dia, permitindo o acesso a serviços diretamente pelo navegador, sem a necessidade de instalação local. Já os sistemas distribuídos possibilitam o processamento de dados e operações de forma descentralizada, aumentando a eficiência e a disponibilidade dos sistemas.

Quando se trata da estrutura dos aplicativos, a arquitetura monolítica é um modelo tradicional, onde todas as funcionalidades estão integradas em um único sistema. Por outro lado, a arquitetura de microsserviços divide o sistema em pequenas partes independentes, permitindo maior escalabilidade e flexibilidade no desenvolvimento.

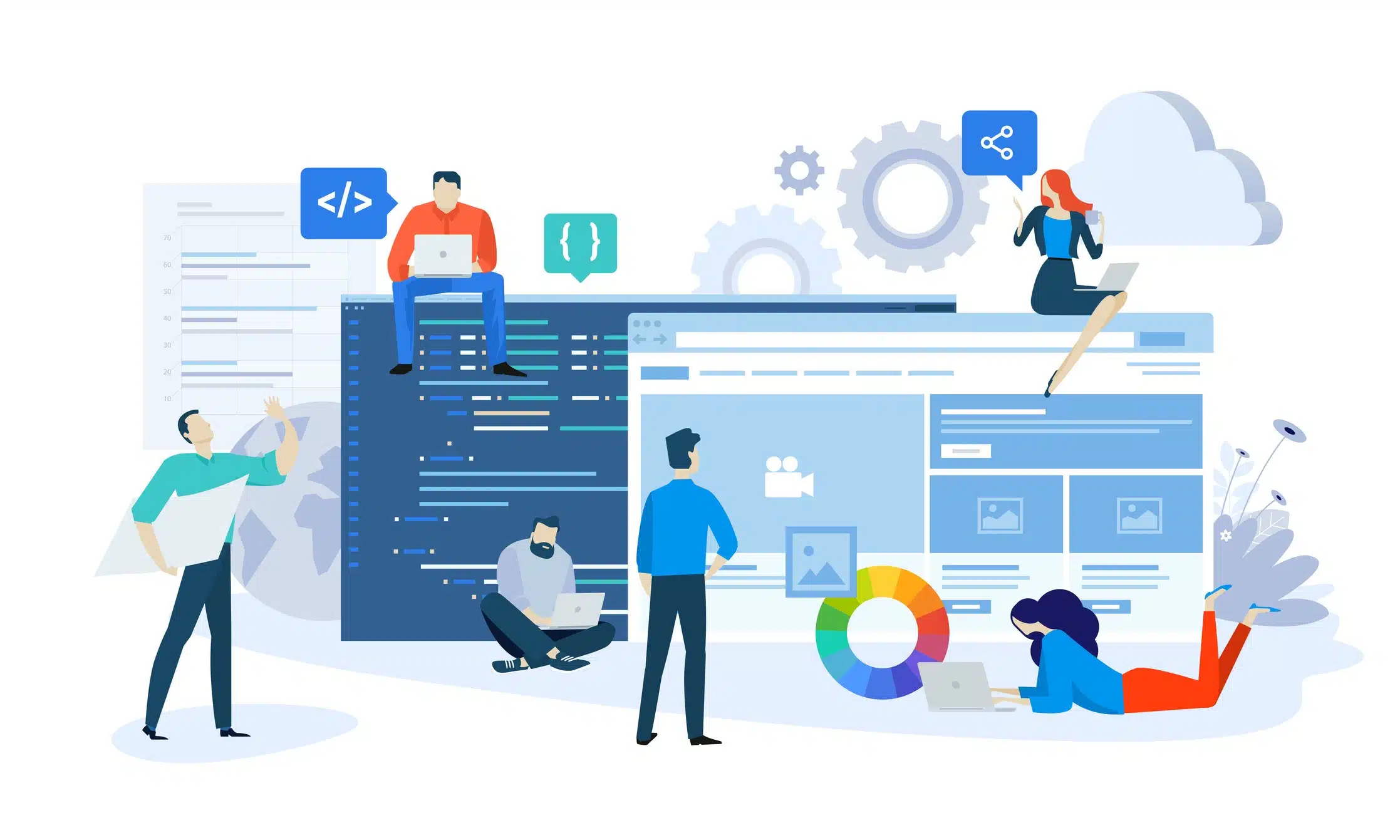
# APLICAÇÕES WEB

## O QUE SÃO?

As aplicações web são softwares que operam a partir de um navegador, necessitando apenas de uma conexão com a internet para funcionar.

Diferentemente de outros sistemas que precisam de instalações no dispositivo do usuário, a aplicação web conecta-se a servidores remotos que permitem a utilização do serviço via internet.

Um exemplo desse tipo de solução é o Google Docs, um editor de texto em nuvem que possibilita o acesso ao usuário a partir de qualquer dispositivo, pois não exige a instalação de um software local. Em contrapartida, o Microsoft Word, uma solução semelhante que também é amplamente usada, exige a instalação do aplicativo para desktop, permitindo o acesso apenas a partir de um aparelho específico.



## CARACTERÍSTICAS

As principais características dos aplicativos para web são a acessibilidade, a interatividade com o sistema, as atualizações em tempo real e a escalabilidade desse tipo de software. A alta acessibilidade garante o acesso aos usuários por qualquer dispositivo conectado à internet, adaptando a solução às especificações e necessidades de cada tipo de dispositivo ou consumidor.

## EXEMPLOS

Exemplos de aplicativos da web incluem webmail, processadores de texto e planilhas.

Além disso, existem programas de e-mail populares como o Gmail e Hotmail, além de serviços de mensagens instantâneas que também são aplicativos web.

Os aplicativos web também permitem que os membros de uma equipe trabalhem juntos no mesmo documento, é o caso do Google Workspace, incluindo Google Docs, Google Slides, Google Sheets e armazenamento em nuvem.



# SISTEMAS DISTRIBUÍDOS

## O QUE SÃO?

Um sistema distribuído é uma coleção de programas de computador que utilizam recursos computacionais em vários pontos centrais de computação diferentes para atingir um objetivo comum e compartilhado.

Também conhecido como computação distribuída ou bancos de dados distribuídos, ele depende de pontos centrais diferentes para se comunicar e sincronizar em uma rede comum. Esses pontos centrais costumam representar dispositivos de hardware físicos diferentes, mas também podem representar processos de software diferentes ou outros sistemas encapsulados recursivos. Os sistemas distribuídos visam remover gargalos ou pontos centrais de falha de um sistema.



## CARACTERÍSTICAS

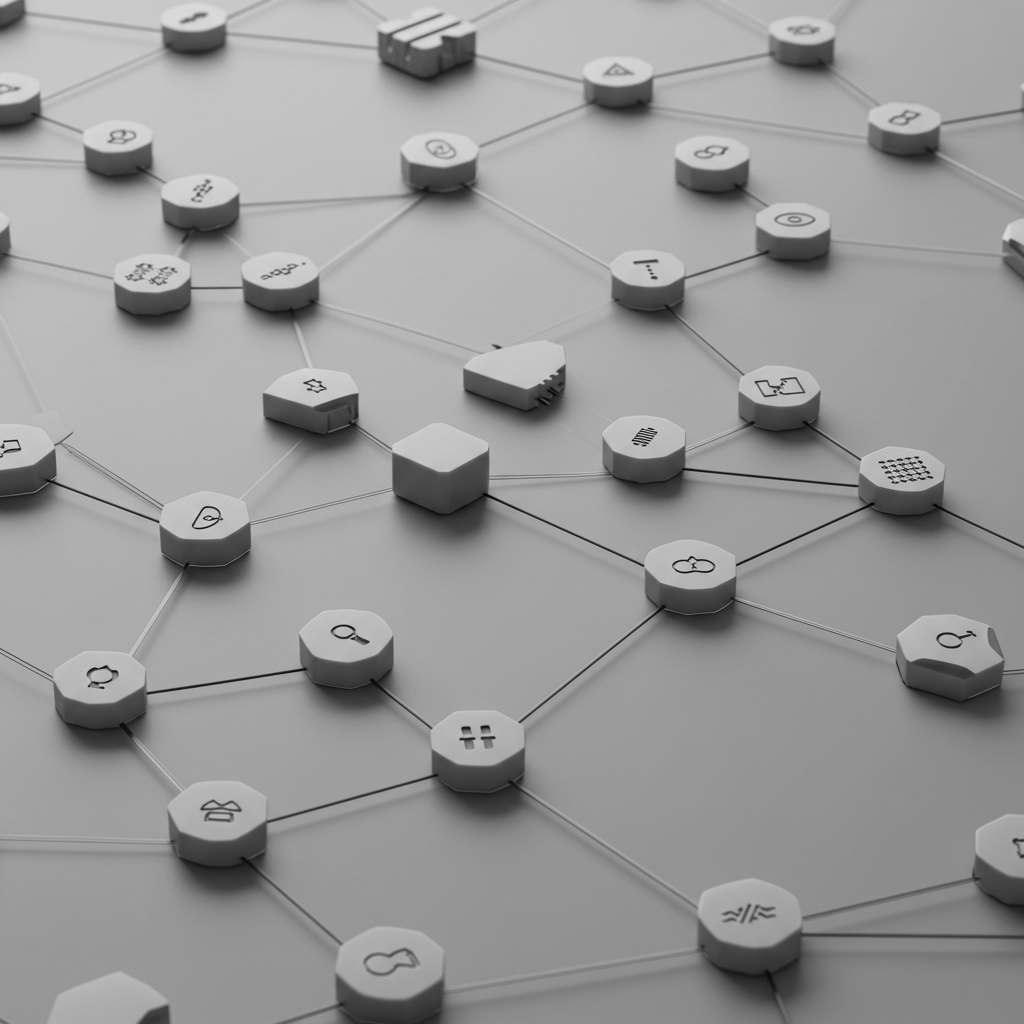
Os sistemas distribuídos possuem características essenciais para seu funcionamento eficiente. O compartilhamento de recursos permite que dispositivos utilizem hardware, software ou dados de forma otimizada. O processamento simultâneo possibilita que várias máquinas executem a mesma função ao mesmo tempo, aumentando o desempenho.

O escalonamento garante que a capacidade do sistema cresça conforme a demanda, adicionando novos recursos quando necessário. A detecção de erros facilita a identificação de falhas, tornando o sistema mais confiável. Além disso, a transparência permite que diferentes pontos do sistema se comuniquem de forma fluida, sem impactar a experiência do usuário.

## EXEMPLOS

Muitos aplicativos modernos utilizam sistemas distribuídos. Aplicativos móveis e da web de alto tráfego são sistemas distribuídos. Os usuários se conectam de maneira cliente-servidor, em que o cliente é um navegador da Web ou um aplicativo móvel. O servidor é, então, seu próprio sistema distribuído. Os servidores da web modernos seguem um padrão de sistema de vários níveis. Um balanceador de carga é usado para delegar solicitações a muitos pontos centrais lógicos do servidor que se comunicam por sistemas de fila de mensagens.

O Kubernetes é uma ferramenta popular para sistemas distribuídos, pois pode criar um sistema distribuído usando uma coleção de contêineres. Os contêineres criam pontos centrais do sistema distribuído, e então o Kubernetes orquestra a comunicação de rede entre os pontos centrais e lida com o escalonamento horizontal e vertical dinâmico dos pontos centrais no sistema.

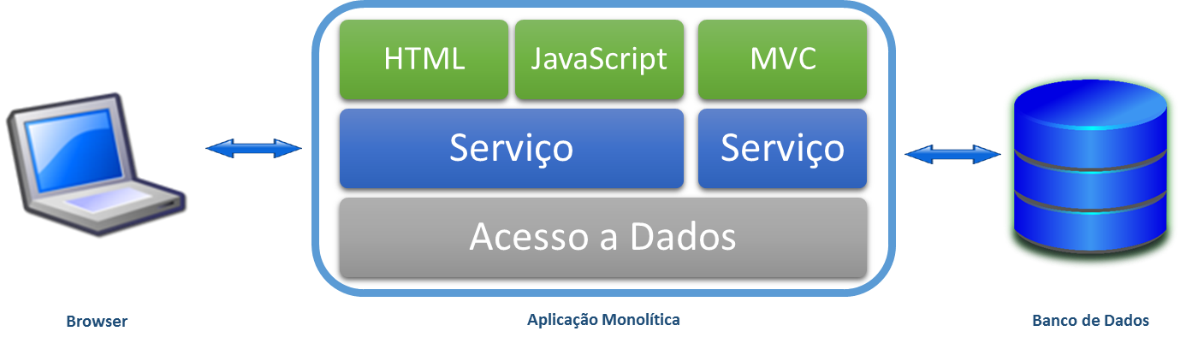


# ARQUITETURA MONOLÍTICA

## O QUE É?

A arquitetura monolítica é um modelo de desenvolvimento de software em que todas as funcionalidades são integradas em um único aplicativo, geralmente com uma única base de código e um único processo de implantação. Normalmente é usada no backend, onde o servidor é responsável por gerenciar as solicitações e fornecer as respostas.

O objetivo principal da arquitetura monolítica é simplificar o desenvolvimento de aplicativos e torná-lo mais fácil de gerenciar. Como todos os componentes estão em um único lugar, os desenvolvedores têm mais controle sobre o código e podem fazer alterações mais facilmente. Além disso, a arquitetura monolítica pode ser mais fácil de testar, implantar e escalar.



## CARACTERISTÍCAS

A arquitetura monolítica é caracterizada por sua estrutura unificada, onde todo o código-fonte está centralizado em um único projeto. Isso facilita o desenvolvimento inicial e a organização do sistema.

Todos os componentes do software rodam em um único processo, permitindo comunicação interna direta sem necessidade de APIs externas. Isso torna a execução mais eficiente, mas pode dificultar a manutenção conforme o sistema cresce.

A implantação é feita de forma única, ou seja, qualquer atualização exige a substituição completa da aplicação. Além disso, a escalabilidade é limitada, pois a única forma de expandir o sistema é replicando toda a aplicação em novos servidores, o que pode não ser ideal para grandes projetos.

## EXEMPLOS

Um exemplo clássico de uma aplicação monolítica é um sistema de gerenciamento de conteúdo (CMS) que possui todas as funcionalidades integradas em um único pacote. Plataformas como WordPress e Joomla! são exemplos de CMS que operam sob uma arquitetura monolítica, onde a interface do usuário, a lógica de negócios e o banco de dados estão todos interligados. Essas plataformas são fáceis de instalar e usar, mas podem enfrentar desafios à medida que as necessidades do usuário se expandem.



# ARQUITETURA DE MICROSERVIÇOS

## O QUE É?

Uma arquitetura de microsserviços é um tipo de arquitetura em que o aplicativo é desenvolvido como uma coleção de serviços. Ela fornece o framework para desenvolver, implantar e manter diagramas e serviços de arquitetura de microsserviços de maneira independente.

Em uma arquitetura de microsserviços, cada microsserviço é um serviço único criado para acomodar um recurso do aplicativo e processar tarefas distintas. Cada microsserviço se comunica com outros serviços por meio de interfaces simples para resolver problemas comerciais.



## CARACTERÍSTICAS

A arquitetura de microsserviços é composta por serviços independentes, permitindo que cada componente seja desenvolvido, alterado e implantado sem impactar o restante do sistema. Isso facilita a implementação de novas funções e acelera o desenvolvimento.

A estrutura é sustentável e testável, pois possibilita a rápida identificação e correção de falhas, além de permitir reverter mudanças caso algo não funcione. Equipes pequenas e autônomas trabalham de forma ágil e descentralizada, seguindo práticas de DevOps para aumentar a eficiência.

Os serviços são organizados com base em funcionalidades comerciais, garantindo que equipes multidisciplinares desenvolvam cada funcionalidade de forma independente. Além disso, a automação da infraestrutura com práticas de CI/CD permite entregas contínuas, reduzindo impactos em outras partes do sistema.

## EXEMPLOS

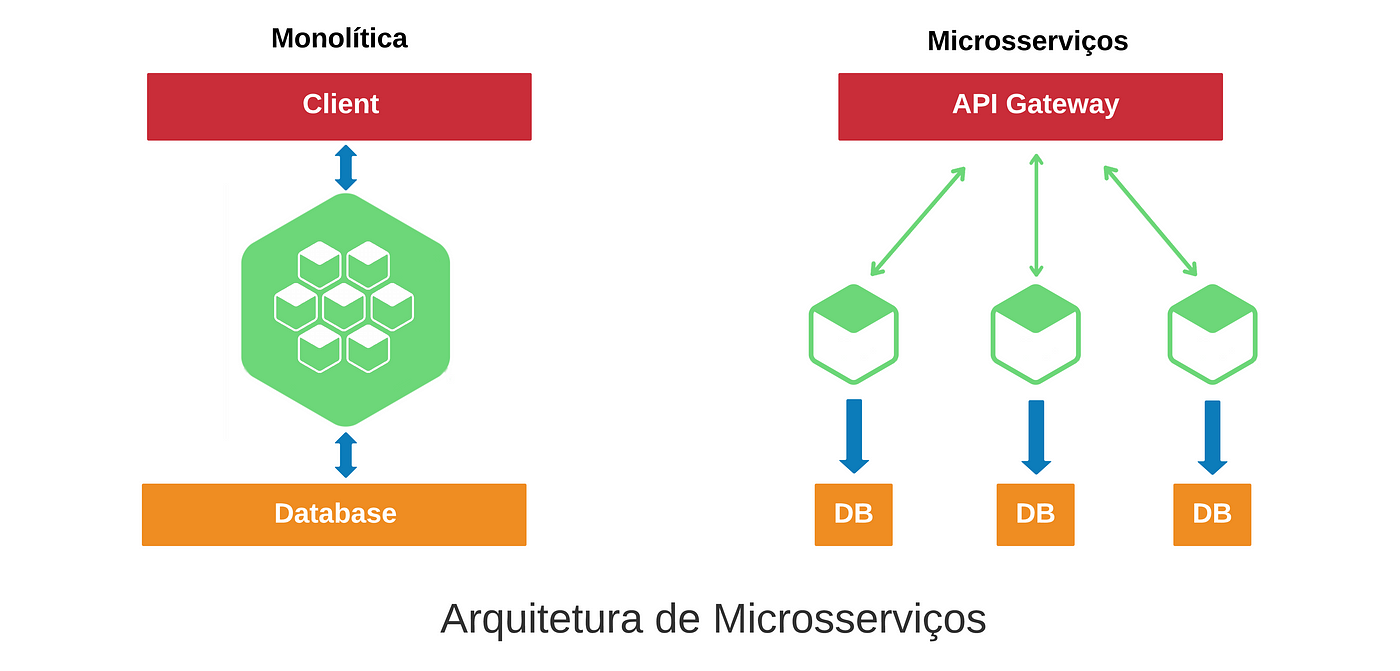
Empresas como Netflix, Amazon e Uber utilizam microsserviços para separar funções como recomendações, pagamentos e roteamento, garantindo escalabilidade e eficiência. Spotify e Google também adotam essa abordagem, permitindo atualizações independentes em serviços como streaming e e-mails.



# ARQUITETURA MONOLÍTICA X MICROSERVIÇO

Arquitetura monolítica é o modelo tradicional do programa de software, que é construído como unidade unificada e é autossuficiente e independente de outros aplicativos. Arquitetura de microsserviços é o contrário, pois usa vários serviços implementáveis com independência. Os monólitos podem ser convenientes no início da vida de um projeto para facilitar o gerenciamento de código, a sobrecarga cognitiva e a implementação. Mas, quando o aplicativo monolítico cresce e se torna complexo, fica difícil escalar, a implementação contínua fica mais desafiadora e as atualizações podem ser complicadas.

Enquanto o aplicativo monolítico é criado como unidade indivisível, os microsserviços dividem essa unidade em uma coleção de unidades independentes que contribuem para o todo. O aplicativo é criado como uma série de serviços implementáveis com independência, descentralizados e desenvolvidos com autonomia.



# CONCLUSÃO

Nesse trabalho foi abordado como assunto as APLICAÇÕES WEB, SISTEMAS DISTRIBUÍDOS, ARQUITETURA MONOLÍTICA E ARQUITETURA DE MICROSERVIÇOS, nele foi abordado assuntos importantes como as suas definições e características, por exemplo.

Além disso, foram apresentados exemplos práticos de cada uma dessas abordagens, demonstrando como são aplicadas no desenvolvimento de software moderno. A comparação entre arquitetura monolítica e microsserviços destacou as vantagens e desafios de cada modelo, ajudando a entender quando e por que optar por um em detrimento do outro.

Com o avanço da tecnologia e a crescente demanda por escalabilidade, flexibilidade e desempenho, as arquiteturas distribuídas e baseadas em microsserviços têm se tornado cada vez mais populares. No entanto, a escolha da abordagem ideal deve levar em consideração fatores como complexidade do projeto, necessidade de manutenção e escalabilidade a longo prazo.

Portanto, compreender essas arquiteturas e suas características é essencial para o desenvolvimento de softwares eficientes, seguros e adaptáveis às demandas do mercado atual.

# BIBLIOGRAFIA

ZETTLER, Kev. **O que é um sistema distribuído?.** Disponível em: https://www.atlassian.com/br/microservices/microservices-architecture/distributed-architecture. Acesso em: 12 fevereiro 2025

PM3. **Aplicação web: o que é, características, princípios e como criar.** Disponível em: https://pm3.com.br/blog/aplicacao-web/?gad\_source=1&gclid=EAIaIQobChMI\_-nGwdS-iwMVXlRIAB21tAI8EAAYASAAEgKQm\_D\_BwE. Acesso em: 12 fevereiro 2025

J, Gabriel. **O que é e para o que serve arquitetura monolítica?.** Disponível em: https://dev.to/gabrielgcj/o-que-e-e-para-o-que-serve-arquitetura-monolitica-3pp4. Acesso em: 12 de fevereiro 2025

SOFTWARE. **O que é Monolithic Architecture e para que serve?.** Disponível em: https://programae.org.br/termos/glossario/o-que-e-monolithic-architecture-e-para-que-serve/#:~:text=Exemplos%20de%20Arquitetura%20Monol%C3%ADtica&text=Plataformas%20como%20WordPress%20e%20Joomla,de%20dados%20est%C3%A3o%20todos%20interligados. Acesso em: 12 de fevereiro 2025

CLOUD, Google. **O que é a arquitetura de microsserviços?.** https://cloud.google.com/learn/what-is-microservices-architecture?hl=pt-BR. Acesso em: 12 de fevereiro 2025

HARRIS, Chandler. **Arquitetura de microsserviços.** https://www.atlassian.com/br/microservices/microservices-architecture. Acesso em: 12 de fevereiro 2025