

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

Paulo Augusto Costa

**PESQUISA SOBRE APLICAÇÕES WEB, SISTEMAS DISTRIBUÍDOS
E ARQUITETURA MONOLÍTICA E EM MICROSERVIÇOS**

**Ribeirão Pires
2025**

Paulo Augusto Costa

**PESQUISA SOBRE APLICAÇÕES WEB, SISTEMAS DISTRIBUÍDOS
E ARQUITETURA MONOLÍTICA E EM MICROSERVIÇOS**

Pesquisa sobre aplicações web, sistemas distribuídos, arquitetura monolítica e arquitetura em microsserviços apresentado ao Curso Técnico em Informática para Internet Integrado ao Ensino Médio na ETEC Prof. Maria Cristina Medeiros, orientado pelo Prof. Anderson Vanin como requisito parcial para obtenção de menção no componente Sistemas Web II.

**Ribeirão Pires
2025**

SUMÁRIO

1	INTRODUÇÃO	3
2	APLICAÇÕES WEB	4
2.1	Benefícios	4
2.2	Exemplos	4
2.3	Funcionamento	5
3	SISTEMAS DISTRIBUÍDOS	6
3.1	Características	6
3.2	Diferenças entre sistema distribuído e sistema centralizado	6
4	ARQUITETURAS	8
4.1	Arquitetura monolítica	8
4.2	Arquitetura em microsserviços	8
4.3	Diferenças entre arquitetura monolítica e em microsserviços	8
5	CONCLUSÃO	10

1 INTRODUÇÃO

Trabalho sobre os conceitos de aplicações webs, sistemas distribuídos, arquitetura monolítica, arquitetura em microsserviços e suas diferenças.

2 APLICAÇÕES WEB

De acordo com a AWS (2025), uma aplicação web é um software executado diretamente em um navegador. As empresas utilizam aplicações web para trocar informações e oferecer serviços remotamente, proporcionando uma experiência conveniente e segura para os clientes. Recursos comuns em sites, como carrinhos de compras, mecanismos de busca e filtragem de produtos, mensagens instantâneas e feeds de notícias em redes sociais, são exemplos de aplicações web. Elas permitem o acesso a funcionalidades avançadas sem a necessidade de instalação ou configuração de softwares.

2.1 Benefícios

Alguns dos benefícios do uso de aplicações Web são:

- **Acessibilidade:** Aplicações Web podem ser acessadas de todos os navegadores da Web e em vários dispositivos pessoais e empresariais.
- **Desenvolvimento eficiente:** O desenvolvimento de aplicações web é relativamente simples e econômico para as empresas. Equipes pequenas conseguem criar e lançar aplicações de forma ágil, tornando esse método uma opção eficiente e acessível para a criação de softwares.
- **Escalabilidade:** Aplicações web permitem adicionar usuários sem custos com infraestrutura, pois armazenam dados na nuvem, eliminando a necessidade de investir em hardware ou armazenamento adicional.

2.2 Exemplos

São exemplos de aplicações Web:

- **Google Docs:** Editor de documentos online que permite criação, edição e colaboração em tempo real sem a necessidade de instalação.

- Netflix: Plataforma de streaming que fornece acesso a filmes e séries diretamente pelo navegador, sem precisar baixar arquivos.
- Gmail: Serviço de e-mail baseado na web que permite o envio, recebimento e organização de mensagens de qualquer dispositivo com internet.

2.3 Funcionamento

As aplicações Web têm uma arquitetura de cliente-servidor. Seu código é dividido em dois componentes: scripts do lado do cliente e scripts do lado do servidor.

Lado do cliente:

- O script do lado do cliente gerencia a interface do usuário, carregando e renderizando elementos gráficos e interativos no navegador. O usuário pode visualizar conteúdos, assistir a vídeos ou preencher formulários.

Lado do servidor:

- Os scripts do lado do servidor processam dados e respondem às solicitações do cliente, como buscar, editar ou salvar informações. Ao clicar em "Ler mais", o servidor retorna o conteúdo solicitado. Já ao clicar em "Enviar", os dados são armazenados no banco de dados. Em alguns casos, o servidor envia a página HTML completa ao cliente, um processo chamado renderização no lado do servidor.

3 SISTEMAS DISTRIBUÍDOS

Segundo Zettler (2025), um sistema distribuído é um conjunto de programas que utilizam recursos computacionais em diversos pontos de processamento distintos para alcançar um objetivo compartilhado. Também chamado de computação distribuída ou bancos de dados distribuídos, ele depende de diferentes pontos para comunicação e sincronização em uma rede comum. Esses pontos podem representar tanto dispositivos de hardware físico quanto processos de software ou outros sistemas encapsulados. O objetivo dos sistemas distribuídos é eliminar gargalos e pontos de falha em um sistema.

3.1 Características

Algumas das características dos sistemas distribuídos são:

- **Compartilhamento de recursos:** um sistema distribuído pode compartilhar hardware, software ou dados.
- **Processamento simultâneo:** várias máquinas podem processar a mesma função ao mesmo tempo
- **Escalonamento:** a capacidade de computação e processamento pode evoluir conforme necessário quando estendida para máquinas adicionais
- **Deteção de erros:** as falhas podem ser detectadas com mais facilidade
- **Transparência:** um ponto central pode acessar os e se comunicar com outros pontos centrais no sistema.

3.2 Diferenças entre sistema distribuído e sistema centralizado

Um sistema de computação centralizado é aquele em que toda a computação é realizada por um único computador em um único local. A principal diferença entre sistemas centralizados e distribuídos está no padrão de comunicação entre os pontos centrais do sistema. Em um sistema centralizado, o estado é armazenado em um único ponto central, acessado de forma personalizada pelos clientes. Todos os pontos centrais do sistema se conectam a esse ponto, o que pode causar congestionamento e lentidão na rede. Além disso, um sistema centralizado apresenta um único ponto de falha, enquanto um sistema distribuído não possui esse tipo de vulnerabilidade.

4 ARQUITETURAS

A arquitetura de software é fundamental para definir como os componentes de um sistema interagem entre si, impactando diretamente a escalabilidade, manutenção e flexibilidade do sistema. Entre as principais abordagens de arquitetura, destacam-se a monolítica e a de microsserviços.

4.1 Arquitetura monolítica

De acordo com AWS (2025), uma arquitetura monolítica é um modelo tradicional de desenvolvimento de software em que uma única base de código é responsável por executar diversas funções comerciais. Nesse modelo, todos os componentes do sistema estão interconectados, o que significa que os dados trocados entre eles dependem uns dos outros. Isso torna a arquitetura monolítica restritiva e difícil de modificar, pois até pequenas alterações podem impactar amplamente diferentes partes da base de código, tornando o processo de atualização ou manutenção mais demorado e complexo.

4.2 Arquitetura em microsserviços

Os microsserviços são uma abordagem arquitetônica que divide o software em pequenos componentes ou serviços autônomos. Cada serviço realiza uma função específica e se comunica com outros por meio de interfaces bem definidas. Como operam de forma independente, é possível atualizar, modificar, implantar ou escalar cada serviço conforme a necessidade.

4.3 Diferenças entre arquitetura monolítica e em microsserviços

As aplicações monolíticas geralmente são compostas por uma interface de usuário no lado do cliente, um banco de dados e uma aplicação no lado do servidor, com todos esses módulos desenvolvidos dentro de uma única base de código. Por outro lado, na arquitetura de microsserviços, cada serviço é responsável por um único recurso ou lógica de negócios. Em vez de compartilhar dados na mesma base de código, os microsserviços se comunicam entre si por meio de APIs.

Categoria	Monolítica	Microserviços
Design	Base de código única com várias funções interdependentes.	Componentes independentes que se comunicam entre si usando APIs.
Desenvolvimento	Requer menos planejamento no início, mas se torna complexo de manter	Requer mais planejamento no início, mas se torna mais fácil de se manter com o tempo
Implantação	Aplicação inteira implantada como uma única entidade	Cada microserviço é uma entidade independente que requer implantação individual em contêineres
Depuração	Rastreia o caminho do código no mesmo ambiente	Requer ferramentas avançadas para rastrear a troca de dados entre microserviços
Modificação	Pequenas mudanças introduzem riscos maiores, pois afetam toda a base de código	É possível modificar microserviços individuais sem afetar toda a aplicação
Escala	É necessário escalar toda a aplicação, mesmo que apenas determinadas áreas funcionais tenham um aumento na demanda	É possível escalar microserviços individuais conforme necessário, economizando custos gerais de escalabilidade
Investimento	Baixo investimento inicial à custa de maiores esforços contínuos e de manutenção	Investimento de tempo e custo para configurar a infraestrutura necessária e desenvolver a competência da equipe porém retorna economia de custos, manutenção e adaptabilidade a longo prazo

5 CONCLUSÃO

Com essa pesquisa foi possível compreender que as aplicações web oferecem uma forma eficiente e acessível para empresas se conectarem com seus clientes, permitindo a troca de informações e serviços de forma remota. Além disso, os sistemas distribuídos se destacam por sua capacidade de compartilhar recursos, processar dados simultaneamente e evitar pontos de falha, apresentando uma arquitetura robusta para ambientes mais complexos.

A comparação entre a arquitetura monolítica e a de microsserviços mostrou as vantagens e desvantagens de ambas. Enquanto a monolítica é mais simples de implementar em fases iniciais de um projeto, a arquitetura de microsserviços oferece maior escalabilidade e flexibilidade a longo prazo, sendo mais eficiente para sistemas em crescimento.

Portanto, entender as diferenças entre essas abordagens é essencial para a escolha da arquitetura mais adequada a cada tipo de projeto, levando em conta suas necessidades de escalabilidade, manutenção e atualização.

REFERÊNCIAS

AMAZON WEB SERVICES. O que é uma aplicação web?. Disponível em: <https://aws.amazon.com/pt/what-is/web-application/>. Acesso em: 23 fev. 2025.

ZETTLER, Kev. O que é um sistema distribuído?. Atlassian. Disponível em: <https://www.atlassian.com/br/microservices/microservices-architecture/distributed-architecture>. Acesso em: 23 fev. 2025.

AMAZON WEB SERVICES. Qual é a diferença entre arquitetura monolítica e de microsserviços?. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-monolithic-and-microservices-architecture/>. Acesso em: 23 fev. 2025.