CENTRO PAULA SOUZA
ETEC PROF. MARIA CRISTINA MEDEIROS
Técnico em Informática para a Internet Integrado Ao Ensino Médio

João Pedro Morangoni

PESQUISA SOBRE CONCEITOS BÁSICOS NA MATÉRIA DE SISTEMAS WEB II

Ribeirão Pires
2025

João Pedro Morangoni

PESQUISA SOBRE CONCEITOS BÁSICOS NA MATÉRIA DE SISTEMAS WEB II

Pesquisa Sobre Conceitos Básicos Apresentados na matéria SW-II apresentado ao Curso Técnico em Informática Para A Internet com Ensino Médio integrado da Etec Prof. Maria Cristina Medeiros, orientado pelo Prof. Anderson Vanin, como requisito parcial para obtenção da menção na matéria de SWII

Ribeirão Pires 2025

“Não creio que haja uma emoção mais intensa para um inventor do que ver suas criações funcionando. Tais emoções o fazem esquecer da realidade, comer e dormir se tornam esquecíveis. ”

Nikola Tesla

OBJETIVOS

Esta pesquisa tem como seu principal objetivo se aprofundar nos conceitos apresentados pelo professor durante o seguimento das aulas de SW, a fim de uma melhor compreensão dos temas abordados.

SUMÁRIO

1 INTRODUÇÃO

A área da computação possui diversas ramificações e conceitos a serem explorados, devido a essa amplitude, iremos definir alguns destes conceitos e explorar parcialmente a sua capacidade

1.1 APLICAÇÕES WEB

Trata-se de uma aplicação de software que roda na internet, em vez de funcionar com base em sistemas operacionais.

Assim, é um sistema com funcionalidades completas, que foi programado a partir de requisitos e dos princípios da engenharia de software. Contudo, seu grande diferencial é que ele é feito para funcionar na internet.

O que permitiu o surgimento de plataformas cada vez mais robustas na web foi, justamente, a computação em nuvem. Ela mudou a forma de pensar aplicações, de um paradigma de produtos para um novo, focado em serviços.

Nesse caso, as aplicações funcionam como um serviço, oferecendo alguma função importante para os usuários. Como exemplo, podemos mencionar o e-mail, os sites de redes sociais e os sites de editores de texto.

Alguns até oferecem um pagamento por demanda, de acordo com o uso, e se encaixam no paradigma SaaS (software como um serviço). Nesse caso, é preciso ter uma assinatura com uma conta para visualizar as informações e executar as funções necessárias.

1.1.1 FUNCIONAMENTO

Uma aplicação web funciona com base na infraestrutura da internet. Por isso, é preciso entender, primeiro, como funciona a web.

O site fica armazenado em um servidor. Quando o usuário solicita acesso por meio de um endereço URL, ele automaticamente consegue conexão com o servidor DNS (servidor com uma lista de domínios) que, por sua vez, acessa o IP que referencia aquele site.

Então, o site envia informações de download para o usuário. Por fim, o usuário pode interagir com a plataforma enviando informações, alterando e salvando novos dados etc.

1.1.2 DIFERENÇAS ENTRE WEB APP E SITE

Um website é um site estático, que foca apresentar informações e permite navegação por meio de links. É mais tradicional, uma vez que segue o formato principal dos sites, desde o começo da internet.

Por sua vez, as aplicações na web são sistemas completos hospedados em um servidor de internet. Assim, entregam serviços aos usuários, permitem a execução de funções e tarefas claras, bem como constante envio e recebimento de informações por meio dos protocolos da rede.

Um exemplo de website é um site comum de portal de notícias ou uma página corporativa. Um web app é uma plataforma como a Netflix, que permite ver filmes, salvar títulos em uma lista, gerenciar a conta, adicionar formas de pagamento, excluir a assinatura, criar perfis etc.

Enquanto um website é uma aplicação que consiste na implementação de um protótipo de design, com a parte lógica e a parte estrutural, uma aplicação é um sistema com funções bem definidas, que depende da ação do usuário.

1.2 SISTEMAS DISTRIBUÍDOS

Sistemas distribuídos pertencem a uma classe de computação chamada computação paralela *(parallel computing)*, classificados de acordo com o nível de paralelismo suportado pelo hardware, além da distância entre os processos pertencentes ao *landscape* de uma aplicação. Estão presentes neste grupo, além dos

sistemas distribuídos, outras categorias de arquiteturas como *Cluster computing, Multi-core computing, Symmetric multiprocessing, Grid computing.*

Em computação distribuída, os elementos computacionais em termos de unidade são descritos como nós (nodes), caracterizando-se como um processo de software, equipado com uma lista de outros processos, capazes de enviar mensagens diretamente uns aos outros, estimulando-os a executar e concluir as operações de negócio descritas pelo sistema.

Sistemas distribuídos são dinâmicos no sentido de que dispositivos podem juntar-se ou deixar uma unidade de execução a qualquer momento, sem impactos na performance ou topologia do cluster. Tratam-se de componentes que colaboram para realizar uma tarefa predeterminada, comunicando-se através do envio de mensagens, podendo referenciar máquinas físicas ou processos de software rodando em nuvem.

1.2.1 ANATOMIA DE UM SISTEMA DISTRIBUÍDO

Encontramos sistemas distribuídos de todos os tamanhos e formas. Atualmente, a maior parte dos sistemas em larga escala são formados por *backend services* trabalhando em conjunto para entregar business features. Existem diversos níveis de abstração pertencentes à jornada de implementação de componentes com natureza distribuída. Porém, antes de dedicarmos esforços para entender seus fundamentos técnicos, o principal desafio da arquitetura para componentes distribuídos é entendermos as diferentes maneiras existentes para decomposição de um bloco de software em partes menores, descobrindo seus relacionamentos e responsabilidades.

A arquitetura de um componente de software difere-se de acordo com a perspectiva observada. Do ponto de vista físico, um sistema distribuído não passa de um grupo de máquinas comunicando-se através de uma rede. Em *runtime*, um sistema pode ser composto por processos comunicando-se via mecanismos *Inter-process communication (IPC)* como HTTP, rodando em um *Operating System (OS).* Já sob ângulo de implementação, um sistema distribuído pode ser descrito como um conjunto de componentes, com baixo ou nenhum acoplamento, publicados e escalando individualmente cada qual com certo nível de autonomia durante a execução de um processo.

1.3 ARQUITETURA MONOLÍTICA

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.

1.3.1 VANTAGENS E DESVANTAGENS

Uma das principais vantagens da arquitetura monolítica é que ela é simples e fácil de entender. Não há necessidade de aprender várias tecnologias e plataformas para desenvolver um aplicativo monolítico. Além disso, é mais fácil encontrar desenvolvedores que saibam trabalhar com essa arquitetura, já que é muito comum em empresas de diversos setores.

Outra vantagem da arquitetura monolítica é que ela é altamente escalável. Como todo o aplicativo é executado em um único processo, é fácil aumentar o número de recursos do servidor para lidar com um maior número de solicitações. Além disso, a maioria dos provedores de hospedagem oferecem suporte a essa arquitetura e podem fornecer soluções para aumentar a capacidade de processamento e memória.

No entanto, a arquitetura monolítica pode ter algumas desvantagens. Uma delas é que pode ser difícil manter o código organizado e escalável à medida que o aplicativo cresce. Além disso, pode ser difícil adicionar novas funcionalidades sem afetar o restante do código. Isso pode levar a problemas de dependência e dificuldades para manter o aplicativo atualizado com as últimas tecnologias.

1.4 MICROSSERVIÇOS

Microsserviços são uma abordagem arquitetônica e organizacional do desenvolvimento de software na qual o software consiste em pequenos serviços independentes que se comunicam usando APIs bem definidas. Esses serviços pertencem a pequenas equipes autossuficientes.

As arquiteturas de microsserviços facilitam a escalabilidade e agilizam o desenvolvimento de aplicativos, habilitando a inovação e acelerando o tempo de introdução de novos recursos no mercado.

1.4.1 DIFERENÇAS ENTRE ARQUITETURA MONOLÍTICA E MICROSSERVIÇOS

Com as arquiteturas monolíticas, todos os processos são altamente acoplados e executam como um único serviço. Isso significa que se um processo do aplicativo apresentar um pico de demanda, toda a arquitetura deverá ser escalada. A complexidade da adição ou do aprimoramento de recursos de aplicativos monolíticos aumenta com o crescimento da base de código. Essa complexidade limita a experimentação e dificulta a implementação de novas ideias. As arquiteturas monolíticas aumentam o risco de disponibilidade de aplicativos, pois muitos processos dependentes e altamente acoplados aumentam o impacto da falha de um único processo.

Com uma arquitetura de microsserviços, um aplicativo é criado como componentes independentes que executam cada processo do aplicativo como um serviço. Esses serviços se comunicam por meio de uma interface bem definida usando APIs leves. Os serviços são criados para recursos empresariais e cada serviço realiza uma única função. Como são executados de forma independente, cada serviço pode ser atualizado, implantado e escalado para atender a demanda de funções específicas de um aplicativo.

2 CONCLUSÃO

Podemos concluir a partir desta pesquisa que existem diversas maneiras de estruturar um sistema na web, os métodos apresentados podem ser usados para obter diversos fins, cada um apresentando vantagens, regras de construção, limitações e benefícios.

REFERÊNCIAS

GONÇALVES, Marcelo M. **. Sistemas Distribuídos: Conceito e Definições**

Disponível em: https://medium.com/sicreditech/sistemas-distribuídos-conceito-e-definições-f2baa4efc88d

Acesso em:17/02/2025

MARTINS, Diana **Aplicação web: o que é, diferença para website, como funciona e mais!**.

Disponível em: https://rockcontent.com/br/blog/aplicacao-web/

Acesso em:17/02/2025

AWS, Amazon. **O que são microsserviços?**

Disponível em: https://aws.amazon.com/pt/microservices/

Acesso em:17/02/2025

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:18/02/2025