



Faculdade de Tecnologia de Sorocaba – Fatec

Tecnologia em Análise e Desenvolvimento de Sistemas

Disciplina de Programação para WEB

Atividade 2: Pesquisa sobre Arquitetura de Software

Professora: Denilce de Almeida Oliveira Veloso

Gabriel Alves Coelho 0030482311011

Sorocaba

Agosto/2024

Sumário

Arquitetura de Software	3
Introdução	3
Conceitos	3
Padrões de Arquitetura de Software	4
Arquitetura Client-Server (Cliente-Servidor)	4
Arquitetura em Camadas (Layers)	4
Arquitetura Model-View-Controller (MVC)	5
Arquitetura Orientada a Serviços (SOA)	5
Pipes and Filters (PF ou Dutos e Filtros)	6
Arquitetura Monolítica	6
Arquitetura baseada em Microserviços	6
Arquitetura Hexagonal	7
Tendências em Arquitetura de Software	9
Arquiteturas Baseadas em Eventos	9
Computação em Nuvem	9
Arquitetura Orientada a Valor	10
Arquitetura Orientada a Microserviços	10
Escolha da Arquitetura de Software	10
Conclusão	11
Referências	12

Arquitetura de Software

Introdução

Atualmente a Tecnologia da Informação é essencial para diversas atividades cotidianas e até mesmo fundamental para a rotina dos indivíduos, desta forma, o desenvolvimento de software se torna cada vez mais complexo visando atender as necessidades de cada pessoa com velocidade e precisão. Assim, devido a esta complexidade, o termo Arquitetura de Software se torna um objeto de interesse na área de TI.

De certa forma, a Arquitetura de Software tem importante papel no processo de desenvolvimento e manutenção do sistema, assim, pode ser definida como a estrutura fundamental de um software, apresentando seus componentes, funcionamento e projeto de expansão.

Em suma, a Arquitetura de Software é essencial no ciclo de vida deste e pode ser dividida em diversos padrões, os quais serão apresentados no desenvolvimento desta pesquisa.

Conceitos

Arquitetura de Software diz respeito à definição dos componentes do software, bem como suas propriedades externas e relacionamentos com outros sistemas. Assim, é a estrutura do software.

Um padrão de Arquitetura de Software diz respeito à peça fundamental para concretizar a abstração e sua escolha deve atender os requisitos do sistema a ser desenvolvido e implementado.

Como anteriormente explicado, a Arquitetura de Software representa a estrutura fundamental de um sistema e pode ser dividida em vários padrões, alguns deles são apresentados a seguir.

Padrões de Arquitetura de Software

Arquitetura Client-Server (Cliente-Servidor)

A ideia de desenvolver um sistema em camadas surgiu na década de 90 com a arquitetura cliente-servidor, a qual, separa a aplicação em duas camadas, o cliente e o servidor. O cliente é a camada responsável por manter a interface e outro código do sistema. Já o servidor é a camada responsável por manter um banco de dados, normalmente, relacional.

Este modelo apresenta um bom funcionamento para atualizações simples de dados e no caso da necessidade de exibir informações. No que se diz respeito à implementação de lógica de domínio, com validações de dados, por exemplo, este modelo não é indicado pois dificulta a manipulação do código.

Arquitetura em Camadas (Layers)

Um sistema com sua organização em camadas tem cada uma destas com responsabilidades e funcionalidades específicas e cada uma delas pode ser modificada de uma maneira independente.

Na Arquitetura em Camadas, as funcionalidades de cada camada dependem dos recursos e serviços disponibilizados pela camada abaixo dela.

Segundo Sommerville(*apud* Pessoa, 2024) este modelo é utilizado para as seguintes situações:

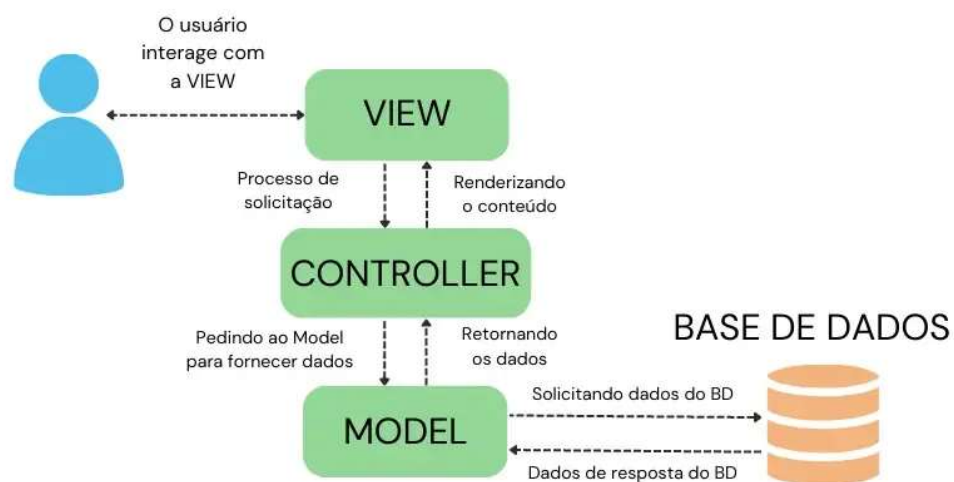
- Quando se deseja construir um recurso novo a partir de um já existente;
- Quando o desenvolvimento é feito por equipes distintas, nas quais cada uma é responsável por uma camada;
- Quando a software tem um requisito de proteção multinível.

Arquitetura Model-View-Controller (MVC)

A Arquitetura Model-View-Controller separa a apresentação e a interação dos dados do sistema, o qual é estruturado em três componentes lógicos, são eles: model (modelo), view (visualização) e controller (controlador), desta forma, os componentes apresentados interagem entre si da seguinte maneira:

O model é o responsável pelas regras de negócio e interação com os dados. O view é o responsável por definir e gerenciar o modo como os dados são apresentados ao usuário. O controller tem a responsabilidade de interagir com o usuário e responder de acordo com esta interação. Sendo assim, o controller é caracterizado por ser a camada intermediária entre o model e o view.

Figura 1: Arquitetura MVC



Fonte: (Medium)

Arquitetura Orientada a Serviços (SOA)

Neste padrão, o sistema é dividido em diversos serviços e cada um deles é responsável por uma funcionalidade. Desta forma, interfaces são desenvolvidas para melhor comunicação entre os serviços, sem que esta dependa de uma linguagem ou plataforma específica.

Pipes and Filters (PF ou Dutos e Filtros)

Esta arquitetura divide o software em componentes independentes chamados de filtros, cada filtro é responsável por uma operação no fluxo dos dados em tempo de execução. O fluxo se inicia com a entrada de um dado em um canal, em seguida ocorre sua transformação e os dados processados são enviados a um canal de saída.

Cada filtro é independente, ou seja, pode ser modificado ou substituído sem interferir no funcionamento dos demais elementos. Este padrão é interessante para aplicações que necessitam realizar o processamento de grandes quantidades de dados de maneira eficiente e escalável. Por outro lado, sua manutenção é caracterizada por alta complexidade.

Arquitetura Monolítica

Arquitetura Monolítica é o padrão que se refere a implementação de todos os componentes, funcionalidades e estruturas em conjunto em um único bloco vertical. Desta forma, todas as funcionalidades são executadas em um único processo, no caso de se apresentarem em modularização, todas as camadas são interdependentes, ou seja, cada parte do software que for modificada interfere no todo.

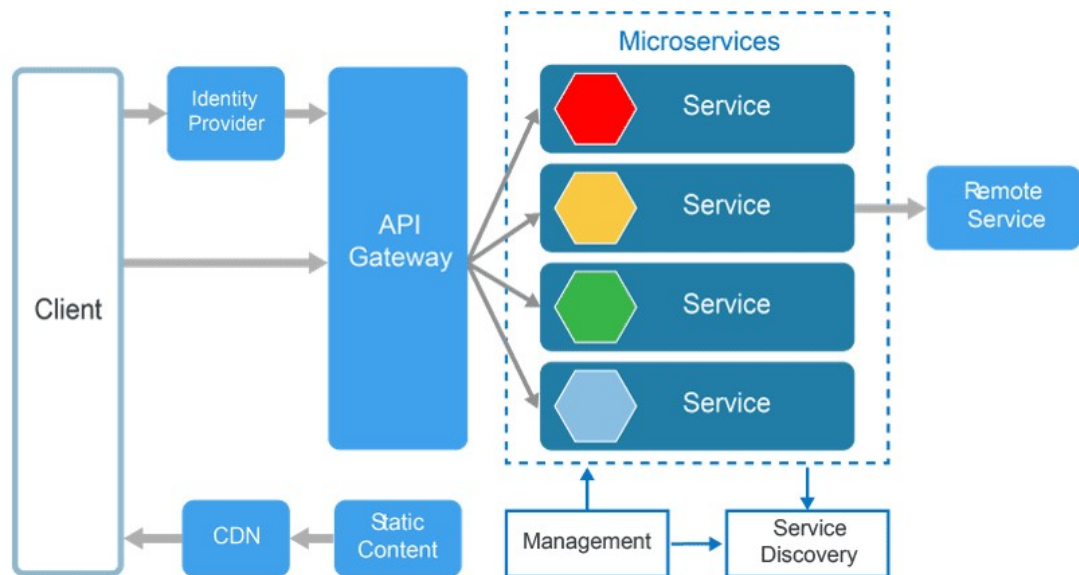
Dentre as vantagens da Arquitetura Monolítica estão a menor complexidade de implementação, deployment de uma unidade e o desenvolvimento rápido.

Arquitetura Baseada em Microsserviços

A Arquitetura baseada em Microsserviços pode ser considerada uma variante da Arquitetura Orientada a Serviços e também uma antagonista da arquitetura Monolítica, pois é possível encontrar serviços ainda menores e mais independentes com linguagens e plataformas diferentes.

Desta forma, Microserviços são aplicações distribuídas que, por sua vez são compostas de outras aplicações menores para formar sistemas complexos.

Figura 2: Arquitetura Baseada em Microserviços



Fonte: (Revista Científica Multidisciplinar Núcleo do Conhecimento)

Arquitetura Hexagonal

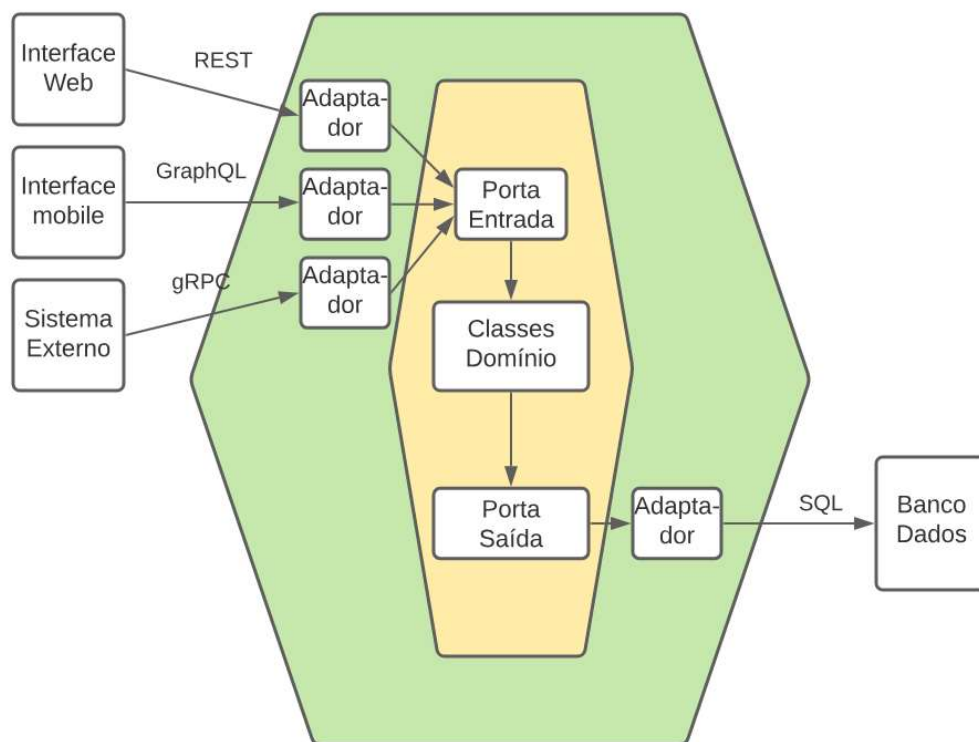
A Arquitetura Hexagonal, também conhecida como ports and adapters (portas e adaptadores) separa a lógica de negócio, fundamental para desenvolvimento das funcionalidades, da lógica de implementação. Desta forma, este padrão apresenta bastante utilidade pois desacopla a aplicação do ambiente, utilizando do conceito de portas e adaptadores, assim, facilita a integração e manutenção com outros componentes da aplicação.

As portas correspondem às interfaces que permitem que a lógica de negócios se comunique com o mundo externo, já os adaptadores têm a responsabilidade de implementar as portas.

O princípio da Arquitetura Hexagonal é proteger a camada de domínio da aplicação, elemento que rege o desenvolvimento dos componentes, no centro e permitir sua conexão com o mundo externo a partir de uma interface padrão.

O referido padrão tem como objetivo solucionar um dos principais problemas no desenvolvimento de software, o acoplamento. Assim, apresenta uma perspectiva diferente sobre a ordem das camadas (geralmente a primeira corresponde à interface e a última ao banco de dados), a ideia deste padrão é entender o front-end e o back-end como camadas externas ao domínio da aplicação.

Figura 3: Arquitetura Hexagonal



Fonte: (Engenharia de Software Moderna)

Tendências em Arquitetura de Software

A Arquitetura de Software é um campo em constante evolução buscando acompanhar as demandas do mercado de Tecnologia da Informação. Tendo em vista esta crescente evolução, são apresentadas, a seguir, algumas tendências no referido assunto.

Arquitetura Baseada em Eventos

A Arquitetura Baseada em Eventos se caracteriza por apresentar a comunicação feita entre os elementos de um sistema através de eventos assíncronos, fato que permite um aumento na flexibilidade de construção de sistemas distribuídos, na escalabilidade e na resiliência. Este padrão de arquitetura é adequado para situações nas quais a integração de sistemas é necessária, bem como em aplicações em tempo real, as quais necessitam apresentar alta disponibilidade e processamento em tempo hábil.

Computação em Nuvem

A possibilidade de se realizar o provisionamento de recursos de computação, armazenamento e serviços sob demanda de forma elástica, características da computação em nuvem, permitem o aumento da flexibilidade para as organizações. De certa forma, a utilização de serviços em nuvem permite que as empresas se concentrem em seu core business e deixem a infraestrutura por conta do provedor dos serviços. Outras vantagens da adoção da computação em nuvem são recursos de escalabilidade automática, tolerância a falhas e monitoramento, atributos que contribuem com a melhoria da disponibilidade e desempenho dos sistemas.

Arquitetura Orientada a Valor

O surgimento de novos princípios em Arquitetura de Software visa a orientação a valor, assim, a arquitetura tem seu planejamento focado na entrega de valor ao cliente final. Esta é uma abordagem que busca compreender aspectos referentes ao sistema que trazem valor ao usuário, assim, busca desenvolver interfaces mais intuitivas, por exemplo.

Arquitetura Orientada a Microsserviços

Como anteriormente definida, a Arquitetura Orientada a Microsserviços tem como objetivo dividir um sistema em serviços pequenos e independentes, aumentando, assim, a flexibilidade, a resiliência e a escalabilidade. Cabe citar que esta arquitetura favorece a utilização de algumas tecnologias emergentes, tais como contêineres e orquestração de contêineres como Kubernetes.

Escolha da Arquitetura de Software

Não existe uma Arquitetura de Software considerada correta para o desenvolvimento de um sistema, desta forma, ela deve ser escolhida de acordo com as características do software, ou seja, é fundamental considerar as peculiaridades do sistema, entre elas, a necessidade de escalabilidade, a facilidade de manutenção e evolução, bem como a necessidade de integração com outros sistemas.

Desta forma, a escolha da Arquitetura de Software se dá de acordo com a complexidade deste, sendo assim, definida de modo que se encaixe e atenda ao escopo do sistema.

Conclusão

A Arquitetura de Software é uma peça chave para este, pois determina sua estrutura, bem como seu funcionamento. A Arquitetura de Software é um determinante para atender os principais requisitos do sistema, tais como, desempenho e escalabilidade.

Dentre os diversos padrões de arquitetura existentes destacam-se a Arquitetura Cliente-Servidor, na qual o cliente mantém a camada da interface e o servidor mantém o banco de dados, Arquitetura em Camadas, as quais podem ser modificadas de forma independente, porém dependem dos recursos oferecidos pelas camadas mais baixas, Arquitetura MVC, dividida em model, responsável pela interação com os dados, view, responsável pela apresentação dos dados e controller, camada intermediária entre model e view, Arquitetura Orientada a Serviços, na qual o sistema é dividido em diversos serviços independentes, Arquitetura Dutos e Filtros, a qual divide o software em elementos chamados filtros, responsáveis por operações dentro do fluxo de dados, Arquitetura Monolítica, a qual implementa todos os elementos e estruturas em conjunto em um único processo, Arquitetura Baseada em Microserviços, considerada uma variante da Arquitetura Orientada a Serviços e Arquitetura Hexagonal, a qual tem como objetivo proteger a camada de domínio da aplicação.

Além dos padrões citados, cabe destacar a Arquitetura em Nuvem, tendência que permite às organizações focar em seu core business, enquanto o provedor de serviços é responsável pela infraestrutura. Destacam-se ainda, como tendências as Arquiteturas Orientada a Valor e Baseada em Eventos.

Em suma, a Arquitetura de Software tem fundamental importância para este, assim, é necessário que esta seja escolhida de forma que atenda os requisitos do software. Logo, a escolha da arquitetura deve considerar o sistema e todos os seus requisitos, para que tal escolha e planejamento ocorram da melhor maneira possível.

Referências

AWARI. **Tendências em Arquitetura de Software: Novas Tecnologias e Práticas Emergentes**. 2023. Disponível em: https://awari.com.br/tendencias-em-arquitetura-de-software-novas-tecnologias-e-praticas-emergentes/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Te nd%C3%AAncias%20em%20Arquitetura. Acesso em: 12 ago. 2024.

AWARI. **Tipos de Arquitetura de Software: Conheça os Mais Usados**. 2023. Disponível em: <https://awari.com.br/tipos-de-arquitetura-de-software-conheca-os-mais-usados/#:~:text=Conclus%C3%A3o-,Ao%20escolher%20a%20arquitetura%20de%20software%20para%20o%20se u%20projeto,de%20integra>. Acesso em: 12 ago. 2024.

BORGES, Gustavo de Jesus. **Uma Proposta de Arquitetura de Microserviços Aplicada em um Case de E-commerce**. 2020. Disponível em: <https://www.nucleodoconhecimento.com.br/tecnologia/arquitetura-de-microservicos>. Acesso em: 12 ago. 2024.

MODERNA, Engenharia de Software. **O que é uma Arquitetura Hexagonal?** Disponível em: <https://engsoftmoderna.info/artigos/arquitetura-hexagonal.html>. Acesso em: 12 ago. 2024.

NORMANDO, Célio. **Arquitetura MVC e princípios de projeto**. 2024. Disponível em: <https://medium.com/@celionormando/arquitetura-mvc-e-princ%C3%ADpios-de-projeto-3d0b278ef910>. Acesso em: 12 ago. 2024.

PESSÔA, Camila. **Padrões arquiteturais: arquitetura de software descomplicada**. 2024. Disponível em: <https://www.alura.com.br/artigos/padroes-arquiteturais-arquitetura-software-descomplicada?srsIid=AfmBOopBqNmsKZDRoBanQuvaEAHNSOr0LkD5orpmADwcRvU0uwgRXu5z>. Acesso em: 12 ago. 2024.

WIKIPÉDIA. **Arquitetura de software**. 2024. Disponível em: https://pt.wikipedia.org/wiki/Arquitetura_de_software. Acesso em: 12 ago. 2024.