**CENTRO UNIVERSITÁRIO BELAS ARTES DE SÃO PAULO**

**GRADUAÇÃO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - EAD**

**FELIPE SCHAITEL - 24112424**

**ARQUITETURA DE SOFTWARE - RELATÓRIO ATV 01**

**FLORIANÓPOLIS, SC**

**2025**

**1 O QUE É ARQUITETURA DE SOFTWARE?**

Arquitetura de Software é um campo de conhecimento que engloba técnicas, conceitos e metodologias. Faz parte da Engenharia de Software que tem o objetivo de estruturar de forma eficiente *softwares*. Robert C. Martin no livro Arquitetura Limpa define “O objetivo da arquitetura de software é minimizar os recursos humanos necessários para construir e manter um determinado sistema” (Martin, p.48).

A arquitetura engloba: protocolos de comunicação, seleção de alternativas de projeto, sincronização, alocação de funcionalidade a componentes, organização e estrutura geral de controle. Segundo Garlan e Shaw, no livro *Perspectives on an Emerging Discipline*, a arquitetura de Software define

– a arquitetura define o que é o sistema em termos de componentes computacionais e, os relacionamentos entre estes componentes, os padrões que guiam a sua composição e restrições. Além da escolha dos algoritmos e estruturas de dados, a arquitetura envolve: decisões sobre as estruturas que formarão o sistema, controle, protocolos de comunicação, sincronização e acesso a dados, atribuição de funcionalidade a elementos do sistema, distribuição física dos elementos escalabilidade e desempenho e outros atibutos de qualidade; e seleção de alternativas de projeto.

(GARLAN, SHAW 1996)

Esses fatores estão ligados a atributos da qualidade do Software pois estão diretamente relacionados à organização e estrutura do sistema que consequentemente também impactam no desempenho.

Há diferentes tipos de arquitetura de software como: Em camadas, Client-Servidor, Orientada a serviços (SOA), Microservices, Monolítica etc.

**2 POR QUE DEFINIR UMA ARQUITETURA DE SOFTWARE É IMPORTANTE?**

Esse campo de conhecimento é importante e necessário pois, o software é algo que está sempre mudando seja por correção de erros, novas tecnologias, na reutilização de códigos, acrescentar ou remover funcionalidades etc. *Soft* significa suave enquanto *ware* - produto portanto, o *software* foi criado para ser algo “suave” planejado para mudar facilmente o comportamento das máquinas. A arquitetura garante a estrutura, consistência, qualidade, mutabilidade e manutenção do *Software*. Martin em Arquitetura limpa afirma que

Quando o software é feito da maneira certa, ele exige só uma fração dos recursos humanos para ser criado e mantido. As mudanças são simples e rápidas. Os poucos defeitos surgem distantes uns dos outros. O esforço é minimizado enquanto a funcionalidade e a flexibilidade são maximizadas.

(MARTIN, p.44)

O livro aborda também que os elementos básicos do código, como a estrutura ser formada por declarações *if, else, while, boolean* etc, mesmo com as novas tecnologias e paradigmas, continua sendo a mesma forma de codificar.

Segundo Robert C. Martin no livro Arquitetura Limpa, define que esses elementos básicos do código (os constructos de programação) são utilizados há muito tempo. Mesmo com as novas tecnologias e paradigmas, essas estruturas essenciais continuam sendo a mesma lógica e forma de codificar.

Quando observa bem de perto a prática de programar computadores, você percebe que muito pouco mudou em 50 anos. As linguagens ficaram um pouco melhores e as ferramentas, fantasticamente melhores. Mas os blocos de construção básicos de um programa de computador não mudaram. (...)

Criei muitas aplicações. Construí muitos sistemas. E a dedicação com que trabalhei em todos esses projetos me ensinou algo surpreendente.Isso é surpreendente: *As regras da arquitetura são sempre as mesmas!* Isso porque os sistemas que criei eram radicalmente diferentes entre si. Então, por que sistemas tão diferentes compartilham regras similares de arquitetura? Cheguei à conclusão de que as regras da arquitetura de software são independentes de todas as outras variáveis.

(MARTIN, p.36)

O autor afirma que ele usa esses constructos da programação desde 1964, em diversos tipos de sistemas e essa lógica é assim até os dias atuais aplicada em diferentes tipos de sistemas, linguagens e paradigmas

Esses são uns dos motivos pelo qual a arquitetura de Software é tão consistente e importante na criação e interoperabilidade de um Software.

**3 QUAIS SÃO AS CARACTERÍSTICAS DO ESTILO ARQUITETURAL “SISTEMAS EM CAMADAS SYSTEMS”**

De acordo com Awari, plataforma de ensino-aprendizagem de tecnologia, O estilo arquitetural de camadas divide o sistema em camadas com funções específicas, cada camada é responsável por um conjunto de funcionalidades.

As suas características e propriedades são:

1- Flexibilidade

2 - Portabilidade;

3 - Facilita a manutenção e evolução do Software;

4 - Permitir a reutilização do código;

5- Simplicidade;

6 - Escalabilidade.

Estruturar em camadas é importante e tem o objetivo de facilitar a alocação de funcionalidades e componentes. Facilita a manutenção e correção de erros. Aprendemos no módulo passado na matéria de Design de Software que codificar em camadas é uma boa prática de programação pois facilita também o trabalho em equipe.

**4 REFERÊNCIAS BIBLIOGRÁFICAS**

**MARTIN, Robert C.** Arquitetura limpa: o guia do artesão para estrutura e design de software. Tradução de Alexandre Huchel. São Paulo: Starlin Alta Editora e Consultoria Eireli, 2020. ISBN 978-85-508-1600-5. Disponível em:<https://github.com/KAYOKG/BibliotecaDev/blob/main/.github/img/Arquitetura%20Limpa%20-%20O%20Guia%20do%20Artes%C3%A3o%20para%20Estrutura%20e%20Design%20de%20Software.svg>. Acesso em: [20/02/2025].

M. Shaw, D. Garlan; Software Architecture. Perspectives on an Emerging Discipline, Prentice Hall, 1996.

**DEVMEDIA**. Arquitetura de software: desenvolvimento orientado para arquitetura. Disponível em:<https://www.devmedia.com.br/arquitetura-de-software-desenvolvimento-orientado-para-arquitetura/8033>. Acesso em: 19 fev. 2025.

**ALURA**. Padrões arquiteturais: a arquitetura de software descomplicada. Disponível em:<https://www.alura.com.br/artigos/padroes-arquiteturais-arquitetura-software-descomplicada>. Acesso em: 19 fev. 2025.

**FERREIRA, Andrey**. Introdução à arquitetura de software. Disponível em:<https://www.inf.ufpr.br/andrey/ci163/IntroduzArquiteturaAl.pdf>. Acesso em: 19 fev. 2025.

**AWARI**. Tipos de arquitetura de software: conheça os mais usados. Disponível em:<https://awari.com.br/tipos-de-arquitetura-de-software-conheca-os-mais-usados/>. Acesso em: 19 fev. 2025.