**Pweb – Atividade 2**

**Arquitetura de Software**

****

**Sorocaba – 2025**

**Introdução**

Talvez você já tenha escutado a frase ou mesmo pensado: “Tenho uma ideia para um aplicativo”. Tudo soa como algo muito fácil de realizar à primeira vista, porém, durante o estudo e a prática em desenvolvimento de software, percebemos que alguns processos são indispensáveis antes de escrever a primeira linha de código.

Desse modo é essencial conhecer os padrões arquiteturais e também conhecer o que é arquitetura de software.

**O que é arquitetura de software?**

As apropriações de nomenclaturas do mundo real são comuns no universo de tecnologia, e não poderia ser diferente com o termo arquitetura. No mundo físico, profissionais de arquitetura constroem projetos e desenhos de casas, prédios, entre outras estruturas, baseando-se em um contexto que envolve diversas variáveis e situações.

Arquitetura de software pode ser entendida como um conjunto de normas, princípios e técnicas para construção de software. De certa forma, assemelha-se à ideia de arquitetura que conhecemos e que delimita os padrões de edificações.

**Qual é a função da arquitetura de software?**

É área da TI é responsável pela análise estratégica dos componentes operacionais que permitem criar soluções viáveis para uma tecnologia. Para isso, leva em conta aspectos como:

* desempenho;
* escalabilidade;
* interoperabilidade;
* compatibilidade;
* performance.

Com a arquitetura de software, é possível entender as diferenças entre as linguagens, sistemas operacionais e ambientes da computação. Ou seja, qualquer componente tecnológico pode ser usado para integrar uma solução arquitetural.

**Padrões em arquitetura de software**

Os padrões de arquitetura de software são peças fundamentais para concretizar toda a abstração, e a escolha adequada ao modelo de negócio é essencial, pois evita retrabalho e alto custo para o cliente e a equipe.

Além disso, podemos trabalhar com padrões de arquitetura diferentes no mesmo software, seja para atender determinados requisitos do negócio, facilitar a manutenção ou aproveitar vantagens de diferentes tecnologias.

Um padrão arquitetural é uma solução já estabelecida para desenvolvimento de softwares, sendo um modelo reutilizável para problemas já conhecidos. Esses padrões já foram estudados, testados em projetos reais e passaram por melhorias, ou seja, são soluções difundidas e aceitas no mercado. Confira algumas vantagens no uso de padrões arquiteturais:

* Maior flexibilidade e escalabilidade de software
* Facilidade de manutenção e evolução
* Segurança
* Melhor desempenho das aplicações
* Redução de custos e riscos

Além de todos esses benefícios, a utilização de padrões de arquitetura facilita o diálogo entre equipes e pessoas por se tratar de soluções conhecidas no mercado. Pensando nisso, vamos conhecer alguns dos principais padrões de arquitetura de software?

**Arquitetura client-server (cliente-servidor)**

Segundo Martin Fowler, no livro Padrões de arquitetura de aplicações corporativas, os sistemas eram separados duas camadas:

**Cliente:** responsável por manter a interface com o usuário e um ou outro código da aplicação. Por exemplo, Delphi e VB disponibilizavam componentes visuais para trabalhar com banco de dados.

**Servidor:** normalmente um banco de dados relacional, como o SQL.

Esse modelo funcionava bem para realizar atualizações mais simples em dados e se precisasse apenas exibir as informações. Porém, se houvesse a necessidade de implementação de lógica de domínio mais complexa, com validações, regras de negócio, verificações e até cálculos a manipulação do código, ficava mais difícil. Esse problema acontecia em muitos casos porque as pessoas desenvolvedoras escreviam a lógica no cliente, favorecendo a duplicação de código e outros problemas, como a criação de muitas telas para alterações simples em processos de buscas.

Outra possibilidade, porém menos usual, era manter a lógica de domínio embutida no banco de dados, no formato de stored procedures (em tradução livre, procedimentos armazenados). No entanto, o código permanecia limitado e engessado.

**Arquitetura em camadas (Layers)**

Em uma organização do sistema em camadas, cada uma delas tem uma responsabilidade e funcionalidade específicas. Porém esse modelo é diferente do padrão MVC, pois é possível modificar suas camadas de forma independente, por exemplo, podemos adicionar ou alterar uma View existente sem fazer alterações nos dados contidos na Model.

Por outro lado, no padrão de arquitetura em camadas, as funcionalidades de uma camada dependem dos recursos e serviços disponibilizados pela camada abaixo dela. São os níveis que fornecem serviços. Vamos entender com um exemplo de uma arquitetura em camadas para um software genérico a partir do diagrama a seguir:

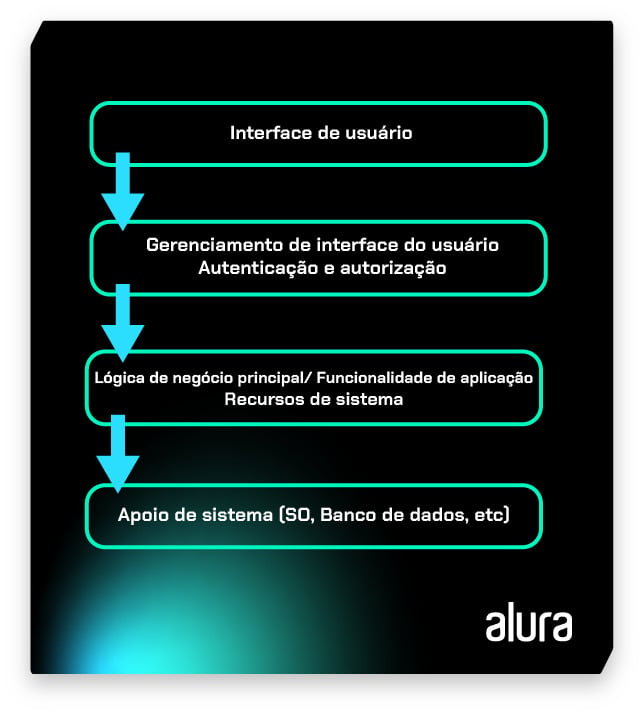


Figura 1 - Arquiteturas em camadas | Disponível em: <https://www.alura.com.br/artigos/padroes-arquiteturais-arquitetura-software-descomplicada?srsltid=AfmBOoqJNz9AROGdzRZfs8e-eFnpwDqXFl8FpQbzsHTyaG1Xze7qi6In>

Na representação, a camada de interface do usuário, que está no topo, é a de apresentação e está ligada à camada abaixo, que por sua vez gerencia o processo de autenticação e autorização entre o usuário e o sistema.

Abaixo da camada de gerenciamento está a que abriga a lógica de negócio principal e os recursos do sistema, e a camada inferior é responsável pelo apoio ao sistema, que pode ser representada pelo armazenamento de dados, sistema operacional, banco de dados, dentre outros.

Dessa forma, temos uma representação que segue de um nível mais alto até o mais baixo, não em grau de importância, mas em relação à arquitetura de um sistema. A quantidade de camadas não é um número ou um modelo fechado, pois podem ser podem ser divididas em mais camadas caso exista a necessidade.

**Arquitetura Model-View-Controller (MVC)**

Talvez o padrão MVC seja um dos primeiros a ser conhecido no aprendizado em desenvolvimento web, pois o padrão é bastante flexível e tem uma alta escalabilidade e reusabilidade.

A arquitetura MVC faz a separação da apresentação e a interação dos dados do sistema, que é estruturado em três componentes lógicos: model (modelo), view (visualização, visão ou vista) e controller (controlador). Esses componentes interagem entre si da seguinte maneira:

* Model é responsável por estabelecer as regras de negócio, interagir com o sistema de dados e fazer as operações associadas a esses dados.
* View define e gerencia como os dados são apresentados ao usuário.
* Controller é a camada intermediária entre model e view, interage com o usuário (por meio de teclas, cliques do mouse, requisições etc.) e é responsável por responder de acordo.

O padrão MVC pode ser representado pelo diagrama a seguir:

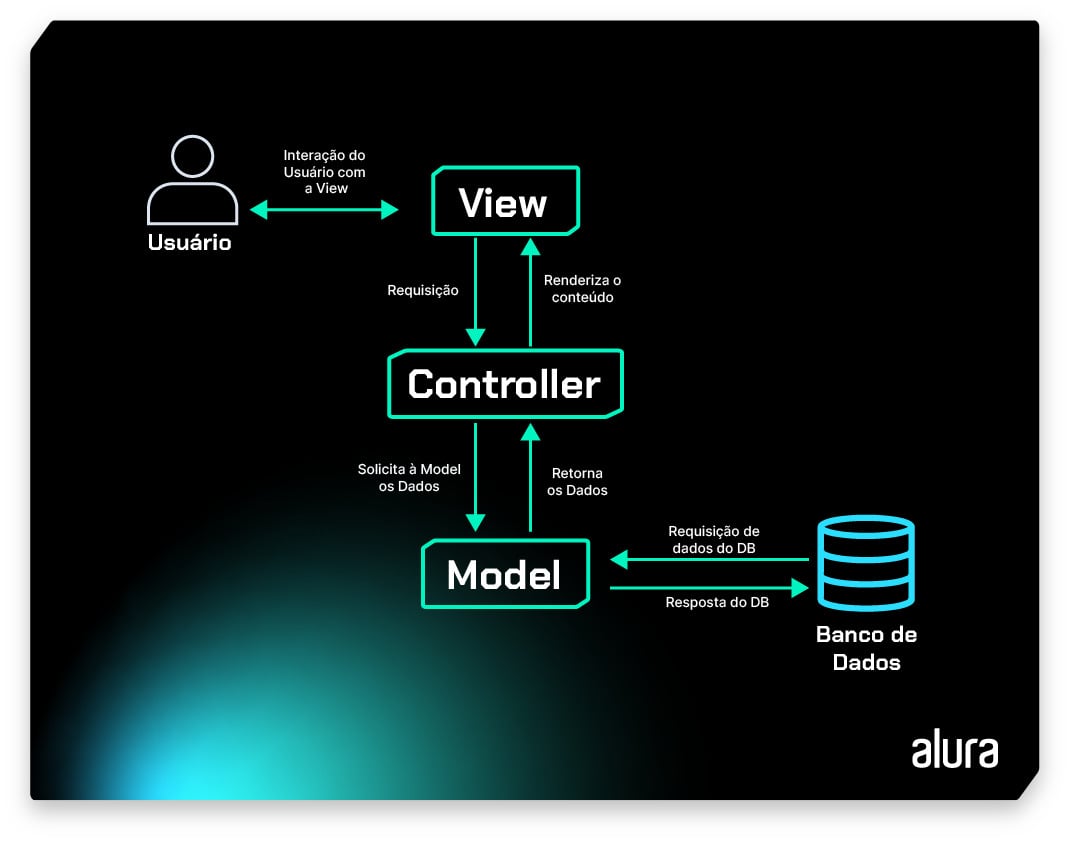


Figura 2 - Arquitetura MVC | Disponível em: <https://www.alura.com.br/artigos/padroes-arquiteturais-arquitetura-software-descomplicada?srsltid=AfmBOoqJNz9AROGdzRZfs8e-eFnpwDqXFl8FpQbzsHTyaG1Xze7qi6In>

**Conclusão**

A arquitetura de software é um pilar fundamental no desenvolvimento de sistemas, pois define a estrutura, a organização e os padrões que orientarão todo o ciclo de vida da aplicação. Conhecer e aplicar corretamente os conceitos e padrões arquiteturais, como Cliente-Servidor, Arquitetura em Camadas e MVC, permite criar soluções mais escaláveis, seguras, de fácil manutenção e alinhadas às necessidades do negócio. Além de otimizar recursos e reduzir riscos, a arquitetura proporciona uma linguagem comum entre equipes, facilitando a comunicação e a colaboração. Portanto, compreender esses fundamentos não apenas contribui para a qualidade técnica do software, mas também garante que ele seja sustentável e adaptável frente às constantes evoluções tecnológicas.

**Referências:**

Alura - Padrões arquiteturais: arquitetura de software descomplicada, disponível em: [https://www.alura.com.br/artigos/padroes-arquiteturais-arquitetura-software-descomplicada?srsltid=AfmBOoqJNz9AROGdzRZfs8e-eFnpwDqXFl8FpQbzsHTyaG1Xze7qi6In](https://www.alura.com.br/artigos/padroes-arquiteturais-arquitetura-software-descomplicada?srsltid=AfmBOoqJNz9AROGdzRZfs8e-eFnpwDqXFl8FpQbzsHTyaG1Xze7qi6In%20)

Faculdade XPE - O que é arquitetura de software? Veja definição e aplicação no mundo corporativo, disponível em: [https://blog.xpeducacao.com.br/o-que-e-arquitetura-de-software/](https://blog.xpeducacao.com.br/o-que-e-arquitetura-de-software/%20)