Projeto de Arquitetura de software

—

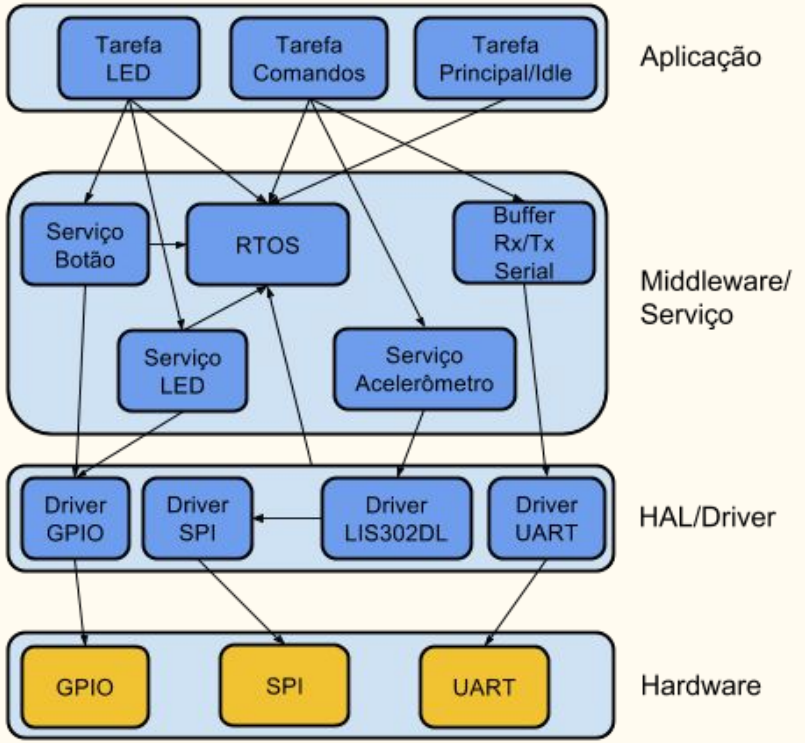
João Vitor Espig e Marco Antonio Samuelsson

Roteiro

- Introdução
- Princípios SOLID
- Padrões de arquitetura
- Considerações Finais

Introdução: O que é?

- Representa o blueprint (plano) de um sistema;
- Organiza e comunica elementos essenciais na construção do software;
- Planos arquiteturais devem ser detalhados para evitar surpresas;
- Arquitetura confiável aumenta as chances de sucesso e facilita correções;
- Considera integridade, restrições econômicas e preocupações estéticas;



Introdução: O que é?

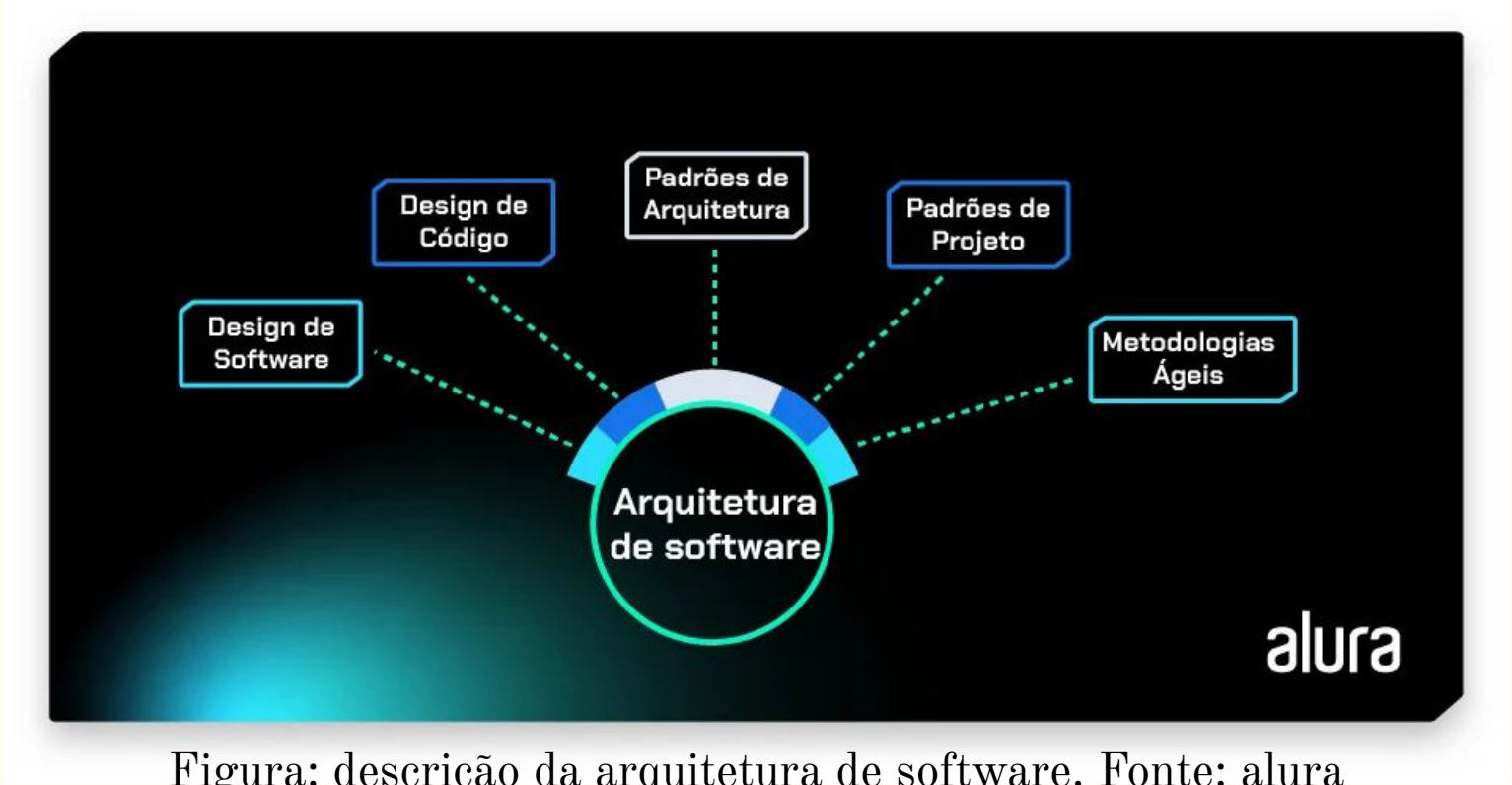
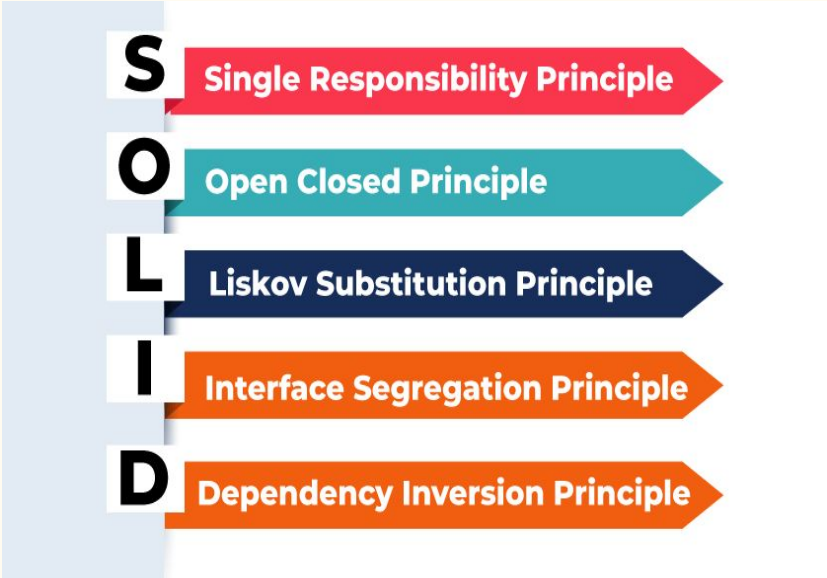


Figura: descrição da arquitetura de software. Fonte: alura

Princípios SOLID

- Origem em 1995, no artigo “The principles of OoD” de Robert C Martin
- Fundamental para POO
- Tem o propósito de melhorar a qualidade do software



Fonte: Medium

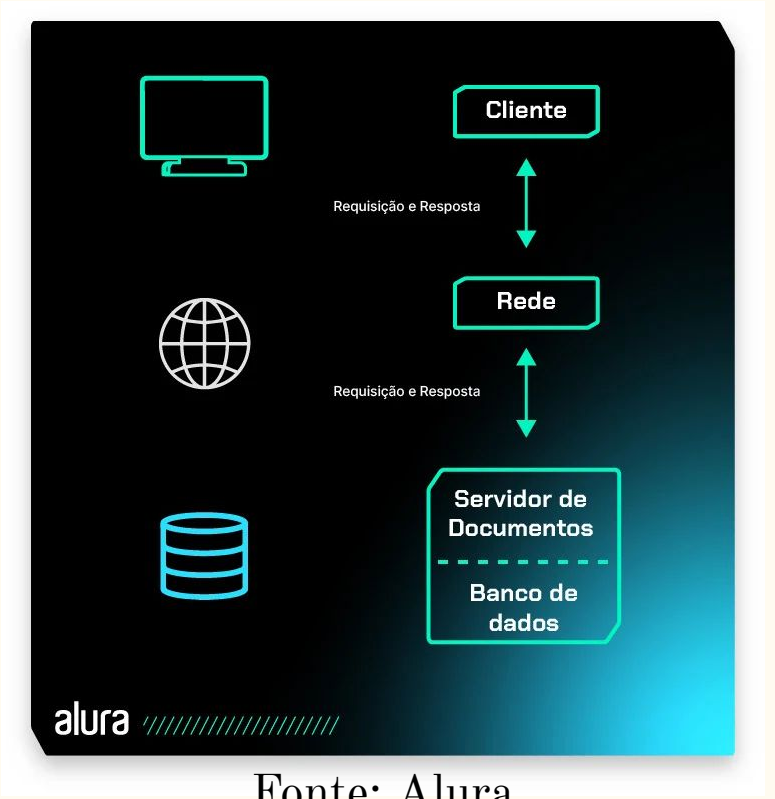
Princípios SOLID

- Responsabilidade Única
  - Uma atribuição por entidade
- Aberto-fechado
  - Ser expansível e independente
- Substituição de Liskov
  - Sistemas independentes (princípio aberto-fechado) devem conseguir substituir outros serviços ou comunicarem com eles
- Segregação de Interface
  - Serviços divididos em pequenas partes
- Inversão de Dependência
  - Módulos de nível superior não devem ser dependentes de módulos inferiores

Padrões de arquitetura

Arquitetura Cliente-Servidor

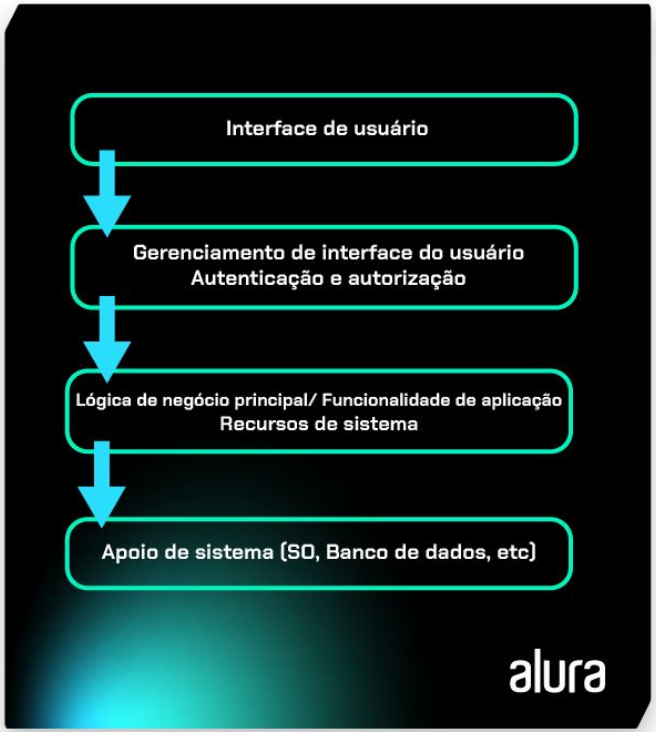
- Surgiu na década de 1990
- Adequado para sistemas simples
- Dificuldades com uma lógica de domínio mais complexa
- Interface com o usuário (Cliente)
- Banco de dados (Servidor)
- **Exemplo**: Serviço de e-mail



Fonte: Alura

Arquitetura em Camadas

- Dividido em camadas de aplicação;
- Cada camada depende da camada abaixo;
- Versão mais independente do padrão MVC;
- Modificação livre das camadas.



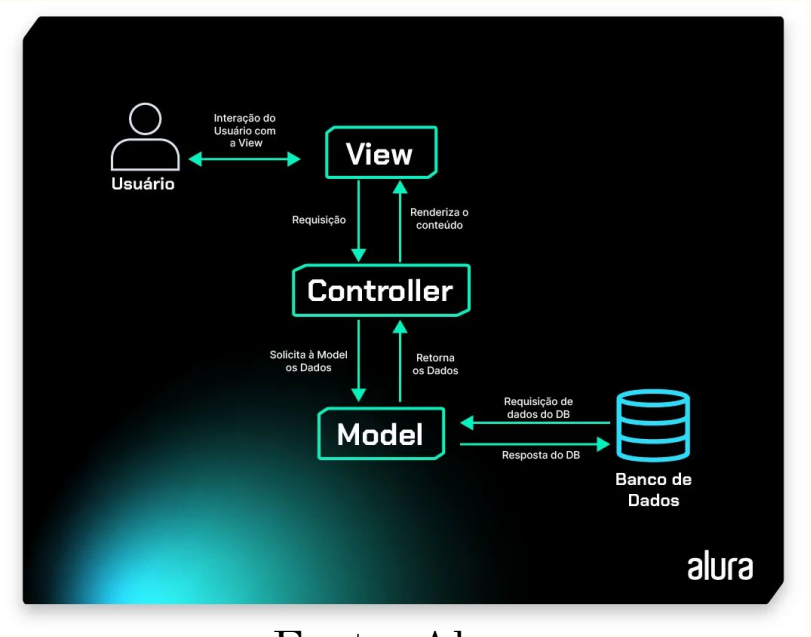
Fonte: Alura

Arquitetura em Camadas

- **Exemplo: Sistemas de Gerenciamento Escolar:**
  - **Camada de Apresentação:** Portais de estudantes, professores e administradores.
  - **Camada de Aplicação:** Processamento de matrículas, alocação de turmas.
  - **Camada de Negócios:** Regras de cálculo de notas e controle de presença.
  - **Camada de Dados:** Bancos de dados contendo registros de alunos, professores e horários.

Arquitetura Model-View-Controller (MVC)

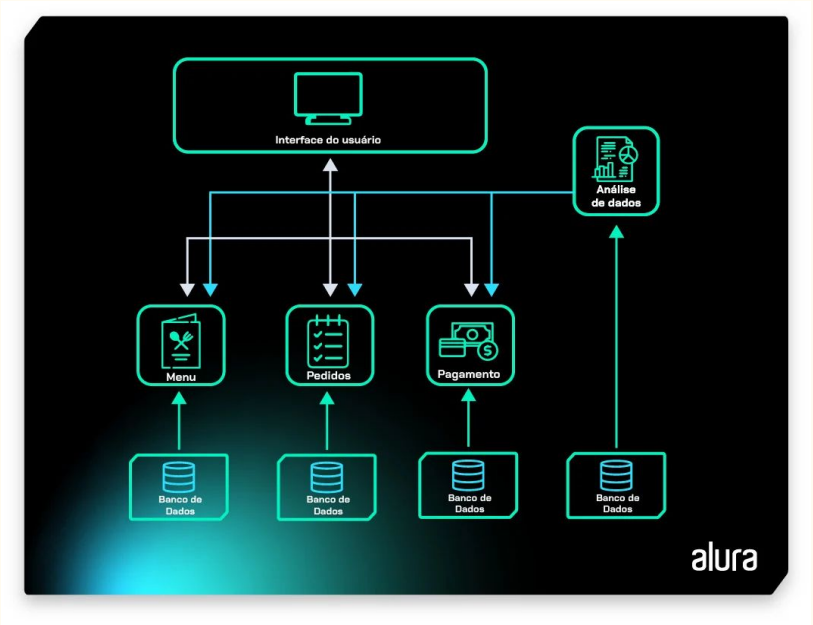
- Lógica de domínio (Model)
- Interface com o usuário (View)
- Camada intermediária que "trata" as ações do usuário (Controller)
- Bastante utilizado em desenvolvimento web
- **Exemplo**: Um sistema de blog, onde o Model contém os posts, a View renderiza a interface com a lista de posts, e o Controller gerencia as interações, como a criação de novos posts.



Fonte: Alura

Arquitetura de Microsserviços

- Sistema grande dividido em pequenos subsistemas;
- Permite uma melhor escalabilidade;
- Permite uso de diferentes tecnologias e linguagens;
- Contrasta a arquitetura monolítica;
- **Exemplo**: Um sistema de e-commerce com micro serviços para carrinho de compras, inventário e pagamentos.



Fonte: Alura

Considerações Finais

- Arquitetura de sistemas está sempre em crescimento e evolução;
- Os princípios SOLID criam sistemas flexíveis e de mais fácil manutenção;
- Fundamental para garantir que um sistema seja eficiente, escalável e adaptável;
- Impacta diretamente a capacidade do sistema de atender às demandas atuais e futuras.
- A utilização de padrões de arquitetura traz benefícios:
  - flexibilidade;
  - escalabilidade;
  - segurança;

Referências

- https://www.monitoratec.com.br/blog/arquitetura-de-software/
- https://www.devmedia.com.br/arquitetura-de-software-desenvolvimento-orientado-para-arquitetura/8033
- https://www.alura.com.br/artigos/padroes-arquiteturais-arquitetura-software-descomplicada
- https://www.alura.com.br/artigos/solid#principio-da-inversao-de-dependencia-(d---dependency-inversion-principle)