

Padrão de Arquitetura em Camadas

Grupo 1

David Samuel Tavares de Moraes
Jorge Abrão Neto
Lucas Mendes da Silva
Paulo Victor Rocha de Almeida

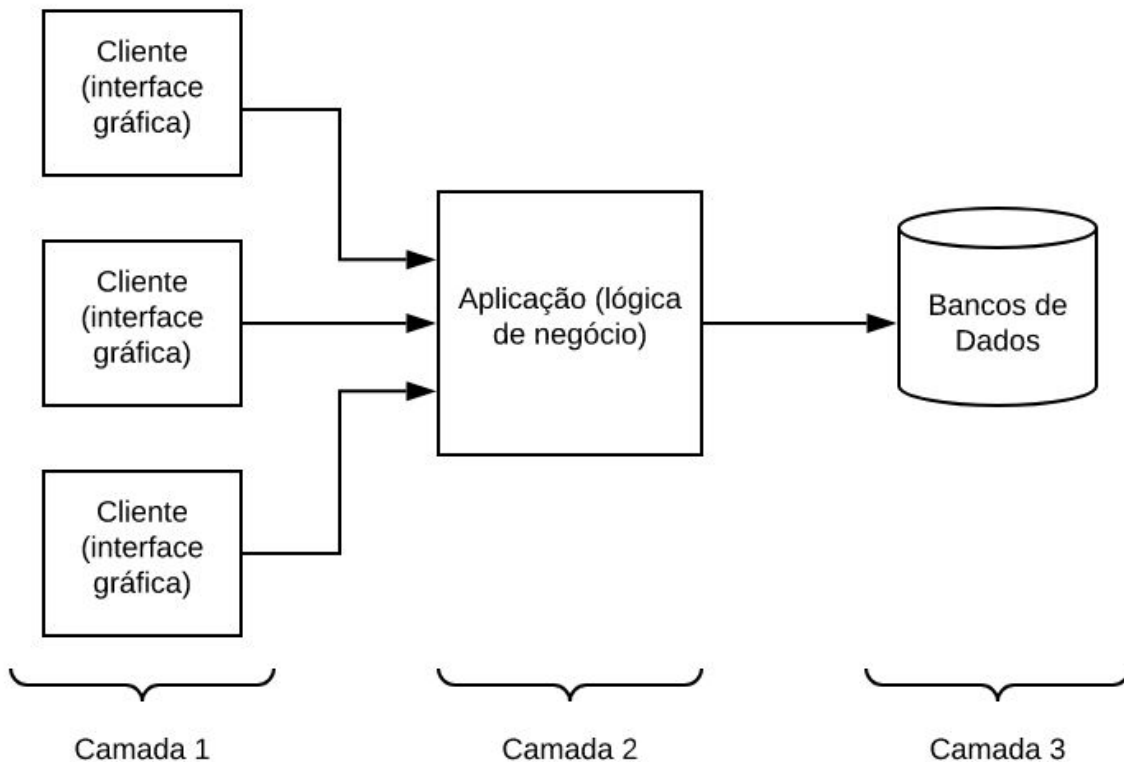
O que é?

Padrão de arquitetura em camadas



- Estilo arquitetural comumente usado no desenvolvimento de software.
- Permite a separação e organização de diferentes componentes e funcionalidades de um sistema em camadas distintas, onde cada camada tem responsabilidades específicas.
- Geralmente é composto por três camadas principais: apresentação (ou interface do usuário), lógica de negócio (ou processamento) e persistência (ou armazenamento). Cada camada tem um propósito e função claramente definidos.

Padrão de arquitetura em camadas

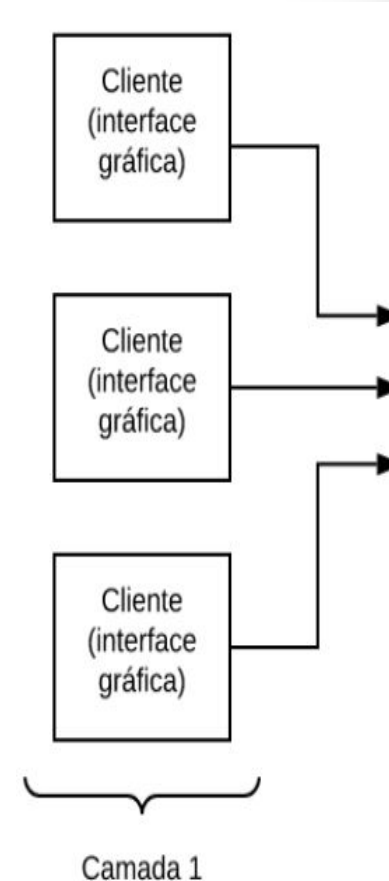


As camadas



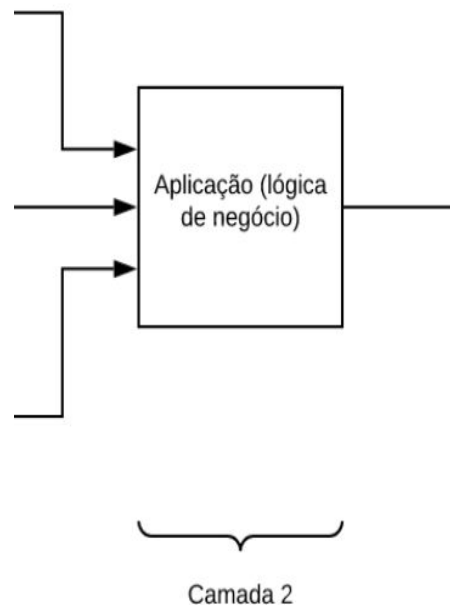
Camada de Interface do usuário

- Também chamada de '*Camada de Apresentação*', é responsável por lidar com a interação direta com o usuário ou outros sistemas externos.
- Pode incluir interfaces gráficas, APIs ou qualquer outra forma de interação.
- Objetivo: receber os inputs do usuário e exibir os resultados ou solicitar ações.
- Exemplo: Em desenvolvimento web, seria um website (HTML, CSS, JavaScript)



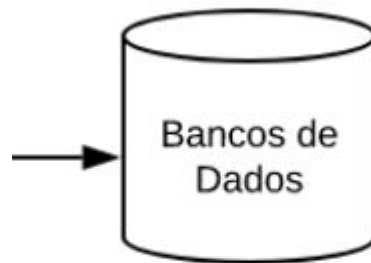
Camada de Lógica de Negócio

- Também chamada de '*Camada do aplicativo*', é responsável por implementar as regras de negócio e a lógica do sistema. Aqui, ocorrem as manipulações dos dados recebidos da camada de apresentação.
- É nesta camada que as operações e transformações são executadas para processar os dados e produzir os resultados desejados.
- A camada do aplicativo é geralmente desenvolvida usando Python, Java, Perl, PHP ou Ruby e se comunica com a camada de dados usando chamadas de API.



Camada de Armazenamento

- Também chamada de '*Camada de Dados*', é responsável pelo armazenamento e gerenciamento dos dados do sistema em um banco de dados ou qualquer outro mecanismo de armazenamento.
- Este pode ser um sistema de gerenciamento de banco de dados relacional, como MySQL, Oracle, Microsoft SQL Server ou em um servidor de banco de dados NoSQL, como CouchDB ou MongoDB.
- A camada de apresentação e a camada de dados não podem se comunicar diretamente entre si.



Camada 3



Vantagens e Desvantagens

Vantagens:



- **Separação de Responsabilidades:** Cada camada é responsável por uma preocupação específica da aplicação. Essa separação clara de responsabilidades facilita a compreensão do sistema como um todo e permite que diferentes equipes ou desenvolvedores trabalhem em paralelo em cada camada, aumentando a eficiência do desenvolvimento.
- **Manutenção facilitada:** A separação clara de responsabilidades e a modularidade da arquitetura em camadas tornam a manutenção do sistema mais fácil. Alterações em uma camada específica têm um impacto limitado nas outras camadas, desde que a interface entre elas seja mantida consistente.
- **Escalabilidade:** É possível adicionar mais instâncias de uma determinada camada ou substituir uma camada por uma implementação mais robusta conforme a demanda cresce.
- **Reutilização de código:** Com a modularidade proporcionada pelo padrão em camadas, é possível reutilizar componentes ou camadas em diferentes partes da aplicação ou até mesmo em projetos diferentes.



Desvantagens:

- **Overhead de comunicação:** Com a separação das camadas, é necessário algum mecanismo de comunicação entre elas, como APIs, interfaces ou eventos. Essa comunicação adiciona uma sobrecarga de processamento e latência ao sistema, especialmente quando as camadas estão em diferentes máquinas físicas ou virtuais. O aumento do tráfego de rede e o processamento adicional para serializar/deserializar dados podem afetar o desempenho do sistema.
- **Rigidez na evolução:** O padrão em camadas pode tornar o sistema mais rígido em termos de evolução devido a seu grande grau de acoplamento.
- **Complexidade:** À medida que um sistema cresce em tamanho e complexidade, a arquitetura em camadas pode se tornar mais difícil de ser gerenciada e compreendida.

Conclusão

Conclusão

- O *Padrão de Arquitetura em Camadas* facilita a manutenção, a escalabilidade e a evolução do sistema, pois a separação em camadas permite que cada uma delas seja desenvolvida, testada e mantida independentemente das outras.
- O *Padrão em Camadas* também promove a reutilização de código e facilita a compreensão da estrutura do sistema, tornando-o mais modular e flexível.
- Vale ressaltar que a arquitetura em camadas não se limita apenas a três camadas. Dependendo da complexidade do sistema, podem ser adicionadas camadas intermediárias ou outras camadas específicas para atender às necessidades do projeto.



Referências Bibliográficas

- [iMasters - Arquitetura em Camadas](#)
- [EngSoftModerna Cap. 7: Arquitetura - Engenharia de Software Moderna](#)



Obrigado !