

# Arquitetura de Software em Camadas

Arquitetura em 3 camadas

# Arquitetura em Três Camadas

## Camada 1

### **Interface com o Usuário**

- conhecida também como camada de apresentação
- responsável pela interação com o usuário
- trata tanto da exibição de informação, como da coleta e processamento de entradas e eventos de interfaces
- Apresentação e interação podem estar implementadas em uma interface gráfica de uma aplicação desktop ou na interação via browser de uma aplicação web

# Arquitetura em Três Camadas

Camada 2

**Lógica de Negócio;**

**camada de aplicação;**

**ou núcleo do sistema**

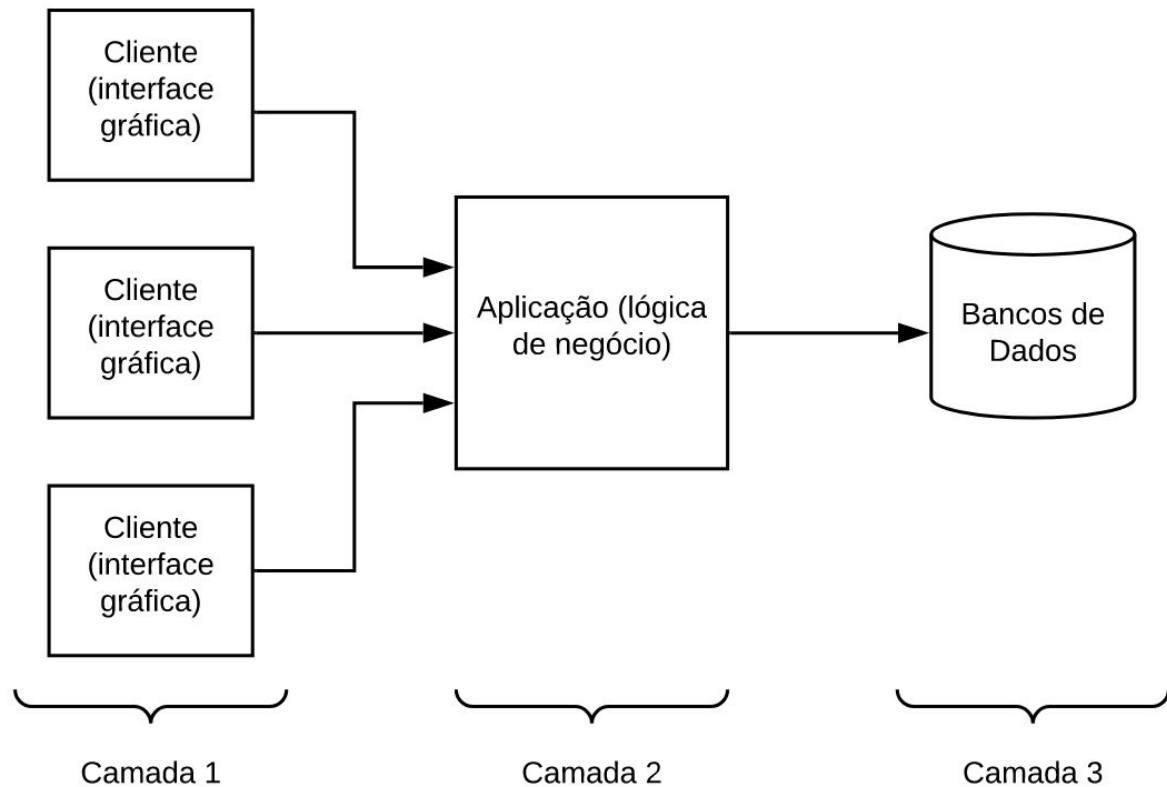
- Implementa as regras de negócio do sistema
- Na implementação, as classes de negócio ou que representam as entidades centrais da aplicação estarão nesta camada

# Arquitetura em Três Camadas

## Camada 3

### **Banco de Dados ou camada de “Persistência”**

- Classes implementadas nesta camada promovem a persistência de dados, ou seja, o armazenamento permanente das informações críticas para o funcionamento do sistema
- Persistência pode ser promovida de diferentes formas, mas, geralmente, será implementada via comunicação com uma base de dados
- Somente as classes nessa camada devem realizar operações sobre a base de dados
- Um padrão bastante utilizado para implementação da persistência é o Data Access Object (DAO) ou Objeto de Acesso aos Dados



A comunicação deve acontecer entre camadas adjacentes!

# Vantagens

- Cada uma das camadas pode ser organizada em um pacote e promove a separação explícita das responsabilidades das classes que compõem a aplicação
- As camadas isolam dão um propósito para as classes dentro delas, dessa forma as classes e podem ser (ou a camada) pode ser reutilizável ou até mesmo substituível
- Palavras chave:
  - Modularidade (manutenção, o reuso de código e a escalabilidade do sistema)
  - Separação de preocupações (compreensão e a modificação de partes isoladas do sistema, sem afetar as demais)
  - Flexibilidade (mudanças são limitadas à camada em que as alterações estão sendo feitas; reduz o impacto de mudanças futuras)
  - Testabilidade (é possível testar cada componente de forma individual; testes automatizados)