

# ARQUITETURA DE SISTEMAS

## Entrega 6.0

### Análise e Pesquisa

O projeto trata-se de um sistema de **Flashcards** para estudo, implementado em Java. O sistema deve:

- Permitir a **criação e gerenciamento de decks**.
- **Adicionar e visualizar flashcards** dentro dos decks.
- Notificar usuários sobre novas cartas adicionadas.
- **Facilitar a manutenção e escalabilidade**, evitando código desorganizado.

Para atender a esses requisitos, foi necessário escolher uma **arquitetura adequada** que garantisse separação de responsabilidades e flexibilidade para futuras expansões.

### Escolha da Arquitetura

Optamos por utilizar a **Arquitetura em Camadas (MVC - Model, View, Controller)**. Isso porque:

- **Separação de responsabilidades**: facilita a manutenção e modularidade.
- **Facilidade para escalar**: novas funcionalidades podem ser adicionadas sem modificar todo o código.
- **Código mais organizado**: melhora a leitura e evita acoplamento entre as partes do sistema.

#### Componentes da Arquitetura:

- **Model (Modelo)**: Representa os dados do sistema (Flashcards, Decks, Banco de Dados).
- **View (Visão)**: Interface com o usuário (interação pelo terminal).
- **Controller (Controle)**: Gerencia as regras de negócio e comunicação entre Model e View.

### Desenvolvimento do Protótipo

A implementação do projeto foi feita conforme a arquitetura escolhida:

#### Modelo (Model)

- **Classe FlashCard**: Representa os flashcards, armazenando pergunta, resposta e nível de dificuldade.

- **Classe Deck:** Gerencia uma coleção de flashcards.
- **Classe DatabaseConnection** (*Singleton*): Simula a conexão com um banco de dados.

### Controle (Controller)

- **Classe Flashcard:**
  - Cria e gerencia decks.
  - Adiciona flashcards usando FlashcardFactory (*Factory Method*).
  - Notifica usuários via SistemaNotificacao (*Observer*).

### Visão (View)

- **Classe Main:**
  - Exibe opções ao usuário.
  - Captura entrada de dados e chama métodos do FlashcardController.