

## 1. DAO (Data Access Object)

**Função:** Acessar e manipular os dados no banco de dados.

**Responsabilidades:**

- Fazer consultas (SELECT), inserções (INSERT), atualizações (UPDATE) e exclusões (DELETE).
- Isolar a lógica de acesso ao banco de dados do resto da aplicação.
- Utiliza geralmente JDBC, JPA ou Hibernate.

**Exemplo:** `UsuarioDAO` pode ter métodos como `buscarPorId()`, `salvar()`, `atualizar()`, `remover()`.

---

## 2. Service

**Função:** Implementar a lógica de negócio da aplicação.

**Responsabilidades:**

- Regras de negócio e validações.
- Chamar métodos do DAO.
- Preparar dados antes de salvar ou retornar.
- É como o “cérebro” da aplicação.

**Exemplo:** `UsuarioService` pode ter métodos como `registrarNovoUsuario()`, `validarLogin()`, `alterarSenha()`.

---

## 3. App (ou Controller, ou Main)

**Função:** Ponto de entrada da aplicação, onde tudo começa a rodar.

**Responsabilidades:**

- No caso de uma aplicação de linha de comando, aqui está o `main()`.

- Em aplicações web, é onde ficam os Controllers que recebem requisições do usuário.
- Chama o Service para realizar ações e devolver uma resposta ao usuário.

**Exemplo:**

- Em app console: `public static void main(String[] args)`
  - Em web: `@RestController` com métodos que recebem requisições HTTP.
- 

## 4. Model (ou Entity)

**Função:** Representar os dados e objetos do mundo real.

**Responsabilidades:**

- Estrutura de dados (atributos + getters/setters).
- Pode ser anotada com JPA (`@Entity`, `@Table`, etc.) para mapeamento com o banco.
- São as "peças" que DAO, Service e App manipulam.

**Exemplo:** `Usuario.java` com atributos como `id`, `nome`, `email`, `senha`.