

# Atividade Prática: Implementação de Funcionalidade em Arquitetura de Camadas

**Disciplina:** Engenharia de Software Avançada

**Tema:** Arquitetura de Software em Camadas

**Objetivo:** Aplicar os conceitos de arquitetura de software em camadas para implementar uma funcionalidade de um aplicativo web, separando as responsabilidades de visualização, controle, lógica de negócio e persistência de dados. A atividade deve ser concluída em 60 a 90 minutos.

## Problema Proposto:

Você deve construir um pequeno aplicativo web para **gerenciar uma lista de tarefas (To-Do List)**. A funcionalidade principal a ser implementada é a **adição de uma nova tarefa**. A solução deve seguir rigorosamente uma arquitetura em quatro camadas:

1. **View (Visualização):** Responsável pela interface do usuário. Deve conter um formulário para inserir a descrição da nova tarefa e exibir uma mensagem de sucesso ou erro após a submissão.
2. **Control (Controlador):** Responsável por receber a requisição do usuário, orquestrar a lógica de negócio e retornar a resposta adequada para a camada de visualização.
3. **Model (Modelo):** Representa a entidade de negócio e a lógica de negócio associada. A classe principal `Tarefa` e a lógica para adicionar uma tarefa devem estar nesta camada.
4. **Services (Serviços):** Responsável exclusivamente pela persistência de dados. A camada de `Services` deve conter a lógica para escrever as tarefas em um arquivo de texto simples (`tarefas.txt`).

## Instruções para a Atividade:

1. Estruture seu projeto com diretórios e arquivos que reflitam claramente a arquitetura de 4 camadas: `view/`, `control/`, `model/`, `services/`.
2. Crie a classe `Tarefa` na camada `model`.
3. Na camada `model`, crie uma classe de modelo de negócio, por exemplo, `GerenciadorDeTarefas`, que conterá a lógica para adicionar uma tarefa.
4. Na camada `services`, crie uma classe de serviço, por exemplo, `PersistenciaArquivo`, que conterá a lógica para salvar os dados no arquivo `tarefas.txt`.

5. Crie um controlador na camada `control` que fará a orquestração entre a `View` e a `Model`. Ele deve invocar o método da `Model` para adicionar a tarefa e, por sua vez, a `Model` deve invocar o serviço de persistência.
6. Crie uma interface na camada `view` com um formulário simples para adicionar a tarefa.
7. O aplicativo pode ser implementado em qualquer linguagem de programação e framework para web.