

# Especificação do Trabalho Final – Disciplina:

## Aplicações para Internet

### Objetivo

Desenvolver uma aplicação RESTful completa, utilizando os conceitos e padrões de design estudados ao longo da disciplina, com foco em boas práticas de desenvolvimento, arquitetura limpa e exposição de serviços via HTTP. O trabalho tem como objetivo consolidar os conhecimentos em arquitetura de software, APIs RESTful, padrões de projeto e consumo de serviços web.

### Requisitos Técnicos

#### 1. Estrutura e Arquitetura

A aplicação deverá ser construída utilizando padrões de design e arquitetural. A seguir temos os padrões estudados, aplique os que mais se adequa ao seu projeto:

- Arquitetura limpa (**Clean Architecture**)
- Aplicação dos princípios **SOLID**
- Aplicação de **padrões de design**:
  - Service
  - Singleton
  - Abstract
  - DAO (Data Access Object)
- Utilização adequada de **injeção de dependência** (quando aplicável)

#### 2. API RESTful

- Os recursos devem ser expostos como uma **API RESTful**.
- Implementar **endpoints HTTP** usando os métodos adequados: GET, POST, PUT, DELETE.
- Retornar códigos de status HTTP apropriados.

- Tratar erros de forma padronizada (ex: 404, 400, 500).
- Os dados devem ser trocados em formato **JSON**.

### 3. Persistência de Dados

- A aplicação deverá utilizar algum mecanismo de persistência (ex: banco de dados relacional como PostgreSQL/MySQL ou banco NoSQL como MongoDB).
- Utilizar o padrão DAO para acesso aos dados.

### 4. Cliente para Consumo da API

A API deverá ser consumida por **pelo menos um cliente**, podendo ser:

- Uma interface **frontend** (ex: em React, Angular, Vue, etc.)
- Ou uma **documentação interativa** da API via **Swagger/OpenAPI**, com suporte a testes.

## Regras de Formação de Grupos

- O trabalho poderá ser feito em **grupos de até 3 pessoas**.
- Grupos com **4 integrantes** serão permitidos apenas mediante aprovação prévia do professor.
- Não serão aceitos trabalhos individuais.
- A apresentação do trabalho é obrigatória. Trabalhos não apresentados terão nota **ZERO**, independentemente da entrega técnica.

## Entrega e Avaliação

**Prazo de entrega: 16/06/2025**

**Valor total: 20 pontos**

- **12 pontos – Avaliação técnica do projeto**
  - Organização do código e uso de padrões (4 pts)
  - Implementação correta da API REST (3 pts)

- Aplicação dos princípios SOLID e Clean Architecture (3 pts)
- Documentação e/ou client funcional (2 pts)
- **8 pontos – Apresentação**
  - Clareza na explicação do projeto
  - Participação de todos os membros
  - Demonstração funcional da aplicação
  - Justificativa das decisões técnicas e uso dos padrões

## Sugestões de Temas para o Projeto

Os alunos têm liberdade para propor o tema do projeto, desde que se enquadre nos requisitos da especificação. Alguns exemplos incluem:

- **Sistema de gerenciamento de tarefas** (tipo Trello simplificado)
- **Catálogo de produtos e gerenciamento de estoque**
- **Sistema de reservas de salas ou agendamento de horários**
- **Controle financeiro pessoal ou familiar**
- **API de filmes e avaliações** (tipo IMDb simplificado)
- **Plataforma de cadastro e busca de vagas de emprego**
- **Sistema de controle de alunos e notas**
- **Biblioteca online para empréstimo de livros**
- **Sistema de votação ou enquetes online**
- **Gerenciador de projetos e equipes**

## Entrega

- O código-fonte deverá ser enviado via repositório GitHub (ou plataforma equivalente).
- Incluir no repositório um **README.md** com:
  - Descrição do projeto

- Como executar a aplicação (backend e client)
- Tecnologias utilizadas
- Instruções para testar os endpoints