

Trabalho 2 de Engenharia de Software:

Desenvolvimento de Aplicação Web com *Clean Architecture*

Objetivo

Desenvolver uma aplicação web utilizando o framework Flask com a linguagem Python, aplicando os princípios da *Clean Architecture*, os princípios SOLID e testes unitários bem estruturados.

Descrição da Tarefa

Cada grupo deve criar uma aplicação web funcional que resolva um problema realista à sua escolha (por exemplo: sistema de gerenciamento de tarefas, biblioteca de livros, controle de estoque, etc.). A aplicação deve ser construída seguindo os princípios da Clean Architecture, promovendo uma separação clara entre camadas (entidades, casos de uso, interfaces e infraestrutura).

Além disso, o código deve evidenciar a aplicação dos princípios SOLID e conter uma cobertura adequada de testes unitários.

Requisitos Técnicos

Tecnologias obrigatórias:

- Python
- Flask
- Pytest
- Git (com repositório versionado no GitHub)

Critérios obrigatórios de arquitetura e qualidade:

- A aplicação deve ter **ao menos três casos de uso** e **ao menos duas entidades**
- Organização do projeto segundo a Clean Architecture (entidades, use cases, interface adapters, frameworks & drivers)
- Aplicação explícita dos princípios SOLID no design e na implementação das classes e métodos
- Testes unitários implementados para os use cases, testando a logica de negócio de forma isolada da infraestrutura.

Critérios de Avaliação (Total: 10,0 pontos)

Critério	Peso
Aplicação funcional com ao menos três casos de uso e ao menos duas entidades	4,0
Aplicação correta da <i>Clean Architecture</i>	2,0
Aplicação correta dos princípios SOLID	2,0
Implementação de testes unitários para os use cases	1,0
Seguir as convenções de estilo da PEP-8	1,0

Entrega

- Apresentação: 25/06/2025

Exemplos de Temas para a Aplicação

Os alunos podem escolher livremente o tema da aplicação, desde que envolva persistência de dados e lógica de negócio razoável. Exemplos:

- Sistema de Gerenciamento de Tarefas (To-do List)
- Catálogo de Livros com Empréstimo
- Controle de Estoque de Produtos
- Sistema de Registro de Frequência Escolar
- Aplicação de Anotações Pessoais com Tags
- Sistema de Reservas de Salas
- Gerenciador Financeiro Pessoal (entradas e saídas)

Sugestão de Estrutura de Pastas - Clean Architecture em Flask

```
project_name/
|
├── app/
|   ├── routes/
|   │   └── task_routes.py
|   └── main.py
|
├── domain/
|   └── task.py
|
└── use_cases/
```

```
|   └─ task_use_case.py
|
├─ infra/
|   └─ repositories/
|       └─ task_repository_sqlite.py
|   └─ db/
|       └─ database.py
|
├─ tests/
|   └─ domain/
|   └─ use_cases/
|
├─ requirements.txt
└─ README.md
```