

Universidade de Pernambuco
Professor: Augusto César Oliveira
Disciplina: Programação para Web
Aluno(a): _____

data: ____/____/____

Projeto de Programação Web

Observações:

* Este projeto deve ser realizado em **dupla** e apresentado em sala de aula em até **7 minutos** no dia **26/06/2025**.

* Todo o material produzido deve ser enviado pelo **Classroom** até às **23h59** do dia que antecede a data de apresentação.

Objetivo

Desenvolver uma aplicação web completa, com front-end, back-end e banco de dados relacional. O projeto deve demonstrar domínio sobre os conceitos fundamentais da programação web, incluindo estrutura de software, comunicação entre sistemas e persistência de dados.

Requisitos do Projeto

1. Front-End + Integração com Back-End (4 pontos)

- Deve conter **pelo menos 3 páginas distintas**.
- O front-end deve ser capaz de **consumir dados** do back-end via API (GET).
- Deve ser possível **enviar dados** ao back-end via API (POST/PUT).
- Os dados enviados devem ser **armazenados no banco de dados**.
- A aplicação deve apresentar **fluxos completos**, como:
 - Cadastro de uma entidade no front-end → envio à API → gravação no banco → exibição no front-end.

Pontuação:

- 1 ponto – Estrutura com 3 páginas funcionais
- 1 ponto – Consumo correto da API (GET)
- 1 ponto – Envio e persistência de dados (POST/PUT)
- 1 ponto – Fluxo completo funcionando de ponta a ponta

2. Back-End (3 pontos)

- Deve implementar **pelo menos 4 casos de uso reais** (ex: criar usuário, listar produtos, autenticar, editar perfil).
- Deve ser exposta uma **API REST** para comunicação com o front-end.
- Utilizar uma **arquitetura em camadas**, por exemplo:

- Camada de Controle (Controllers)
- Camada de Serviço (Services/Use cases)
- Camada de Acesso a Dados (Repositories)

Pontuação:

- 1 ponto – Implementação correta de 4 casos de uso
- 1 ponto – Organização do código em camadas
- 1 ponto – API funcional e acessível via HTTP

3. Banco de Dados Relacional (3 pontos)

- Deve conter **pelo menos 3 tabelas**.
- Deve haver **pelo menos 1 relacionamento** entre essas tabelas (ex: usuários e pedidos).
- As operações da aplicação devem refletir corretamente no banco de dados (inserir, consultar, atualizar, deletar).

Pontuação:

- 1 ponto – Estrutura correta das tabelas
- 1 ponto – Relacionamento implementado e funcional
- 1 ponto – Integração com o back-end (persistência dos dados)

Critérios Gerais de Avaliação

- O projeto deve ser **original** (não plágio).
- O código deve estar **funcional e organizado**.
- O código do projeto deve estar no **GitHub**, com um **README** contendo:
 - **Principais funcionalidades**
Descreva o que o sistema faz e quais problemas ele resolve.
 - **Características do sistema desenvolvido**
Destaque aspectos técnicos, arquitetura adotada e tecnologias utilizadas.
 - **Instruções de execução**
Informe como instalar, configurar e executar o projeto localmente (pré-requisitos, comandos, etc.).

Entrega

- Enviar o **link do repositório** no GitHub na atividade do **Google Classroom** até a data limite.
- Certifique-se de que o repositório contenha:
 - Código do front-end
 - Código do back-end
 - README completo, com todas as informações exigidas acima.