

# Projeto – Programação de Sistemas para Internet

---

## Tema

**Livre escolha**, desde que atenda aos requisitos técnicos obrigatórios.

## Período de execução

04/07 a 13/08 Entrega final até 13/08.

## Objetivo

Desenvolver uma aplicação web com Flask, usando:

- Rotas
- request, make\_response, redirect, url\_for
- Autenticação de usuários
- Banco de dados SQLite
- Templates Jinja2 com extends e includes
- Versionamento no GitHub
- Documento de requisitos funcionais

## Entregas semanais sugeridas

### Semana 1 (04/07–10/07)

- Criação do repositório no GitHub
- Escolha do tema do sistema
- Documento de Requisitos Funcionais completo no repositório
- Estrutura inicial do projeto (ambiente virtual, app.py, requirements.txt)
- Configuração inicial do banco SQLite (tabela de usuários)

### Semana 2 (11/07–17/07)

- Implementar autenticação:
  - Página de registro
  - Página de login
  - Senha com hash seguro
  - Sessão ou Flask-Login para manter usuário autenticado
- Logout funcional
- Templates com extends/includes para base e navbar

### Semana 3 (18/07–24/07)

- Implementar CRUD do recurso escolhido:

- Criar
  - Listar
  - Editar
  - Excluir
- Restrição de acesso: cada usuário só pode ver/editar/excluir seus dados

## Semana 4 (25/07–31/07)

- Uso refinado de Flask:
  - Uso de request para formulários/dados
  - Uso de redirect e url\_for
  - Uso de make\_response (exemplo: cookies ou headers customizados)
- Tratamento de erros:
  - Páginas personalizadas para 404 e 500

## Semana 5 (01/08–13/08)

- Testes manuais completos
- Estilização básica com CSS ou Bootstrap
- Ajustes finais no código
- README bem elaborado:
  - Instruções para rodar localmente
  - Dependências
  - Screenshots ou gifs opcionais
- Revisão final e entrega no GitHub

## Requisitos técnicos obrigatórios

- Autenticação (registro, login, logout com hash de senha)
- Banco de dados SQLite com tabelas adequadas
- CRUD completo de um recurso
- Uso de request, make\_response, redirect, url\_for
- Templates Jinja2 com extends/includes
- Tratamento de erros (404 e 500 customizados)
- Versionamento no GitHub com histórico de commits e entregas semanais
- README com instruções claras

## Observações

- Tema é livre, mas precisa atender aos requisitos técnicos.
- Trabalhos em grupo devem deixar claro quem fez o quê no README.

- A pontuação pode ser dada de forma parcial (exemplo: autenticação implementada mas sem hash → nota parcial).