

# Planejamento: Plataforma de Controle de Oficinas de Ensino

Este documento descreve o plano de desenvolvimento para uma plataforma web destinada a gerenciar o ciclo de vida das oficinas de ensino do projeto ELLP, envolvendo Professores, Tutores, Alunos, e a emissão de certificados.

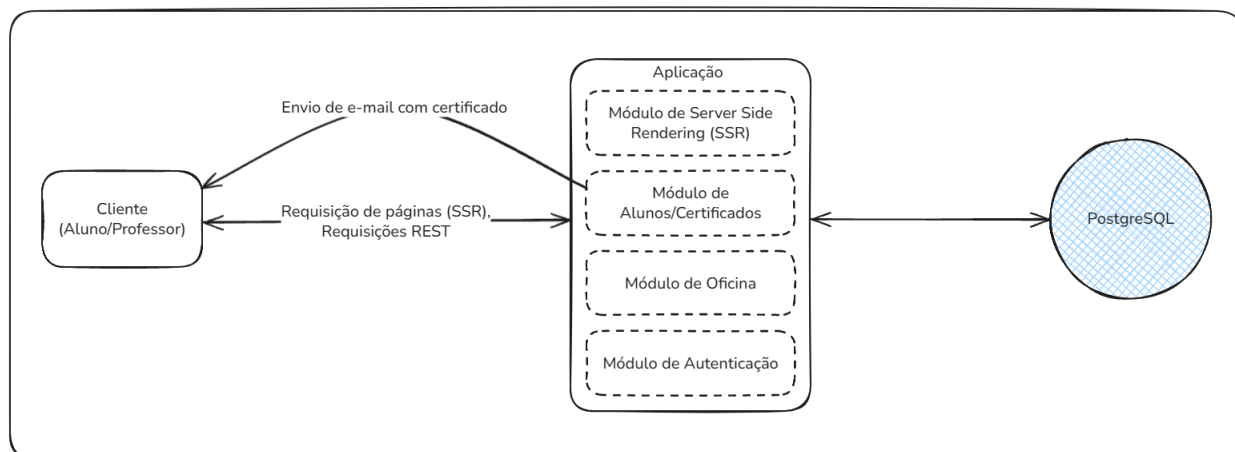
## 1. Definição de Requisitos Funcionais

Os requisitos funcionais definem as capacidades do sistema e são segmentados por perfil de usuário para maior clareza.

Requisito	Título	Prioridade
RF-001	Usuário pode ser um administrador (permissão que faz com que o usuário consiga cadastrar outros usuários e alterar os templates do certificado)	Média
RF-002	Deve poder realizar login e logout de forma segura.	Alta
RF-003	Deve possuir um painel para gerenciar as oficinas que ministra.	Alta
RF-004	Deve poder criar e editar oficinas, definindo tema, descrição, datas, horários, limite de vagas e pré-requisitos.	Alta
RF-005	Deve poder publicar ou despublicar uma oficina do catálogo.	Alta
RF-006	Deve poder visualizar a lista de alunos inscritos em cada oficina.	Alta
RF-007	Deve poder registrar a lista de presença dos alunos em cada aula/encontro.	Alta
RF-008	Deve poder marcar uma oficina como "Concluída" ao final, o que libera a emissão de certificados para os alunos aprovados.	Alta

<b>RF-009</b>	Aluno deve poder verificar sua participação e comprovante da oficina (se houver).	Baixa
<b>RF-010</b>	Assim que concluído, o comprovante deve ser enviado no e-mail do aluno.	Alta

## 2. Definição da Arquitetura em Alto Nível do Sistema



### Componentes:

#### 1. Frontend (Server-side Rendering):

- Para simplificar o processo de desenvolvimento e recursos, foi optado pela criação de uma aplicação server-side rendering, onde o servidor back-end será responsável por gerar páginas front-end e respondê-las em requisições

#### 2. Backend (Servidor):

- **Módulo de Autenticação e Usuários:** Gerencia o cadastro, login e permissões dos usuários.
- **Módulo de Oficinas:** Contém toda a lógica de negócio para criar, gerenciar, listar e se inscrever em oficinas.
- **Módulo de Alunos/Certificados:** Gera os arquivos PDF sob demanda, preenchendo um template com os dados do aluno e da oficina e fornecer dados sobre alunos.
- **Módulo de Server-Side Rendering:** responsável por gerenciar todo o front-end.

#### 3. Banco de Dados:

- **Banco de Dados Relacional (PostgreSQL):** Será o principal banco para armazenar dados estruturados como usuários, oficinas, inscrições e presenças, garantindo consistência e integridade referencial.

### 3. Definição da Estratégia de Automação de Testes

Adotaremos a abordagem da Pirâmide de Testes para garantir a qualidade do software de forma eficiente.

#### 1. Testes de Unidade:

- **Objetivo:** Validar as menores partes (métodos/funções) do código de forma isolada.
- **Ferramentas:** Jest com React Testing Library para o frontend e Pytest para o backend.

#### 2. Testes de Integração:

- **Objetivo:** Garantir que diferentes módulos do sistema se comunicam corretamente.
- **Ferramentas:** React Testing Library e Pytest-Django para a API.

#### 3. Testes de Ponta a Ponta:

- **Objetivo:** Validar os fluxos de ponta a ponta.
- **Ferramentas:** Cypress ou Playwright.

### 4. Definição das Tecnologias a Serem Utilizadas

As tecnologias foram escolhidas devido a familiaridade.

#### ● Frontend:

- **Framework:** React.js (puro, utilizando Hooks e Context API para estado).
- **Estilização:** Material-UI.
- **Requisições HTTP:** Axios.

#### ● Backend:

- **Plataforma:** Python.
- **Framework:** FastAPI

#### ● Banco de Dados:

- **SGBD:** PostgreSQL.

### 5. Definição de Cronograma

#### Etapa de planejamento (até 15/09):

- Definição do documento de planejamento.
- Criação dos repositórios (setup inicial).

#### Sprint 1:

##### 1. Setup inicial

- Modelagem final do banco de dados.

##### 2. Gestão de Usuários

- Cadastro de professores.

- Cadastro de alunos.
- CRUD de criação de usuários/participantes.
- 3. Gestão de Oficinas (Professor/Tutor)**
  - CRUD de oficinas (tema, descrição, vagas, datas).
  - Publicar/remover oficinas.
- 4. Catálogo e Inscrição (Aluno)**
  - Tela de catálogo de oficinas.

## **Sprint 2:**

- 1. Presença e Finalização da Oficina**
  - Professor registra presença dos alunos.
  - Marcar oficina como concluída.
- 2. Certificados**
  - Geração simples de PDF com dados do aluno e oficina.
  - Tela de histórico do aluno com certificados.
- 3. Ajustes e Qualidade**
  - Testes finais.
  - Correções de bugs e melhorias de UX.