

**INSTITUTO
FEDERAL**
Triângulo Mineiro

PP - LabESOF

Descrição do Mini cenário: **Site de Publicação de Livros**

Data: 29/07/2025

Autores: Gabriel Lucas Silva Seabra,
Jussie Lopes da Silva,
Luiz Alberto Cury Andalécio,
Vitor Gabriel Resende Lopes Oliveira,
Alexandre Monteiro Londe.

Sumário

1. Introdução
2. Visão Geral do Sistema
 - a. Objetivo Geral
 - b. Objetivos Específicos
 - c. Resumo da Solução
3. Product Backlog - Histórias de Usuários (Nível de Negócio)
4. Diagrama de Casos de Uso
5. Diagrama de Atividade ou BPMN
6. Identidade Visual
 - a. Design Inicial
 - b. Logotipo com Slogan
7. Arquitetura e Tecnologias a Serem Utilizadas
 - a. Arquitetura
 - b. Tecnologias
8. Ambiente e Configurações a Serem Definidas
 - a. Ambiente de Desenvolvimento
 - b. Etapas de Configuração
 - c. Proposta de Deploy



Plano de Projeto - PP

O Plano de Projeto - PP é o documento que direciona todo o desenvolvimento do Projeto, nele consta todos os elementos que envolvem o desenvolvimento pleno das atividades do projeto.

Este artefato considera a Entrega 1, e conta com um valor de nota apresentado na seção de Distribuição de Notas - DN no final deste documento.

Para compor o PP, os itens abaixo são considerados obrigatórios. Porém, o time pode acrescentar algo que não conste no modelo apresentado, caso desejem enriquecer mais ainda o trabalho.

Como este documento é o artefato norteador do projeto, todos os membros do time estão engajados neste, para entender todos os passos que serão desenvolvidos no projeto.

Abaixo os itens compostos no PP:

1. Introdução

Este documento apresenta o Plano de Projeto do Readowl, um software desenvolvido com o objetivo de fomentar a literatura amadora no Brasil. Ele será o guia principal de direcionamento para todas as fases do desenvolvimento do sistema, contendo as diretrizes, requisitos, diagramas, identidade visual, arquitetura e ambiente necessário para construção do produto.

Atualmente, considerando todo tipo de leitura, desde jornais até livros de autoajuda, apenas cerca de 45% da população brasileira possui o hábito de ler. Quando o foco se volta para histórias fictícias, esse número cai drasticamente e demonstra um nicho já escasso dentro de um costume pouco cultivado, e com isso, a quantidade de autores dispostos a oferecer narrativas de qualidade também vem diminuindo, afinal, se poucos têm o costume de ler, pra que escrever e postar uma obra?

Entretanto, o problema vai além da falta de leitores ou escritores: encontrar uma plataforma confiável onde um autor possa publicar sua obra gratuitamente e receber um pouco de apoio financeiro ainda é um desafio considerável, principalmente quando o público-alvo é brasileiro. As plataformas existentes, em sua maioria, apresentam falhas que frustram leitores, escritores e até administradores, seja no sistema de leitura pouco personalizável, na divulgação ineficiente (e muitas vezes injusta), no processo de aprovação que pode gerar conflitos entre avaliador e escritor, ou na interface confusa para publicação. Tudo isso desestimula quem deseja criar e compartilhar.

Então, pensando nessas complicações, nasce o Readowl: um software criado para acolher escritores iniciantes e crescer a cultura literária no Brasil.

2. Visão Geral do Sistema

a. Objetivo Geral

Desenvolver uma plataforma de publicação e leitura de livros com o propósito de aproximar leitores de escritores iniciantes, ampliando a visibilidade dessas obras e fortalecendo o cenário da literatura amadora brasileira. Inclusive, ao incentivar novos autores e aumentar o

interesse pela escrita, o sistema também contribui para a formação de futuros escritores profissionais, o que eleva a qualidade e o alcance da produção literária nacional.

b. Objetivos Específicos

Estrutura básica e acesso

- **Landing Page** para receber visitantes, apresentar a proposta do site e incentivar o cadastro.
- **Sistema de cadastro e login seguro** com autenticação e recuperação de senha.
- **Tela inicial para usuários logados** com acesso rápido às funcionalidades, configurações e suporte.
- **Configuração de perfis de usuário** com opções para editar informações, definir privacidade e foto de perfil.

Publicação e organização de obras

- **CRUD de livros** para criar, visualizar, editar e excluir obras.
- **CRUD de volumes** para organizar capítulos.
- **CRUD de capítulos** com editor de texto formatado (negrito, itálico, cabeçalhos, imagens).
- **Sistema de rascunho e publicação** para que autores possam salvar textos sem publicar imediatamente.

Descoberta e interação com conteúdos

- **Vitrine automatizada dos livros e capítulos em destaque** na página inicial do usuário logado com **filtros de conteúdo** por popularidade, avaliação, data e gênero literário.
- **Sistema de busca avançada** por título, autor, gênero e status (em andamento, completo).
- **Sistema de favoritos/biblioteca pessoal** para o leitor salvar obras de interesse.
- **Notificações de acordo com as obras favoritas/seguidas.**

Engajamento e feedback

- **Sistema de avaliação** para leitores deixarem notas nos livros.
- **Nota geral** para a obra **calculada automaticamente** com base na média das avaliações.
- **Curtidas** para capítulos e livros.
- **CRUD de comentários** na página do livro e nos capítulos para promover interação.

Conteúdo educativo e suporte

- **Aba de artigos e tutoriais** com dicas de escrita de um guia.
- **Central de ajuda/FAQ** com respostas para dúvidas comuns e tutoriais de uso da plataforma.

Administração e moderação

- **Gerenciamento de usuários** para administradores (edição, bloqueio, redefinição de senha).
- **Moderação de conteúdo** para análise e exclusão de obras ou comentários inapropriados.
- **Sistema de denúncia** para que usuários reportem conteúdos que violem as regras.

c. Resumo da Solução

O Readowl será uma aplicação web responsiva que oferecerá uma experiência intuitiva para leitores e escritores. O sistema será estruturado em três camadas: apresentação (frontend), lógica de negócio (backend) e persistência de dados (banco de dados), e será desenvolvido com tecnologias modernas, utilizando metodologias ágeis (Scrum) para organização e entrega contínua.

3. Product Backlog - Histórias de Usuários (Nível de Negócio)

3.1. Histórias de Usuário

ID	História	Prioridade
1	Como visitante, quero visualizar uma Landing Page que apresente a proposta do site, para entender os benefícios antes de me cadastrar.	Alta
2	Como visitante, quero poder criar uma conta de forma simples e segura, para acessar as funcionalidades da plataforma.	Alta
3	Como visitante, quero poder fazer login com autenticação segura e recuperação de senha, para garantir acesso contínuo à minha conta.	Alta
4	Como usuário, quero acessar uma tela inicial personalizada após o login, para encontrar facilmente minhas funções e conteúdos em uma navbar.	Alta
5	Como usuário, quero configurar meu perfil, alterando informações pessoais, foto e privacidade, para me apresentar da forma que desejar.	Alta
6	Como autor, quero criar, visualizar, editar e excluir meus livros, para gerenciar minhas obras publicadas.	Alta

- | | | |
|----|--|-------|
| 7 | Como autor, quero criar, editar e excluir volumes, para organizar os capítulos de maneira lógica. | Alta |
| 8 | Como autor, quero criar, editar e excluir capítulos usando um editor de texto formatado, para melhorar a apresentação do conteúdo. | Alta |
| 9 | Como autor, quero salvar capítulos como rascunho antes de publicar, para revisar e aprimorar meu texto antes de disponibilizá-lo. | Baixa |
| 10 | Como usuário, quero visualizar livros e capítulos em destaque na minha página inicial, para descobrir novos conteúdos. | Alta |
| 11 | Como usuário, quero buscar e filtrar obras por nome, popularidade, avaliação, data e gênero literário, para encontrar leituras que me interessem. | Média |
| 12 | Como usuário, quero receber telas de erro, para caso eu me depare com uma página falha. | Baixa |
| 13 | Como leitor, quero salvar livros na minha biblioteca/favoritos, para acompanhá-los futuramente. | Média |
| 14 | Como leitor, quero receber notificações sobre novos capítulos ou atualizações dos livros que sigo, para me manter informado. | Baixa |
| 15 | Como leitor, quero avaliar livros atribuindo notas e um comentário geral, para ajudar outros leitores e dar feedback ao autor. | Alta |
| 16 | Como sistema, quero contabilizar as visualizações de cada capítulo e exibir a soma delas na página inicial do livro, para exibir um ranking justo. | Média |
| 17 | Como leitor, quero curtir capítulos, para demonstrar que gostei do conteúdo. | Média |
| 18 | Como leitor, quero comentar no capítulo de um livro ou capítulo, para interagir com o autor e outros leitores. | Alta |
| 19 | Como visitante ou usuário, quero acessar artigos e tutoriais, incluindo um guia de escrita, para aprender e melhorar minha produção literária. | Baixa |
| 20 | Como usuário, quero acessar uma central de ajuda/FAQ, para encontrar respostas rápidas para minhas dúvidas. | Baixa |
| 21 | Como administrador, quero visualizar, editar, bloquear e redefinir senhas de usuários, para manter a comunidade segura. | Alta |
| 22 | Como usuário, quero denunciar livros, capítulos ou comentários, para ajudar a manter o ambiente saudável. | Baixa |
| 23 | Como administrador, quero revisar e remover obras ou comentários inadequados, para preservar a qualidade do conteúdo. | Média |

3.2. Épicos / Funcionalidades

Épico / Funcionalidade	Histórias Relacionadas	Prioridade
1º - Estrutura Básica e Acesso	1, 2, 3, 4, 5	Alta
2º - Publicação de Livros	6, 7, 8, 9	Alta
3º - Divulgação de Livros	10, 11, 13, 14	Alta
4º - Feedback de Leitores	15, 16, 17, 18	Alta
5º - Suporte	12, 19, 20	Média
6º - Administração	21, 22, 23	Alta

3.3. Tarefas de cada épico (exemplo detalhado)

1º Épico – Estrutura Básica e Acesso

Tarefa 1: Criar layout da Landing Page

Subtarefas:

1. Definir estrutura visual e conteúdo (wireframe).
2. Desenvolver a definição com HTML/CSS.
3. Implementar animações e efeitos visuais (framer-motion ou CSS transitions).
4. Integrar links para login e cadastro.
5. Otimizar imagens e SEO.
6. Realizar testes de usabilidade.

Tarefa 2: Implementar sistema de cadastro e login com autenticação

Subtarefas:

1. Criar formulário de cadastro com validação no frontend.
2. Implementar rota de criação de usuário no backend.
3. Implementar criptografia de senha.
4. Implementar login com geração de token armazenado de forma segura.
5. Realizar testes unitários e de integração para autenticação.

Tarefa 3: Implementar recuperação de senha via e-mail

Subtarefas:

1. Criar formulário de solicitação de redefinição de senha.
2. Implementar envio de e-mail com link seguro (nodemailer + token temporário).
3. Criar página de redefinição de senha.
4. Validar token de redefinição no backend.
5. Testar o fluxo.

Tarefa 4: Criar a navbar da tela inicial para usuário logado.

Subtarefas:

1. Criar layout com logo.
2. Criar atalhos para as principais funcionalidades: home, pesquisar, notificações, perfil, publicar, minhas publicações.
3. Testar o Fluxo

Tarefa 5: Implementar CRUD de perfil do usuário

Subtarefas:

1. Criar página de edição de perfil.
2. Implementar upload e tratamento de imagem de perfil.
3. Adicionar campos de biografia, redes sociais e preferências de leitura.

2º Épico – Publicação de Livros

Tarefa 1: Implementar CRUD de livros

Subtarefas:

1. Criar página para o formulário de criação/edição de livro.
2. Adicionar campos para título, sinopse, gênero e capa.
3. Implementar upload e compressão de imagem de capa.
4. Criar endpoint de exclusão ou cancelamento seguro.

5. Mostrar o livro na página inicial do criador.
6. Testar a criação.

Tarefa 2: Implementar CRUD de volumes

Subtarefas:

1. Criar página para o formulário de criação/edição de volumes.
2. Adicionar campo para edição do nome.
3. Criar endpoint de criação, exclusão ou cancelamento seguro.
4. Listar volume em ordem alfabética em seu respectivo livro.
5. Testar a criação.

Tarefa 3: Implementar CRUD de capítulos com editor de texto formatado

Subtarefas:

1. Criar página com a listagem dos capítulos criados.
2. Criar página para o formulário de criação/edição de capítulos.
3. Adicionar campos para título, volume, conteúdo, etc.
4. *Implementar a função de “Salvar como rascunho”.
5. *Criar coluna no banco para status do capítulo (publicado/rascunho).
6. Criar endpoint de criação, exclusão ou cancelamento seguro.
7. Listar capítulo em ordem alfabética nos seus respectivos volumes.
8. Testar a criação.

3º Épico – Divulgação de Livros

Tarefa 1: Desenvolver vitrine automatizada de livros e capítulos em destaque

Subtarefas:

1. Implementar carrossel com destaque por popularidade (filtro dia/mês/ano).
2. Implementar carrossel com destaque por últimos lançamentos.

3. Implementar carrossel com destaque por nota.
4. Criar tabela de últimos capítulos lançados.
5. Integrar vitrine com API.

Tarefa 2: Implementar filtros de conteúdo

Subtarefas:

1. Criar formulário de pesquisa com dropdowns para gênero, status e data de publicação.
2. Implementar lista de respostas de acordo com o filtro.
3. Testar a busca.

Tarefa 3: Implementar favoritos/biblioteca pessoal

Subtarefas:

1. Criar botão “Seguir”.
2. Criar página de biblioteca pessoal com os livros favoritos adicionados.
3. *Permitir organização por filtros personalizados.

Tarefa 5: Implementar sistema de notificações

Subtarefas:

1. *Criar tabela de notificações no banco.
2. *Implementar push notifications no frontend.
3. *Criar página/lista de notificações.

4º Épico – Feedback de Leitores

Tarefa 1: Implementar sistema de avaliação

Subtarefas:

1. Criar componente de estrelas para o usuário deixar uma avaliação geral na tela inicial do livro.

2. Criar CRUD de comentário para o usuário deixar uma avaliação geral na tela inicial do livro.
3. Salvar avaliações no banco com ID do usuário.
4. Permitir apenas uma avaliação por usuário por livro.
5. Permitir remoção de avaliação para o administrador
6. Criar função para calcular média das notas do livro e exibi-las na tela inicial.
7. Atualizar nota geral do livro ao inserir ou alterar avaliação.
8. Testar avaliações.

Tarefa 2: Implementar cálculo automático da nota geral

Subtarefas:

1. *Criar função para contabilizar a visualização do capítulo a cada clique na tela, com verificações. Não conta a visualização nos seguintes casos: o autor visualiza os próprios capítulos, usuário recarregue a página.
2. *Implementar visualizações gerais do livro baseado na soma de visualizações de cada capítulo.

Tarefa 3: Implementar curtidas nos capítulos

Subtarefas:

1. Criar botão de curtir com contador para capítulos.
2. Criar tabela para relacionar curtidas a usuários.
3. Permitir remoção de curtida para o usuário que a fez
4. Permitir apenas uma curtida por usuário por capítulo.
5. Testar avaliações.

Tarefa 4: Implementar CRUD de comentários para capítulos

Subtarefas:

1. Criar campo para adicionar comentário no capítulo.
2. Implementar listagem de comentários com paginação no fim do capítulo.
3. Adicionar função de exclusão de comentários pelo autor ou administrador.

4. *Criar botão de curtir comentário com contador.
5. Salvar comentário no banco.
6. *Permitir remoção de curtida para o usuário que a fez.
7. *Permitir apenas uma curtida por usuário por comentário.
8. Testar comentários e suas funcionalidades.

5º Épico – Suporte

Tarefa 1: Implementar telas de erro

Subtarefas:

1. Implementar view de detecção automática de erro com as imagens personalizadas.
2. Testar erros.

Tarefa 2: Criar página de artigos e tutoriais

Subtarefas:

1. *Criar botão para CRUD de artigos visíveis somente por administradores.
2. *Implementar tela de CRUD de artigos para administradores.
3. Criar estrutura de listagem de artigos em cards.
4. Criar uma página de visualização individual do texto.
5. *Implementar editor de artigos para administradores.

Tarefa 3: Criar footer de ajuda/FAQ

Subtarefas:

1. Implementar footer com listagem de navegações, perguntas frequentes, redes sociais, informações dos criadores e etc.

6º Épico – Administração

Tarefa 1: Criar painel de gerenciamento dos usuários para administradores

Subtarefas:

1. Implementar lista com todos os usuários registrados (para administradores).

2. Possibilitar formulário de edição de informações da conta.
3. Possibilitar botão de exclusão de conta.

Tarefa 2: Implementar sistema de denúncias

Subtarefas:

1. Adicionar botão de “Denunciar” em livros, capítulos e comentários.
2. Criar tabela de denúncias no banco.

Tarefa 3: Criar painel de moderação de conteúdo para administradores

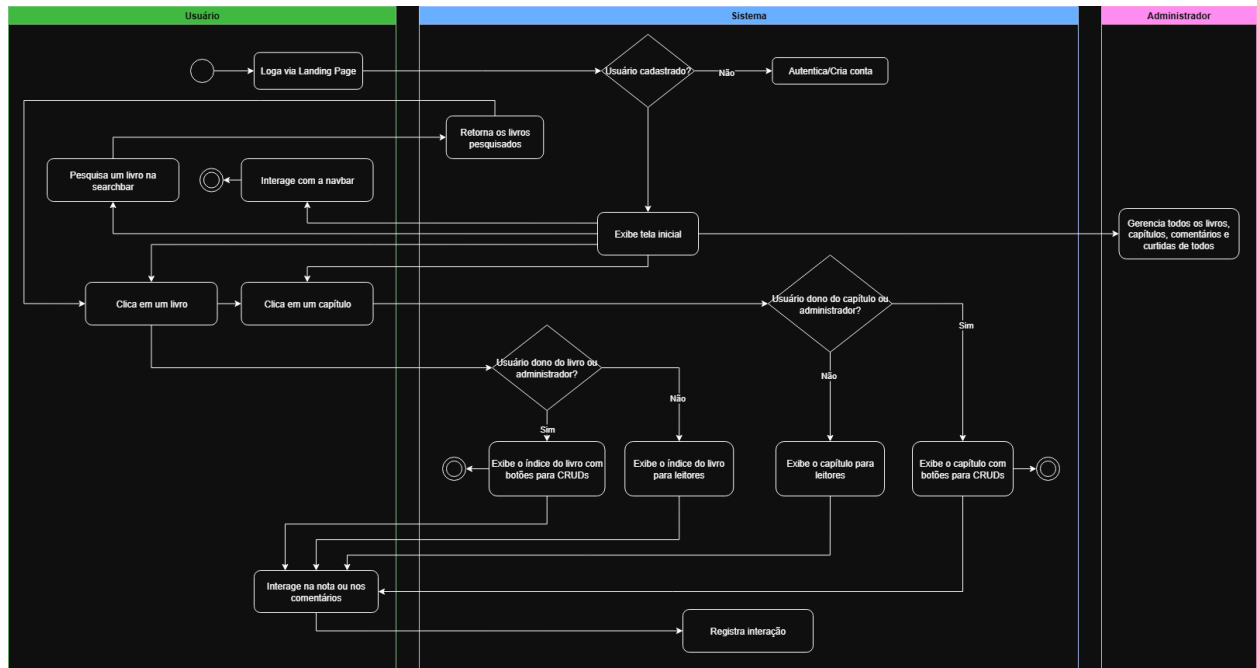
Subtarefas:

1. Criar listagem de obras e comentários denunciados.
2. Adicionar botão de exclusão.

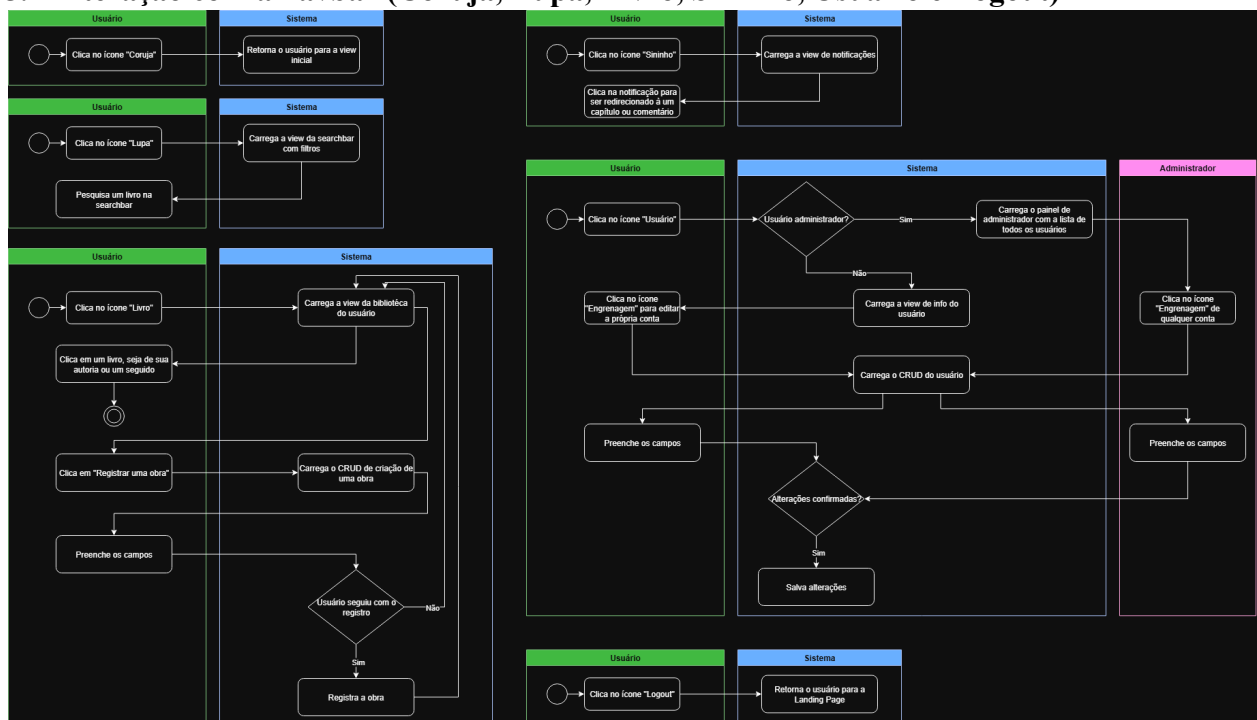
4. Diagrama de Casos de Uso

O Diagrama anexado em versão visual representa as funcionalidades que os autores usuário no sistema :

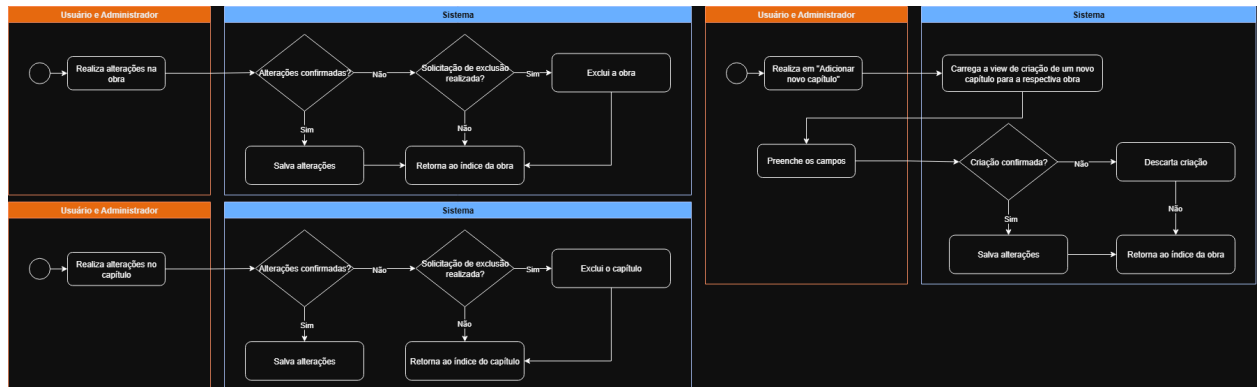
5.1 Visão macro



5.2 Interação com a navbar (Coruja, Lupa, Livro, Sininho, Usuário e Logout)



5.3 CRUDs (Edição/exclusão de livros/capítulos criados e adição de capítulos)



6. Identidade Visual

Identidade Visual do Readowl

Cores Principais:

- Roxo (#6750A4),
- Roxo Médio (#836DBE),
- Roxo Escuro (#23004B),
- Roxo Claro (#9E85E0),
- Branco (#FFFFFF),
- Branco Arroxado (#F0EAFB).

Tipografia

- Conteúdo Informativo: Poppins
- Conteúdo de Leitura: Yusei Magic

Botões

- Arredondados, com ícones e animações sutis ao passar o mouse para uma experiência interativa e amigável.

Logotipo com Slogan



- **Nome:** Readowl - uma fusão de "read" (ler) e "owl" (coruja), simbolizando sabedoria e conhecimento através da leitura.
- **Slogan:** "Cultivando literatura para qualquer lugar."
- **Mascote:** Uma coruja mago.

Prototipação no [Figma!](#)

7. Arquitetura e Tecnologias a Serem Utilizadas

Arquitetura:

3 camadas.

Frontend:

React (com Vite): Base da interface do usuário.

React Router DOM: Navegação entre páginas (Home, Livro, Perfil, etc.).

Tailwind CSS: Estilização rápida e responsiva, seguindo a identidade visual.

TanStack Query: Comunicação com o backend, cache e atualização de dados.

React Hook Form: Construção de todos os formulários (login, cadastro, publicação).

TipTap: Editor de texto para os autores escreverem seus capítulos.

Backend:

Node.js (com Express): Servidor e lógica de negócio.

TypeScript: Adiciona tipagem estática, evitando bugs.

Prisma: ORM para comunicar com o banco de dados PostgreSQL.

Zod: Validação de dados de forma consistente no front e no back.

JWT + Bcrypt.js: Sistema de autenticação e segurança de senhas.

Multer + Cloudinary: Upload e armazenamento das capas dos livros e fotos de perfil.

Nodemailer: Envio de e-mails para recuperação de senha.

Banco de Dados:

PostgreSQL: Armazenamento de todos os dados.

Ambiente:

Docker, Git, VS Code.

8. Ambiente e Configurações a Serem Definidas

a. Ambiente de Desenvolvimento

- Editor: VS Code
- Controle de versão: Git + GitHub
- Gerenciador de pacotes: npm
- Containerização: Docker (Node + PostgreSQL)

b. Etapas de Configuração

1. Clonar repositório
2. Instalar dependências com npm install
3. Configurar .env com variáveis locais
4. Subir containers com docker-compose up
5. Rodar migrations do banco de dados
6. Iniciar servidor com npm run dev

c. Proposta de Deploy

- Frontend: Vercel ou Netlify
- Backend: Render ou Railway
- Banco de Dados: Railway ou Supabase
- Domínios sugeridos: readowl.[app.br](https://readowl.app.br) ou readowl.[com.br](https://readowl.com.br)