

# DOCUMENTAÇÃO

Backlog de Jogos Digitais

GamesLog

**Responsável: Luiz Fernando Brito Ferreira**

# **Introdução**

**Nome do Projeto:** GamesLog

## **Descrição Geral:**

Com o crescimento exponencial da indústria de jogos digitais e a ampla variedade de plataformas e títulos disponíveis, organizar e gerenciar o progresso dos jogos tornou-se um desafio para muitos jogadores. Este projeto visa criar um site intuitivo e moderno para ajudar os usuários a centralizar o gerenciamento de seu backlog de jogos. O sistema permitirá que os usuários registrem seus jogos, acompanhem o progresso (não iniciado, jogando, finalizado), visualizem estatísticas personalizadas e consultem informações detalhadas de jogos através de integrações com APIs externas.

## **Objetivo Geral:**

Desenvolver uma plataforma que simplifique e otimize o gerenciamento de backlog de jogos digitais para os usuários.

## **Objetivos Específicos:**

- Fornecer uma interface amigável e responsiva para cadastro e edição de jogos.
- Implementar funcionalidades de filtros e buscas para facilitar o acesso às informações desejadas.
- Integrar uma API externa para preencher automaticamente detalhes sobre jogos (como descrição, gênero e tempo estimado de conclusão).
- Apresentar estatísticas personalizadas, como status dos jogos e gêneros mais jogados.

## ❓ Funcionalidades Principais

- Gerenciamento de Backlog:
  - Adicionar;
  - Editar;
  - Remover;
  - Listar jogos.
- Atualizar o status do jogo:
  - Desejo;
  - Em Espera;
  - Jogando;
  - Pausado;
  - Dropado;
  - Finalizado;
  - 100%;
  - Platinado;
- Pesquisa:
  - Pesquisa por nome do jogo.
- Integração com API de Jogos:
  - Buscar automaticamente informações (descrição, gênero, tempo de conclusão) ao adicionar um jogo.
- Estatísticas:
  - Jogos por status (ex.: 3 finalizados, 5 em progresso).
  - Filtros (Geral ou por ano):
    - Finalizados;
    - Dropados;
    - Plataforma;
    - ...
  - Gêneros mais jogados.
  - Quantidade de horas médias.

# Requisitos

## 3.1. Funcionais

O usuário deve ser capaz de:

- Adicionar, visualizar, editar e remover jogos.
- Filtrar e pesquisar jogos.
- Visualizar informações detalhadas dos jogos via API.

## 3.2. Não Funcionais

- Interface responsiva para dispositivos móveis e desktop.
- Backend eficiente e seguro para gerenciar dados.
- Tempo de resposta rápido, especialmente nas requisições às APIs externas.

# Tecnologias

## FrontEnd:

- React.js: Para criar a interface do usuário.
- Tailwind CSS: Para estilização responsiva.

## BackEnd:

- Java (Spring Boot): Para a lógica de negócios e API do sistema.
- PostgreSQL: Banco de dados relacional.

## APIs Externas:

- RAWG: Para buscar informações sobre jogos.
- Hobby API: Para buscar informações sobre o tempo médio para finalizar jogos.

# Modelagem de Dados

Modelo inicial para o banco de dados:

Campo	Tipo	Descrição
<b>id</b>	Inteiro	Identificador único do jogo.
<b>title</b>	String	Nome do jogo.
<b>platform</b>	String	Plataforma do jogo (ex.: PC, PS5).
<b>genre</b>	String	Gênero do jogo (ex.: Ação, RPG).
<b>status</b>	String	Status ( <i>Não iniciado, Jogando</i> ).
<b>estimated_time</b>	Inteiro	Tempo médio para concluir o jogo.
<b>notes</b>	Texto	Observações adicionais do usuário.

## Wire frames

# 📌 Roadmap do Desenvolvimento

## Fase 1: Configuração Inicial

- Configurar o ambiente do projeto:
- Criar repositórios no GitHub.
- Configurar React (frontend) e Spring Boot (backend).
- Criar o banco de dados PostgreSQL.
- Implementar o CRUD básico no backend.

## Fase 2: Funcionalidades Avançadas

- Implementar filtros, busca e integração com APIs externas.
- Exibir informações detalhadas de cada jogo.

## Fase 3: Estilização

- Adicionar responsividade e design moderno com Tailwind CSS.

## Fase 4: Deploy

- Frontend: em serviços como Vercel ou Netlify.
- Backend: em serviços como Railway ou Heroku.
- Banco de dados no Railway ou Supabase.



## Referências

Documentação do React: [React](#)

Documentação do Tailwind: [Documentation - Tailwind CSS](#)

Documentação do Spring Boot: [Spring Boot](#)

RAWG API: [Explore RAWG Video Games Database API • RAWG](#)

Hobby API: [GitHub - ckatzorke/howlongtobeat: A simple api for https://howlongtobeat.com/](#)