

LUCAS TITO SAMPAIO

# DOCUMENTO DE REQUISITOS

Gamecrate

[Github auxiliar](#)

2025

# PROBLEMA (ANÁLISE DO DOMÍNIO)

## CONTEXTO

Este projeto surgiu de uma ideia pessoal para treinar e aprofundar meus conhecimentos em desenvolvimento full stack utilizando Vue.js, Spring Boot, PostgreSQL e Docker. Mais do que um exercício técnico, a aplicação nasce da vontade de tirar uma ideia do papel: criar uma plataforma onde eu possa registrar minhas experiências com jogos.

Com o tempo, percebi que esquecia detalhes sobre os jogos que finalizei — o que gostei, o que me incomodou, quanto tempo levei, e até mesmo qual plataforma utilizei. Ter um espaço próprio para dar notas, escrever comentários e organizar esses registros se tornou uma necessidade pessoal.

## ALTERNATIVAS E CONCORRÊNCIA

### Backloggd

**Descrição:** plataforma inspirada no Letterboxd, focada em reviews e organização de jogos.

**Nicho/foco da alternativa:** jogadores que gostam de registrar e comentar suas experiências.

**Funcionalidades principais:** marcações como "jogado", "finalizado", "desejo jogar", notas e listas.

**Diferencial declarado pelos desenvolvedores:** forte senso de comunidade.

### GGApp

**Descrição:** aplicação para registrar jogos, com sistema de backlog e notas.

**Nicho/foco da alternativa:** jogadores que querem organizar jogos já jogados ou ainda não iniciados.

**Funcionalidades principais:** marcações como "jogado", "finalizado", "desejo jogar", notas e listas.

**Diferencial declarado pelos desenvolvedores:** visual minimalista e foco na organização pessoal.

### HowLongToBeat

**Descrição:** plataforma voltada para estimativas de tempo de jogo.

**Nicho/foco da alternativa:** jogadores que desejam saber quanto tempo levarão para zerar determinado título.

**Funcionalidades principais:** tempos médios, listas e categorias de progresso.

**Diferencial declarado pelos desenvolvedores:** foco na duração dos jogos.

## ENTENDENDO OS USUÁRIOS

### **Backloggd**

**Pontos Fortes:** sistema robusto de review e lista, comunidade ativa.

**Pontos Fracos:** interface pode ser confusa para novos usuários.

### **GGApp**

**Pontos Fortes:** boa organização pessoal, fácil navegação.

**Pontos Fracos:** poucas opções de personalização e recursos limitados na versão gratuita.

### **HowLongToBeat**

**Pontos Fortes:** útil para planejamento de tempo.

**Pontos Fracos:** não possui foco em avaliações nem categorização flexível.

## IDEAÇÃO (VISÃO DO PRODUTO)

### Requisitos (Objetivos) de Negócio

#### **[RN-1] Consolidar o Aprendizado em Desenvolvimento Full Stack**

Desenvolver e lançar uma aplicação web completa utilizando Vue.js, Spring Boot, PostgreSQL e Docker, aplicando boas práticas de arquitetura, versionamento e deploy até o final do segundo semestre de 2025.

#### **[RN-2] Criar um Sistema Funcional de Registro de Jogos**

Disponibilizar uma plataforma pessoal onde seja possível registrar, avaliar e organizar pelo menos 50 jogos diferentes no primeiro trimestre de uso, incluindo notas, comentários, tempo de finalização e status de término.

#### **[RN-3] Explorar Integrações com APIs Externas de Jogos**

Realizar pelo menos uma integração com API pública (como IGDB ou RAWG.io) para busca automatizada de jogos até o segundo mês de desenvolvimento da aplicação.

#### **[RN-4] Documentar e Versionar o Projeto Completamente**

Manter toda a estrutura do projeto documentada em repositório público (como GitHub), incluindo README completo, instruções de uso e estrutura modularizada com Clean Architecture até o fim do desenvolvimento.

# ESPECIFICAÇÃO

## REQUISITOS FUNCIONAIS

- [HU-1] Como usuário, posso criar uma conta para acessar a plataforma.
  - Conversas/detalhes:
    - Durante o processo de cadastro, devem ser solicitadas informações básicas como nome, e-mail e senha.
  - Confirmações:
    - Após o preenchimento dos campos obrigatórios, o sistema deve retornar mensagem de sucesso.
  - Caminho de exceção:
    - Se o usuário não preencher todos os campos obrigatórios, o sistema deve exibir uma mensagem de erro informando quais campos estão faltando.
    - Se o e-mail já estiver associado a outra conta, o sistema deve exibir um erro informando que o e-mail já está em uso.

☐ Implementado

☐ Validado

- [HU-2] Como usuário, posso buscar jogos em uma API externa para adicioná-los ao meu catálogo.
  - Conversas/detalhes:
    - Ao digitar um nome, o sistema exibe sugestões da API.
  - Confirmações:
    - Jogo adicionado com sucesso ao catálogo.
  - Caminho de exceção:
    - Falha na API.
    - Jogo não encontrado.

☐ Implementado

☐ Validado

- [HU-3] Como usuário, posso adicionar jogos manualmente (sem usar API).
  - Conversas/detalhes:
    - Ao buscar pelo nome do jogo e nenhum nome for encontrado, mostrar opção de adicionar manualmente.
    - O botão de adicionar jogos deve ser um split button, que conta com um dropdown para exibir a opção de adicionar manualmente.
    - Formulário com nome, plataforma, ano, descrição, imagem.

- Confirmações:
  - Jogo salvo no catálogo pessoal.
- Caminho de exceção:
  - Campos obrigatórios não preenchidos.

☐ Implementado

☐ Validado

- [HU-4] Como usuário, posso avaliar um jogo com uma nota, comentário, tempo de finalização e status de término da história principal.

- Conversas/detalhes:
  - Campo de nota (0–10).
  - Campo de comentário (texto livre).
  - Campo de tempo estimado de finalização (ex: 20h).
  - Campo “Terminei a história principal?” com opções: *Sim*, *Não*, *Em andamento*.
- Confirmações:
  - Avaliação vinculada ao jogo.
- Caminho de exceção:
  - Nota fora do intervalo, tempo mal formatado, status inválido.

☐ Implementado

☐ Validado

- [HU-5] Como usuário, posso criar categorias personalizadas para agrupar meus jogos.

- Conversas/detalhes:
  - Criação de nova categoria e seleção de jogos.
- Confirmações:
  - Jogos listados na categoria criada.
- Caminho de exceção:
  - Nome de categoria duplicado.

☐ Implementado

☐ Validado

- [HU-6] Como usuário, posso listar e filtrar meus jogos por plataforma, nota, data de adição, tempo de finalização, status de término ou categoria.
  - Conversas/detalhes:
    - Opções de filtro visíveis e aplicáveis.
  - Confirmações:
    - Exibição atualizada da lista.
  - Caminho de exceção:
    -

☐ Implementado

☐ Validado

## REQUISITOS NÃO-FUNCIONAIS

- [RNF-1]: Interface Intuitiva
  - A interface da plataforma deve ser clara, moderna e fácil de navegar para usuários de todas as idades e níveis de habilidade tecnológica, garantindo uma experiência de usuário agradável e eficiente.

☐ Implementado

☐ Validado

- [RNF-2]: Responsividade
  - O design da plataforma deve se adaptar a diferentes tamanhos de tela e dispositivos, incluindo smartphones, tablets, e desktops, garantindo uma experiência de usuário consistente em todos os dispositivos.

☐ Implementado

☐ Validado

- [RNF-3]: Segurança de Dados
  - Devem ser implementadas medidas de segurança avançadas para proteger os dados pessoais e sensíveis dos usuários, incluindo criptografia de dados em trânsito e em repouso, e protocolos de autenticação segura.

☐ Implementado

☐ Validado

- [RNF-4]: Acessibilidade
  - A plataforma deve seguir as melhores práticas de acessibilidade web, garantindo que usuários com diferentes capacidades possam navegar e utilizar o sistema sem barreiras.

☐ Implementado

☐ Validado

- [RNF-5]: Integração com Serviços Externos
  - A integração com serviços de terceiros (como a API de jogos utilizada para busca) deve ser estável e confiável, com tratamento de erros caso o serviço esteja indisponível.
    - ☐ Implementado
    - ☐ Validado

## PROTÓTIPOS DE INTERFACE DE USUÁRIO

[Protótipo](#)