



Análise e elicitação de requisitos do projeto

Grupo 03: Adla, Clara, Daniel, Francisco e Heloisa.

Autor(a): Adla.

INCLUDE ENGENHARIA

ANÁLISE DE REQUISITOS - Sistema de coleção de Pokémon

1. Elicitação de requisitos

- Planejamento para reunir as funções que serão necessárias para atender o objetivo do sistema. Dessa forma, serão separados, nesse documento, os requisitos funcionais e não funcionais do sistema, seguidos de análises prévias de funções dentro do projeto.
- **Requisitos funcionais**

- Tabela 1 – Requisitos Funcionais do sistema.

ID	Nome	Descrição
RF01	Menu	É necessário um menu para guiar o usuário durante sua navegação no site e levá-lo às funcionalidades.
RF02	Coleta de pokemons	O colecionador terá uma lista de pokemons em que pode escolher e colecionar. Essa lista deve ser gerada de uma forma aleatória visando diversificar as coleções entre os jogadores. (dados -> PokéAPI com aleatoriedade definida conforme definição do grupo, sendo limitado a 3 pokemons por



		coleccionador.)
RF03	Adição, atualização, remoção e listagem de Pokémon.	O usuário deve conseguir adicionar, listar, atualizar e deletar os pokémon que coletar;
RF04	Interface	Deve ter uma interface que contenha o Nome, Descrição, Tipo, HP (Health Points) e PC (Pontos de Combate) dos pokémon, assim como também deve ter uma interface que é possível visualizar quais pokémon o colecionador possui, e seus dados (Nome, ID, Número de Pokémon).
RF05	Cadastro e Login de usuários	O sistema deverá contar com uma área de realização de cadastro de novos usuários e de posterior login.
RF06	Sistema de batalha	Usuários podem batalhar entre si selecionando dois Pokémon. Regras da batalha baseadas em HP e PC, com o PC tendo maior peso. Mensagens de erro caso o adversário não possua Pokémon suficientes ou esteja indisponível.

● Fonte: Autoria própria



- **Requisitos não funcionais**

- Tabela 2 – Requisitos Não Funcionais..

ID	NOME	DESCRIÇÃO
RNF01	Desempenho	O sistema deve exibir o conteúdo disponível de forma rápida.
RNF02	Utilização obrigatória de tecnologias	Utilização obrigatória de tecnologias: ReactJS para o front-end. Spring Boot para o back-end. PostgreSQL para banco de dados. PokéAPI para consultas de dados.
RNF03	Responsividade	O sistema deve ser compatível com uma variedade de tamanhos de tela, garantindo uma experiência consistente em diferentes aparelhos.
RNF04	Usabilidade	O conteúdo do sistema deve estar contido em um sistema de fácil uso, com funções intuitivas e textos de fácil compreensão e visualização.

- Fonte: Autoria própria.

2. Atribuições dos Papéis no Fluxo

1. Analista de Requisitos

- Levantar requisitos e documentar cenários de uso.
- Criar diagramas de casos de uso e descrever os fluxos do sistema.
- Garantir que o time esteja alinhado aos objetivos.



2. UI/UX Designer

- Criar protótipos das interfaces no Figma ou outra ferramenta.
- Garantir que o design seja intuitivo e reflita a temática Pokémon.
- Validar a usabilidade das telas.

3. Desenvolvedor Front-end

- Implementar as telas e funcionalidades usando ReactJS.
- Integrar com APIs (PokéAPI e endpoints back-end).
- Garantir responsividade e funcionalidade no front-end.

4. Desenvolvedor Back-end

- Criar endpoints REST usando Spring Boot.
- Implementar lógica de negócios, como batalhas.
- Configurar conexão com o banco de dados.

5. Especialista em Banco de Dados

- Modelar e criar o banco de dados no PostgreSQL.
- Garantir integridade e eficiência nas operações CRUD.
- Manter backups e otimizações.

6. Tester/QA

- Desenvolver casos de teste para verificar funcionalidades.
- Validar o comportamento esperado nos cenários de erro.
- Realizar testes de integração e desempenho.

7. Gerente de Projeto

- Organizar tarefas e gerenciar o cronograma.
- Realizar reuniões e garantir a comunicação entre os membros.
- Verificar entregas parciais e realizar ajustes conforme necessário.

Planejamento

- **Objetivo:** Definir o escopo, priorizar funcionalidades e alinhar a equipe.
 - Reunião inicial para:
 - Entender os requisitos funcionais e não funcionais.
 - Escolher os responsáveis por cada parte do sistema.



- Estabelecer prazos e milestones no GitLab.
- Ferramentas: **Pipefy**, **Discord** ou **Telegram**.

Design

- **Objetivo:** Planejar a arquitetura e design do sistema.
 - Arquitetura:
 - Front-end: Componentes React.
 - Back-end: Endpoints REST com Spring Boot.
 - Banco de Dados: Estruturar tabelas no PostgreSQL.
 - UI/UX:
 - Interface para CRUD de Pokémon e batalhas.
 - Coerência com a temática Pokémon.
 - Ferramentas: **Figma**, **Lucidchart**, ou **Draw.io** para diagramas.

Desenvolvimento

1. **Front-end:**
 - Criar interfaces:
 - Cadastro e login.
 - Listagem e gerenciamento de Pokémon.
 - Tela para batalhas.
 - Consumo da PokéAPI.
2. **Back-end:**
 - Implementar endpoints:
 - CRUD de Pokémon.
 - Gerenciamento de usuários.
 - Lógica de batalhas (cálculo de HP/PC).
3. **Banco de Dados:**
 - Criar tabelas:
 - Usuários.
 - Pokémon coletados.
 - Configurar conexão com Spring Boot.

Teste e Integração

- Garantir que cada módulo funcione individualmente e em conjunto.
 - Testes unitários para lógica de batalhas e CRUD.
 - Testes de integração entre front-end e back-end.



- Verificar conexões com a PokéAPI.

Entrega

- Consolidar:
 - Código no **GitLab**.
 - Documentação: Atas de reuniões, relatórios e roadmap.