Documento de Requisitos do Sistema (Pokédex)

**Cronograma do Projeto**

|  |  |
| --- | --- |
| **Data** | **Descrição** |
| 16/05/2023 | Definição da equipe e tema do projeto |
| 09/05/2024 | Definir os papeis de cada um no grupo |
| 09/05/2024 | Começar a produçao do html e css |
| 14/05/2024 | Integração da API com o sistema |
| 16/05/2024 | Ajustes de documentação e aprimoramento do sistema |
| 20/05/2024 | Terminar o projeto |
| 23/05/2024 | Entrega do Projeto |

**Equipe de desenvolvimento**

|  |  |
| --- | --- |
| **Nome** | **Função** |
| Guilherme Mendonça Silva | WebDesign  / CSS |
| Gustavo Henrique Rocha O. | Desenvolvedor JavaScript |
| Hennan Heim Falcão | Documentação e Suporte |
| João Vitor Amaral | Desenvolvedor HTML |

# Introdução

Este documento especifica os requisitos do sistema *Pokédex,* fornecendo aos desenvolvedores e stakeholders as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

* 1. **O Projeto consiste na criação de um repositório digital, com base na Pokédex, desenvolvido com HTML, CSS e JavaScript, com intuito de fazer com que os interessados no assunto, tenham como acessar informações sobre as criaturinhas digitais de maneira lúdica**.

Linkdorepositório:<https://github.com/yScroww/Pokedex-Site.git>

Link para simulação da aplicação: [Pokédex](https://yscroww.github.io/Pokedex-Site/)

# Descrição geral do sistema

O sistema deve permitir acessar informações acerca dos Pokémon por meio de integração à PokéAPI.

Uma API (Interface de Programação de Aplicações, do inglês Application Programming Interface) é um conjunto de regras e definições que permite que diferentes softwares se comuniquem entre si. Basicamente, uma API define os métodos e dados que os desenvolvedores podem usar para interagir com um serviço, aplicativo ou plataforma, facilitando a integração e a interoperabilidade entre diferentes sistemas.

O sistema deve permitir a entrada do usuário, em seguida, o usuário é direcionado para a página principal onde são exibidos os Pokémon.

A pesquisa é feita por meio de uma barra de busca, em que é possível acessar todos os Pokémon disponíveis para consulta. Ademais, na página principal será exibida uma lista com os Pokémon que é atualizada conforme o usuário desce a página. Essa lista muda a depender da geração de Pokémon que o usuário seleciona. Ao selecionar um Pokémon, é aberta uma seção, na mesma página, com informações específicas acerca dele. Informações como nome e tipos serão exibidas.

# Requisitos funcionais

Como é um sistema simples, a dependência de dados é restringida ao cadastro de usuários a parte voltada para o gerenciamento de dados foi toda desenvolvida com NodeJS. Os requisitos funcionais são: cadastro de usuários, a consulta de informações específicas sobre cada Pokémon, alteração da geração a qual se deseja consultar os Pokémon, busca.

* 1. **Cadastro de usuários**

O usuário deve informar e-mail e criar uma senha. A partir disso, é possível fazer o login.

* 1. **Informações específicas**

Ao clicar sobre cada Pokémon será possível obter informações específicas a respeito do Pokémon desejado, como nome, tipos, atributos, seu som e sexo.

* 1. **Alteração da geração dos Pokémon**

Na lateral direita da tela é possível escolher qual geração de Pokémon o usuário deseja visualizar

* 1. **Busca**

O campo de busca permite que o usuário digite o nome do Pokémon e o sistema aplica um filtro e exibe os Pokémon que contenham os caracteres digitados pelo usuário

# Requisitos não-funcionais

* 1. **Usabilidade**

Usabilidade é um conceito amplo que avalia a qualidade da experiência do usuário com um produto ou sistema. Quando aplicações são desenvolvidas, é crucial considerar a usabilidade para que as pessoas possam compreendê-las e utilizá-las da melhor forma possível. Aqui estão os principais pilares da usabilidade:

* **Facilidade de Aprendizado**: Quanto mais fácil for para os usuários aprenderem a usar a interface, melhor. Isso envolve a curva de aprendizado e o tempo necessário para compreender o sistema.
* **Eficiência e Memorização:** Após o primeiro contato, avaliamos se as atividades são facilmente memorizadas e executadas sem consultar a documentação. Uma interface eficiente é aquela que permite que os usuários realizem tarefas com facilidade.
* **Erros:** Avaliamos os erros cometidos pelos usuários durante a utilização. É importante entender a gravidade desses erros e se os usuários conseguem corrigi-los sem dificuldade.
* **Reminiscência**: Como os usuários se comportam após um período sem usar a aplicação? Isso ajuda a entender se há dificuldades em um novo contato com a interface.
* **Satisfação**: Medir o nível de satisfação é essencial. [Uma interface amigável e bem projetada contribui para a satisfação dos usuários](https://blog.betrybe.com/desenvolvimento-web/usabilidade/)

Como se trata de um projeto simples e para fins didáticos, esse processo foi simplificado. No mais, o projeto visou apresentar uma interface amigável e de simples compreendimento por parte do usuário.

# Diagramação e prototipaçãos UML

Esboço de Página Inicial



Figura 1 - Imagem ilustrativa

Página Inicial após primeira alteração



Figura 2 - Imagem ilustrativa

Página 3



Figura 3 - Imagem Ilustrativa



Figura 4 - Imagem Ilustrativa