

API POKÉMON GO

PADRÕES TÉCNICOS DE COMUNICAÇÃO DA API
E SOFTWARES UTILIZADOS

Nícolas Guarino Santana

Índice

Sumário

| | |
|--|----------|
| 1. INTRODUÇÃO..... | 2 |
| 2. ARQUITETURA..... | 3 |
| 3. SOFTWARES E LINGUAGENS DE PROGRAMAÇÃO..... | 3 |
| 3.1. NodeJs..... | 3 |
| 3.1.1. Jasmine..... | 4 |
| 3.2. MongoDB..... | 4 |
| 3.2.1. mLab..... | 4 |
| 3.2.2. Studio 3T..... | 5 |
| 3.3. Postman..... | 5 |
| 3.4. Git..... | 5 |
| 4. INSTALAÇÃO..... | 6 |
| 5. UTILIZAÇÃO..... | 6 |

1. INTRODUÇÃO

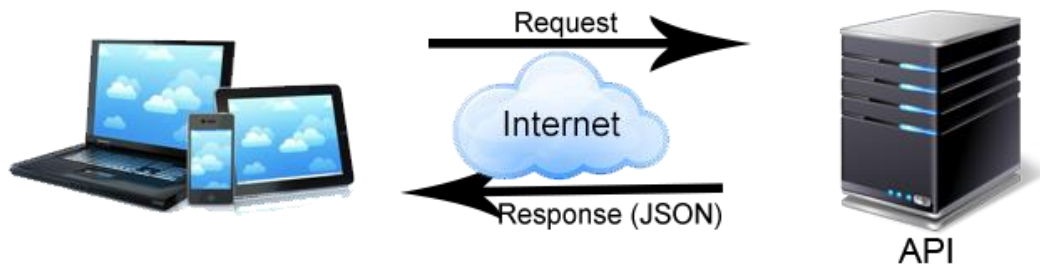
Guia técnico para implementação e utilização da API via JSON e softwares utilizados para o desenvolvimento.

A API Pokémon Go tem como objetivo fornecer dados para que possam ser consumidos onde é disponibilizado informações de Pokémon. Este serviço permite a qualquer solução customizada de TI seja facilmente integrada aos recursos disponíveis pela API.

Este documento contém os padrões técnicos para utilização e comunicação da API, uma breve descrição da tecnologia e suas linguagens utilizadas e um guia de consulta para o consumo da API.

2. ARQUITETURA

A arquitetura de software dos dados fornecidos pela API segue os padrões e os protocolos de comunicações, conforme a figura abaixo.



O serviço que será detalhado neste documento utiliza o padrão de resposta encode UTF-8, que é a forma mais amplamente utilizada para representar texto Unicode nas páginas da Web.

3. SOFTWARES E LINGUAGENS DE PROGRAMAÇÃO

Para o desenvolvimento da API foram utilizados diversos softwares e linguagens de programação, são eles:

3.1. Node.JS



O Node.JS foi escolhido para o desenvolvimento das aplicações devido sua facilidade de desenvolvimento, por trabalhar com Java Script, sendo assim mantendo um projeto fácil para realizar manutenções.

- **3.1.1. Jasmine**



O Jasmine é uma biblioteca do Node.JS utilizada para realizar testes em sua aplicação Java Script, foi escolhida por possuir uma sintaxe intuitiva o que faz com que escrever seus testes se torne uma tarefa fácil.

3.2. MongoDB



O MongoDB é um banco de dados orientado a documento que armazena dados em documentos JSON com esquemas dinâmicos. Isso significa que você pode armazenar seus registros sem se preocupar com a estrutura de dados, número de campos ou tipos de campos para armazenar valores.

- **3.2.1. mLab**



O mLab é um *hosting* de banco de dados e foi utilizado para hospedar o banco de dados na nuvem.

- **3.2.2. Studio 3T**



O Studio 3T é um software que oferece interface de usuário com base NoSQL para o MongoDB, que foi utilizado para realizar as manutenções no banco de dados.

3.3. Postman



O Postman é um software que permite enviar e receber dados via requisições HTTP, e para que possamos também realizar testes em nossas API's para verificar se o comportamento dela está dentro do esperado.

3.4. Git



O Git é um Sistema de controle de versões distribuído, utilizado para o desenvolvimento de softwares. Foi utilizado para criar as versões do projeto e manter o código seguro.

4. INSTALAÇÃO

- Faça um clone do projeto que está no [GitHub](#)
- Abra o Prompt de Comando na pasta onde o projeto foi baixado
- Execute os comandos `npm install`
- Após isso execute o comando `npm start`
- Ao iniciar a aplicação, você já pode visualizar o funcionamento da API diretamente do browser.

5. UTILIZAÇÃO

Após instalar o projeto em sua máquina, você pode utilizar os links abaixo para visualização do funcionamento da API.

- <http://localhost:3000/pokemons> - Listará todos os Pokémons registrados;

Lista os Pokémons com o ID desejado

- <http://localhost:3000/pokemons/id> – Modifique o final da URL que contém `/id` pelo código identificador do Pokémon: Ex: `/25` listando assim o Pokémon referente ao id digitado;

Lista os Pokémons com o nome desejado

- <http://localhost:3000/pokemons/nome/> - Ao final da URL digite o nome do Pokémon que deseja buscar:
Ex: <http://localhost:3000/pokemons/nome/Venusaur>;

Lista os Pókemons referentes a sua geração

- <http://localhost:3000/geracao/> - Ao final da URL insira a geração dos Pokémons que deseja listar:
Ex: <http://localhost:3000/geracao/1>

Lista os Pókemons de forma paginada

- <http://localhost:3000/pagina/quantidade> - Ao final da URL insira a página e a quantidade de itens por página que deseja buscar:
Ex: <http://localhost:3000/geracao/1/10>