

**Nome:** Mara Fabiana Pinheiro Alcântara

**RGM:** 27055931

## 1. Introdução

Este projeto é um sistema web que permite buscar usuários do GitHub e retornar suas informações e repositórios públicos. A aplicação será capaz de consultar a API do GitHub, coletar dados de usuários e seus repositórios, e exibir essas informações de forma clara e acessível para o usuário.

## 2. Visão Geral do Sistema

A plataforma terá as seguintes funcionalidades principais:

1. **Busca de Usuários:** O usuário poderá buscar outros usuários do GitHub utilizando o nome de usuário.
2. **Exibição de Dados do Usuário:** Retornará o nome, descrição e os repositórios.
3. **Exibição de Repositórios:** O sistema listará os repositórios públicos do usuário.

## 3. Arquitetura do Sistema

O sistema será desenvolvido utilizando a arquitetura cliente-servidor, onde o servidor terá uma API RESTful que irá interagir com o GitHub para obter as informações dos usuários e repositórios. O frontend irá consumir essa API e exibir as informações de forma interativa.

### 3.1. Componentes Principais

- **Frontend (Cliente):** Interface web que permite ao usuário buscar por usuários do GitHub e visualizar suas informações e repositórios.
- **Backend (Servidor):** Serviço que se comunica com a API pública do GitHub e fornece dados ao frontend.
- **Banco de Dados:** Pode ser utilizado para armazenar informações de autenticação e histórico de buscas.

### 3.2. Tecnologias Usadas

- **Frontend:**
  - HTML
  - CSS
  - JavaScript
- **Backend:**
  - Node.js com Express para a criação da API.

- GitHub API para buscar usuários e repositórios.

---

## 4. Funcionalidades

### 4.1. Busca de Usuários

- **Descrição:** A funcionalidade de busca permitirá que o usuário digite o nome de usuário do GitHub, e o sistema exibirá as informações relevantes do usuário e seus repositórios.
- **Requisitos Funcionais:**
  1. O usuário pode digitar o nome do usuário do GitHub na barra de busca.
  2. O sistema deve validar o nome fornecido.
  3. O sistema fará uma requisição à API do GitHub para buscar o usuário e seus repositórios.
  4. Se o usuário não for encontrado, o sistema deve exibir uma mensagem de informando que o usuário não existe.
- **Requisitos Não Funcionais:**
  1. A busca deve ser rápida, com uma resposta de até 2 segundos.
  2. A interface deve ser amigável e responsiva.

### 4.2. Exibição das Informações do Usuário

- **Descrição:** Ao encontrar um usuário, o sistema deve exibir as informações relevantes do perfil do GitHub, como nome, biografia, foto de perfil, etc...
- **Requisitos Funcionais:**
  1. O sistema exibirá o nome, foto, localização e bio do usuário.
- **Requisitos Não Funcionais:**
  1. A exibição deve ser clara e de fácil leitura.
  2. O layout deve ser adaptável a diferentes dispositivos.

### 4.3. Exibição de Repositórios

- **Descrição:** Após a busca por um usuário, o sistema deverá listar os dez últimos repositórios públicos do GitHub daquele usuário.
- **Requisitos Funcionais:**
  1. O sistema fará uma requisição à API do GitHub para listar os repositórios.
  2. A lista de repositórios exibirá o nome em um botão clicável.

3. O usuário pode clicar no nome do repositório para abrir a página do repositório no GitHub.
- **Requisitos Não Funcionais:**
    1. A exibição dos repositórios deve ser paginada para facilitar a navegação em grandes quantidades de dados.

---

## 5. Arquitetura Técnica

### 5.1. Backend

O backend será responsável por interagir com a API do GitHub e fornecer os dados para o frontend.

- **Endpoints principais:**
  - GET /users/:username: Retorna as informações básicas de um usuário (nome, bio, etc.).
  - GET /users/:username/repos: Retorna os repositórios públicos de um usuário.

### 5.2. Frontend

A interface será construída utilizando HTML, CSS e JavaScript, permitindo uma experiência interativa e dinâmica.

- **Principais componentes:**
  - **Barra de busca:** Permite ao usuário pesquisar por nome ou ID de usuário.
  - **Painel de resultados:** Exibe as informações do usuário e seus repositórios.
  - **Página de detalhes do repositório:** Exibe detalhes sobre um repositório quando o usuário clicar em um link.

### 5.3. Integração com a API do GitHub

A API pública do GitHub será utilizada para buscar dados sobre os usuários e seus repositórios.

- **Exemplo de requisição para obter informações de usuário:**
    - GET `https://api.github.com/users/{username}`
  - **Exemplo de requisição para obter repositórios de usuário:**
    - GET `https://api.github.com/users/{username}/repos`
-

## 6. Fluxo de Dados

1. O usuário entra na página e faz uma busca utilizando o nome de usuário ou ID.
  2. O frontend envia uma requisição para o backend.
  3. O backend consulta a API do GitHub e retorna os dados sobre o usuário e seus repositórios.
  4. O frontend exibe as informações ao usuário de forma estruturada e interativa.
- 

## 7. Testes

### 7.1. Testes Unitários

Serão escritos testes unitários para verificar a funcionalidade de cada componente individualmente, como a função que interage com a API do GitHub e a função que processa os dados de resposta.

### 7.2. Testes de Integração

Serão realizados testes de integração para verificar se o sistema de backend, frontend e a API do GitHub estão funcionando corretamente em conjunto.

### 7.3. Testes de Interface

Serão realizados testes de usabilidade para garantir que a interface do usuário seja intuitiva e funcional.

---

## 8. Conclusão

Este documento apresenta uma visão geral do sistema a ser desenvolvido, incluindo as funcionalidades principais, arquitetura, tecnologias e fluxos de dados. Com base neste plano, a equipe de desenvolvimento pode seguir com a implementação do projeto de forma eficiente, garantindo que todos os requisitos sejam atendidos.