|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REV. | DATA | Descrição | POR | Verif. |
| A | 14/06/2024 | Emissão Inicial | DCP |  |
| B | 18/07/2024 | Inclusão de diagramas e CDU | DCP |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Índice de Revisões

# Sumário

[1 Índice de Revisões 1](#_Toc172277842)

[2 Sumário 2](#_Toc172277843)

[3 Introdução 4](#_Toc172277844)

[3.1 Objetivo 4](#_Toc172277845)

[3.2 Escopo 4](#_Toc172277846)

[4 Visão do Projeto 4](#_Toc172277847)

[5 Requisitos do Projeto 4](#_Toc172277848)

[5.1 Requisitos Funcionais 4](#_Toc172277849)

[5.2 Requisitos Não-funcionais 5](#_Toc172277850)

[6 Arquitetura e Design do Sistema 5](#_Toc172277851)

[7 Modelagem de Dados 5](#_Toc172277852)

[7.1 Diagramas de Entidade-Relacionamento 5](#_Toc172277853)

[7.2 Diagrama de Classe 5](#_Toc172277854)

[7.3 Diagrama de Caso de Uso 5](#_Toc172277855)

[7.4 Descrição dos Casos de Uso 5](#_Toc172277856)

[7.4.1 Caso de Uso: Registrar Usuário 6](#_Toc172277857)

[7.4.2 Caso de Uso: Registrar Visita 7](#_Toc172277858)

[7.4.3 Caso de Uso: Consultar Visitas 8](#_Toc172277859)

[7.4.4 Caso de Uso: Registrar Notificação 9](#_Toc172277860)

[7.4.5 Caso de Uso: Consultar Notificações 10](#_Toc172277861)

[7.4.6 Caso de Uso: Encerrar Notificações 11](#_Toc172277862)

[7.4.7 Caso de Uso: Autenticar Usuário 12](#_Toc172277863)

[8 Rotas da Aplicação 13](#_Toc172277864)

[9 Permissões 14](#_Toc172277865)

[10 Interface de Usuário 14](#_Toc172277866)

[10.1 Wire frames e Protótipos 14](#_Toc172277867)

[10.2 Padrões de Design 14](#_Toc172277868)

[10.3 Acessibilidade e Usabilidade 14](#_Toc172277869)

[11 Banco de dados 15](#_Toc172277870)

[11.1 Script de criação do banco 15](#_Toc172277871)

[11.2 Tabelas 15](#_Toc172277872)

[12 Plano de testes 15](#_Toc172277873)

[12.1 Caso de Teste 15](#_Toc172277874)

[12.2 Ferramentas de Teste 15](#_Toc172277875)

[12.3 Testes de interface: 15](#_Toc172277876)

[12.4 Testes unitários: 15](#_Toc172277877)

# Introdução

## Objetivo

O presente documento tem como objetivo fornecer uma visão clara da proposta do projeto e o roteiro para seu desenvolvimento.

## Escopo

Desenvolver uma plataforma WEB de notificações em tempo real para os diversos parque nacionais do Brasil.

# Visão do Projeto

A plataforma permitirá que visitantes do parque nacional registrem notificações sobre algo que encontraram durante sua visita ao parque, por exemplo: lixos no local, pássaros nativos da região, problema ao acessar a trilha etc. Assim, o sistema servirá de grande apoio para a instituição que mantém o parque preservado, agilizando o processo de tomada de decisão, facilitando o monitoramento de descarte de resíduos inadequados e preservação das rotas de trilhas.

Com isso, esperamos aumentar a colaboração com a preservação do parque, fomentar o turismo, instigar a colaboração entre os visitantes e passar o sentimento de contribuição com o ponto turístico.

# Requisitos do Projeto

## Requisitos Funcionais

* O sistema deve permitir que o usuário se autentique utilizando sua conta google;
* Os usuários devem poder registrar-se usando e-mail e senha;
* O sistema deve permitir que o usuário registre notificações na plataforma;
* O usuário deve poder escolher o Parque Nacional desejado.
* O sistema deve exibir o mapa do Parque Nacional escolhido pelo usuário;
* O sistema deve exibir os pontos de notificações dos usuários no mapa;
* O sistema deve ser capaz de filtrar a exibição dos pontos de notificações no mapa por cada tipo de notificação.
* O usuário trabalhador do parque deve ter acesso específico;
* O usuário trabalhador deve ser capaz de retornar um feedback do tipo ‘Resolvido’ na notificação;
* O sistema deve exibir em forma de lista todas as notificações registradas no sistema.

## Requisitos Não-funcionais

* ~~Em caso de perda de conexão com internet, o sistema deve armazenar o registro em uma fila e concluir o registro quando recuperar a conexão;~~ (Para possível versão mobile)
* O sistema deve ter um tempo de resposta inferior a dois segundos.

# Arquitetura e Design do Sistema

Arquitetura: MVC;

Linguagens de programação: C# e TypeScript;

Banco de dados: SQL Server

Hospedagem de código: GitHub;

Controle de versão: Git;

Framework: .NET 8 e Bootstrap;

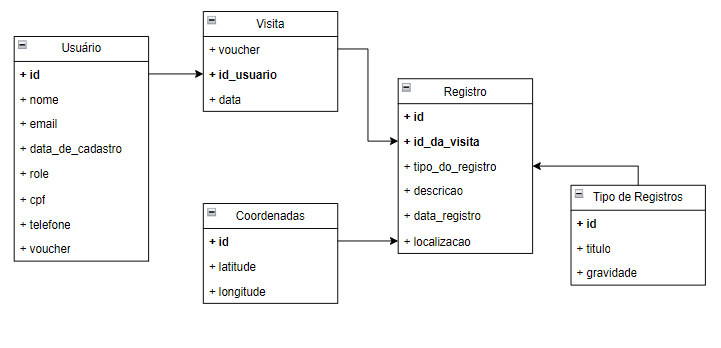
Biblioteca: React.js;

API’s externas: Google OAuth para realizar a autenticação com conta google; Maps JavaScript API para exibição e utilização do mapa; avaliar a possibilidade de implementar uma API de previsão do tempo.

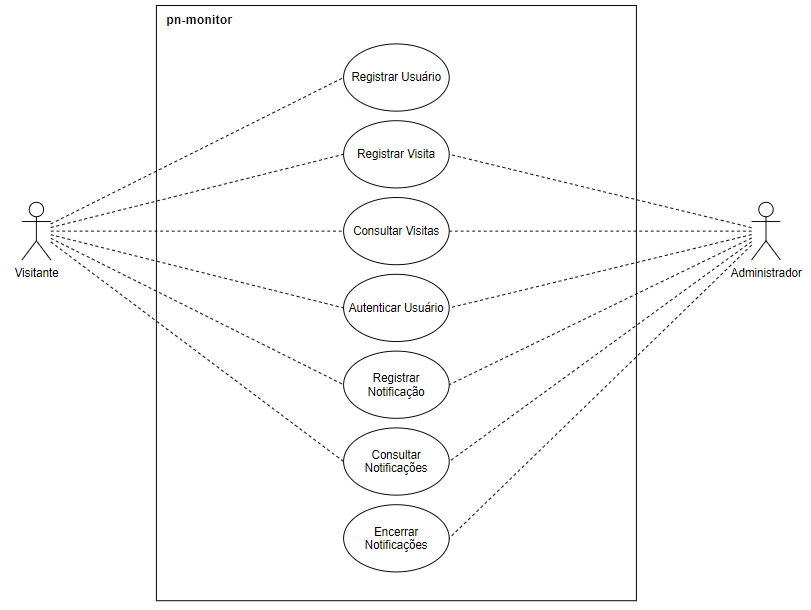
# Modelagem de Dados

## Diagramas de Entidade-Relacionamento

## Diagrama de Classe



## Diagrama de Caso de Uso



## Descrição dos Casos de Uso

### Caso de Uso: Registrar Usuário

**Objetivo:**

Permitir que os usuários se registrem no sistema, criando uma conta com suas informações pessoais.

**Ator Principal:**

Novo usuário

**Pré-condições:**

* O dispositivo do usuário deve ter acesso à internet.

**Fluxo Principal:**

* O novo usuário acessa a página de registro no sistema.
* O sistema exibe um formulário de registro.
* O usuário preenche os campos obrigatórios do formulário, incluindo:
  + Nome completo
  + Endereço de e-mail
  + Nome de usuário
  + Senha
  + Confirmação de senha
* O usuário envia o formulário
* O sistema valida as informações fornecidas, verificando se o e-mail e o nome de usuário já não estão em uso.
* O sistema ativa a conta do usuário e confirma que o registro foi concluído com sucesso.

**Fluxos Alternativos:**

**A1:** O usuário não preenche todos os campos obrigatórios.

* O sistema exibe uma mensagem de erro solicitando o preenchimento dos campos obrigatórios.

**A2:** O e-mail do usuário já está em uso.

* O sistema exibe uma mensagem de erro informando que o e-mail já está em uso e solicita revisão.
* O usuário corrige a localização e reenvia o formulário.

**Pós-condições:**

* A conta do usuário é criada com sucesso.
* O usuário pode se autenticar utilizando suas novas credenciais.

### Caso de Uso: Registrar Visita

**Objetivo:**

Permitir que os usuários registrem suas visitas ao parque nacional incluindo detalhes como data, hora e voucher da visita.

**Ator Principal:**

Usuário do Sistema (visitante, guarda-parque, pesquisador etc.)

**Pré-condições:**

* O usuário deve estar autenticado no sistema.
* O dispositivo do usuário deve ter acesso à internet.

**Fluxo Principal:**

* O usuário acessa a funcionalidade de registro de visitas no sistema.
* O sistema exibe um formulário para o registro da visita.
* O usuário preenche os campos obrigatórios do formulário, incluindo:
  + Data da visita
  + Hora início e término da visita
  + Voucher fornecido pela equipe responsável.
* O usuário envia o formulário.
* O sistema valida as informações fornecidas.
* O sistema registra a visita no banco de dados.
* O sistema confirma ao usuário que a visita foi registrada com sucesso.

**Fluxos Alternativos:**

**A1:** O usuário não preenche todos os campos obrigatórios.

* O sistema exibe uma mensagem de erro solicitando o preenchimento dos campos obrigatórios.

**A2:** O usuário não consegue validar a localização fornecida.

* O sistema exibe uma mensagem de erro solicitando a correção da localização
* O usuário corrige a localização e reenvia o formulário.

**Pós-condições:**

* A visita é registrada no banco de dados.
* O usuário pode visualizar o histórico de suas visitas no sistema.

### Caso de Uso: Consultar Visitas

**Objetivo:**

Permitir que os usuários consultem o histórico de suas visitas ao parque nacional, visualizando detalhes como data, hora e localização de cada visita.

**Ator Principal:**

Usuário do Sistema (visitante, guarda-parque, pesquisador etc.)

**Pré-condições:**

* O usuário deve estar autenticado no sistema.
* O dispositivo do usuário deve ter acesso à internet.

**Fluxo Principal:**

* O usuário acessa a funcionalidade de consulta de visitas no sistema.
* O sistema exibe uma lista de visitas registradas pelo usuário, com filtros opcionais para data e localização.
* O usuário seleciona os filtros desejados e aplica a busca.
* O sistema exibe os resultados da busca, mostrando uma lista de visitas que atendem aos critérios selecionados.
* O usuário seleciona uma visita específica para visualizar detalhes.
* O sistema exibe os detalhes da visita, incluindo:
  + Data da visita
  + Hora de início e término da visita
  + Localização da visita no mapa
* O usuário pode optar por visualizar a localização da visita em um mapa interativo.
* O sistema exibe a localização exata da visita no mapa.
* O usuário pode retornar à lista de visitas ou aplicar novos filtros para outra consulta.

**Fluxos Alternativos:**

**A1:** O usuário não encontra visitas que atendam aos critérios selecionados.

* O sistema exibe uma mensagem informando que não há visitas correspondentes aos filtros aplicados.
* O usuário pode ajustar os filtros e tentar novamente.

**Pós-condições:**

* As visitas são exibidas conforme os critérios de busca aplicados.
* A localização das visitas é visualizada no mapa interativo.

### Caso de Uso: Registrar Notificação

**Objetivo:**

Permitir que os usuários registrem notificações sobre eventos ou incidentes ocorridos no parque nacional e visualizem a localização dessas notificações em um mapa.

**Ator Principal:**

Usuário do Sistema (visitante, guarda-parque, pesquisador etc.)

**Pré-condições:**

* O usuário deve estar autenticado no sistema.
* O dispositivo do usuário deve ter acesso à internet.

**Fluxo Principal:**

* O usuário acessa a funcionalidade de registro de notificações no sistema.
* O sistema exibe um formulário para o registro da notificação.
* O usuário preenche os campos obrigatórios do formulário, incluindo:
  + Tipo de notificação (ex.: avistamento de animal, incêndio, vandalismo etc.)
  + Descrição detalhada do evento ou incidente.
  + Data e hora do ocorrido.
  + Localização (o sistema deve capturar a localização atual do usuário).
* O usuário envia o formulário.
* O sistema valida as informações fornecidas.
* O sistema registra a notificação no banco de dados.
* O sistema exibe a notificação no mapa, indicando a localização exata do evento ou incidente.
* O sistema confirma ao usuário que a notificação foi registrada com sucesso.

**Fluxos Alternativos:**

**A1:** O usuário não preenche todos os campos obrigatórios.

* O sistema exibe uma mensagem de erro solicitando o preenchimento dos campos obrigatórios.

**A2:** O usuário não consegue validar a localização fornecida.

* O sistema exibe uma mensagem de erro solicitando a correção da localização
* O usuário corrige a localização e reenvia o formulário.

**Pós-condições:**

* A notificação é registrada no banco de dados.
* A notificação é exibida no mapa com a localização correta.

### Caso de Uso: Consultar Notificações

**Objetivo:**

Permitir que os usuários consultem notificações registradas sobre eventos ou incidentes ocorridos no parque nacional e visualizem a localização dessas notificações em um mapa.

**Ator Principal:**

Usuário do Sistema (visitante, guarda-parque, pesquisador etc.)

**Pré-condições:**

* O usuário deve estar autenticado no sistema.
* O dispositivo do usuário deve ter acesso à internet.

**Fluxo Principal:**

* O usuário acessa a funcionalidade de consulta de notificações no sistema.
* O sistema exibe uma lista de notificações registradas, com filtros opcionais para tipo de notificação.
* O usuário seleciona os filtros desejados e aplica a busca.
* O sistema exibe os resultados da busca, mostrando uma lista de notificações que atendem aos critérios selecionados.
* O usuário seleciona uma notificação específica para visualizar detalhes.
* O sistema exibe os detalhes da notificação, incluindo:
  + Tipo de notificação,
  + Descrição detalhada do evento ou incidente,
  + Data e hora do ocorrido,
  + Localização no mapa.

O usuário pode optar por visualizar a localização da notificação em um mapa interativo.

O sistema exibe a localização exata da notificação no mapa.

O usuário pode retornar à lista de notificações ou aplicar novos filtros para outra consulta.

**Fluxos Alternativos:**

**A1:** O usuário não encontra notificações que atendam aos critérios selecionados.

* O sistema exibe uma mensagem informando que não há notificações correspondentes aos filtros aplicados.
* O usuário pode ajustar os filtros e tentar novamente.

**Pós-condições:**

* As notificações são exibidas conforme os critérios de busca aplicados.
* A localização das notificações é visualizada no mapa interativo.

### Caso de Uso: Encerrar Notificações

**Objetivo:**

Permitir que os usuários encerrem registros de notificações após a resolução de eventos ou incidentes no parque nacional.

**Ator Principal:**

Usuário do Sistema (visitante, guarda-parque, pesquisador etc.)

**Pré-condições:**

* O usuário deve estar autenticado no sistema.
* O dispositivo do usuário deve ter acesso à internet.
* Deve existir um registro de notificação previamente registrado.

**Fluxo Principal:**

* O usuário acessa a funcionalidade de gerenciamento de notificações no sistema.
* O sistema exibe uma lista de notificações registradas pelo usuário.
* O usuário seleciona a notificação que deseja encerrar.
* O sistema exibe os detalhes da notificação selecionada.
* O usuário clica na opção para encerrar a notificação.
* O sistema solicita a confirmação do encerramento da notificação.
* O usuário confirma o encerramento.
* O sistema registra a data e hora do encerramento e atualiza o status da notificação para ‘Encerrada’.
* O sistema confirma ao usuário que a notificação foi encerrada com sucesso.

**Fluxos Alternativos:**

**A1:** O usuário decide não encerrar a notificação após visualizar os detalhes.

* O usuário retorna à lista de notificações sem fazer alterações.

**Pós-condições:**

* A notificação é atualizada no banco de dados com o status “Encerrada”.
* A data e hora do encerramento são registradas.
* A notificação não aparece mais na lista de notificações ativas.

### Caso de Uso: Autenticar Usuário

**Objetivo:**

Permitir que os usuários se autentiquem no sistema utilizando login e senha registrados na plataforma ou através de uma conta Google.

**Ator Principal:**

Usuário do Sistema (visitante, guarda-parque, pesquisador etc.)

**Pré-condições:**

* O dispositivo do usuário deve ter acesso à internet.
* O usuário deve ter uma conta registrada na plataforma ou uma conta Google válida.

**Fluxo Principal:**

* O usuário acessa a página de login do sistema.
* O sistema exibe duas opções de autenticação:
  + Login com usuário e senha
  + Login com conta Google
* O usuário seleciona a opção desejada:
  + Login com usuário e senha:
    - O usuário insere seu nome de usuário e senha.
    - O sistema valida as credenciais fornecidas.
    - Se as credenciais forem válidas, o sistema autentica o usuário e redireciona para a página inicial.
  + Login com conta Google:
    - O usuário clica no botão “Login com Google”.
    - O sistema redireciona o usuário para a página de autenticação do Google.
    - O usuário insere suas credenciais do Google e autoriza o acesso.
    - O sistema recebe a confirmação do Google e autentica o usuário, redirecionando para a página inicial.

**Fluxos Alternativos:**

**A1:** As credenciais de login com usuário e senha são inválidas.

* O sistema exibe uma mensagem de erro informando que as credenciais estão incorretas.
* O usuário pode tentar novamente ou optar por autenticar com a conta Google.

**A2:** O usuário cancela a autenticação com a conta Google.

* O sistema retorna à página de login, permitindo que o usuário escolha outra forma de autenticação.

**Pós-condições:**

* O usuário é autenticado no sistema e tem acesso às funcionalidades disponíveis.
* O sistema mantém um registro da última autenticação do usuário.

# Rotas da Aplicação

|  |  |  |
| --- | --- | --- |
| Usuario | | |
| UserController | **GET** | /login |
| **GET** | /register |
| **POST** | /login |
| **POST** | /register |
| **GET** | /user |
| Notificacoes | | |
| NotificationController | **POST** | /notificacao |
| **GET** | /notificacoes |
| **DELETE** | /notificacao/{notificacaoID} |
| **GET** | /notificacao/{notificacaoID} |
| **PUT** | /notificacao/{notificacaoID} |
| Visita | | |
| VisityController | **POST** | /visita |
| **GET** | /visitas |
| **DELETE** | /visita/{visitaID} |
| **GET** | /visita/{visitaID} |
| **PUT** | /visita/{visitaID} |
| Tipo de Registro | | |
| RegisterTypeController | **POST** | /tipo\_registro |
| **GET** | /tipo\_registros |
| **DELETE** | /tipo\_registro/{tpRegistroID} |
| **GET** | /tipo\_registro/{tpRegistroID} |
| **PUT** | /tipo\_registro/{tpRegistroID} |
| Coordenada | | |
| CoordinateController | **POST** | /coordenada |
| **GET** | /coordenadas |
| **DELETE** | /coordenada/{coordenadaID} |
| **GET** | /coordenada/{coordenadaID} |
| **PUT** | /coordenada/{coordenadaID} |

# Permissões

|  |  |  |
| --- | --- | --- |
| **Função** | **Visitante** | **Administrador** |
| Registrar Usuário | X | X |
| Registrar Visita | X | X |
| Consultar Visitas | X | X |
| Autenticar Usuário | X | X |
| Registrar Notificação | X | X |
| Consultar Registros | X | X |
| Encerrar Registro |  | X |

# Interface de Usuário

## Wire frames e Protótipos

## Padrões de Design

## Acessibilidade e Usabilidade

# Banco de dados

## Script de criação do banco

[A preencher]

## Tabelas

# Plano de testes

## Caso de Teste

## Ferramentas de Teste

## Testes de interface:

## Testes unitários: