UNIVERSIDADE DE VILA VELHA
SISTEMA DE INFORMAÇÃO

MATHEUS SOUZA MELO
YAM GASPARINO CORTELETTI

DOCUMENTAÇÃO APP

VILA VELHA
2024

MATHEUS SOUZA MELO
YAM GASPARINO CORTELETTI

DOCUMENTAÇÃO APP

Trabalho do Segundo bimestre da Universidade Vila Velha, como requisito parcial para obtenção de nota para aprovação na disciplina.
Professor: Wagner de Andrade Perin .

VILA VELHA
2024

.

1 INTRODUÇÃO

O **Aero Reff** é um aplicativo desenvolvido com o intuito de fornecer informações detalhadas sobre a vida marinha e as condições das praias ao redor do mundo. Utilizando **React Native**, uma das tecnologias mais populares para desenvolvimento de aplicativos móveis, o Aero Reff oferece uma experiência multiplataforma, permitindo que o usuário acesse informações sobre os oceanos Atlântico e Pacífico, como peixes e flora marinha, de maneira interativa e intuitiva.

Além disso, o aplicativo integra dados em tempo real sobre as condições das praias, com informações sobre a altura das ondas, a altura máxima das ondas e a velocidade da correnteza. Esses dados são fornecidos pela **Open-Meteo Marine API**, uma API especializada que oferece previsões meteorológicas e marítimas detalhadas, com cobertura global. A API coleta informações de fontes meteorológicas globais e as processa para fornecer dados sobre o clima e as condições do oceano, incluindo variáveis importantes como vento, temperatura da água, altura das ondas e correntes.

Com essa integração, o Aero Reff permite que os usuários acessem informações precisas e atualizadas sobre as condições do mar em várias praias ao redor do mundo, ajudando tanto turistas quanto profissionais do setor a tomarem decisões mais informadas. O uso do **React Native** garante que o aplicativo seja de fácil acesso em dispositivos iOS e Android, oferecendo uma interface amigável e funcionalidades práticas para quem deseja explorar a biodiversidade marinha e planejar suas atividades à beira-mar com segurança

1.1 OBJETIVOS

1.1.1 Geral

Desenvolver um aplicativo mobile que forneça informações detalhadas sobre a fauna e flora marinha nos oceanos Atlântico e Pacífico, além de apresentar dados atualizados sobre as condições das praias ao redor do mundo, como altura das ondas, altura máxima das ondas e velocidade das correntes, utilizando a **Open-Meteo Marine API** para acesso a dados em tempo real.

1.1.2 Específicos

- Integrar dados sobre peixes e flora marinha;
- Utilizar a Open-Meteo Marine API**;**
- Desenvolver uma interface de usuário intuitiva e interativa;
- Oferecer funcionalidades de busca e filtro;
- Promover a educação ambiental.

2 DESCRIÇÃO DA APLICAÇÃO

2.1 COMPONENTE

2.1.1 Cadastro (auth.js)

Este componente é responsável pelo **cadastro de usuários** no aplicativo. Ele usa o **AsyncStorage** para armazenar informações de login, como nome de usuário e senha, de maneira persistente no dispositivo. O fluxo de cadastro inclui:

- **Validação de entrada**: Verifica se o nome de usuário e a senha foram fornecidos antes de tentar registrar o usuário.
- **Armazenamento de dados**: Se o usuário não existir previamente, seus dados (nome de usuário e senha) são salvos no **AsyncStorage**.
- **Erro**: Se houver algum erro, como o nome de usuário já estar registrado ou falhas no armazenamento, uma mensagem de erro é exibida.

Funções principais:

- validateInput: Verifica se os campos de usuário e senha foram preenchidos.
- Cadastro: Salva um novo usuário no **AsyncStorage**.
- Login: Verifica o login do usuário comparando os dados salvos no **AsyncStorage**.
- Logout: Remove as informações do usuário e finaliza a sessão

2.1.2 Filtro

O componente Filtro permite ao usuário aplicar filtros de busca de forma dinâmica. Ele contém um botão que, ao ser pressionado, exibe ou oculta os filtros disponíveis.

- **Exibição condicional:** A visibilidade dos filtros é controlada por um estado (visivel), permitindo que o usuário visualize ou oculte as opções.
- I**nteratividade:** O componente recebe uma lista de filtros e uma função de callback (onSelectFiltro) que é chamada quando um filtro é selecionado.

Funcionalidades:

- Exibe filtros com base em uma lista de filtros fornecida como propriedade (filtros).
- Permite selecionar um filtro específico para realizar uma ação, como buscar por espécies de peixes ou plantas.

2.1.3 Flora

O componente **Flora** exibe informações detalhadas sobre espécies de flora marinha. Ele apresenta dados como o nome, imagem, habitat e uma descrição da planta.

- **Exibição de imagem e texto**: Exibe uma imagem e textos informativos sobre a planta.
- **Dados**: O componente recebe informações como nome, imagem, habitat e uma descrição da flora como propriedades, que são renderizadas na interface.

Funcionalidades:

- Exibe o nome e a imagem da planta.
- Exibe os habitats onde a planta pode ser encontrada.
- Exibe uma descrição sobre a planta.

2.1.4 Oceanos

O componente **Oceanos** exibe informações sobre um oceano específico, como seu nome, descrição, localização geográfica (latitude e longitude) e fuso horário.

- **Exibição de dados oceânicos**: Exibe o nome do oceano, uma breve descrição, a latitude e longitude, e o fuso horário do oceano.
- **Interface visual**: Assim como no componente **Flora**, o componente **Oceanos** também utiliza **Image** para mostrar imagens associadas ao oceano.

Funcionalidades:

- Exibe o nome e descrição do oceano.
- Exibe coordenadas geográficas (latitude e longitude) e o fuso horário do oceano.

2.1.5 Peixes

O componente **Peixes** funciona de forma semelhante ao componente **Flora**, mas para exibir informações sobre espécies de peixes. Ele apresenta o nome, habitat e descrição do peixe, além de sua imagem.

- **Exibição de informações sobre peixes**: Exibe dados sobre espécies de peixes marinhos.
- **Interatividade**: O componente permite exibir as espécies de peixes em detalhes, com foco no habitat e características principais.

Funcionalidades:

- Exibe o nome e a imagem do peixe.
- Exibe o habitat onde o peixe pode ser encontrado.
- Exibe uma descrição detalhada sobre a espécie de peixe.

2.2 TELAS

2.2.1 Tela Cadastro

A tela de **CadastroScreen** permite que o usuário crie uma nova conta, fornecendo um nome de usuário e senha. O código de implementação usa o React e React Native, juntamente com hooks como *useState* e *useEffect* para gerenciar estados e realizar navegação condicional.

No início, o componente usa o hook *useState* para controlar os estados de usuário, senha, erros e o estado de login (logado). O *useEffect* é utilizado para redirecionar o usuário automaticamente para a tela de login após um cadastro bem-sucedido. Se o estado logado for alterado para true, o navigation.replace('Login') é chamado, realizando a navegação.

2.2.2 Tela Login

A tela de **LoginScreen** permite que o usuário entre na plataforma fornecendo suas credenciais. O código segue uma estrutura semelhante à da tela de cadastro, utilizando os hooks *useState* para gerenciar os estados de nome de usuário, senha, erro e login. Após uma tentativa de login bem-sucedida, o *useEffect* é acionado para redirecionar o usuário para a tela inicial (Home).

2.2.3 Tela Home

A tela de **HomeScreen** exibe informações sobre o usuário logado e oferece navegação para outras seções do aplicativo, como Peixes, Praias e Flora. A tela também tem a funcionalidade de logout, permitindo que o usuário saia da plataforma. O código da tela de home inclui um botão de logout que, ao ser pressionado, chama a função Logout, e redireciona o usuário para a tela de login.

2.2.4 Tela Flora

A Tela de Flora exibe uma lista de espécies de plantas, com detalhes sobre cada uma, como nome, características e informações adicionais. O funcionamento da tela segue a mesma lógica das telas de peixes e praias, com o uso de FlatList para exibir a lista de plantas e a navegação para detalhes adicionais ao selecionar um item.

2.2.5 Tela Praias

A tela de Praia permite ao usuário adicionar e visualizar informações sobre praias, como a altura das ondas e a velocidade das correntes. Para isso, ela utiliza duas APIs principais: a **Open Meteo Marine API**, que fornece dados meteorológicos relacionados ao mar, como a altura das ondas, a velocidade das correntes e a altura máxima das ondas; e a **Nominatim OpenStreetMap API**, que busca as coordenadas geográficas de uma praia a partir do nome inserido pelo usuário.

Quando o usuário adiciona o nome de uma praia, a aplicação faz uma requisição para a API do OpenStreetMap para obter as coordenadas (latitude e longitude) e a cidade correspondente à praia. Em seguida, utiliza essas coordenadas para fazer uma nova requisição à Open Meteo Marine API, buscando informações sobre as ondas e

correntes oceânicas. Esses dados são então apresentados em cards, que exibem o nome da praia, a cidade e as métricas relacionadas.

Os dados das praias são armazenados localmente usando **AsyncStorage**, garantindo que as informações sejam preservadas mesmo após o fechamento do aplicativo. A tela também oferece um modal para adicionar novas praias, onde o usuário pode inserir o nome da praia e adicionar um novo card. Durante o processo de requisição das APIs, um indicador de carregamento é mostrado para informar o usuário que a busca está em andamento.

Essa estrutura permite que o usuário crie uma lista personalizada de praias, com informações detalhadas sobre cada uma, sem perder os dados entre as sessões do aplicativo.