



*Prova de Aptidão Profissional*

*Técnico de Gestão e Programação de*  
*Sistemas Informáticos*

*Manual de desenvolvedor*



WECYCLE

**Elaborado por**

Nome: Giulia Nataly Torres

N.º: 2220119      Triénio: 20/23

**Professor orientador**

Carla Macedo

*30 / 06 / 2023*

## *Índice*

1- Introdução.....	3
2 – Arquitetura e Design .....	4
3 – Guia de desenvolvimento.....	4
3.1 – Configuração .....	4
3.2 – Navegação .....	6
3.3 - Ecrãs .....	6

## 1- Introdução

Neste manual, você vai encontrar as informações essenciais para entender e contribuir com o desenvolvimento da aplicação WeCycle.

A WeCycle foi projetada para preencher a lacuna de conhecimento, fornecendo informações relevantes e promovendo uma participação mais efetiva dos utilizadores na prática da reciclagem. Destacamos algumas de suas funcionalidades principais:

- **Feed de notícias sobre a natureza e reciclagem:** Os utilizadores têm acesso a um feed de notícias atualizado, abordando temas relacionados à preservação ambiental, sustentabilidade e práticas de reciclagem. Dessa forma, eles podem se manter informados sobre os últimos avanços, descobertas e campanhas nessa área.
- **Leitura de códigos de barras de embalagens:** A WeCycle permite que os utilizadores façam a leitura do código de barras de uma embalagem e obtenham informações detalhadas sobre sua composição, além de instruções sobre como realizar a reciclagem corretamente. Isso ajuda a evitar a contaminação dos materiais recicláveis e a promover a separação adequada dos resíduos.
- **Mapa com os ecopontos mais próximos:** A aplicação conta com um recurso de localização que permite identificar os ecopontos mais próximos da localização do utilizador. Esse mapa facilita o acesso a pontos de coleta específicos, onde os utilizadores podem descartar corretamente seus resíduos recicláveis, contribuindo para aumentar a taxa de reciclagem.
- **Sistema de conta:** A WeCycle também possui um sistema de conta pessoal, que permite aos utilizadores criar uma conta, excluir sua conta e fazer logout. Cada conta pessoal inclui informações como e-mail, nome e nome de utilizador.

Essas funcionalidades da WeCycle visam enfrentar o desafio da falta de conhecimento sobre a importância da reciclagem. Através da disposição de informações atualizadas, acesso facilitado a pontos de coleta e um sistema que incentiva o engajamento individual, a aplicação busca aumentar a conscientização e a participação dos utilizadores na prática da reciclagem, contribuindo para a redução do impacto ambiental causado pelos resíduos urbanos.

## 2 – Arquitetura e Design

O projeto foi desenvolvido com base na Framework React Native, utilizando JavaScript como linguagem de programação. Para a organização e teste do projeto, foi utilizada a plataforma Expo, que permite a realização de testes diretamente em dispositivos móveis. Para isso, basta conectar-se à mesma rede do dispositivo que contém o projeto e utilizar a aplicação "Expo Go", disponível para sistemas operativos Android e iOS.

No que diz respeito às ferramentas de desenvolvimento, o editor de texto escolhido foi o Visual Studio Code. Por meio dele, é possível clonar o projeto diretamente do GitHub ou transferir o arquivo que contém o projeto. É importante destacar que a instalação do projeto requer a execução do comando "npm i" para garantir a instalação correta de todas as dependências, juntamente com suas respectivas versões.

O projeto também utiliza o Firebase como base de dados. A Framework é utilizada para armazenar e gerenciar as informações necessárias, como dados de utilizadores, configurações e informações sobre produtos. Ela oferece uma API fácil de utilizar e recursos avançados de desenvolvimento.

O projeto pode ser dividido em três partes principais: configuração, ecrãs e navegação. Cada uma dessas partes desempenha um papel fundamental na estruturação e funcionamento da aplicação. No próximo tópico, vamos explorar em detalhes cada uma delas, fornecendo informações e diretrizes para o desenvolvimento adequado do projeto.

## 3 – Guia de desenvolvimento

Como mencionado no tópico anterior vamos dividir essa secção em três partes principais: configuração, navegação e ecrãs, a começar pela configuração:

### 3.1 – Configuração

As configurações da aplicação são coesas e simplificadas, destaco principalmente que deve-se certificar se possui instalados corretamente os seguintes ficheiros: "package-lock.json" e "package.json" que são ambos responsáveis pelo bom funcionamento da aplicação, é neles que podemos encontrar toda a informações sobre a versão das extensões e pacotes instalados no projeto; também é de suma importância existência do diretório "node\_modules" que carrega os módulos e bibliotecas necessários para o projeto.

Caso ocorra um erro a esses arquivos mencionados, você pode tentar excluí-los, em seguida, utilizar o comando "npm i" para os reinstalá-los novamente conforme o que se encontra no projeto.

Outra configuração de extrema importância é a da base de dados. Essa configuração é realizada por meio do arquivo "firebase.js", onde são definidas informações como a chave de API, o nome da base de dados, entre outras.

Deixo a seguir um exemplo de uma chamada à base de dados, todas as outras utilizações da base de dados seguem o mesmo modelo apresentado (os exemplos apresentados foram retirados do ficheiro “ProductScreen” e tem como intuito retirar a informação da base de dados referente à tabela “composition” que possui o código de barras de um produto, o material que é feito e qual a parte do produto se refere):

```
//Composition
const compositionRef = collection(db, "composition");
const [materialComposition, setMaterialComposition] = useState([]);
```

1-Referencia à base de dados e inicialização do objeto.

```
const getComposition = async () => {
  //composition
  const compositionData = await getDocs(compositionRef);
  setMaterialComposition(
    compositionData.docs.map((doc) => ({ ...doc.data(), id: doc.id })))
  );
};
```

2-Função para percorrer a informação da base de dados e adicioná-la ao objeto.

```
//Use hook and store products
useEffect(() => {
  getProduct();
  getComposition();
  getContainer();
}, []);
```

3-Chamada da função através de um useEffect hook, certificando que a informação é carregada assim que a página é iniciada.

Deixo também em anexo o modelo da base de dados não relacional:

### WeCycle-db

**Coleção:** “composition”

**Documento:** “id” (string)

**Campo:** “codigo\_barras” (string)

**Campo:** “material” (string)

**Campo:** “parte” (string)

**Coleção:** “container”

**Documento:** “id” (string)

**Campo:** “cor” (string)

**Campo:** “material” (string)

**Coleção:** “ecopontos”

**Documento:** “id” (string)

**Campo:** “latitude” (number)

**Campo:** “longitude” (number)

**Coleção:** “products”

**Documento:** “id” (string)

**Campo:** “fabricante” (string)

**Campo:** “img” (string)

**Campo:** “instrucoes” (string)

**Campo:** “link” (string)

**Campo:** “material\_principal” (string)

**Campo:** “nome” (string)

**Campo:** “separar” (boolean)

**Coleção:** “users”

**Documento:** “id” (string)

**Campo:** “email” (string)

**Campo:** “nome” (string)

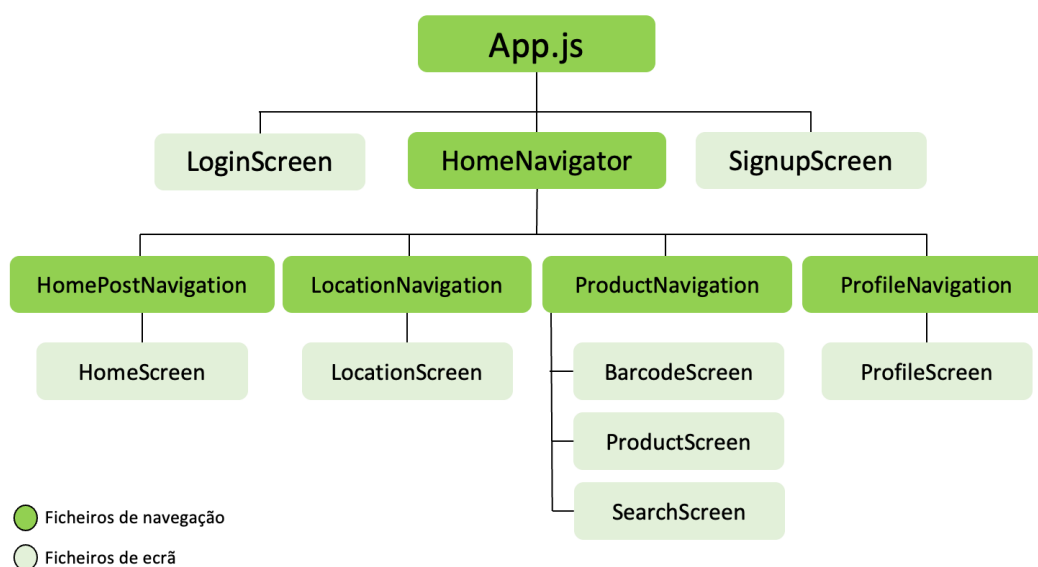
**Campo:** “username” (string)

### 3.2 – Navegação

A navegação desempenha um papel fundamental na experiência do utilizador na aplicação. No projeto em questão, a navegação é gerida a partir do arquivo "app.js". Neste arquivo, são definidos os principais fluxos de navegação, incluindo a transição para o ecrã de login (LoginScreen), o ecrã de registo (SignupScreen) e a navegação principal (HomeNavigator).

O ficheiro "app.js" serve como ponto de partida para a navegação dentro da aplicação. A partir dele, é possível direcionar o utilizador para as diferentes partes da aplicação, garantindo uma experiência de utilização fluída e intuitiva.

É a partir do ficheiro "HomeNavigator" que podemos aceder aos outros ficheiros de navegação que levam aos determinados ecrãs, cada ecrã possui o seu próprio navegador, como apresentado no modelo a seguir:



Além dos ficheiros de ecrã individuais, vale ressaltar a presença da chamada de dois ficheiros dentro do 'HomeNavigation': 'TabBarButton.js' e 'BottomTabBarButton'. O ficheiro 'TabBarButton.js' é responsável pela barra de navegação entre os ecrãs, permitindo ao utilizador alternar facilmente entre as diferentes secções da aplicação. Já o ficheiro 'BottomTabBarButton' é responsável pela estilização da barra de navegação, garantindo uma visualização atraente e coerente em toda a aplicação.

### 3.3 - Ecrãs

Neste seguimento, vamos explorar os diferentes ficheiros de ecrã presentes na aplicação WeCycle, destacando a função principal de cada um deles. Esses ficheiros desempenham um papel fundamental na interação do utilizador com a aplicação, fornecendo interfaces visuais e funcionalidades específicas para cada parte do fluxo de utilização.

- **LoginScreen:** Esta página é responsável pelo processo de login do utilizador. A autenticação é realizada através de uma função pré-definida da Firebase

chamada 'signInWithEmailAndPassword'. O utilizador insere o seu e-mail e senha para fazer login na aplicação.

- **SignupScreen:** Nesta página, os utilizadores podem criar uma conta. São solicitados o nome, nome de utilizador, e-mail e senha. Essas informações são armazenadas na autenticação da Firebase e também na coleção "users". A senha é criptografada para garantir a segurança dos dados.
- **HomeScreen:** A página HomeScreen é destinada à visualização de notícias relacionadas à natureza. As notícias são obtidas através de uma chamada HTTP feita pela biblioteca Axios, utilizando a API "http://servicodados.ibge.gov.br/api/v3/noticias/?qtd=20".
- **LocationScreen:** Esta página solicita permissão para aceder a localização do utilizador e exibe o ponto de recolha mais próximo com base nessa localização. As funções utilizadas são fornecidas pela biblioteca "expo-location", e a exibição do mapa é feita pela biblioteca "react-native-maps".
- **BarcodeScreen:** Nesta página, o utilizador é solicitado a permitir o acesso à câmara do dispositivo e pode fazer a leitura do código de barras de um produto. Essa funcionalidade é possível graças à biblioteca "expo-barcode-scanner".
- **ProductScreen:** A página ProductScreen permite buscar e exibir informações sobre um determinado produto com base no código de barras lido anteriormente. As informações são obtidas a partir das informações na base de dados.
- **SearchScreen:** A página SearchScreen é utilizada para pesquisar produtos. Ela permite filtrar e exibir as opções de produtos com base nos critérios de pesquisa fornecidos pelo utilizador.
- **ProfileScreen:** A página ProfileScreen exibe as informações do utilizador, como nome, e-mail e outras informações relevantes. A partir desta página, o utilizador pode modificar as suas informações, excluir a sua conta e efetuar logout da aplicação.