



Pokédex DSM

Aplicativo mobile com Expo (React Native + TypeScript)

Feito por: Gabriel Pessoni

Visão Geral

- ◉ Aplicativo no estilo Pokédex
- ◉ Funcionalidades:
 - Buscar Pokémons
 - Visualizar detalhes (stats, tipos, habilidades, movimentos)
 - Filtrar e ordenar
 - Comparar Pokémons lado a lado
 - Capturar e manter coleção pessoal persistida



Estrutura de Pastas



```
src/  
  pages/           # Telas (login, cadastro, pokémon, etc.)  
  routes.tsx       # Navegação stack + modal logout  
  services/api.ts  # Cliente HTTP (PokeAPI)  
  styles.ts        # Estilos (styled-components)  
App.tsx            # Root + Toast provider  
index.ts           # Registro raiz (Expo)
```



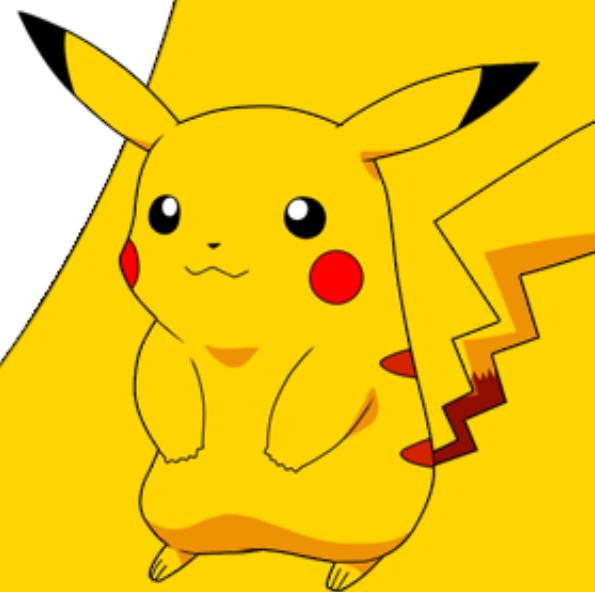
Bibliotecas Utilizadas

- 🍷 Core: React Native + Expo
- 🍷 Navegação: React Navigation Stack
- 🍷 UI: styled-components, @expo/vector-icons
- 🍷 Forms: react-native-mask-text, picker
- 🍷 Storage: AsyncStorage
- 🍷 HTTP: Axios
- 🍷 Feedback: react-native-toast-message



Rotas e Páginas

- 📍 login: autenticação local
- 📍 cadastro: criação de usuário
- 📍 main: Pokédex (busca, filtros, coleção)
- 📍 all-pokemons: listagem ampliada
- 📍 pokemon: detalhes do Pokémon
- 📍 pokemon-moves: movimentos
- 📍 compare-pokemons: comparação



Fluxo de Autenticação

- 🔴 Cadastro (cadastro.tsx):

Usuário preenche: nome, email, senha, CPF, telefone e curso

Validação básica (não permite campos vazios)

Usuário é salvo no AsyncStorage na chave users

- 🔴 Login (login.tsx):

Lê users no AsyncStorage

Busca por email + senha correspondentes

Se sucesso → salva o usuário na chave currentUser

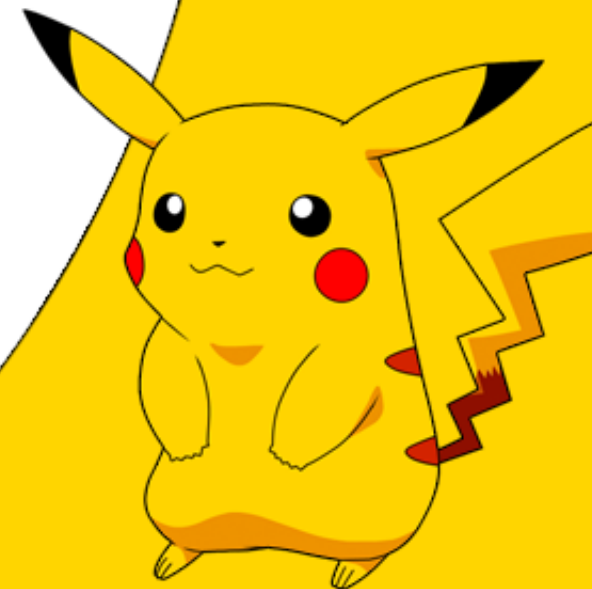
Se falha → exibe erro via Toast

- 🔴 Sessão inicial:

Ao abrir o app, rota inicial é decidida lendo currentUser

Se existir → vai direto para main

Senão → redireciona para login



Persistência de Dados

- 📍 Onde e como os dados ficam salvos:

Utiliza AsyncStorage como banco local (chaves/valores em JSON)

- 📍 Chaves principais:

users → lista de todos os usuários cadastrados

currentUser → usuário logado no momento

pokemons:{email} → coleção de Pokémons de cada usuário (ex.: pokemons:ash@kanto.com)

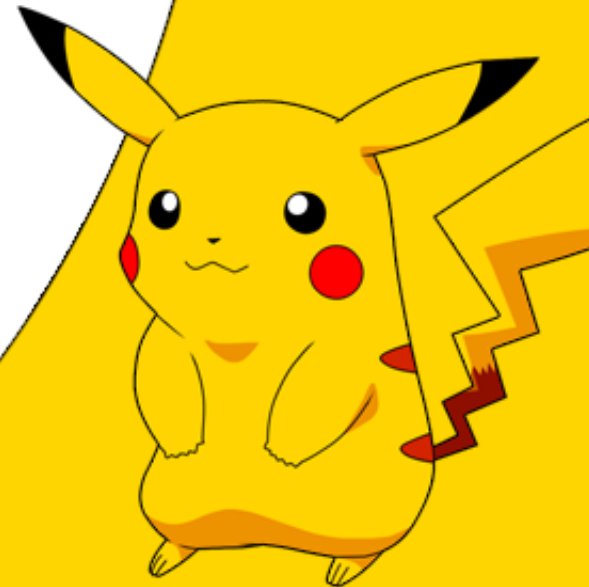
- 📍 Como funciona:

Carregamento (componentDidMount):

Ao abrir a main.tsx, lê currentUser

Monta a chave pokemons:{email}

Carrega a lista de Pokémons salvos para esse usuário e exibe na tela



Persistência de Dados

```
async componentDidMount(): Promise<void> {  
  const currentUserStr = await AsyncStorage.getItem("currentUser");  
  if (!currentUserStr) return;  
  const currentUser = JSON.parse(currentUserStr);  
  const key = `pokemons:${currentUser.email}`;  
  const pokemons = await AsyncStorage.getItem(key);  
  if (pokemons) {  
    const parsedPokemons = JSON.parse(pokemons);  
    this.setState({  
      pokemons: parsedPokemons,  
      filteredPokemons: parsedPokemons  
    });  
  }  
}
```



Demonstração

Da nossa pokédex

