Prof. André Aparecido da Silva



Meu App Complexo

Acesse o site principal https://reactnative.dev

A instalação esta neste link: Set Up Your Environment · React Native

https://reactnative.dev/docs/set-up-your-environment

App um pouco mais Complexo

Um aplicativo considerado um pouco mais complexo para um desenvolvedor iniciante em React Native é aquele que vai além de apresentar textos, imagens e botões. Ele precisa combinar diferentes componentes, controlar estados, permitir navegação entre telas, receber informações do usuário e executar ações conforme as interações.

App um pouco mais Complexo

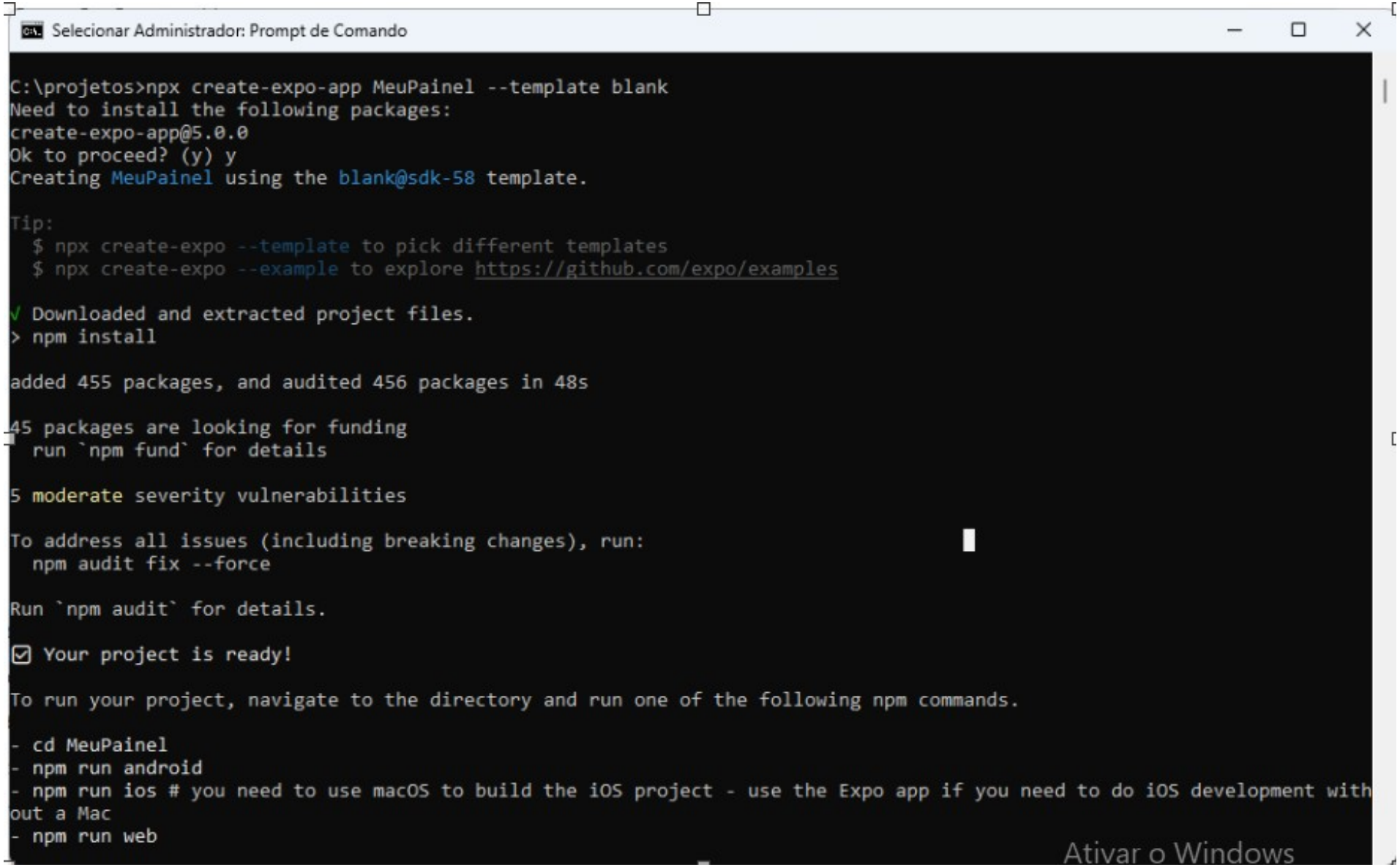
Não precisa ser um aplicativo profissional ou utilizar banco de dados. A ideia é desenvolver um projeto que reúna vários conceitos fundamentais do React Native em uma única aplicação.

Níveis de Complexidade dos Aplicativos

Nível 1 – Básico
Iniciante
→ Uma única tela.
→ View e Text.
→ Image.
→ Button.
→ StyleSheet.
→ TextInput.
→ Poucas interações.

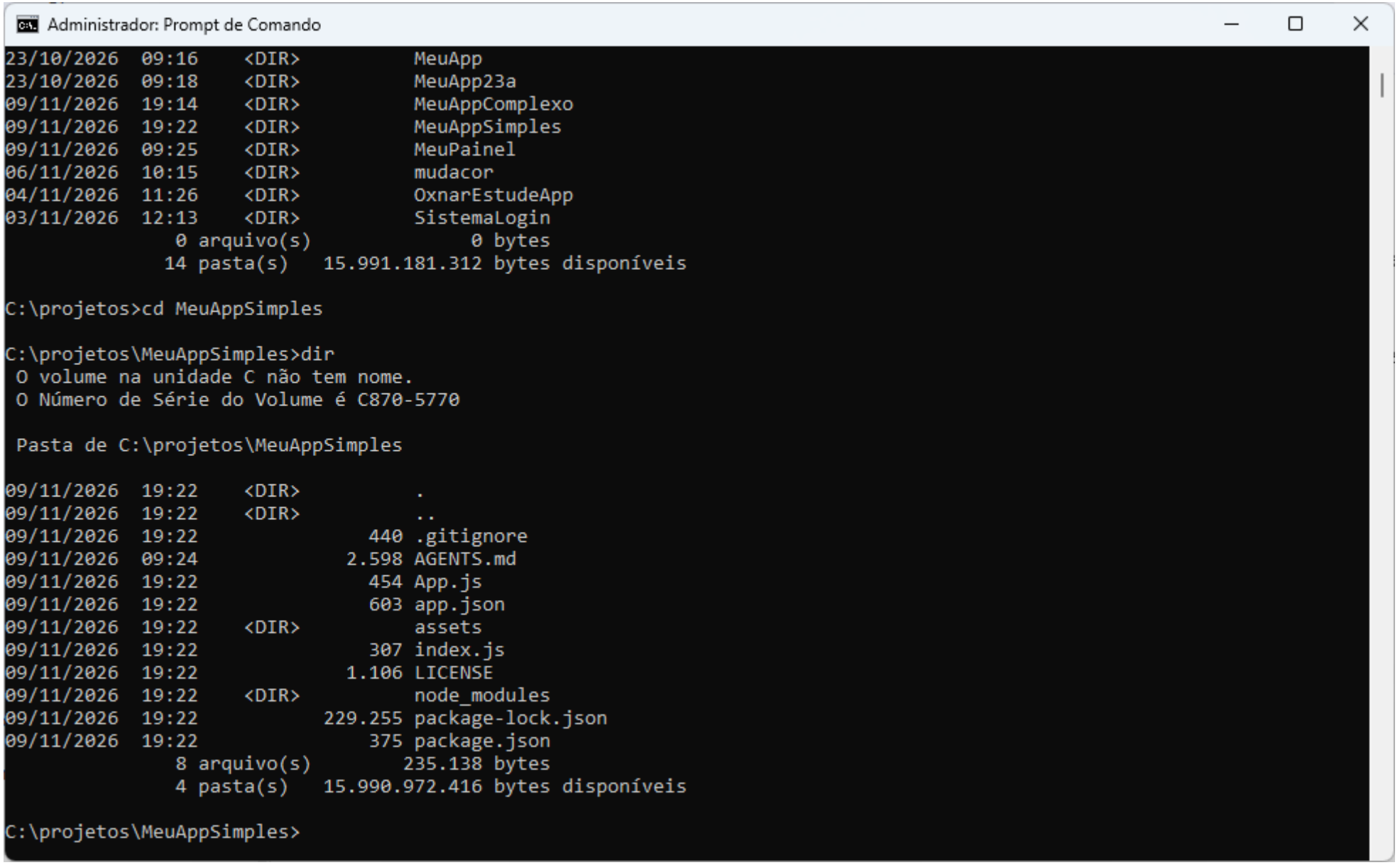
Exemplo App Simples

npx create-expo-app MeuAppSimples --template blank



A estrutura de diretorios

npx create-expo-app MeuAppSimples --template blank



Editando App.js

JavaScript

```
import { View, Text, Image, TextInput, Button, StyleSheet, Alert } from 'react-nativ

export default function App() {
  const [nome, setNome] = useState('');
  const [mensagem, setMensagem] = useState('');

  const handleEnviar = () => {
    if (nome.trim() === '') {
      Alert.alert('Aviso', 'Por favor, digite seu nome!');
      return;
    }
    setMensagem(`Olá, ${nome}! Seja bem-vindo(a).`);
  };

  return (
    <View style={styles.container}>
      {/* Exemplo de Image */}
      <Image
        source={{ uri: 'https://reactnative.dev/img/tiny_logo.pt' }}
        style={styles.logo}
      />

      {/* Exemplo de Text */}
      <Text style={styles.titulo}>Meu Primeiro App</Text>

      {/* Exemplo de TextInput */}
      <TextInput
        style={styles.input}
        placeholder="Digite seu nome..."
        placeholderTextColor="#888"
        value={nome}
        onChangeText={setNome}
      />

      {/* Exemplo de Button */}
      <Button title="Entrar" onPress={handleEnviar} color="#61dafb" />

      {/* Mensagem exibida após a interação */}
      {mensagem !== '' && <Text style={styles.resultado}>{mensagem}</Text>}
    </View>
  );
}
```

Editando App.js (Continuação)

```
// Exemplo de StyleSheet
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#1e1e24',
    alignItems: 'center',
    justifyContent: 'center',
    padding: 20,
  },
  logo: {
    width: 80,
    height: 80,
    marginBottom: 20,
  },
  titulo: {
    fontSize: 24,
    fontWeight: 'bold',
    color: '#ffffff',
    marginBottom: 20,
  },
  input: {
    width: '100%',
    height: 50,
    backgroundColor: '#ffffff',
    borderRadius: 8,
    paddingHorizontal: 15,
    fontSize: 16,
    marginBottom: 15,
    color: '#000000',
  },
  resultado: {
    marginTop: 20,
    fontSize: 18,
    color: '#4caf50',
    fontWeight: '600',
  },
});
```

O App ficou assim

npx expo start

Pressiona w e depois instale os pacotes para web

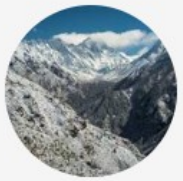
npx expo install react-dom react-native-web

Novamente npx expo start



MeuAppSimples

localhost:8081



André Aparecido da Silva

André Aparecido da Silva

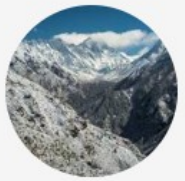
❤️ Gostos: 16

TDM - LIKE

MeuAppSimples

localhost:8081

Chat



Turminha do Mal

Turminha do Mal

❤️ Gostos: 23

TDM - LIKE

Pesquisar

Pesquisar

App um pouco mais Complexo

Nível 2 – Intermediário inicial

→ Um pouco mais complexo

→ Duas a quatro telas.

→ Navegação entre telas.

→ Formulários.

→ useState.

→ FlatList.

→ ScrollView.

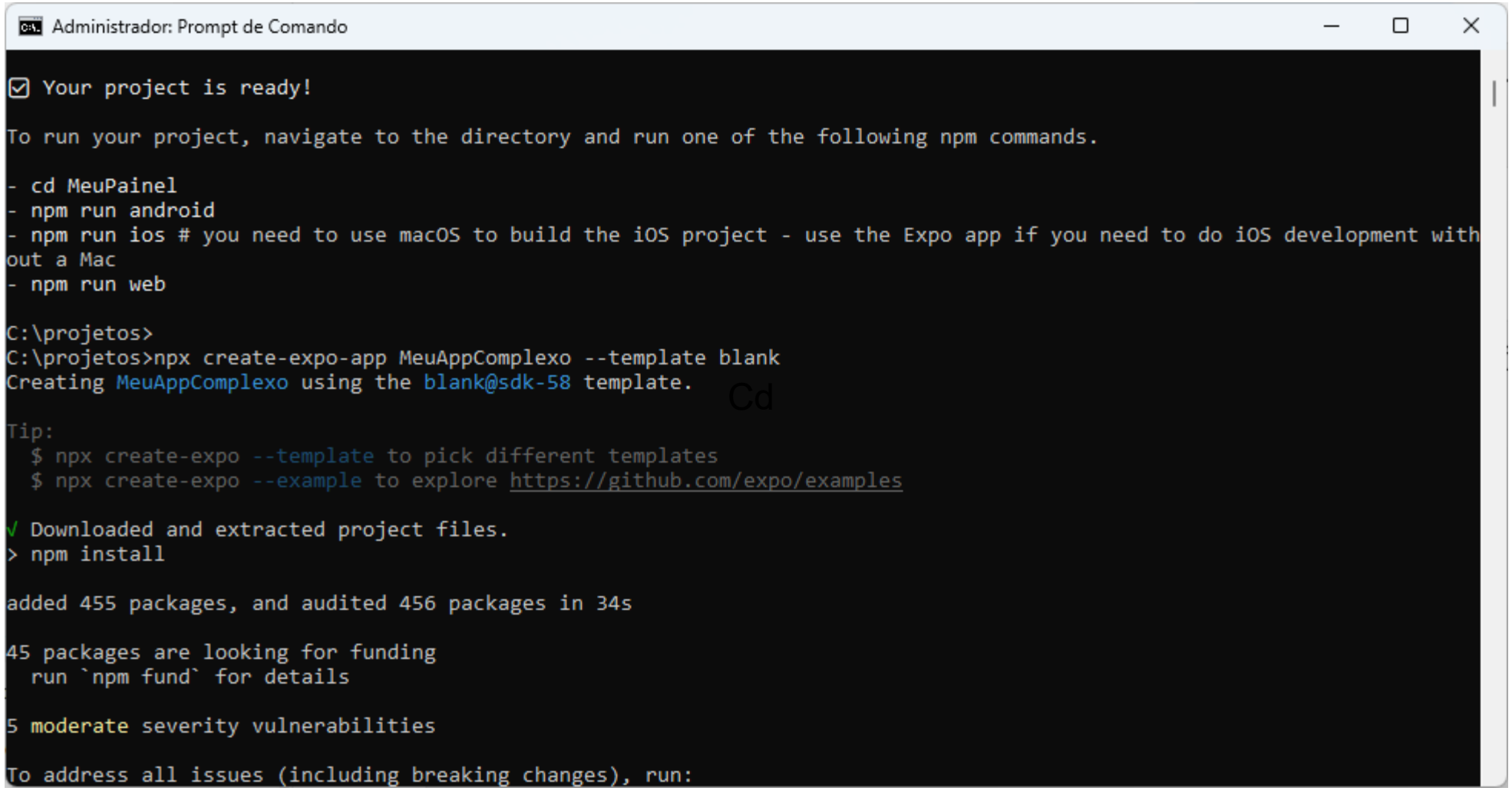
→ Switch e TouchableOpacity.

→ Switch e TouchableOpacity.

→ Validação de informações.

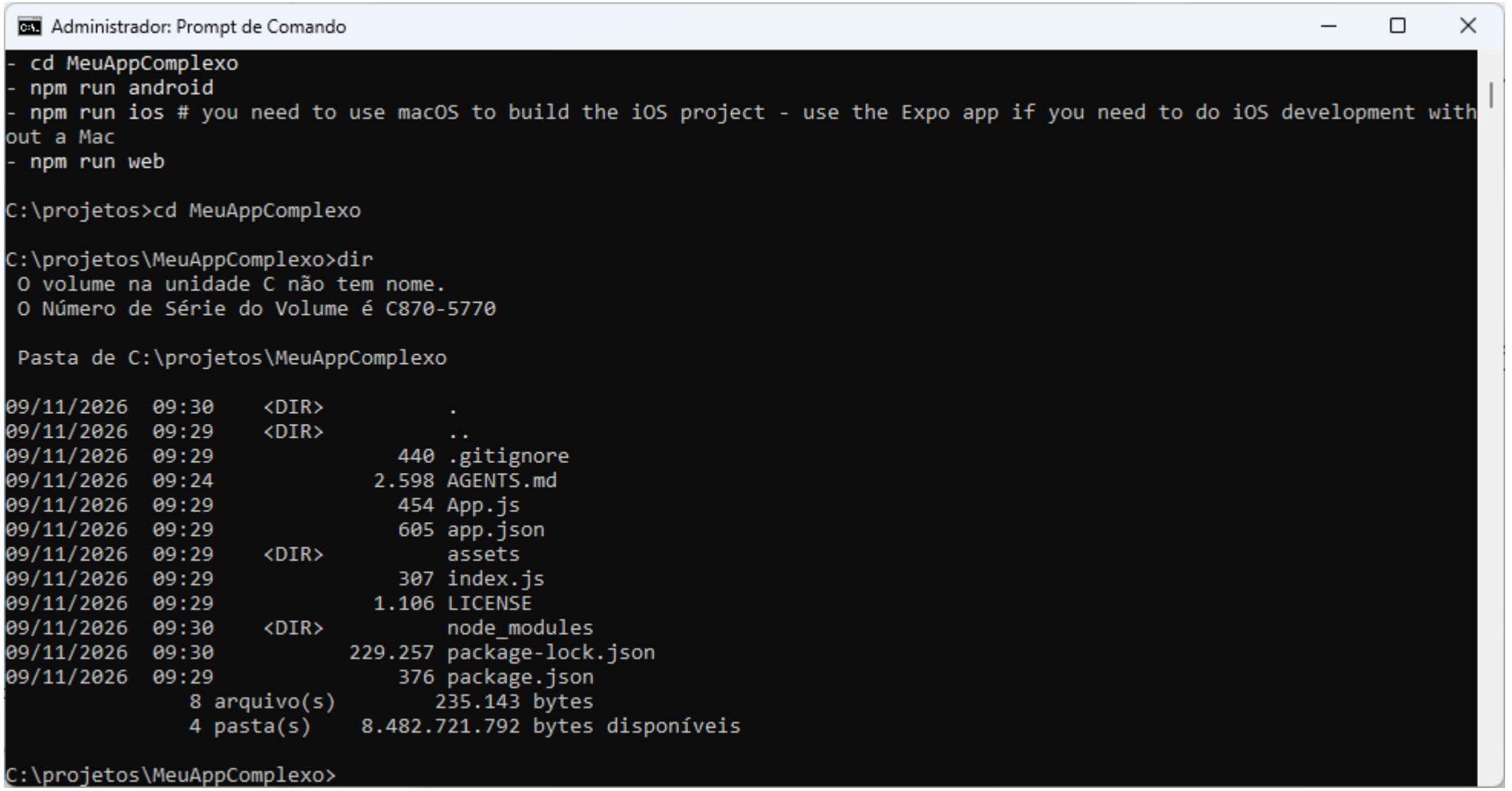
App um pouco mais Complexo

npx create-expo-app MeuAppComplexo --template blank



A estrutura de diretorios foi criada

Novamente iremos editar o arquivo App.js



Executando o App

npx expo start

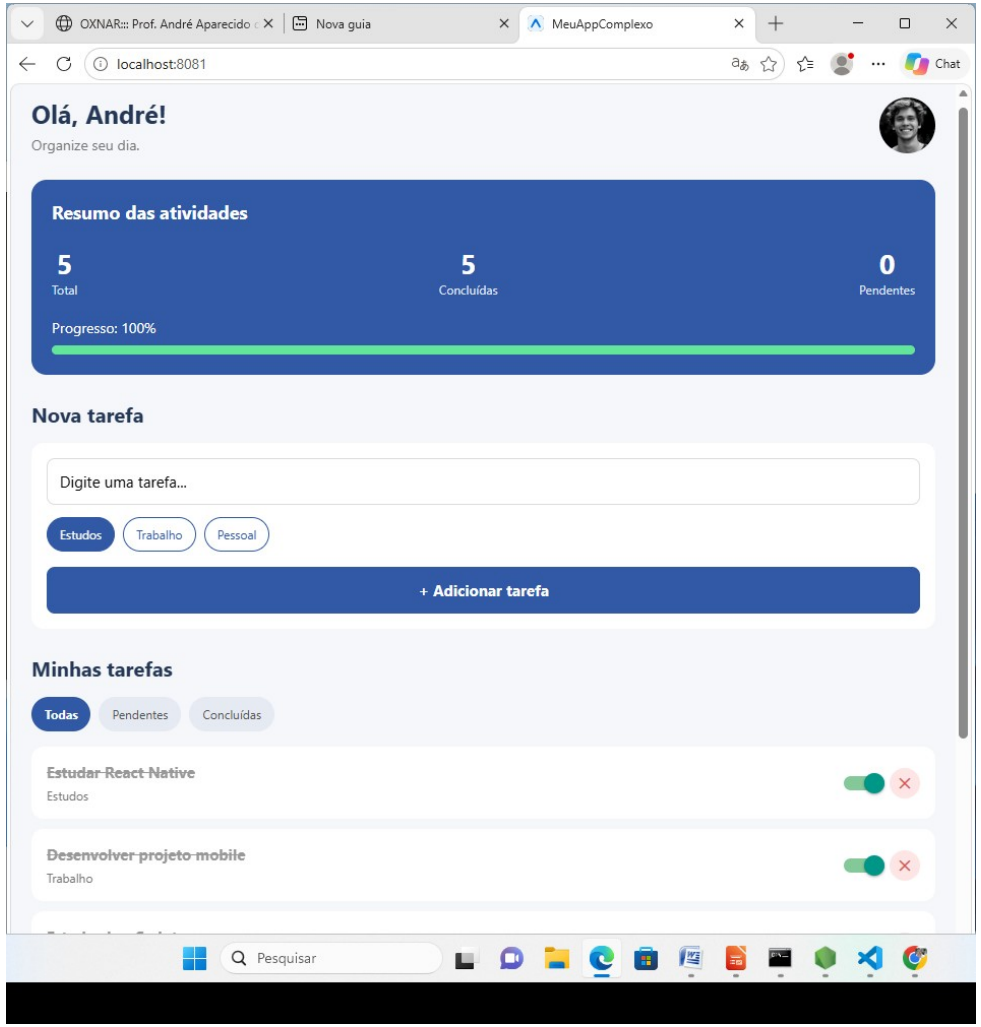
w

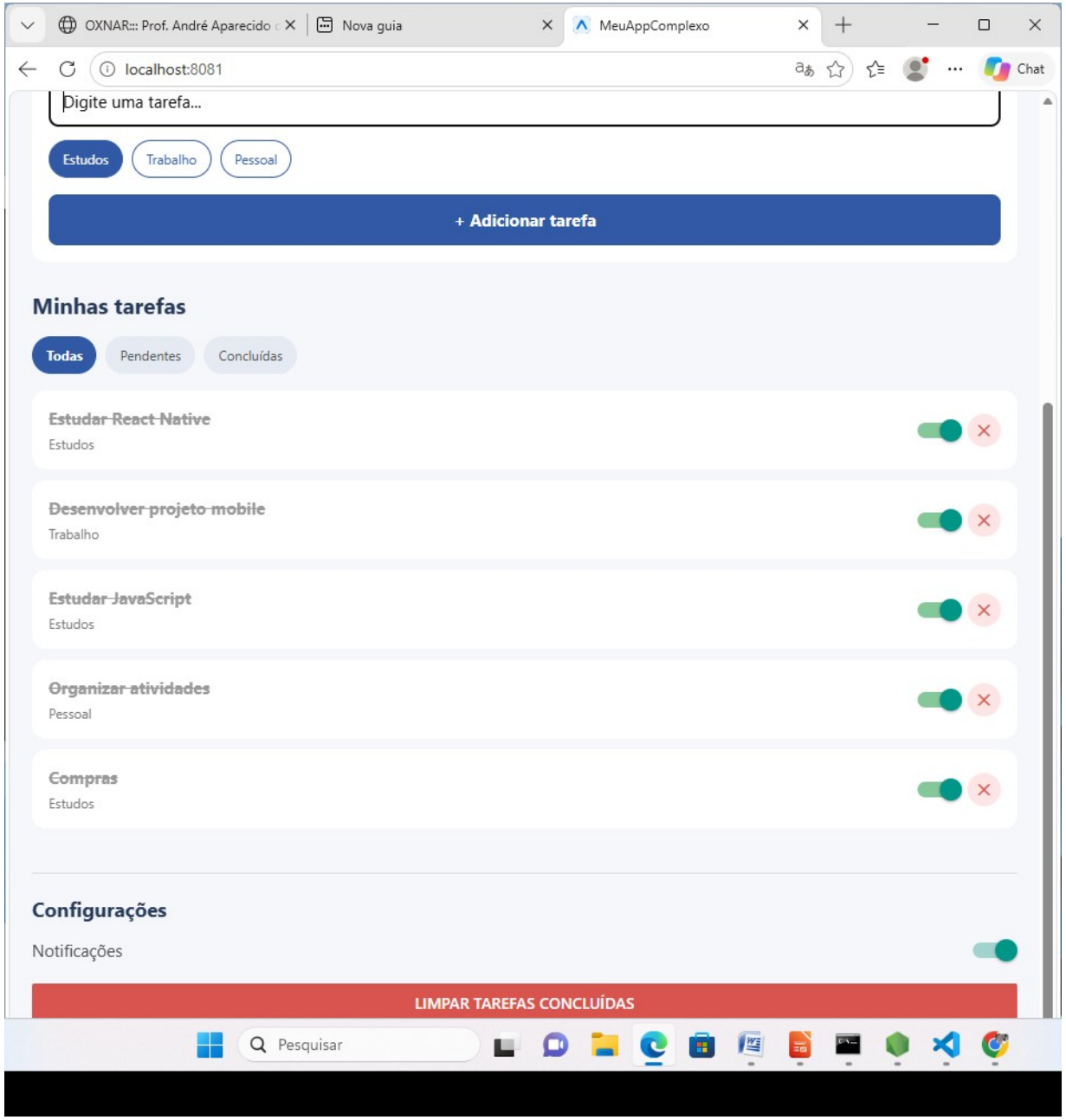
npx expo install react-dom react-native-web

Npx expo start

O código será disponibilizado separado por ser muito grande

Executando o App





Avançando na complexidade

Objetivo: desenvolver um aplicativo para controlar receitas, despesas e saldo financeiro.

Funcionalidades:

Tela inicial com saldo disponível.

Cadastro de receitas.

Cadastro de despesas.

Histórico de movimentações.

Exclusão de movimentações.

Filtro por receita ou despesa.

Cálculo automático do saldo.

Navegação entre telas.

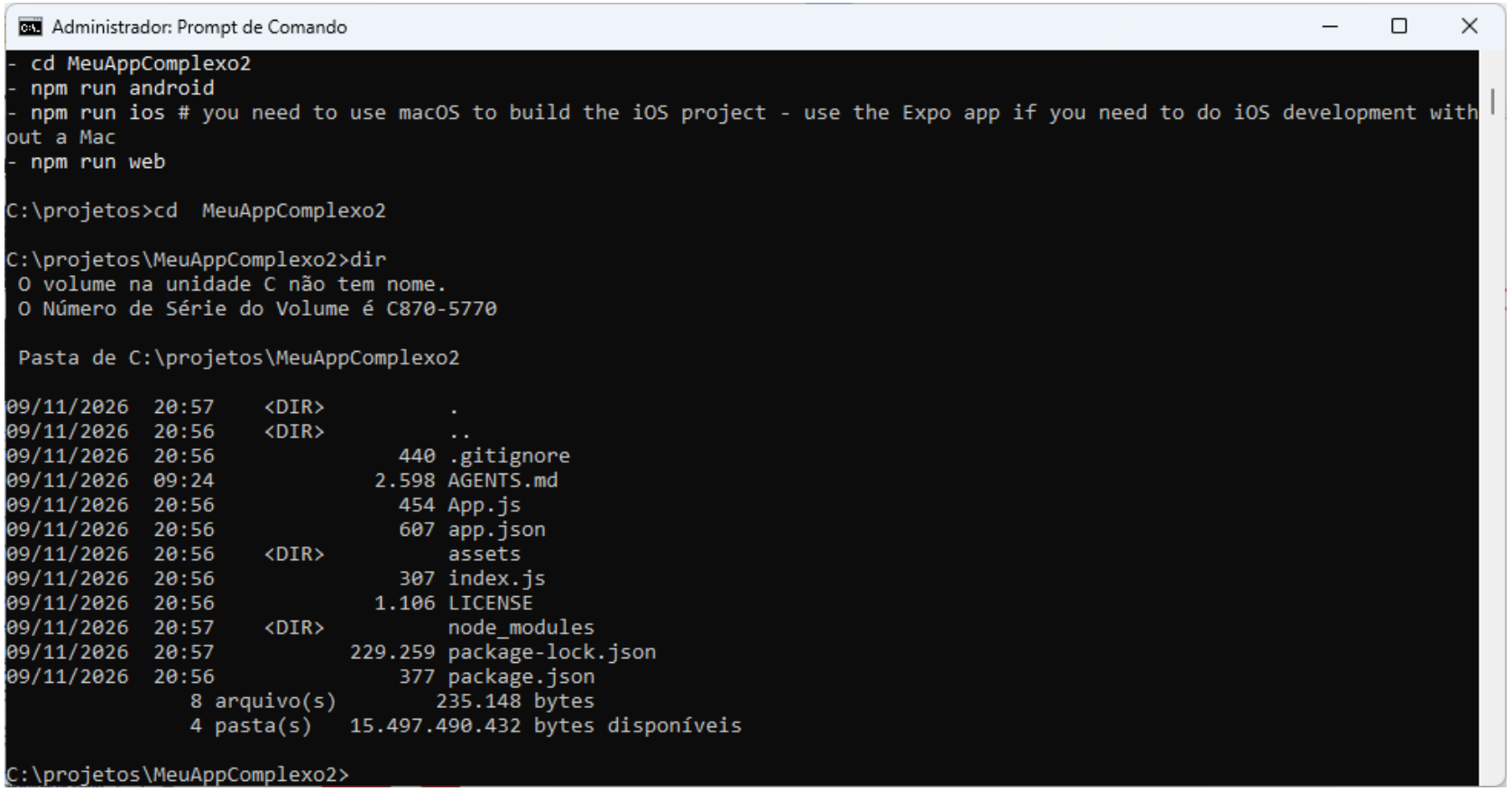
Categorias financeiras.

Resumo financeiro.

Linhas para criação do App

C:\projetos>npx create-expo-app MeuAppComplexo2 --template blank

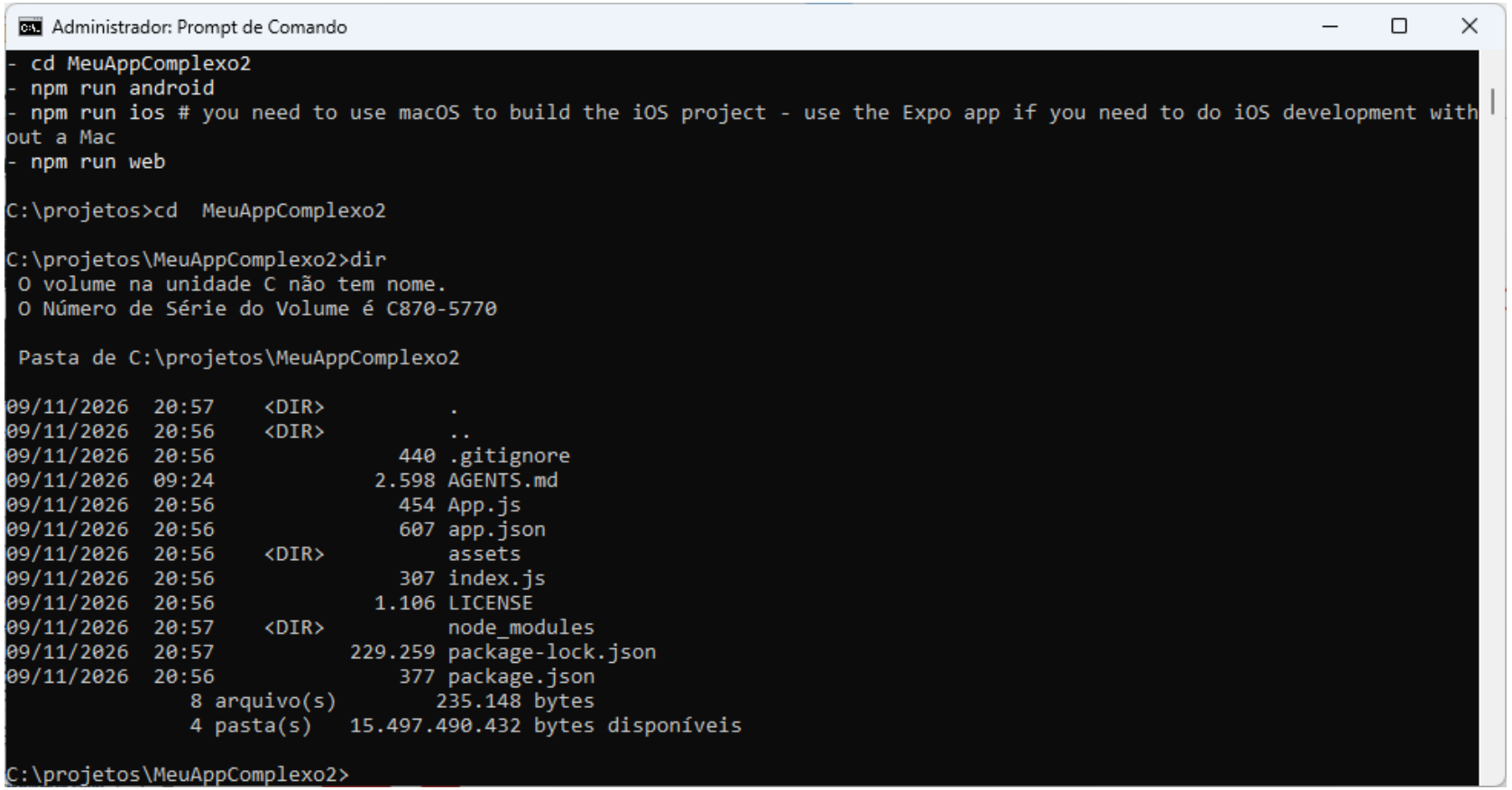
C:\projetos> cd MeuAppComplexo2



O código será disponibilizado a parte

C:\projetos>npx create-expo-app MeuAppComplexo2 --template blank

C:\projetos> cd MeuAppComplexo2



Executando o App

npx expo start

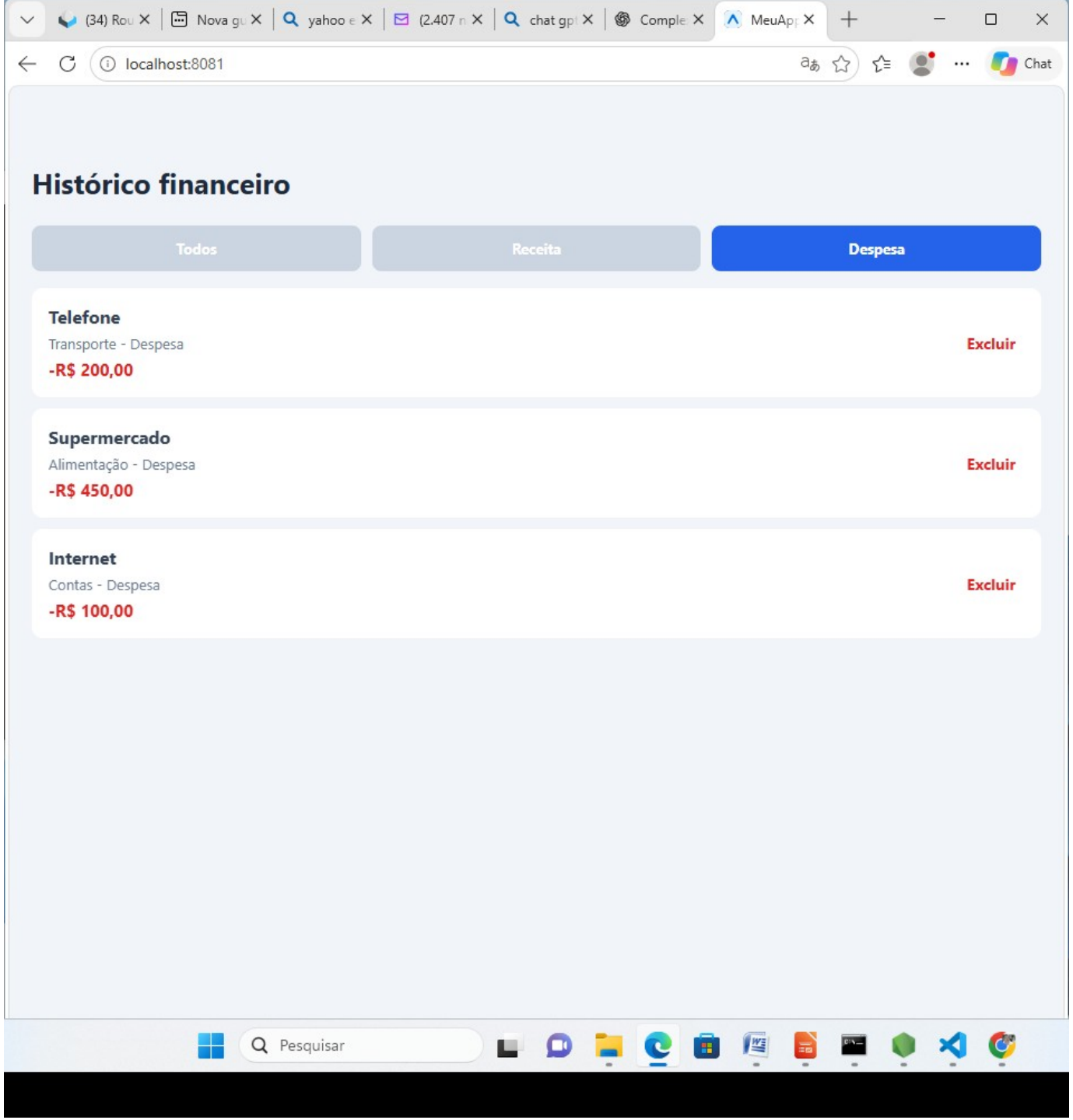
w

npx expo install react-dom react-native-web

Npx expo start

O código será disponibilizado separado por ser muito grande

O App3


localhost:8081
Chat
Histórico financeiro
Todos
Receita
Despesa
Telefone
Transporte - Despesa
-R$ 200,00
Excluir
Supermercado
Alimentação - Despesa
-R$ 450,00
Excluir
Internet
Contas - Despesa
-R$ 100,00
Excluir
Pesquisar

Um Jogo de campo minado

c:\projetos\MeuAppComplexo2>npx create-expo-app MeuAppCampoMinado --template blank

npx expo start

W

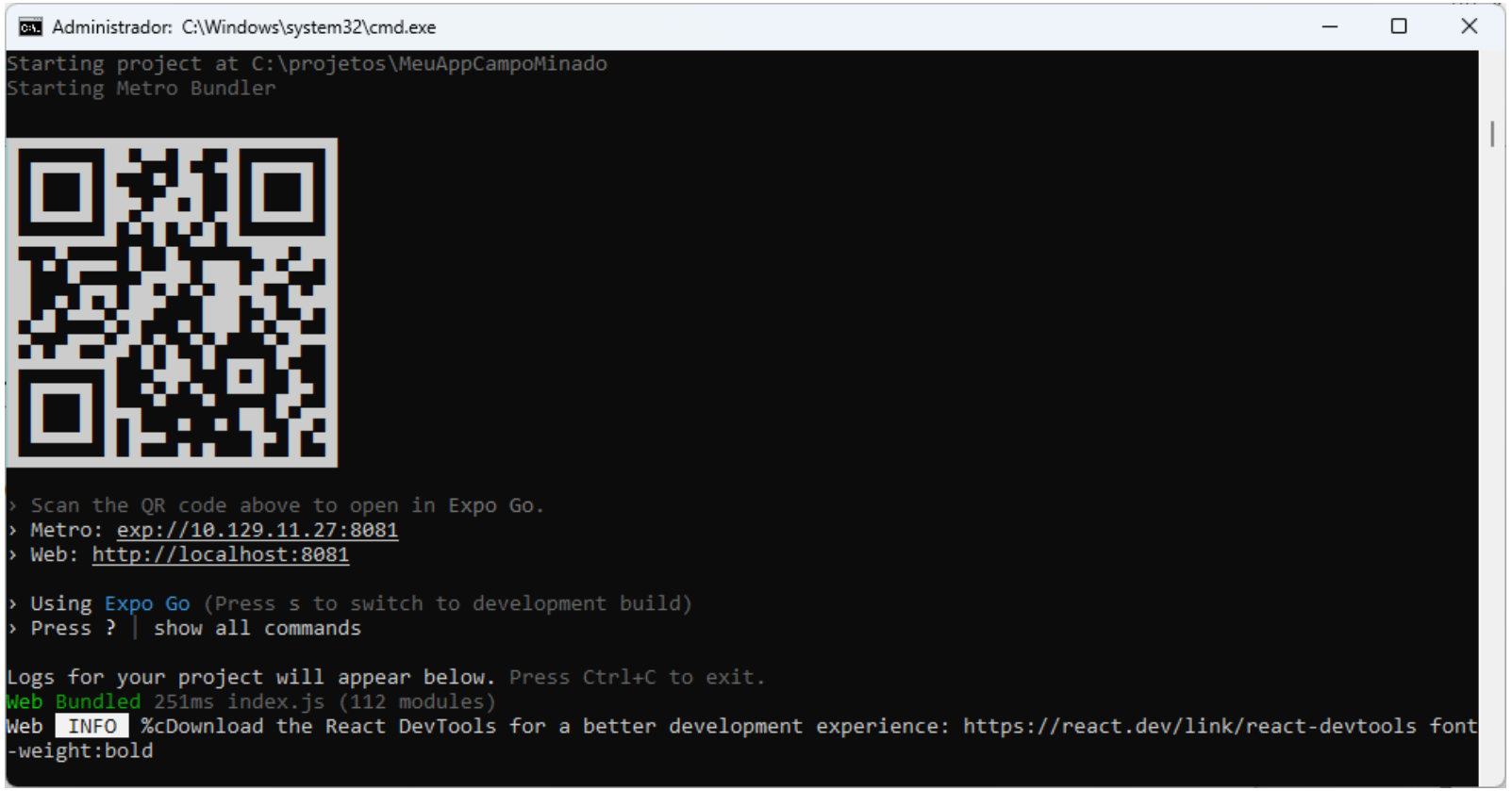
npx expo install react-dom react-native-web

(ATENÇÃO: ESTE APP FOI CRIADO UTILIZANDO IA)

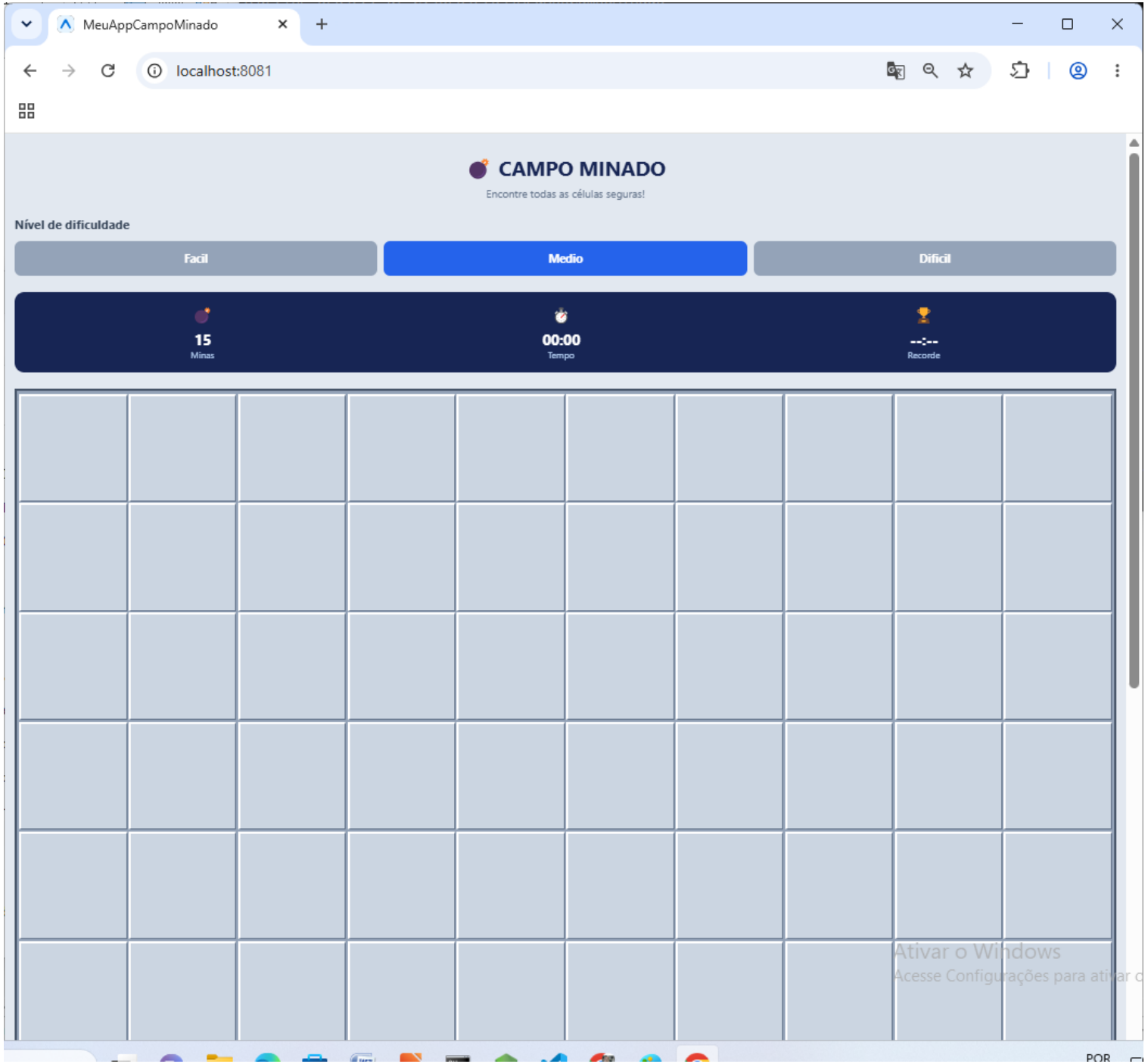
Abrindo o App

npx expo install react-dom react-native-web

npx expo start ou acesse o endereço pelo navegador



Ficou assim


MeuAppCampoMinado
localhost:8081
CAMPO MINADO
Encontre todas as células seguras!
Nível de dificuldade
Facil
Medio
Difícil
15
Minas
00:00
Tempo
--:--
Recorde
Ativar o Windows
Acesse Configurações para ativar o

Atividade de hoje

Elabore 4 apps utilizando react Native

- → 2 app mais simples;
- → 1 app mais com complexidade intermediária;
- → 1 app mais mais complexo;

* Você define como o App será e o que ele fará.

* Valor 2,0 pontos

* Entrega: 13/11/2026.

* Poste no classroom

XXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXX

Exercício 4 (Estilização)

Templates no react native

A documentação oficial do Expo lista cinco templates principais. Entretanto, o `blank` é o mais adequado para criar projetos simples em JavaScript. Os demais templates principais incluem configuração TypeScript por padrão, podendo exigir adaptações para uma turma que está aprendendo apenas JavaScript. (Como é a nossa situação).

Templates no react native

Template	Linha de comando	Descrição
Blank	`npx create-expo-app@latest Exerc5 --template blank`	Projeto básico com `App.js`, ideal para exercícios de componentes e `useState`.
Blank TypeScript	`npx create-expo-app@latest Exerc5 --template blank-typescript`	Projeto simples, mas com TypeScript e arquivo inicial normalmente chamado `App.tsx`.
Default	`npx create-expo-app@latest Exerc5 --template default`	Estrutura para aplicativos com várias telas e navegação usando Expo Router. Inclui TypeScript.
Tabs	`npx create-expo-app@latest Exerc5 --template tabs`	Projeto inicial com navegação por abas e Expo Router. Inclui TypeScript.
Bare Minimum	`npx create-expo-app@latest Exerc5 --template bare-minimum`	Projeto mínimo que gera também as pastas nativas Android e iOS. Confira a configuração de linguagem gerada pela versão utilizada.

Outros modelos prontos para JavaScript

Além dos templates, você pode criar projetos a partir de exemplos funcionais do Expo. Esses exemplos podem ser mais úteis quando quiser estudar uma funcionalidade específica.

Exemplo	Linha de comando	Descrição
Expo Router	npx create-expo-app@latest MeuRouter --example with-router	Demonstra navegação entre telas usando Expo Router.
React Navigation	npx create-expo-app@latest MeuNavegador --example with-react-navigation	Demonstra navegação usando a biblioteca React Navigation.