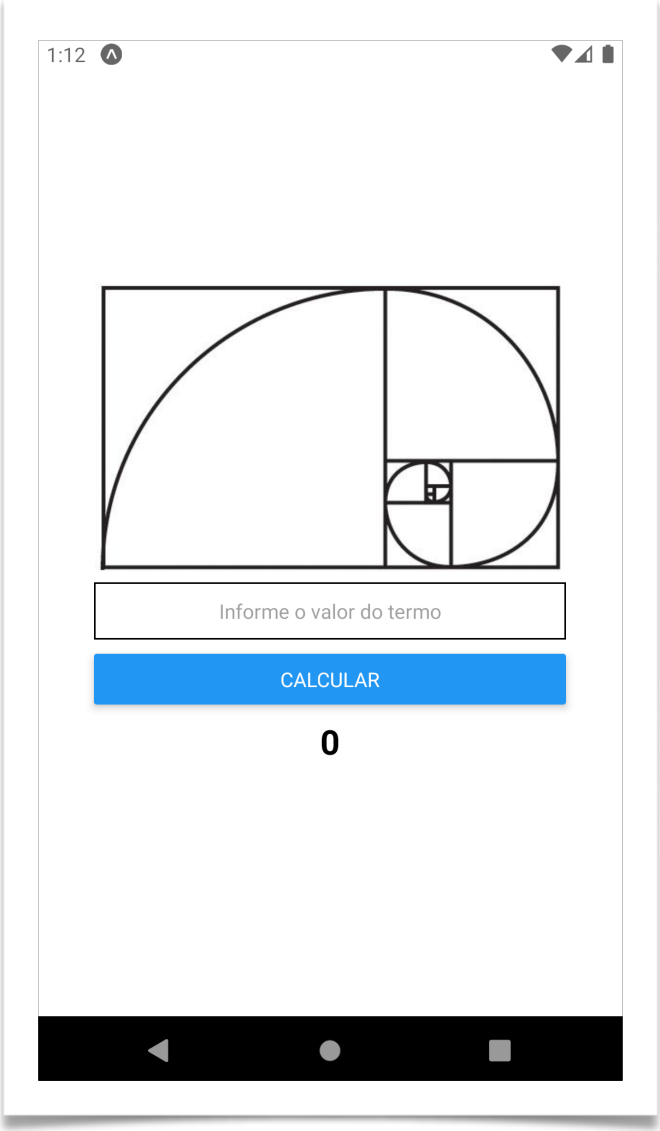
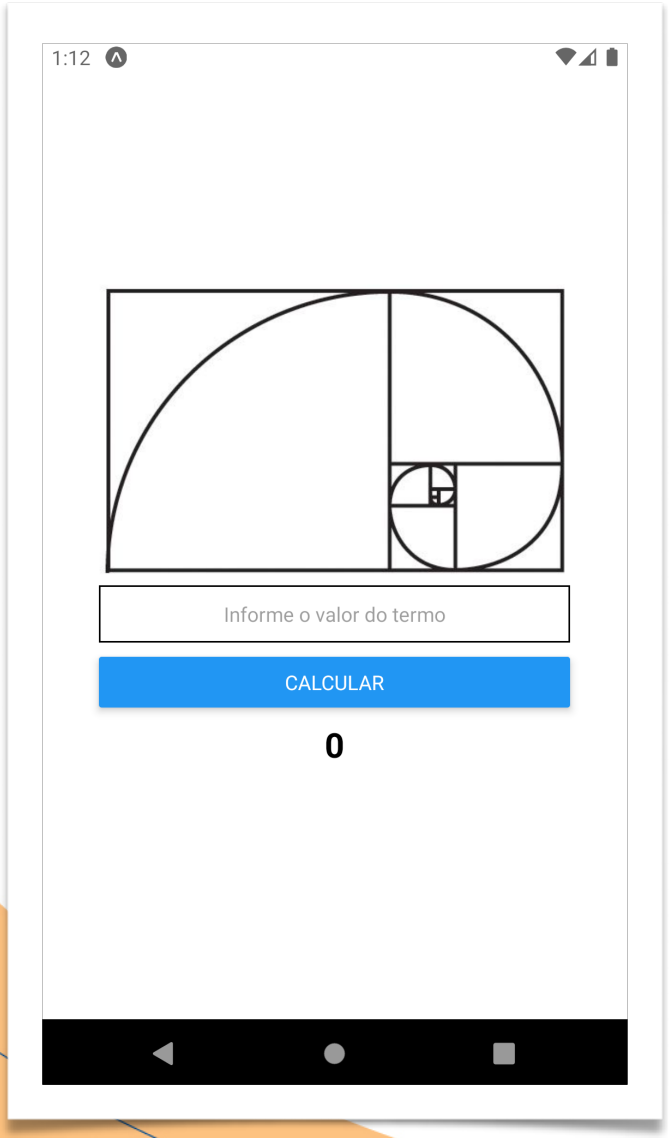
Prog Disp Móveis



Prog Disp Móveis



Primeiro passo será criar o projeto a partir de um template.

```
npx create-expo-app --template
```

```
? Choose a template: › - Use arrow-keys. Return to
submit.
❯   Blank - a minimal app as clean as an empty canvas
    Blank (TypeScript)
    Navigation (TypeScript)
    Blank (Bare)
```

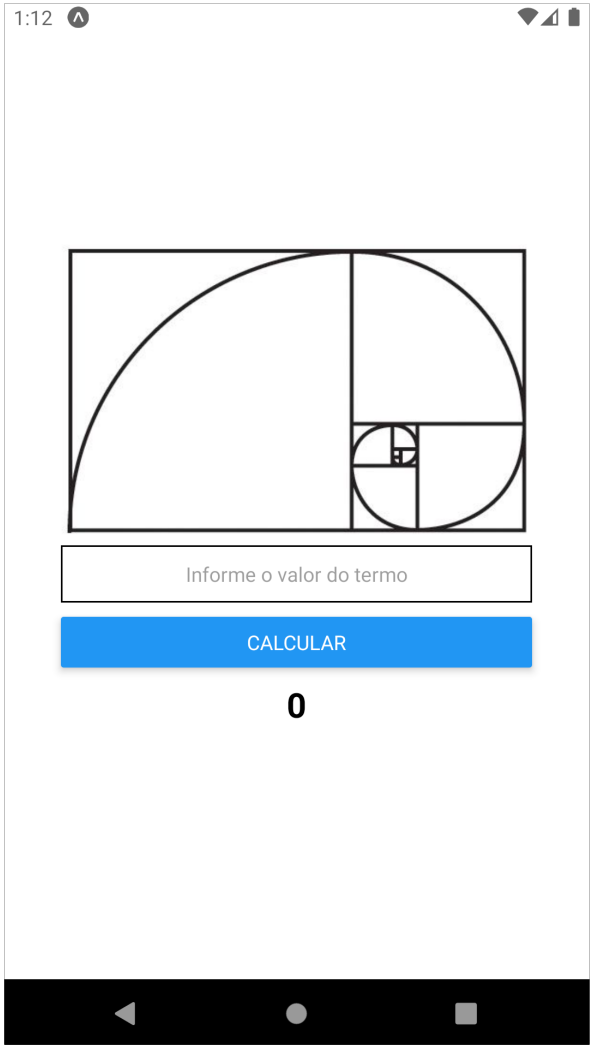
Selecionar o tipo do projeto

```
? Choose a template: › - Use arrow-keys. Return to
submit.
    Blank
❯   Blank (TypeScript) - blank app with TypeScript
enabled
    Navigation (TypeScript)
    Blank (Bare)
```

Fornecer o nome do projeto

```
✔ Choose a template: › Blank (TypeScript)
? What is your app named? › Fibonacci
```

Prog Disp Móveis



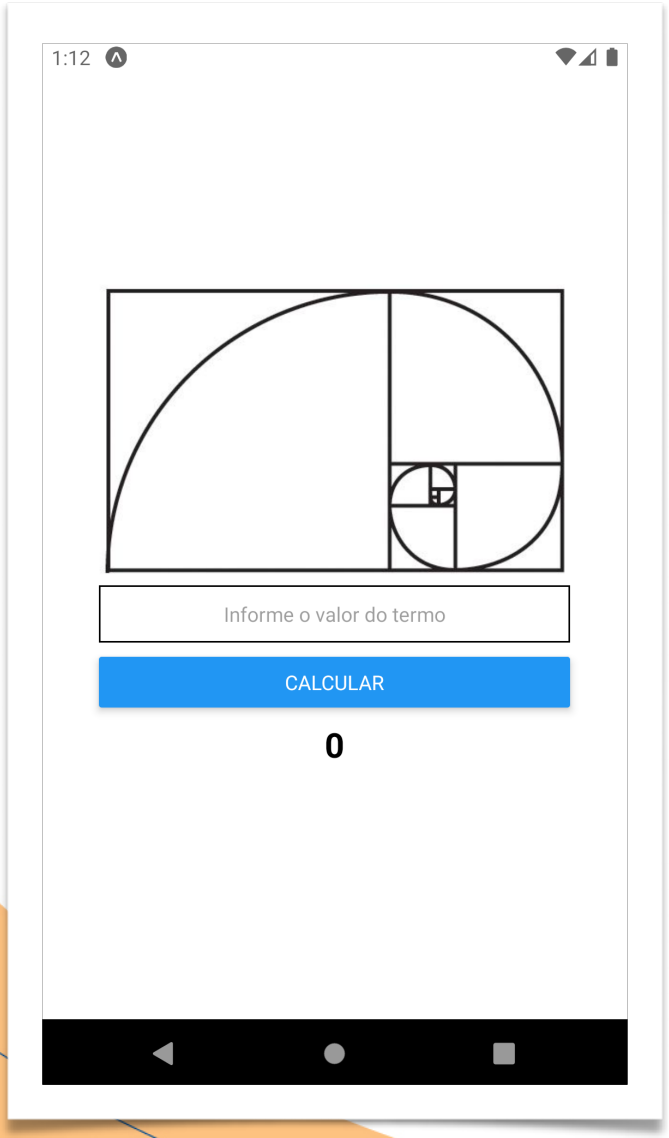
Será realizado o download do projeto.

```
✔ Choose a template: › Blank (TypeScript)
✔ What is your app named? … Fibonacci
✔ Downloaded and extracted project files.
> npm install

added 1221 packages, and audited 1222 packages in 41s


✅ Your project is ready!

To run your project, navigate to the directory and run one
of the following npm commands.

- cd Fibonacci
- npm run android
- npm run ios
- npm run web
```

Ir para a pasta do projeto e executar os comandos para desconsiderar as verificações do SSL/TLS

```
npm config set strict-ssl=false

set NODE_TLS_REJECT_UNAUTHORIZED=0
```

Prog Disp Móveis

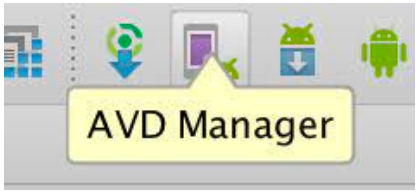


Abrir o VSCode na pasta do projeto.

Usando o terminal (externo ao VSCode) execute o comando abaixo

- **`cd Fibonacci`**

Abra o Android Studio. Execute o Virtual Device Manager

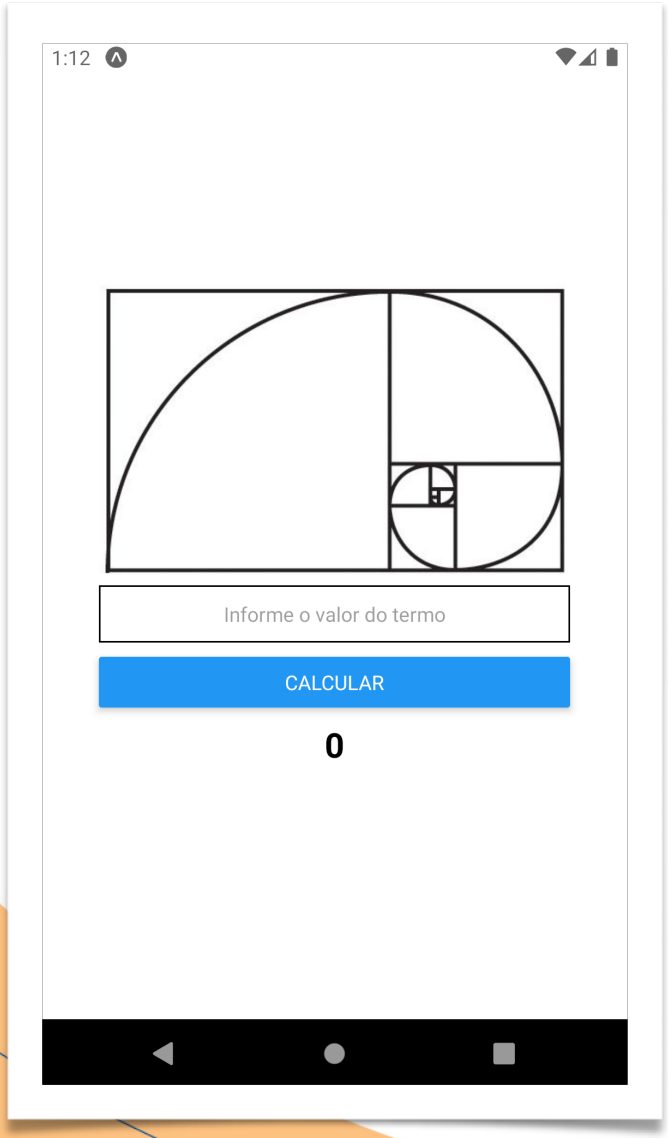


Faça a inicialização de um dos emuladores.

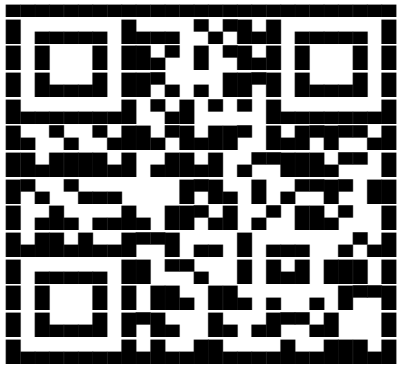
Após o emulador estar totalmente operacional, dê o comando abaixo.

- **`npm run android`**

Prog Disp Móveis

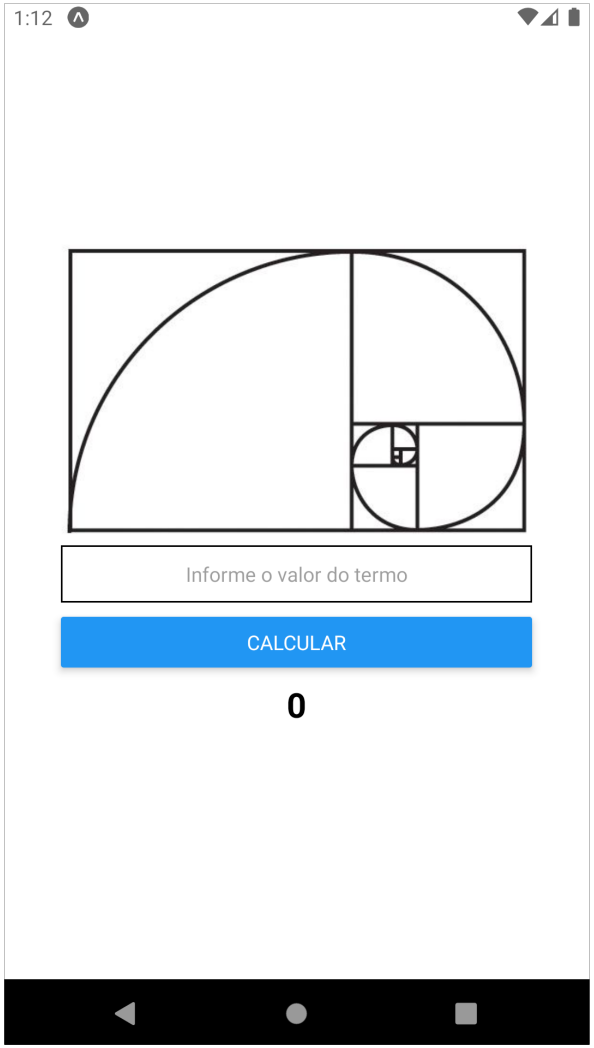


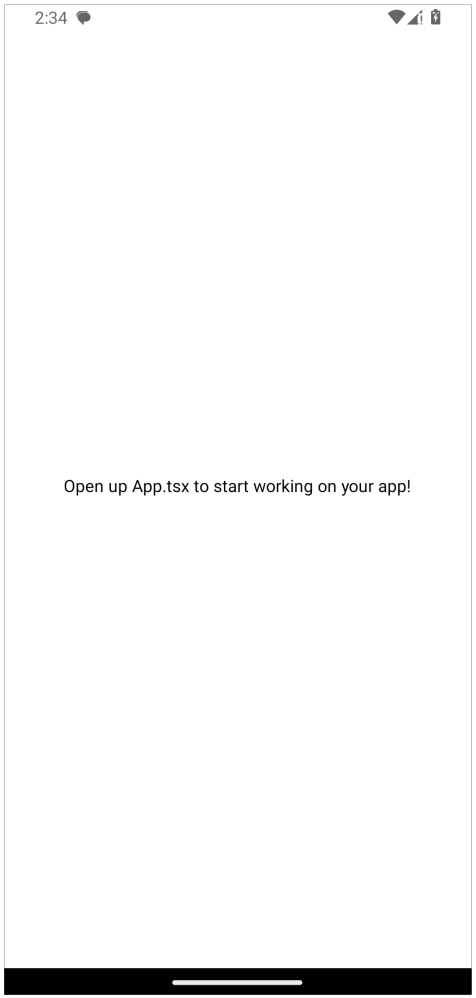

```
- npm run android

> fibonacci@1.0.0 android
> expo start --android

Starting project at /Users/anderson/Dropbox/Mac (2)/Downloads/Fibonacci
Starting Metro Bundler
› Opening exp://192.168.0.22:19000 on API_33
```



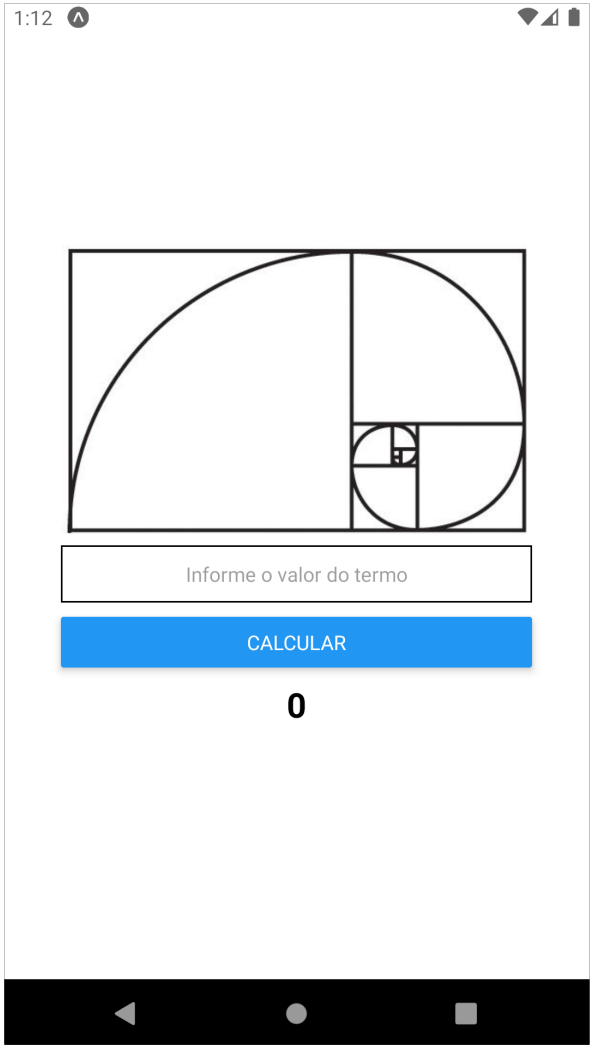
```
› Metro waiting on exp://192.168.0.22:19000
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

› Press a │ open Android
› Press i │ open iOS simulator
› Press w │ open web

› Press j │ open debugger
› Press r │ reload app
› Press m │ toggle menu

› Press ? │ show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
```

Prog Disp Móveis

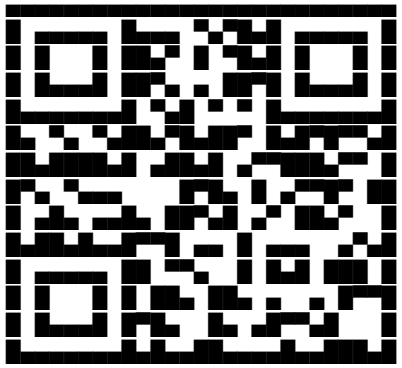




Prog Disp Móveis

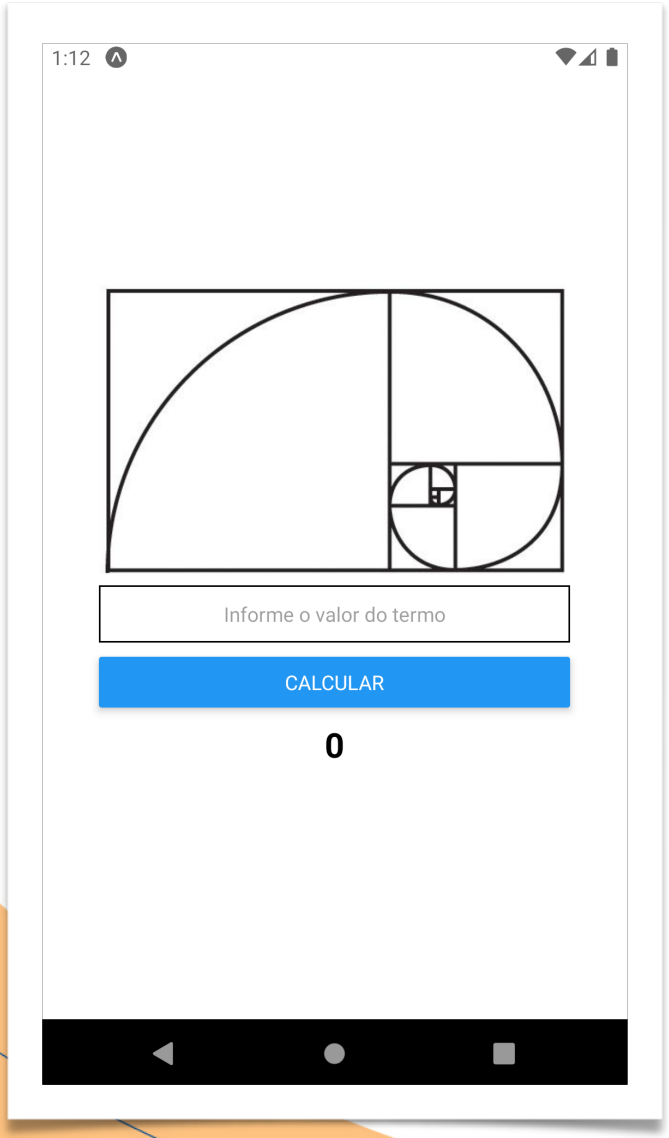



```
- npm run android

> fibonacci@1.0.0 android
> expo start --android

Starting project at /Users/anderson/Dropbox/Mac (2)/Downloads/Fibonacci
Starting Metro Bundler
› Opening exp://192.168.0.22:19000 on API_33
```

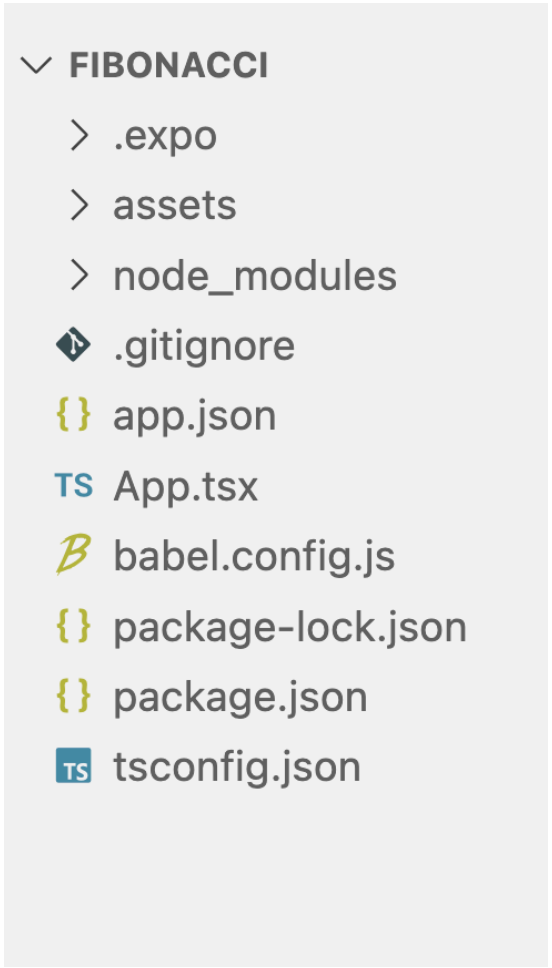


```
› Metro waiting on exp://192.168.0.22:19000
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

› Press a | open Android
› Press i | open iOS simulator
› Press w | open web

› Press j | open debugger
› Press r | reload app
› Press m | toggle menu

› Press ? | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
```

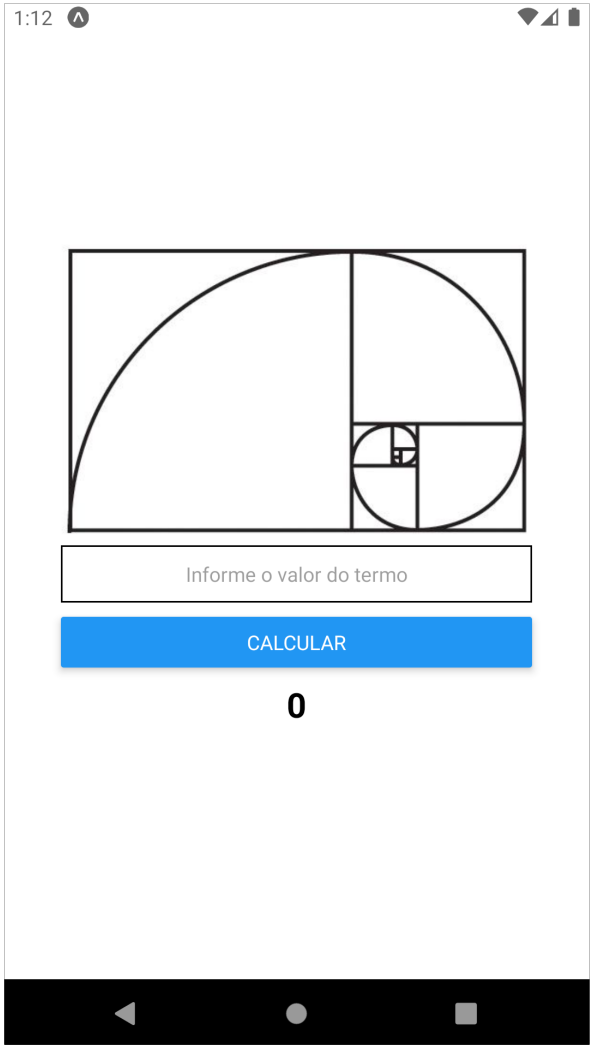
Prog Disp Móveis



Agora no VSCode irá aparecer assim…



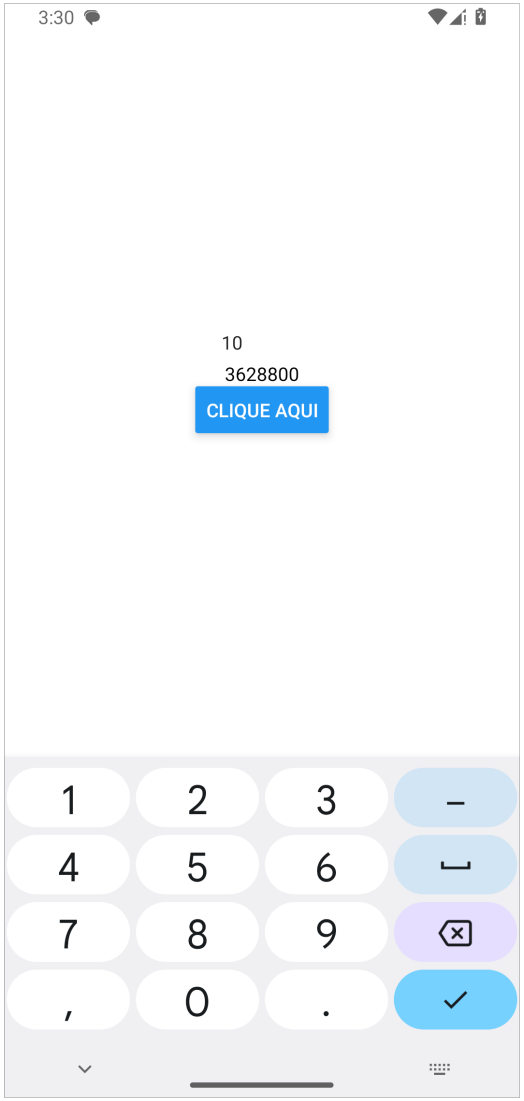
Prog Disp Móveis




```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default function App() {
  return (
    <View style={styles.container}>
      <Text>Open up App.tsx to start working on your app!</Text>
      <StatusBar style="auto" />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

Prog Disp Móveis

```
import React, { useState } from "react";
import { StyleSheet, Text, View, TextInput, Button} from 'react-native';

export default function App() {
  const [n, setN] = useState("");
  const [fatorial, setFatorial] = useState("");

  const Fatorial = () => {
    let fat=1;
    if(parseInt(n)<2) return(setFatorial(fat.toString()));
    for(let i=2; i<=parseInt(n); i++)
      fat = fat*i;
    setFatorial(fat.toString());
  }

  return (
    <View style={styles.container}>
      <TextInput
        placeholder="Informe n"
        onChangeText={(text: string) => setN(text)}
        keyboardType="numeric"
      />
      <Text>{fatorial}</Text>
      <Button title="clique aqui" onPress={() => Fatorial()} />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```



Prog Disp Móveis