

Tutorial

Como fazer um app com React Native

Primeiro → instale todos os pacotes necessários através do seguinte link: <https://reactnative.dev/docs/environment-setup>

- Depois de tudo instalado vamos criar o projeto:

#Criando um projeto

1. Crie uma pasta para a instalação do projeto
2. Entre na pasta através do prompt de comando no modo administrador
3. inicie o projeto com o comando `npx react-native@latest init AwesomeProject`
4. Logo depois, espere a finalização da criação do projeto

```
C:\> npm exec react-native@latest init Tutorial
Microsoft Windows [versão 10.0.22621.2428]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Windows\System32>cd C:\Users\55899\Documents\PROGRAMAÇÃO 5\trabalho-tutorial

C:\Users\55899\Documents\PROGRAMAÇÃO 5\trabalho-tutorial>npx react-native@latest init Tutorial
Need to install the following packages:
react-native@0.72.6
Ok to proceed? (y) y
[ ] | idealTree:metro: sill placeDep ROOT source-map@0.7.4 OK for: hermes-profile-transformer@0.0.6 want: ^0.7.3
```

Esta é a tela de confirmação do projeto, ou seja, a criação do projeto deu certo

```
Administrator: Prompt de Comando

#####
###      #####      ###
##      ###      ##
##      #####      ##
##      #####      ##
##      ##      ##
##      #####      ##
##      #####      ##
#####
#####      ###      ###      #####
###      ##      ##      ##      ##
###      #####      #####      ##
##      #####      #####      ##
##      #####      #####      ##
###      #####      #####      ##
#####      #####      #####
##      #####      #####      ##
##      #####      #####      ##
##      #####      #####      ##
##      #####      #####      ##
###      #####      #####      ##
#####      #####      #####

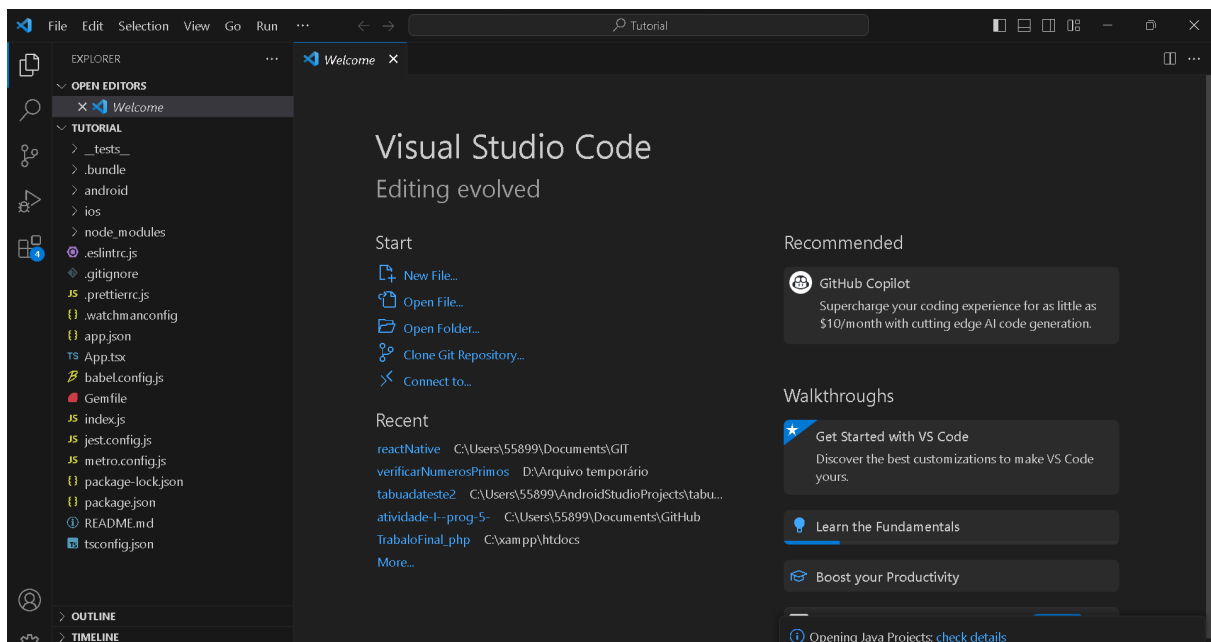
Welcome to React Native!
Learn once, write anywhere

✓ Downloading template
✓ Copying template
✓ Processing template
✓ Installing dependencies

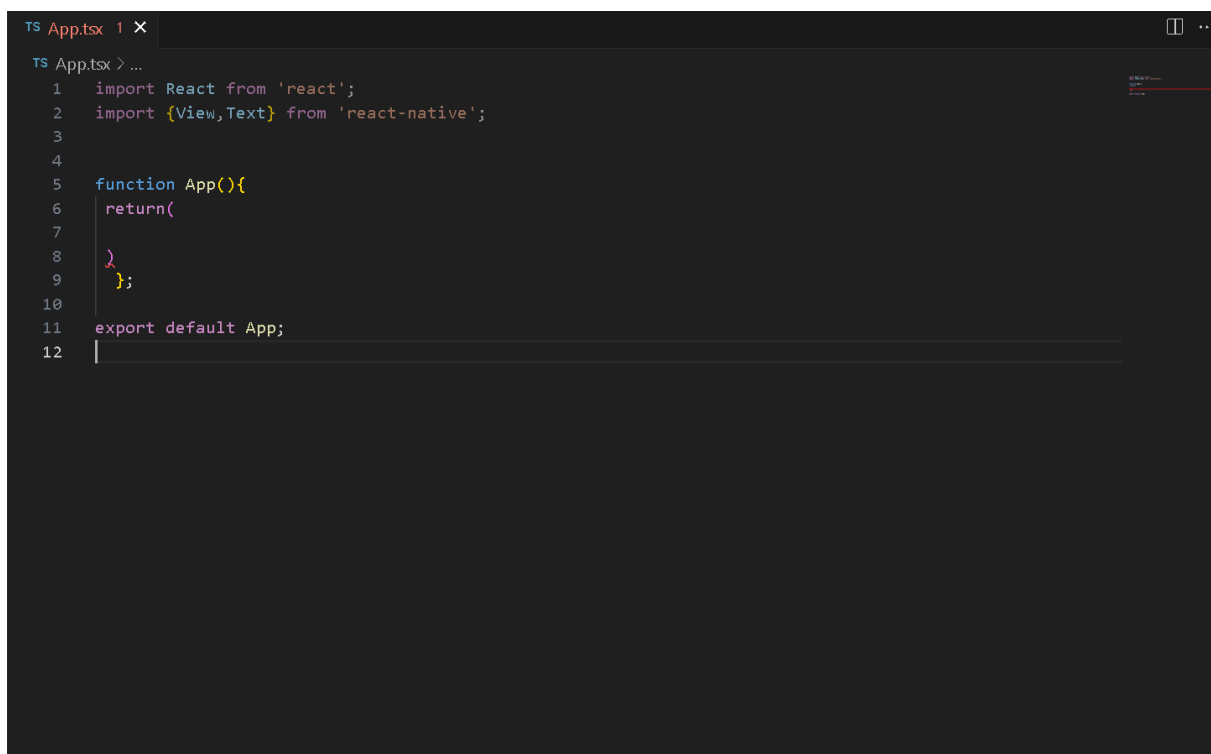
Run instructions for Android:
  • Have an Android emulator running (quickest way to get started), or a device connected.
  • cd "C:\Users\55899\Documents\PROGRAMAÇÃO 5\trabalho-tutorial\Tutorial" && npx react-native run-android

Run instructions for Windows:
  • See https://aka.ms/ReactNativeGuideWindows for the latest up-to-date instructions.
```

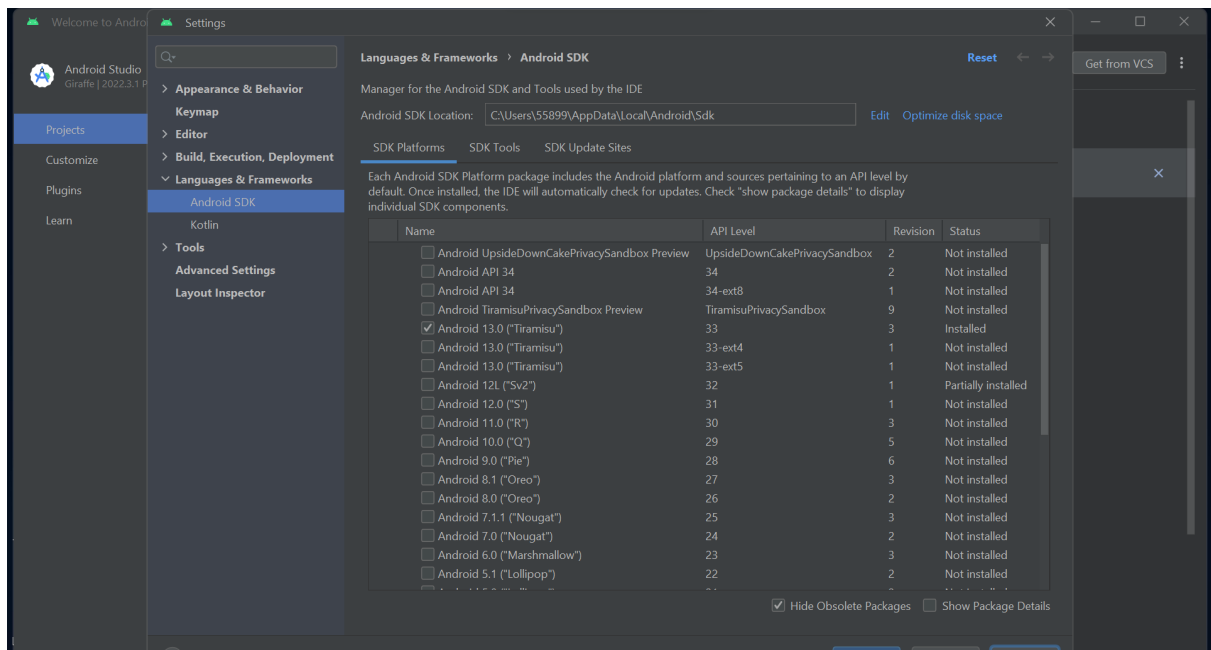
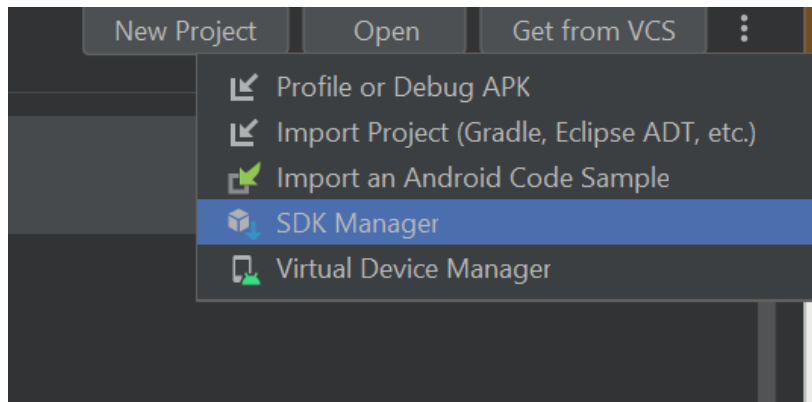
Em seguida, abra a pasta do seu computador que contém o projeto no `Visual Studio Code`



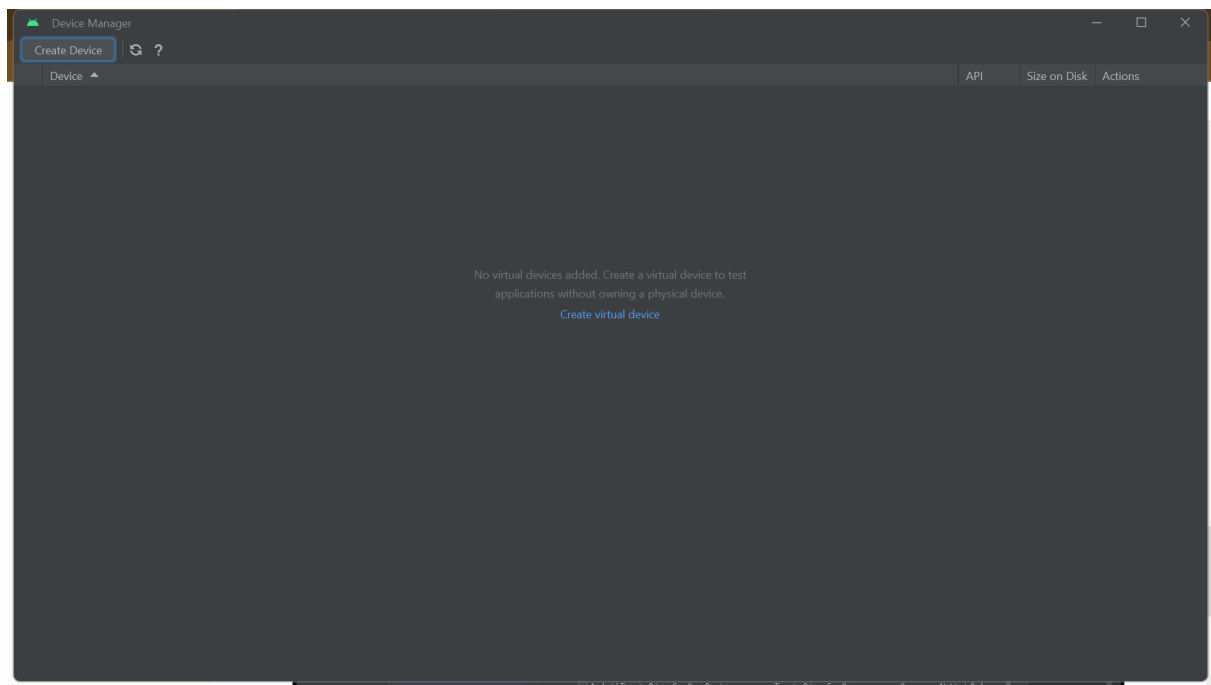
Em seguida abra o arquivo `App.tsx` e apague os códigos que forem desnecessários, deixando apenas esta estrutura básico



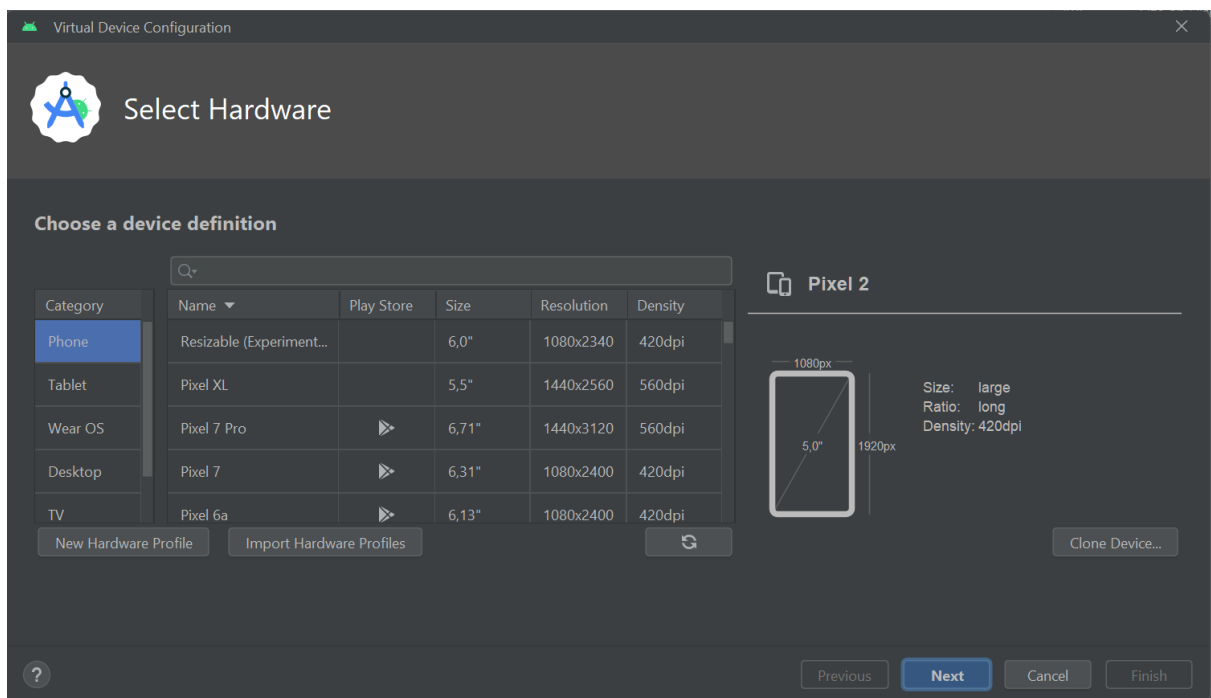
Em seguida é preciso abrir o Android Studio (se não tiver, faça a instalação pelo seu navegador), abra o sdk manager e faça a instalação da API 33



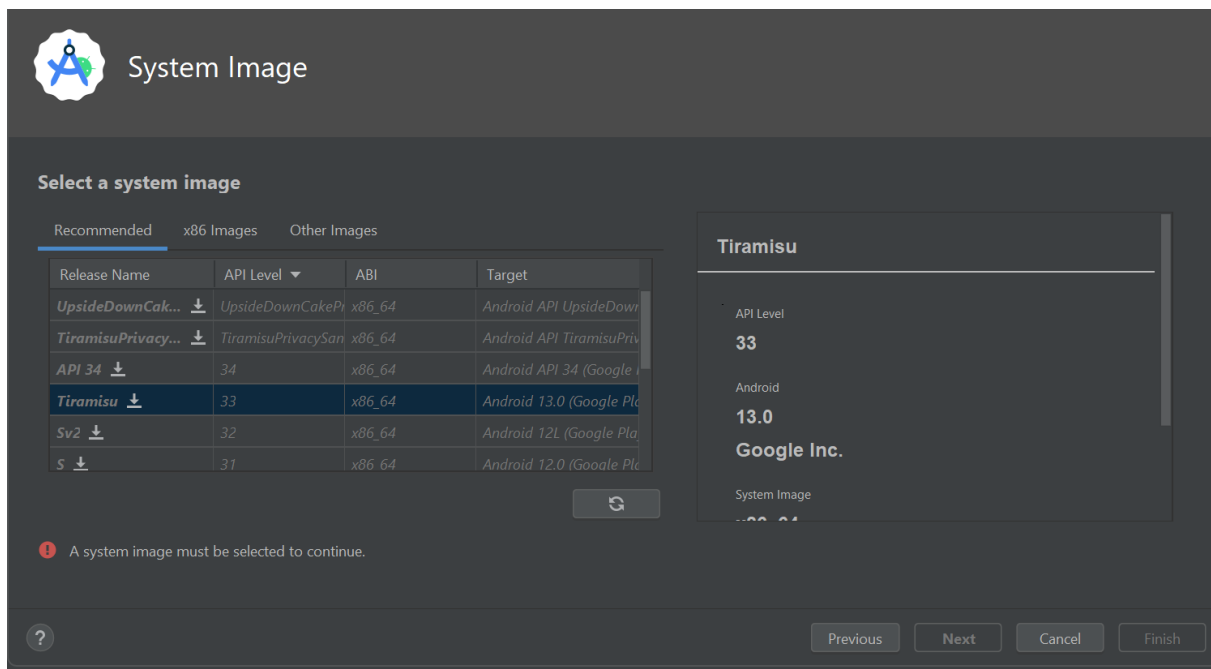
Depois entre no Virtual Device Manager para baixar o emulador, aonde será rodada a aplicação



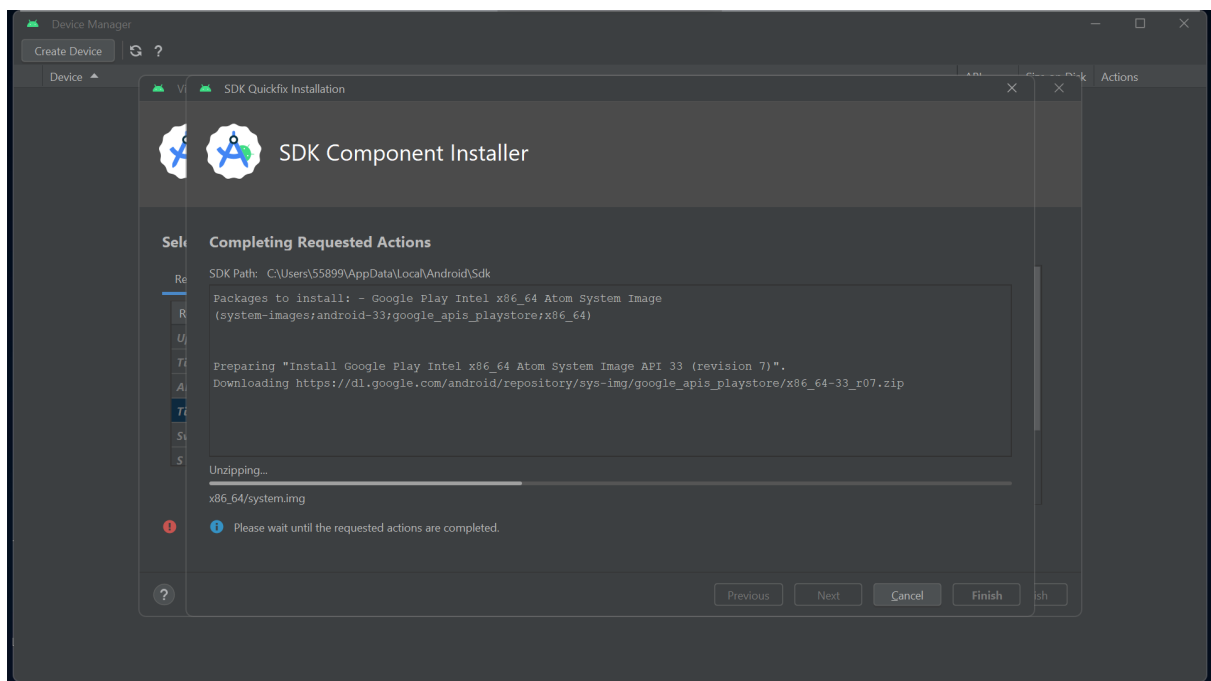
Em seguida será aberta esta tela, aonde você escolherá a layout do seu emulador



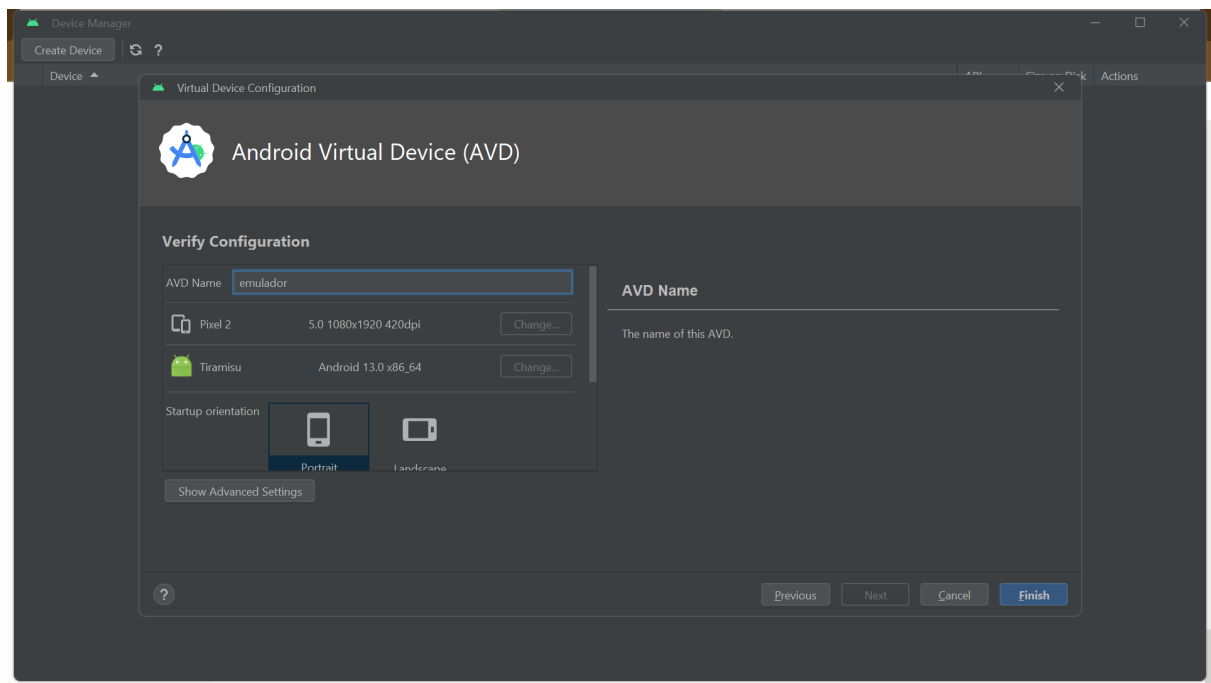
Em seguida selecione a versão android do seu emulador



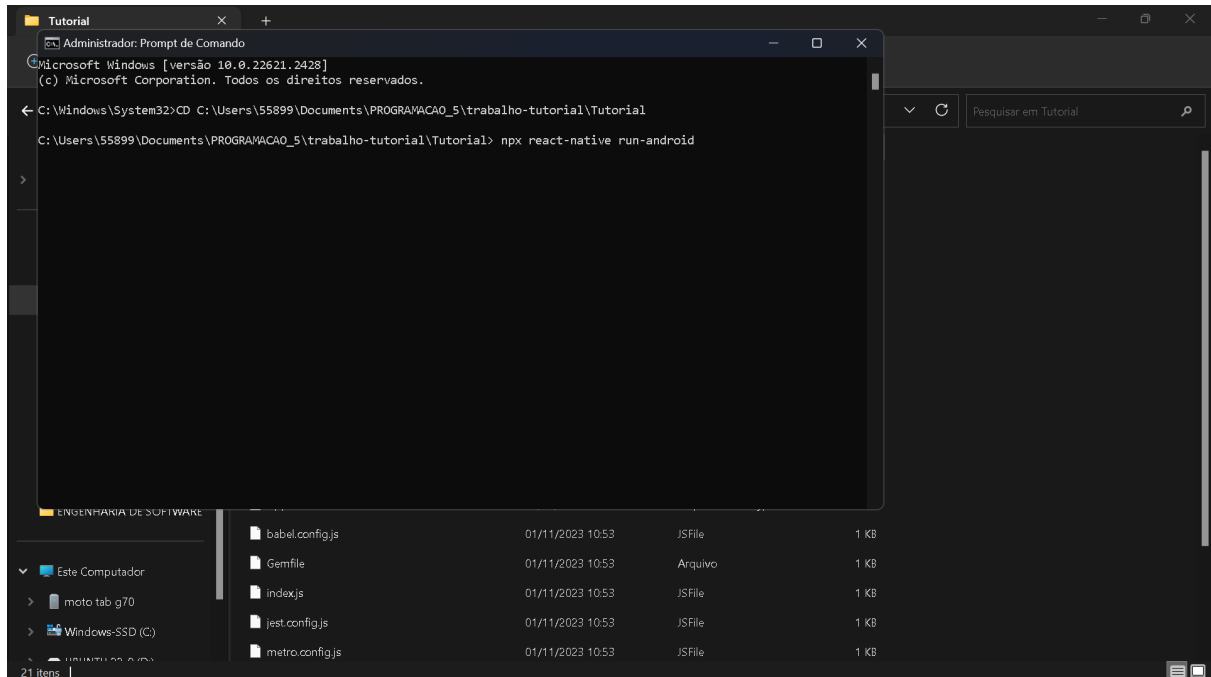
Em seguida será baixada a versão do android que você escolheu



Nesta tela você pode escolher a orientação do seu emulador, se ele ficará na horizontal ou na vertical, além de poder colocar um nome no seu emulador



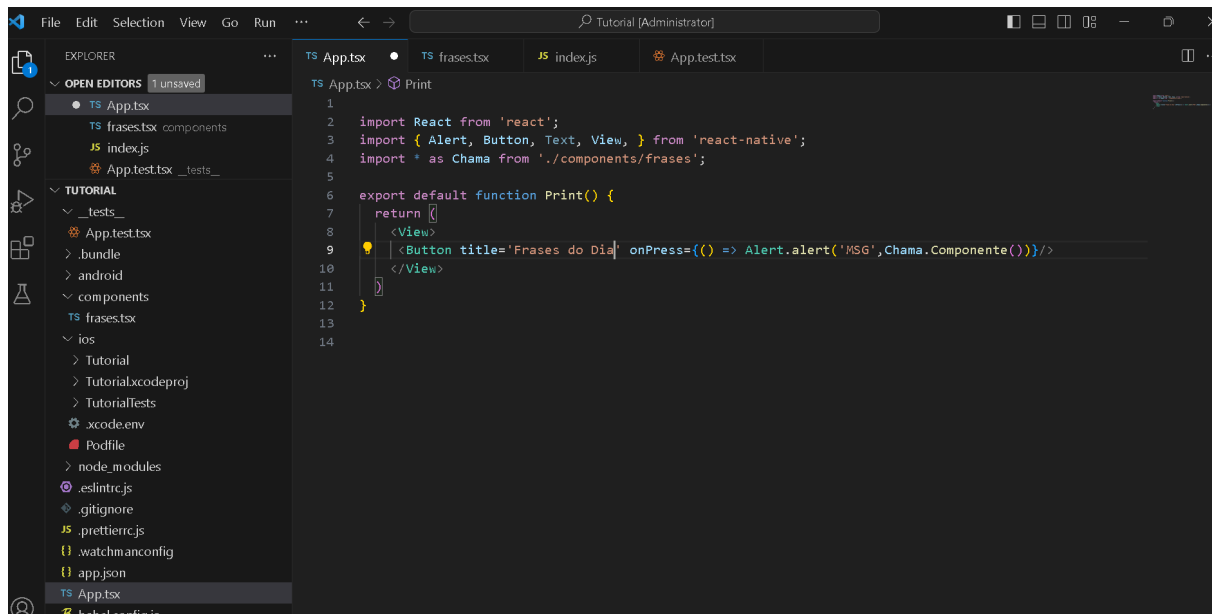
Para rodar o projeto, você deve entrar no diretório do projeto via cmd e codar `npx react-native run-android`, se o dispositivo android estiver conectado via USB ele executará o projeto no dispositivo via USB, se não estiver conectado em nenhum dispositivo ele executará no emulador que você fez a instalação



Este é o arquivo que starta o projeto, ,ou seja, que inicia.

```
import React from 'react' //importando o modulo React da biblioteca react
import { Alert, Button, Text, View, } from 'react-native' //importando os componentes que irá compor a interface da aplicação
import * as chama from './components/frases'; //importando todas as funções do diretório frases.tsx
```

```
//essa é responsável por criar o botão que aparecerá na interface da aplicação
export default function print(){
  return(
    <View>
      <Button title = 'Frases do Dia' onPress = {() => Alert.alert('MSG',chama.componente())}/>
    </View>
  )
}
```



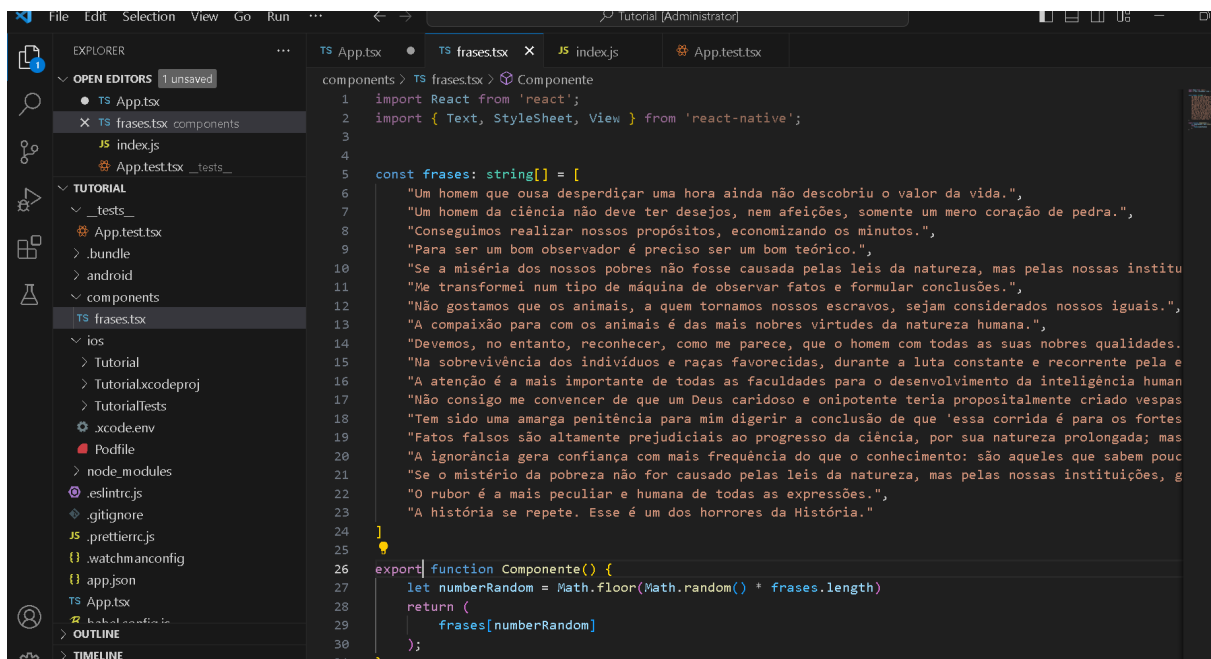
O projeto criado é uma aplicação para mostrar ao usuário frases diárias, frases que sejam motivacionais. No projeto foi criado o seguinte arquivo `frases.tsx`, que contém todas as frases que irão ser apresentadas ao usuário de maneira aleatória quando o usuário clicar no botão para mostrar a frase.

```
//a constante frases é um vetor que contém todas as frases que irão ser mostradas para o usuário
const frases : string [] = [
  ...
]

//essa função é responsável por criar um número aleatória
// A palavra-chave export significa que esta função pode ser importada e usada em outros arquivos.
export function componente() { //Esta linha define uma função chamada componente que pode ser usada em outras partes do código.
  let numberRandom = Math.floor(Math.random() * frases.length)
  return (

    //Esta linha retorna uma frase aleatória com base no número aleatório gerado.
    //A frase é escolhida do array frases usando o número aleatório como índice.
    frases[numberRandom]

  )
}
```

Introdução ao React-Native

React Native é uma das estruturas de JavaScript mais populares do mundo hoje. É famoso por sua utilidade para a criação de aplicativos nativos nas plataformas Android e iOS. O React Native encontra suas raízes no React; é uma biblioteca JavaScript que o Facebook criou para projetar a interface de aplicativos móveis. React Native pode ser combinado com JavaScript para construir aplicativos com recursos semelhantes aos nativos. Como os códigos React Native são compartilháveis, é uma excelente plataforma para o desenvolvimento de aplicativos de plataforma cruzada.

A renderização do aplicativo desta estrutura usa IU móvel em vez de WebView. Ele fornece ao desenvolvedor uma interface para interagir com as APIs da plataforma do aplicativo. É por isso que o React Native pode integrar recursos nativos, como acesso ao GPS e à câmera do dispositivo. Atualmente, ele cria aplicativos nativos para plataformas Android e iOS e pode oferecer suporte para plataformas de aplicativos adicionais. A capacidade de reutilização do código economiza aos desenvolvedores muitas horas de desenvolvimento. Muitos usuários que enfrentam aplicativos no Coinbase e no Facebook são feitos com React Native.

A principal vantagem do React Native é a capacidade de desenvolver aplicativos nativos para diferentes plataformas a partir de um único código-fonte. Isso significa que os desenvolvedores podem escrever uma vez e implantar em várias plataformas, economizando tempo e esforço no desenvolvimento e manutenção de aplicativos móveis.

React Native também fornece uma experiência de desenvolvimento rápida, com a capacidade de atualizar rapidamente a interface do usuário e visualizar as mudanças em tempo real. Além disso, os aplicativos React Native têm um visual e um desempenho próximos aos dos aplicativos nativos, graças à capacidade de acessar componentes nativos e funcionalidades específicas de cada plataforma.

No entanto, é importante notar que o React Native não é uma solução única para todos os cenários. Em alguns casos, aplicativos que exigem um desempenho extremamente otimizado ou recursos nativos altamente especializados podem se beneficiar mais do desenvolvimento nativo. No entanto, para muitos aplicativos, o React Native oferece uma abordagem eficaz e eficiente para a criação de aplicativos móveis multiplataforma.

Características do React-Native

O React Native é uma estrutura de desenvolvimento móvel que possui várias características distintas que o tornam uma escolha popular para a criação de aplicativos móveis multiplataforma. Aqui estão algumas das principais características do React Native:

1. **Desenvolvimento Multiplataforma:** Com o React Native, é possível criar aplicativos para iOS e Android a partir de um único código-fonte, economizando tempo e esforço.
2. **Componentes Reutilizáveis:** O React Native utiliza uma abordagem baseada em componentes, onde você pode criar componentes reutilizáveis que podem ser usados em todo o aplicativo, tornando o desenvolvimento mais modular e escalável.
3. **Desempenho de Alto Nível:** Os aplicativos React Native têm um desempenho próximo ao dos aplicativos nativos, uma vez que os componentes são traduzidos em código nativo durante a execução, e você pode otimizar partes críticas do aplicativo usando módulos nativos.
4. **Acesso a Recursos Nativos:** O React Native permite o acesso a recursos e funcionalidades nativas do dispositivo, o que significa que você pode utilizar câmera, GPS, sensores e outros recursos específicos de cada plataforma.
5. **Rápido Desenvolvimento:** O desenvolvimento no React Native é conhecido por ser mais rápido em comparação com o desenvolvimento nativo, pois oferece uma recarga instantânea de código durante o desenvolvimento (conhecida como "Hot Reloading"), o que permite visualizar as alterações em tempo real.
6. **Comunidade Ativa:** O React Native possui uma comunidade ativa e grande quantidade de bibliotecas de terceiros, que facilitam o desenvolvimento e a expansão de recursos em aplicativos.
7. **Suporte a JavaScript e TypeScript:** Você pode escrever aplicativos React Native em JavaScript ou TypeScript, o que oferece flexibilidade na escolha da linguagem de programação.
8. **Facilidade de Migração:** É possível integrar gradualmente o React Native em aplicativos existentes, permitindo uma migração mais suave e custo-eficaz.
9. **Suporte Corporativo:** Empresas como Facebook, Airbnb, e outras têm adotado o React Native, garantindo um suporte robusto e contínuo.
10. **Documentação Rica:** O React Native possui uma documentação abrangente e uma ampla base de conhecimento que auxiliam os desenvolvedores a aprender e usar a estrutura de forma eficaz.

Em resumo, o React Native é uma escolha atraente para o desenvolvimento de aplicativos móveis devido à sua capacidade de criar aplicativos multiplataforma com bom desempenho, reutilização de código e acesso a recursos nativos, tudo isso com uma comunidade ativa e suporte sólido.

Vantagens de usar React-Native

- **Desenvolvimento multiplataforma:** Uma das maiores vantagens do React Native é a capacidade de criar aplicativos para iOS e Android a partir de um único código-fonte. Isso economiza tempo e esforço, uma vez que os desenvolvedores podem reutilizar a maior parte do código em ambas as plataformas.
- **Rápido desenvolvimento:** React Native permite um desenvolvimento mais rápido em comparação com o desenvolvimento nativo. Ele fornece componentes pré-construídos e uma grande comunidade de desenvolvedores que compartilham bibliotecas e soluções prontas, acelerando o processo de criação de aplicativos.
- **Economia de custos:** Ao compartilhar código entre plataformas, as empresas podem economizar dinheiro no desenvolvimento e na manutenção de aplicativos para iOS e Android.
- **Compartilhamento de código:** A maioria da lógica de negócios e parte da interface do usuário podem ser compartilhadas entre as plataformas, o que facilita a manutenção e atualização do aplicativo.

- **Reutilização de habilidades:** Desenvolvedores que já estão familiarizados com JavaScript e React podem aplicar suas habilidades no desenvolvimento móvel com React Native, o que facilita a transição.
- **Open-Source:** Dado o fato de que React Native é uma plataforma de código aberto licenciada pelo MIT, ele dá aos desenvolvedores acesso para usar bibliotecas e frameworks gratuitamente. Ele impõe algumas restrições à capacidade de reutilização do software, mas também fornece proteção legal para desenvolvedores.
- **Hot Reloading:** O recurso de recarregamento a quente é um dos mais úteis e atraentes para os desenvolvedores. Ele permite que o desenvolvedor implemente alterações em um código e veja o efeito no aplicativo em tempo real. Portanto, é possível atualizar um aplicativo que já está ativo. Tudo o que é necessário é editar o código-fonte, e a atualização vai ao ar após salvar o arquivo. Portanto, os desenvolvedores podem criar atualizações sem um único tempo de inatividade.

Desvantagens de usar React-Native

- **Desempenho:** Apesar dos avanços, os aplicativos React Native podem ter um desempenho ligeiramente inferior em comparação com aplicativos nativos, especialmente para aplicativos mais complexos ou que fazem uso intensivo de recursos do dispositivo.
- **Acessibilidade a recursos nativos:** Para acessar recursos ou funcionalidades nativas específicas de uma plataforma, às vezes é necessário escrever módulos nativos personalizados, o que pode ser complicado.
- **Dependência da comunidade:** A qualidade das bibliotecas e módulos de terceiros pode variar. Algumas bibliotecas podem não ser tão maduras ou bem mantidas quanto suas contrapartes nativas.
- **Tamanho do aplicativo:** Os aplicativos React Native tendem a ser maiores em tamanho do que os aplicativos nativos, devido à necessidade de incluir o mecanismo JavaScript.
- **Atualizações de plataforma:** As atualizações do sistema operacional podem quebrar funcionalidades existentes nos aplicativos React Native, e às vezes é necessário aguardar atualizações de bibliotecas para mantê-los funcionando corretamente.
- **Desafios de depuração e compatibilidade:** Apesar de todos os excelentes recursos, React Native ainda está na versão Beta! É por isso que ele ainda apresenta alguns problemas aparentes, como a complexidade de depurar aplicativos, entre outras limitações, incluindo problemas de compatibilidade.
- **Depende do Facebook:** O fato de React Native ter sido desenvolvido pelo Facebook é impressionante e uma das deficiências da plataforma. Suponha que o Facebook pare de fornecer backup para a plataforma, ela desmoronaria como um baralho de cartas. No entanto, muitas outras plataformas oferecem funções semelhantes ao React Native.

Em resumo, a escolha entre React Native e o desenvolvimento nativo depende das necessidades do projeto, do desempenho necessário e do conhecimento da equipe de desenvolvimento. Para muitos aplicativos, o React Native pode ser uma excelente opção, permitindo economia de tempo e recursos. No entanto, para aplicativos que requerem desempenho máximo ou recursos nativos altamente especializados, o desenvolvimento nativo ainda pode ser a melhor escolha.

Método e Desenvolvimento

O desenvolvimento no React Native envolve a criação de aplicativos móveis multiplataforma usando JavaScript ou TypeScript e a biblioteca React. O processo de desenvolvimento no React Native pode ser resumido em algumas etapas principais:

1. Configuração do Ambiente:

- Instalação do Node.js: Certifique-se de que o Node.js esteja instalado no seu sistema, pois o React Native depende dele.
- Instalação do React Native CLI: Você precisará instalar o CLI do React Native usando o seguinte comando: `npm install -g react-native-cli` ou `npx react-native init MyApp` para criar um novo projeto.

2. Criação de um Novo Projeto:

- Use o comando `npx react-native init NomeDoProjeto` para criar um novo projeto React Native.
- O comando criará a estrutura básica do projeto, incluindo arquivos e pastas necessários.

3. Desenvolvimento do Aplicativo:

- A maior parte do desenvolvimento é feita na pasta `src` do projeto. Você criará componentes, telas e funcionalidades dentro desta pasta.
- Os componentes são criados usando JavaScript ou TypeScript e a biblioteca React. Você pode usar os componentes predefinidos do React Native, como `View`, `Text` e `Button`, ou criar seus próprios componentes personalizados.
- Você pode acessar recursos nativos do dispositivo, como câmera, GPS e sensores, usando módulos nativos ou bibliotecas de terceiros.

4. Testes e Debugging:

- Use emuladores ou dispositivos físicos para testar o aplicativo em tempo real.
- O React Native oferece ferramentas de debugging, como o React DevTools, para identificar e corrigir problemas.

5. Gerenciamento de Estado:

- O React Native oferece opções para gerenciamento de estado, como o uso de `useState`, `useReducer`, ou bibliotecas de gerenciamento de estado como Redux ou Mobx.

6. Estilo e Layout:

- Você pode aplicar estilos aos componentes usando CSS-in-JS ou uma linguagem de estilização como o StyleSheet do React Native.

7. Navegação:

- Para criar um fluxo de navegação entre telas, você pode usar bibliotecas de navegação populares, como React Navigation.

8. Build e Implantação:

- Para construir o aplicativo para iOS, você precisa de um Mac e pode usar o Xcode.
- Para construir o aplicativo para Android, você pode usar o Android Studio.
- O processo de implantação envolve a publicação do aplicativo nas lojas de aplicativos, como a App Store (iOS) e o Google Play (Android).

9. Manutenção e Atualizações:

- Após o lançamento, você deve continuar a manter e atualizar o aplicativo conforme necessário, seja para adicionar novos recursos ou corrigir problemas.

10. Documentação e Comunidade:

- A documentação oficial do React Native e a comunidade de desenvolvedores são recursos valiosos para obter ajuda e aprender mais sobre a plataforma.

O React Native oferece uma abordagem eficaz para o desenvolvimento de aplicativos móveis, permitindo que os desenvolvedores compartilhem a maior parte do código entre plataformas, economizando tempo e recursos. É uma escolha popular para empresas que desejam criar aplicativos multiplataforma de alto desempenho.

Conclusão

Em resumo, o React Native é uma escolha sólida para o desenvolvimento de aplicativos móveis, oferecendo um equilíbrio entre eficiência, desempenho e acessibilidade a recursos nativos. No entanto, a decisão de usar o React Native ou o desenvolvimento nativo deve depender das necessidades específicas do projeto. É importante avaliar esses fatores ao decidir se o React Native é a escolha certa para um projeto específico.