

App para comprar comida

Nessa aula vamos criar um app para comprar comida e para isso vamos criar um projeto. Abra o VS Code na pasta onde você deseja criar o seu projeto e em seguida abra o terminal do Vs Code e digite o comando:

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

Na hora de colar no terminal, certifique-se que tem dois traços (--) antes da palavra "template", para não dar erro na hora da instalação.

Em seguida escolha a opção "**Blank**" e dê um nome ao seu projeto.

Depois de criado o seu projeto, **entre na pasta do seu projeto para realizar a instalação das dependências** que serão utilizadas no projeto, vá até o arquivo **package.json**:

package.json

```
{
  "name": "food",
  "main": "expo-router/entry",
  "version": "1.0.0",
  "scripts": {
    "start": "expo start",
    "reset-project": "node ./scripts/reset-project.js",
    "android": "expo start --android",
    "ios": "expo start --ios",
    "web": "expo start --web",
    "test": "jest --watchAll",
    "lint": "expo lint"
  },
  "jest": {
    "preset": "jest-expo"
  },
  "dependencies": {
    "@expo/vector-icons": "^14.0.2",
```

```
"@react-navigation/native": "^6.0.2",
"expo": "~51.0.16",
"expo-constants": "~16.0.2",
"expo-font": "~12.0.7",
"expo-linking": "~6.3.1",
"expo-router": "~3.5.17",
"expo-splash-screen": "~0.27.5",
"expo-status-bar": "~1.12.1",
"expo-system-ui": "~3.0.6",
"expo-web-browser": "~13.0.3",
"json-server": "^1.0.0-beta.1",
"nativewind": "4.0.1",
"react": "18.2.0",
"react-dom": "18.2.0",
"react-native": "0.74.2",
"react-native-gesture-handler": "~2.16.1",
"react-native-pager-view": "6.3.0",
"react-native-reanimated": "~3.10.1",
"react-native-safe-area-context": "4.10.1",
"react-native-screens": "3.31.1",
"react-native-web": "~0.19.10",
"tailwindcss": "^3.4.4"
},
"devDependencies": {
  "@babel/core": "^7.20.0",
  "@types/jest": "^29.5.12",
  "@types/react": "~18.2.45",
  "@types/react-test-renderer": "^18.0.7",
  "jest": "^29.2.1",
```

```
"jest-expo": "~51.0.3",
"react-test-renderer": "18.2.0",
"typescript": "~5.3.3"
},
"private": true
}
```

Nesse projeto estaremos utilizando a biblioteca tailwind para a estilização e para isso precisamos fazer algumas configurações. Primeiro vamos criar na **raiz** do nosso projeto um novo arquivo.

tailwind.config.js

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  // NOTE: Update this to include the paths to all of your component files.
  content: ["/src/**/*.js,jsx,ts,tsx"],
  presets: [require("nativewind/preset")],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

Agora vamos criar o arquivo **metro.config.js**

metro.config.js

```
const { getDefaultConfig } = require("expo/metro-config");
const { withNativeWind } = require('nativewind/metro');

const config = getDefaultConfig(__dirname)

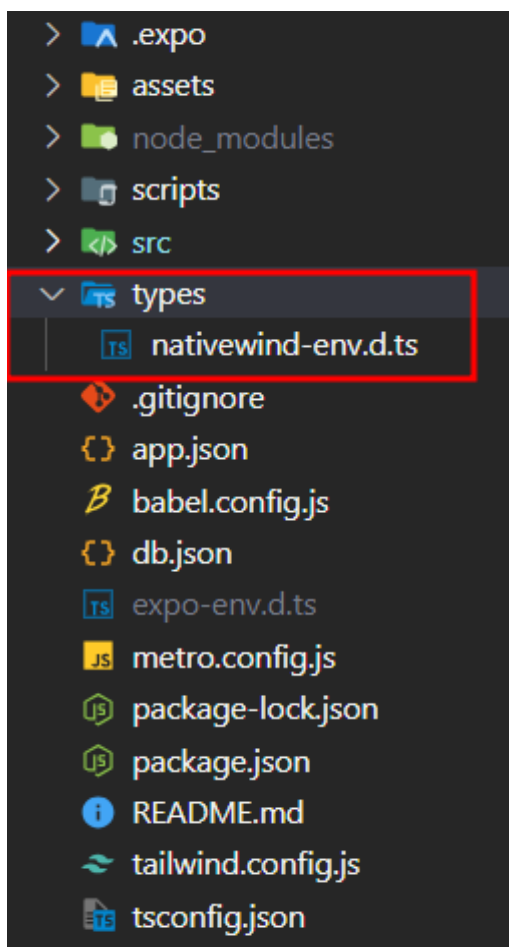
module.exports = withNativeWind(config, { input: './src/styles/global.css' })
```

O próximo passo é configurar o arquivo **babel.config.js**

babel.config.js

```
module.exports = function (api) {  
  api.cache(true);  
  return {  
    presets: [  
      ["babel-preset-expo", { jsxImportSource: "nativewind" }],  
      "nativewind/babel",  
    ],  
  };  
};
```

Verifique se os arquivos estão criados no local correto e se existe a pasta “types” e o respectivo arquivo como na imagem abaixo, caso não exista então crie a pasta e o arquivo.



nativewind-env.d.ts

```
/// <reference types="nativewind/types" />
```

Agora vamos criar a nossa pasta “**src**” onde ficarão armazenados o código da nossa aplicação. Crie a pasta “**styles**” e dentro dela o arquivo “**global.css**”

./src/styles/global.css

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

Em seguida, dentro da pasta “**src**” vamos criar nossa pasta “**app**” com os arquivos “**_layout.tsx**” e “**index.tsx**”

./src/app/_layout.tsx

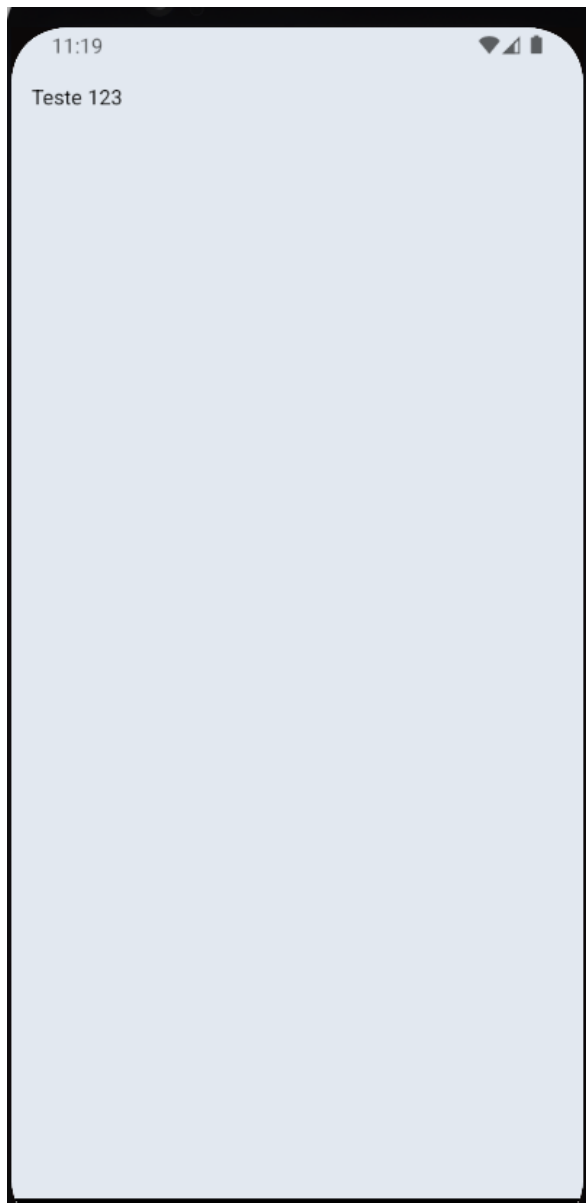
```
import '../styles/global.css'  
import { Slot } from "expo-router";  
  
export default function RootLayout() {  
  return <Slot/>  
}
```

./src/app/index.tsx

```
import { Text, View, ScrollView } from "react-native";  
import Constants from 'expo-constants'  
const statusBarHeight = Constants.statusBarHeight;  
  
export default function Index() {  
  return (  
    <ScrollView  
      style={{ flex: 1 }}  
    >  
      <Text>  
        Hello World!  
      </Text>  
    </ScrollView>  
  )  
}
```

```
        className="bg-slate-200"
        showsVerticalScrollIndicator={false}
    >
        <View className="w-full px-4" style={{ marginTop: statusBarHeight + 8 }}>
<Text>Teste 123<
    </View>
    </ScrollView>
);
}
```

Com isso, vamos testar se nossa aplicação está funcionando corretamente. Execute “**npm start**” no terminal e a seguinte tela deverá estar aparecendo:



Agora, vamos começar a construir nossos componentes. Começando pelo no **header** para isso, crie a pasta **components** e nela a pasta “**header**” e nela o arquivo “**index.tsx**”

./src/components/header/index.tsx

```
import { View, Pressable, Text } from "react-native";
import { Ionicons, Feather } from "@expo/vector-icons";

export function Header() {
  return (
```

```

<View className="w-full flex flex-row items-center justify-between">
  <Pressable className="w-10 h-10 bg-white rounded-full flex justify-center items-center">
    <Icons name="menu" size={20} color="#121212" />
  </Pressable>

  <View className="flex flex-col items-center justify-center">
    <Text className="text-center text-sm text-slate-800">Localização</Text>

    <View className="flex-row items-center justify-center gap-1">
      <Feather name="map-pin" size={14} color="#FF0000" />
      <Text className="text-lg font-bold">Joinville</Text>
    </View>
  </View>

  <Pressable className="w-10 h-10 bg-white rounded-full flex justify-center items-center">
    <Feather name="bell" size={20} color="#121212" />
  </Pressable>
</View>
);
}

```

E vamos chamar agora o componente nosso arquivo **index.tsx** da pasta app.

./src/app/index.tsx

```

import { Text, View, ScrollView } from "react-native";
import Constants from 'expo-constants'

```

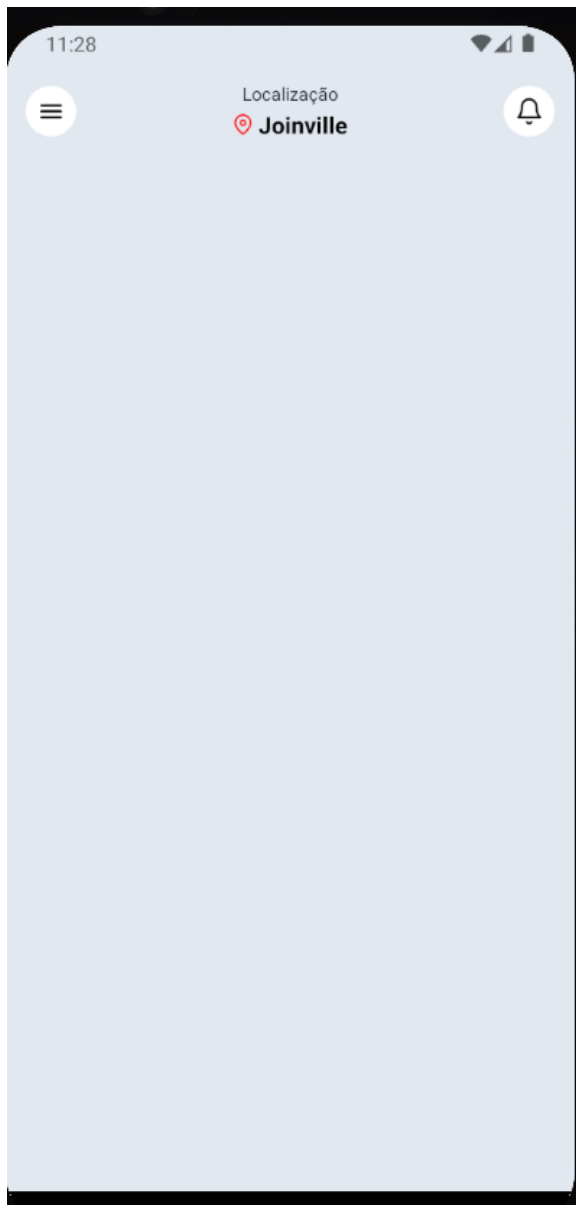


```
import { Header } from "../components/header";

const statusBarHeight = Constants.statusBarHeight;

export default function Index() {
  return (
    <ScrollView
      style={{ flex: 1 }}
      className="bg-slate-200"
      showsVerticalScrollIndicator={false}
    >
      <View className="w-full px-4" style={{ marginTop: statusBarHeight + 8 }}>
        <Header />
      </View>
    </ScrollView>
  );
}
```

Com isso o nosso header já deve estar sendo mostrado em tela.



Agora vamos construir o nosso banner que ficará abaixo do header.

`./src/compenents/banner/index.tsx`

```
import { View, Pressable, Image } from 'react-native';
import PagerView from "react-native-pager-view"

export function Banner() {
  return (
    <View className="w-full h-36 md:h-60 rounded-2xl mt-5 mb-4">
```

```

<PagerView style={{ flex: 1 }} initialPage={0} pageMargin={14}>

  <Pressable
    className="w-full h-36 md:h-60 rounded-2xl"
    key="1"
    onPress={() => console.log("CLICOU NO BANNER 1")}
  >

    <Image
      source={require("../assets/banner1.png")}
      className="w-full h-36 md:h-60 rounded-2xl"
    />
  </Pressable>

  <Pressable
    className="w-full h-36 md:h-60 rounded-2xl"
    key="2"
    onPress={() => console.log("CLICOU NO BANNER 2")}
  >

    <Image
      source={require("../assets/banner2.png")}
      className="w-full h-36 md:h-60 rounded-2xl"
    />
  </Pressable>

</PagerView>
</View>
);
}

```

E agora importe no index.tsx do app

./src/app/index.tsx

```
import { Text, View, ScrollView } from "react-native";
import Constants from 'expo-constants'

import { Header } from "../components/header";
import { Banner } from "../components/banner";

const statusBarHeight = Constants.statusBarHeight;

export default function Index() {
  return (
    <ScrollView
      style={{ flex: 1 }}
      className="bg-slate-200"
      showsVerticalScrollIndicator={false}
    >
      <View className="w-full px-4" style={{ marginTop: statusBarHeight + 8 }}>
        <Header />
        <Banner />
      </View>
    </ScrollView>
  );
}
```

Agora veja se está mostrando o banner no emulador.



O próximo passo é implementar a nossa barra pesquisa.

./src/components/search/index.tsx

```
import { View, TextInput } from 'react-native';
import { Feather } from '@expo/vector-icons'

export function Search() {
  return (
    <View className='w-full flex-row border border-slate-500 h-14 rounded-full items-center gap-2 px-4 bg-transparent'>
      <Feather name='search' size={24} color="#64748b"/>

      <TextInput
        placeholder="Procure sua comida..."
        className='w-full h-full flex-1 bg-transparent'
      />
    </View>
  );
}
```

./src/app/index.tsx

```
import { Text, View, ScrollView } from "react-native";
import Constants from 'expo-constants'
```

```
import { Header } from "../components/header";
import { Banner } from "../components/banner";
import { Search } from "../components/search";

const statusBarHeight = Constants.statusBarHeight;

export default function Index() {
  return (
    <ScrollView
      style={{ flex: 1 }}
      className="bg-slate-200"
      showsVerticalScrollIndicator={false}
    >
      <View className="w-full px-4" style={{ marginTop: statusBarHeight + 8 }}>
        <Header />
        <Banner />
        <Search />
      </View>
    </ScrollView>
  );
}
```



Agora vamos criar o componente responsável por renderizar as nossas sections que aqui no nosso caso serão três.

`./src/components/section/index.tsx`

```
import { View, Text, Pressable } from 'react-native';

interface Props{
  name: string;
  size: "text-lg" | "text-xl" | "text-2xl";
  label: string;
  action: () => void;
}

export function Section({ name, size, label, action }: Props) {
  return (
    <View className='w-full flex flex-row items-center justify-between px-4'>
      <Text className={`$${size} font-semibold my-4 self-start`}>
        {name}
      </Text>

      <Pressable onPress={action}>
```

```
    <Text>{label}</Text>
  </Pressable>
</View>
);
}
```

Da forma como foi construída, podemos usá-la de forma dinâmica, pois no export default estamos passando propriedades para que ao importar no index do app nós possamos personalizar cada vez que utilizarmos esse componente. Esse componente vai esperar receber “**{ name, size, label, action }**” que passaremos no index.tsx do app.

./src/app/index.tsx

```
import { Text, View, ScrollView } from "react-native";
import Constants from "expo-constants";

import { Header } from "../components/header";
import { Banner } from "../components/banner";
import { Search } from "../components/search";
import { Section } from "../components/section";

const statusBarHeight = Constants.statusBarHeight;

export default function Index() {
  return (
    <ScrollView
      style={{ flex: 1 }}
      className="bg-slate-200"
      showsVerticalScrollIndicator={false}
    >
      <View className="w-full px-4" style={{ marginTop: statusBarHeight + 8 }}>
```



```
<Header />

<Banner />

<Search />
</View>

<Section
  name="Comidas em alta"
  label="Veja mais"
  action={() => console.log("CLICOU NO VEJA MAIS")}
  size="text-2xl"
/>

<Section
  name="Famosos no DevFood"
  label="Veja todos"
  action={() => console.log("CLICOU NO Famosos")}
  size="text-xl"
/>

<Section
  name="Restaurantes"
  label="Veja todos"
  action={() => console.log("CLICOU NO RESTAURANTES")}
  size="text-xl"
/>
</ScrollView>

);
```

}

No emulador deverá estar assim:



Agora, vamos simular como se o nosso projeto estivesse consumindo uma api. Para isso vamos criar o que será o nosso banco de dados. Na raiz do projeto crie o arquivo “**db.json**”

db.json

```
{
  "restaurants": [
    {
      "id": "1",
      "name": "KFC",
      "image": "https://i.imgur.com/sZ1Jye3.png"
    },
  ],
}
```

```
{
  "id": "2",
  "name": "McDonald's",
  "image": "https://i.imgur.com/su4iB3p.png"
},
{
  "id": "3",
  "name": "Outback",
  "image": "https://i.imgur.com/jg3GJVN.png"
},
{
  "id": "4",
  "name": "Sushi Dev",
  "image": "https://i.imgur.com/ffYWFBX.png"
},
{
  "id": "5",
  "name": "Burguer Grill",
  "image": "https://i.imgur.com/kmQvG2T.png"
},
{
  "id": "6",
  "name": "Marmitas express",
  "image": "https://i.imgur.com/uXRuDFD.png"
}
],
"foods": [
  {
    "id": "1",
```

```
"name": "Frango assado",  
"price": 39.90,  
"time": "50-60 min",  
"delivery": 5.99,  
"rating": 4.8,  
"image": "https://i.imgur.com/CEVUdju.png",  
"restaurantId": "1"
```

```
},
```

```
{
```

```
"id": "2",  
"name": "Marmita fit",  
"price": 29,  
"time": "30-40 min",  
"delivery": 2.99,  
"rating": 4.9,  
"image": "https://i.imgur.com/uXRuDFD.png",  
"restaurantId": "6"
```

```
},
```

```
{
```

```
"id": "3",  
"name": "Burguer Dev",  
"price": 40,  
"time": "50-60 min",  
"delivery": 5.99,  
"rating": 4.5,  
"image": "https://i.imgur.com/O5c3lyU.png",  
"restaurantId": "5"
```

```
},
```

```
{
  "id": "4",
  "name": "Picanha Grill",
  "price": 36,
  "time": "50-60 min",
  "delivery": 5.99,
  "rating": 4.4,
  "image": "https://i.imgur.com/0UjG3wk.png",
  "restaurantId": "3"
},
{
  "id": "5",
  "name": "Açaí 300ml",
  "price": 20,
  "time": "20-30 min",
  "delivery": 2.99,
  "rating": 4.4,
  "image": "https://i.imgur.com/BaUUCkc.png",
  "restaurantId": "3"
}
]
```

Em seguida, abra o cmd do windows e execute o comando “ipconfig” para obter o ipv4 do seu computador.



Copie o número do ipv4 e substitua no código onde está “localhost” no código que criaremos mais abaixo que ficará em `./src/components/trending/index.tsx` .

Agora abra uma segunda aba no seu terminal e execute o seguinte comando para iniciar o banco da nossa aplicação: “npx json-server db.json”

Com o banco rodando vamos prosseguir. Crie a pasta **trending** dentro de src e dentro de trending, crie a pasta **food** e nela o arquivo **index.tsx**

./src/components/trending/food/index.tsx

```
import { View, Pressable, Text, Image } from 'react-native';
import { Ionicons } from '@expo/vector-icons'

import { FoodProps } from '..'

export function CardHorizontalFood({ food }: { food: FoodProps }) {
  return (
    <Pressable className='flex flex-col rounded-xl relative'>
      <Image
        source={{ uri: food.image }}
        className="w-44 h-36 rounded-xl"
      />

      <View className='flex flex-row bg-neutral-900/90 gap-1 rounded-full absolute top-2 right-3 px-2 py-1 items-center justify-center'>
        <Ionicons name="star" size={14} color="#ca8a04"/>
        <Text className='text-white text-sm'>{food.rating}</Text>
      </View>

      <Text className='text-green-700 font-medium text-lg'>R$ {food.price}</Text>
      <Text className='text-black mt-1'>{food.name}</Text>
      <Text className='text-neutral-600 text-sm'>{food.time} - R$ {food.delivery}</Text>
    </Pressable>
  );
}
```

Esse componente fica responsável por passar dinamicamente os dados de cada alimento que recebemos do banco de dados.

Agora dentro da pasta trending crie o arquivo index.tsx. Ele será responsável por armazenar os nossos alimentos. Não esqueça de trocar o “localhost” pelo seu ipv4.

./src/components/trending/index.tsx

```
import { useState, useEffect } from "react";
import { FlatList } from "react-native";
import { CardHorizontalFood } from "../food";

export interface FoodProps {
  id: string;
  name: string;
  price: number;
  time: string;
  delivery: number;
  rating: number;
  image: string;
  restaurantId: string;
}

export function TrendingFoods() {
  const [foods, setFoods] = useState<FoodProps[]>([]);

  useEffect(() => {
    async function getFoods() {
      const response = await fetch("http://localhost:3000/foods");
      const data = await response.json();
      setFoods(data);
    }
  })
}
```

```

    getFoods();
  }, []);

  return (
    <FlatList
      data={foods}
      renderItem={({ item }) => <CardHorizontalFood food={item} />}
      horizontal={true}
      contentContainerStyle={{ gap: 14, paddingLeft: 16, paddingRight: 16 }}
      showsHorizontalScrollIndicator={false}
    />
  );
}

```

E agora importe no index.tsx do app.

./src/app/index.tsx

```

import { Text, View, ScrollView } from "react-native";
import Constants from "expo-constants";

import { Header } from "../components/header";
import { Banner } from "../components/banner";
import { Search } from "../components/search";
import { Section } from "../components/section";
import { TrendingFoods } from "../components/trending";

const statusBarHeight = Constants.statusBarHeight;

export default function Index() {
  return (

```



```
<ScrollView
  style={{ flex: 1 }}
  className="bg-slate-200"
  showsVerticalScrollIndicator={false}
>
  <View className="w-full px-4" style={{ marginTop: statusBarHeight + 8 }}>
    <Header />

    <Banner />

    <Search />
  </View>
```

```
<Section
  name="Comidas em alta"
  label="Veja mais"
  action={() => console.log("CLICOU NO VEJA MAIS")}
  size="text-2xl"
/>
```

```
<TrendingFoods />
```

```
<Section
  name="Famosos no DevFood"
  label="Veja todos"
  action={() => console.log("CLICOU NO Famosos")}
  size="text-xl"
/>
```

```
<Section
```

```

    name="Restaurantes"

    label="Veja todos"

    action={() => console.log("CLICOU NO RESTAURANTES")}

    size="text-xl"

  />
</ScrollView>

);
}

```

Com isso deverá ser mostrado a seguinte tela no emulador.



Agora vamos fazer algo semelhante para as duas outras section no index.tsx da pasta app.

Dentro de src crie a pasta “**restaurants**” e nela a pasta “**horizontal**” e dentro o arquivo index.tsx

./src/components/restaurants/horizontal/index.tsx

```
import { View, Pressable, Text, Image } from 'react-native';
import { RestaurantsProps } from '..';

export function RestaurantItem({ item }: { item: RestaurantsProps }) {
  return (
    <Pressable
      className='flex flex-col items-center justify-center'
      onPress={() => console.log("CLICOU NO RESTAURANTE " + item.name)}
    >
      <Image
        source={{ uri: item.image }}
        className='w-20 h-20 rounded-full'
      />
      <Text
        className='text-sm mt-2 w-20 text-center leading-4 text-black'
        numberOfLines={2}
      >
        {item.name}
      </Text>
    </Pressable>
  );
}
```

Em seguida, na pasta **restaurants** crie o arquivo **index.tsx** e nele mude também o “localhost” para o número do seu **ipv4**.

./src/components/restaurants/index.tsx

```
import { FlatList } from "react-native";
import { useEffect, useState } from "react";
```

```
import { RestaurantItem } from "../horizontal";

export interface RestaurantsProps {
  id: string;
  name: string;
  image: string;
}

export function Restaurants() {
  const [restaurants, setRestaurants] = useState<RestaurantsProps[]>([]);

  useEffect(() => {
    async function getFoods() {
      const response = await fetch("http://localhost:3000/restaurants");
      const data = await response.json();
      setRestaurants(data);
    }

    getFoods();
  }, []);

  return (
    <FlatList
      data={restaurants}
      renderItem={({ item }) => <RestaurantItem item={item} />}
      horizontal={true}
      contentContainerStyle={{ gap: 14, paddingLeft: 16, paddingRight: 16 }}
      showsHorizontalScrollIndicator={false}
    />
  );
}
```

```
);  
}
```

E agora importe no index.tsx do app

./src/app/index.tsx

```
import { Text, View, ScrollView } from "react-native";  
import Constants from "expo-constants";  
  
import { Header } from "../components/header";  
import { Banner } from "../components/banner";  
import { Search } from "../components/search";  
import { Section } from "../components/section";  
import { TrendingFoods } from "../components/trending";  
import { Restaurants } from "../components/restaurants";  
  
const statusBarHeight = Constants.statusBarHeight;  
  
export default function Index() {  
  return (  
    <ScrollView  
      style={{ flex: 1 }}  
      className="bg-slate-200"  
      showsVerticalScrollIndicator={false}  
    >  
      <View className="w-full px-4" style={{ marginTop: statusBarHeight + 8 }}>  
        <Header />  
  
        <Banner />  

```

```
<Search />

</View>

<Section
  name="Comidas em alta"
  label="Veja mais"
  action={() => console.log("CLICOU NO VEJA MAIS")}
  size="text-2xl"
/>

<TrendingFoods />

<Section
  name="Famosos no DevFood"
  label="Veja todos"
  action={() => console.log("CLICOU NO Famosos")}
  size="text-xl"
/>

<Restaurants />

<Section
  name="Restaurantes"
  label="Veja todos"
  action={() => console.log("CLICOU NO RESTAURANTES")}
  size="text-xl"
/>
</ScrollView>

);
}
```

No seu emulador deverá ter a seguinte tela:



E agora vamos fazer o mesmo para a nossa última section. Dentro de **src** crie a pasta **list** e dentro dela a pasta **item** e o arquivo **index.tsx**

./src/components/list/item/index.tsx

```
import { View, Pressable, Text, Image } from 'react-native';  
import { RestaurantsProps } from '..'
```

```

import { Ionicons } from '@expo/vector-icons'

export function RestaurantItem({ item }: { item: RestaurantsProps }) {
  return (
    <Pressable className='flex flex-row items-center justify-start gap-2'>
      <Image
        source={{ uri: item.image }}
        className='w-20 h-20 rounded-full'
      />

      <View className='flex gap-2'>
        <Text className='text-base text-black leading-4 font-bold' numberOfLines={2}>
          {item.name}
        </Text>

        <View className='flex-row items-center gap-1'>
          <Ionicons name='star' size={14} color="#ca8a04" />
          <Text className='text-sm'>4.5</Text>
        </View>

      </View>
    </Pressable>
  );
}

```

E dentro da pasta **list** crie o arquivo **index.tsx** e não esqueça de também mudar “**localhost**” para o seu ipv4.

./src/components/list/index.tsx


```
import { View, Text } from "react-native";
import { useState, useEffect } from "react";
import { RestaurantItem } from "../item";

export interface RestaurantsProps {
  id: string;
  name: string;
  image: string;
}

export function RestaurantVerticalList() {
  const [restaurants, setRestaurants] = useState<RestaurantsProps[]>([]);

  useEffect(() => {
    async function getFoods() {
      const response = await fetch("http://localhost:3000/restaurants");
      const data = await response.json();
      setRestaurants(data);
    }

    getFoods();
  }, []);

  return (
    <View className="px-4 flex-1 w-full h-full mb-11 gap-4">
      {restaurants.map((item) => (
        <RestaurantItem item={item} key={item.id} />
      ))}
    </View>
  );
}
```

```
    </View>

  );
}
```

E agora vamos importar no index.tsx do app

./src/app/index.tsx

```
import { Text, View, ScrollView } from "react-native";
import Constants from "expo-constants";

import { Header } from "../components/header";
import { Banner } from "../components/banner";
import { Search } from "../components/search";
import { Section } from "../components/section";
import { TrendingFoods } from "../components/trending";
import { Restaurants } from "../components/restaurants";
import { RestaurantVerticalList } from "../components/list";

const statusBarHeight = Constants.statusBarHeight;

export default function Index() {
  return (
    <ScrollView
      style={{ flex: 1 }}
      className="bg-slate-200"
      showsVerticalScrollIndicator={false}
    >
```

```
<View className="w-full px-4" style={{ marginTop: statusBarHeight + 8 }}>
```

```
<Header />
```

```
<Banner />
```

```
<Search />
```

```
</View>
```

```
<Section
```

```
  name="Comidas em alta"
```

```
  label="Veja mais"
```

```
  action={() => console.log("CLICOU NO VEJA MAIS")}
```

```
  size="text-2xl"
```

```
/>
```

```
<TrendingFoods />
```

```
<Section
```

```
  name="Famosos no DevFood"
```

```
  label="Veja todos"
```

```
  action={() => console.log("CLICOU NO Famosos")}
```

```
  size="text-xl"
```

```
/>
```

```
<Restaurants />
```

```
<Section
```

```
  name="Restaurantes"
```

```
  label="Veja todos"
```

```
        action={() => console.log("CLICOU NO RESTAURANTES")}  
        size="text-xl"  
    />  
  
    <RestaurantVerticalList />  
  </ScrollView>  
);  
}
```

Com isso teremos a renderização da lista de restaurantes da nossa última section e deverá ficar assim no emulador:



Com isso, finalizamos esse projeto. Parabéns!!