

# React Native com Tailwind e Typescript


Criando um app com nossa stack favorita:

- React Native
- Expo Router
- Typescript
- NativeWind → [Site Oficial](#)

## Criando o App

1. Inicie um novo projeto:

```
npx create-expo-app@latest meu-app --example with-router  
cd meu-app  
touch tsconfig.json  
npx expo
```

2. Rode o app pelo menos uma vez para instalar o TS 

3. Tudo pronto para começar!



# Criando Arquivos Base

## index.tsx (Home):

```
import { Text } from 'react-native';

const Index = () => {
  return <Text>Eu sou o index de teste</Text>;
};

export default Index;
```




Este é o ponto de entrada do seu app.

## \_layout.tsx (Layout Global):

```
import { Stack } from 'expo-router';

export default function Layout() {
  return (
    <Stack screenOptions={{ headerShown: true }}>
      <Stack.Screen name="index" />
    </Stack>
  );
}
```

 Aqui você controla a navegação e pode importar estilos futuramente.

# Configurando o NativeWind

Instale tudo que precisa:

```
npx expo install nativewind tailwindcss react-native-reanimated react-native-safe-area-context
```

Crie o arquivo `tailwind.config.js` com o comando:

```
npx tailwindcss init
```

## ❑ Configurando `tailwind.config.js`

Adicione este código:

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ["../app/**/*.{js,jsx,ts,tsx}"],
  presets: [require("nativewind/preset")],
  theme: {
    extend: {},
  },
  plugins: [],
};
```

📌 Se preferir, troque `./app/**` por `./src/**`.

# ✨ Estilos Globais

Crie `styles/global.css` :

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

## Configurações Adicionais na Raiz

babel.config.js

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



## metro.config.js

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

const config = getDefaultConfig(__dirname);

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

# Configuração para Typescript

Crie `types/nativewind-env.d.ts` :

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

⚠️ Necessário para o `className` funcionar corretamente.

## Finalizando

Agora você pode usar `className` no React Native como no React! 🎉

```
<View className="bg-blue-500">  
  <Text className="text-white">Olá, NativeWind! 🙌</Text>  
</View>
```

## Referências

- **YT - Tailwind com Expo - Rodrigo Gonçalves**  
Aprenda sobre customização de cores e fontes.
- **YT - Tab Bar Routing - Simpn Grimm**  
Navegação e criação de Tab Bar.

## Alternativa: **react-native-tailwindcss**

### Instalação:

```
npm install react-native-tailwindcss
```

### Exemplo:

```
import { t } from 'react-native-tailwindcss';

export default function Teste() {
  return (
    <View style={[t.bgPurple300]}>
      <Text style={[t.bgGray400, t.textPurple500]}>Olá mundo! 🌍</Text>
    </View>
  );
}
```