

Bibliotecas para instalar:

```
tailwind-react-native-classnames';  
@react-navigation/native  
@react-navigation/native-stack  
react-native-nfc-manager (NfcManager)  
@react-native-async-storage/async-storage  
react-native-webview
```

ADICIONAR ESSAS LINHAS NO *AndroidManifest.xml*:

```
/projeto/android/app/src/main/AndroidManifest.xml  
  
<uses-permission  
android:name="android.permission.ACCESS_FINE_LOCATION" />  
<uses-permission  
android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

EXPLICAÇÃO DA PAGINAÇÃO:

RegistrarVisitante.tsx: Responsável por registrar os visitantes em uma página, e salva no banco de dados no supabase.

FuncoesNFC.tsx: Em teoria seria responsável pelas funções NFC (Write, Read, Delete), porém não funcionam de forma precisa na paginação. Então, as funções estão incluídas no ***PaginaProntuarios.tsx***.

PaginaProntuarios.tsx: Página que contém as informações dos visitantes inscritos no banco de dados, inclui também as funções do NFC, responsáveis pela a escrita e leitura das tags.

IMPORTANTE DESTACAR: OS VISITANTES ESTÃO SENDO ESCRITOS ATRAVÉS DE UM HANDLE DEEP LINK NAS TAGS NFC. O que é isso?

Basicamente um método de escrita no React Native que permite escrever um link nas tags que cria um localhost nas mesmas, isolando os 'dados' de cada visitante.

ProntuarioScreen.tsx: Mostra as telas dos dados dos visitantes que estão escritas nas tags. (Atualmente, está bugada por causa da mudança da paginação)

checkpoint.tsx: Página que define os checkpoints dos usuários (Policiais), em qual pavilhão eles estarão. (Atualmente define que o pavilhão 1 é o Hall, ou seja, apenas os policiais que estiverem no Hall (Pavilhão 1), poderão registrar visitantes.

Falta adicionar a modificação de que, os outros policiais em outros pavilhões, irão apenas checar e registrar a passagem dos visitantes pelo seu pavilhão.

login.tsx e **cadastro.tsx:** Fazem o login e o cadastro dos usuários (policiais), talvez seja melhor tirar ou melhorar a definição do cadastro de cada usuário. Deve-se também adicionar melhores designs nessas partes. Na verdade, em todas as telas.

No **login.tsx**, ele devolve um padrão de verificação para o **UserContext.tsx**, que será utilizado para pegar o usuário logado (Para adicionar em padrões de segurança e registrar as movimentações dos usuários).

UserContext.tsx: Pega o usuário logado.

inputprontuario.tsx: Antigo arquivo com todas as funções.

Configurações do Supabase: `/projeto/services/supabase.ts`