

Como abrir:

É necessário ter Node 8 ou maior para executar todos os comandos.

Domusapi:

É o primeiro projeto que deve ser aberto, execute o comando `npm install` para que as dependências sejam instaladas, logo após execute `npm start`.

Possíveis erros: Falha de conexão de banco de dados. Confira a pasta `config` e o arquivo `db.json`, onde as credencias de seu MSSQL estão resguardadas.

Após a api entrar em execução, uma mensagem em seu terminal apareça contendo seu `IPV4:PORTA`, copie essa informação pois usaremos para abrir os demais projetos.

Não se esqueça de executar o arquivo `TCC.sql`, tendo em vista que é necessário executar em 3 etapas, a `CREATE DATABASE TCC`, a `USE TCC` e a etapa onde são selecionados todos os `create table`.

Domusweb:

É o segundo projeto a ser executado, opcionalmente pois contem as mesmas funcionalidades de `domusapp`, é necessário abrir o arquivo `./src/redux/api.js` e o alterar pelo IP copiado na etapa anterior.

Logo em seguida salve e de `npm install` para instalar as dependencias(Esse processo pode demorar, e muito, portanto recomendo o uso do pacote `yarn`). Após dependências instaladas, existem duas formas a de se utilizar a aplicação. Em modo de desenvolvimento e em modo de produção, para modo de desenvolvimento execute `npm run start` ou `yarn start`, a aplicação estará pronta para ser acessa em `localhost:3000`, para modo de produção execute `npm run build` ou `yarn build`, aguarde o processo terminar e logo após instale o pacote `serve` globalmente

Através do comando `npm install -g serve`, terminado execute, `serve -s build` dentro da raiz do projeto, a aplicação estará pronta para ser acessada em `localhost:5000`.

Domusapp:

Essa etapa é complexa pois precisa de um ambiente muito especializado.

Em caso de Mac(Apple) para teste iOS:

Certifique-se que possui o `xcode` instalado e que pelo menos uma vez foi aberto o simulador de iOS. Após essa verificação, abra o projeto em seu editor de preferencia ou terminal e execute `npm install`(Novamente aconselho uso do `yarn`), logo após rode `react-native run-ios`.

Em caso de Windows, Mac ou Linux para teste Android:

Certifique-se que possui `Android Studio` instalado em sua máquina e que o `sdk` de versão 26 do Android esteja instalado, após, defina em suas variáveis de ambiente o caminho para esse `sdk`. Em caso de emulador, certifique-se de instalar `genymotion` em `personal mode`,

Gerir então emuladores de `custom phone` da mesma versão Android instalada no `Android studio`, após isso abra a configuração e vá até a aba `ADB`, cole então o

caminho do adb de Android studio. Após essas configurações, abra o emulador e execute na pasta raiz do projeto, Npm install(Yarn ai pessoal) e logo após react-native run-android. Em caso de cabo, conecte seu celular com cabo usb e pule apenas a etapa de genymotion, execute react-native run-android.

Por fim, nas duas versões, seja iOS ou Android, entre em ./src/redux/api.js e altere pelo IP copiado na etapa inicial., já na api do portão, só é possível o teste junto com o dispositivo desenvolvido com raspberry pi 3, pois ele é quem provê essa api para abertura da porta.

Para mais informações sobre emuladores e ambiente react-native: <https://rocketseat.com.br/assets/files/ambiente-de-desenvolvimento-rn.pdf> (Detalhe, esse arquivo disponibilizado a público para compartilhamento de conhecimento, pertence a Rocketseat e portanto serve de citação para a configuração do usuário testador). Em caso de dúvida ou erro ao iniciar: ceo.paludetto@gmail.com Carlos Eduardo, líder do projeto.