

Passo a Passo de Conexão ao Aplicativo.

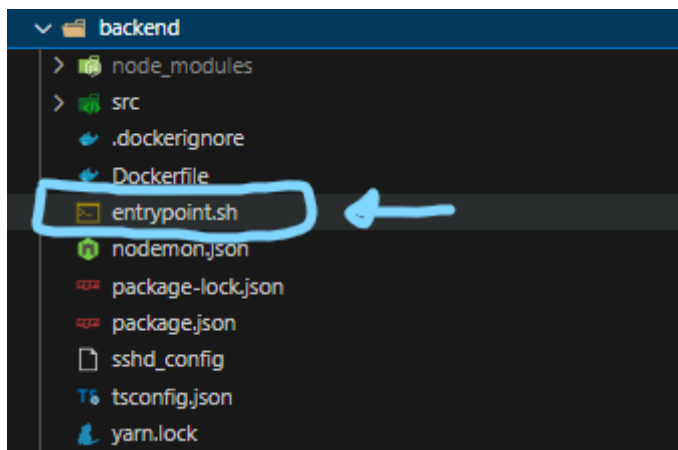
Versão de Desenvolvimento

Iniciando a API local:

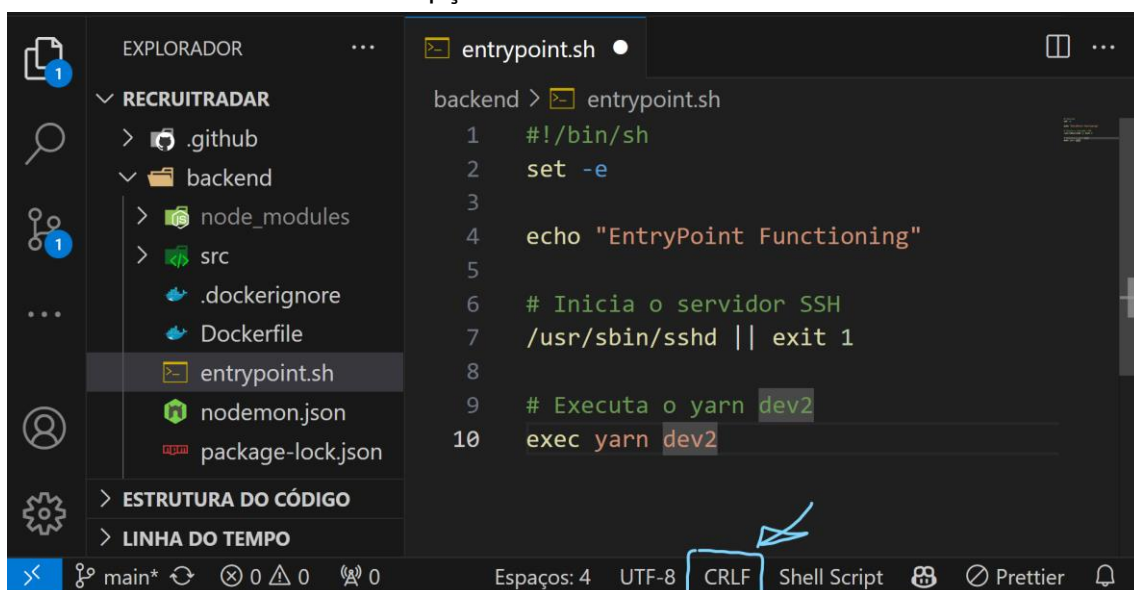
1) Clonar o repositório da aplicação na sua máquina local:

[Clonar um repositório do GitHub para o GitHub Desktop - GitHub Docs](#)

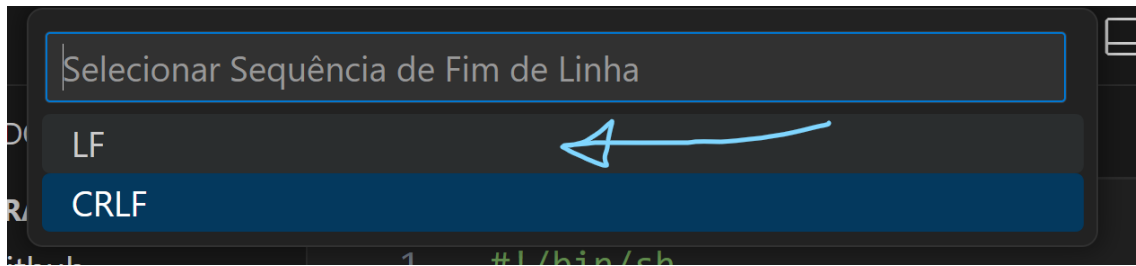
2) Editar um arquivo de conexão que é alterado devido a clonagem de repositório do GIT.



Selecionar na barra inferior a opção CRLF:

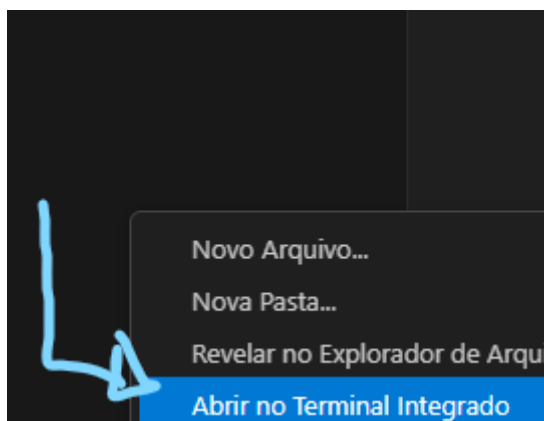
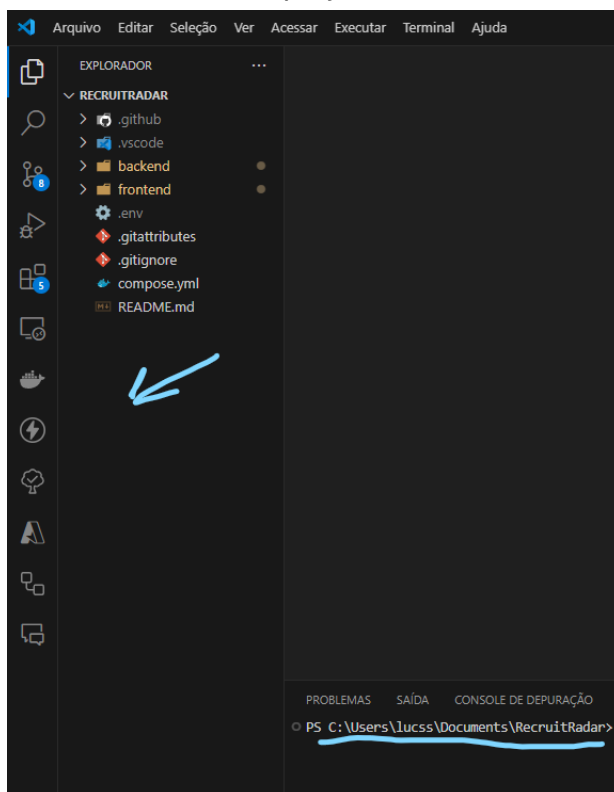


selecionar a opção LF:



2) Após este ajuste, abra o terminal na pasta raiz da aplicação:

Selecionar a raiz do projeto:



Digitar o comando az acr login --name recruitradar

```
az acr login --name recruitradar
```

Obs: para rodar esse comando é necessário o [az CLI na máquina local](#).

Digitar o comando Docker compose up:

```
PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO  TERMINAL  PORTAS  AZURE  COMENTÁRIOS
PS C:\Users\lucss\Documents\RecruitRadar> docker compose up
```

É necessário a conexão com o az acr, pois é onde esta armazenado as imagens do Docker do projeto.

Esperar criar o container e conexão a rede e depois esperar os dois logs que o server está funcionando e o data source está conectado.

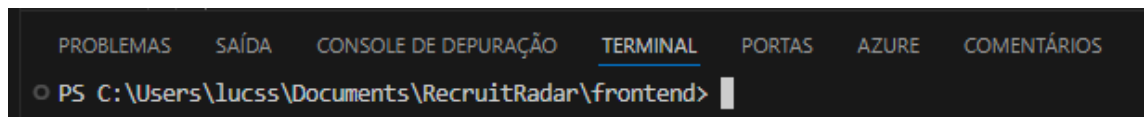
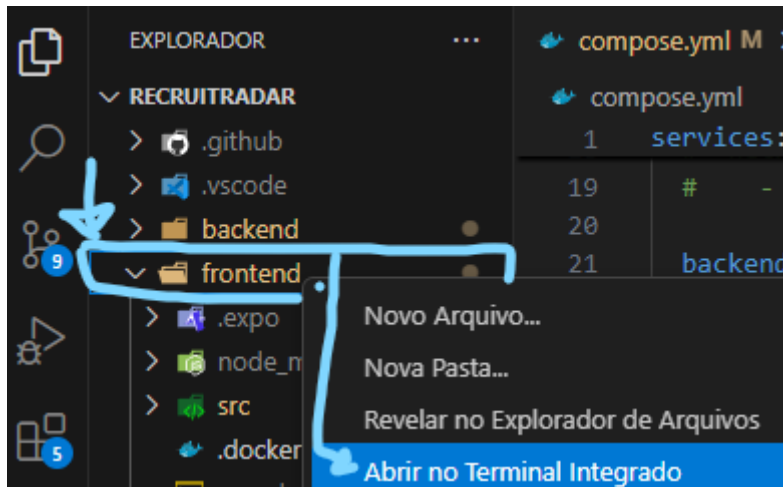
```
PS C:\Users\lucss\Documents\RecruitRadar> docker compose up
[+] Running 2/2
 ✓ Network recruitradar_dev Created
 ✓ Container backend_node Created
Attaching to backend_node
backend_node | EntryPoint Functioning
backend_node | yarn run v1.22.22
backend_node | $ nodemon -L --exitcrash src/config/server.ts
backend_node | [nodemon] 3.1.7
backend_node | [nodemon] to restart at any time, enter `rs`
backend_node | [nodemon] watching path(s): src/**/*.
backend_node | [nodemon] watching extensions: ts,js
backend_node | [nodemon] starting `ts-node ./src/config/server.ts src/config/server.ts`
backend_node | Server is running!
backend_node | Data Source has been initialized!
```

OBSERVAÇÕES:

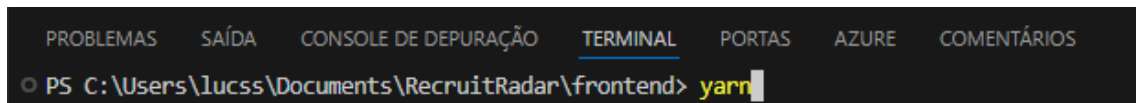
- É necessário que esteja instalado o Docker desktop na máquina local: [Docker - Instalação, Configuração e Primeiros Passos | balta.io](#)
- Primeira inicialização, será demorada, devido a necessidade de download da imagem direto do azure repositories e depois a instalação do mesmo e por fim a inicialização. Com o comando Docker compose up, tudo isso deverá ocorrer de maneira automática.

Iniciando a Aplicação local:

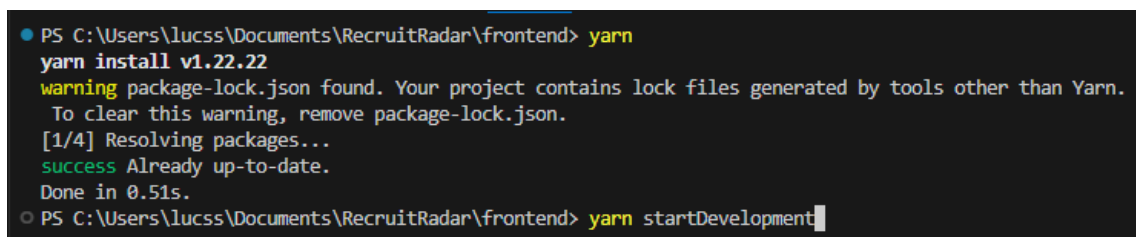
Abra o terminal na pasta frontend



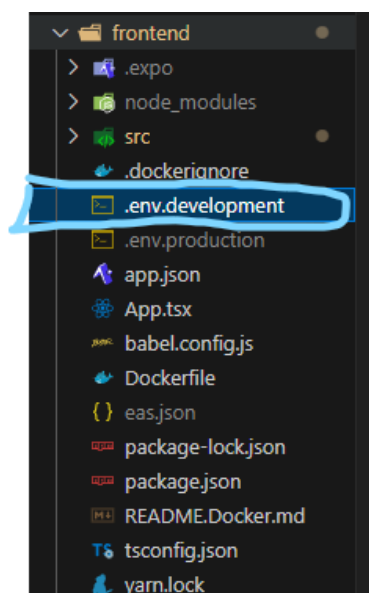
Digite **yarn**



espere a instalação dos pacotes para digitar, **yarn startDevelopment**:



OBS: verifique se na pasta raiz do /frontend esteja o arquivo **.env.development**:



Caso não esteja, você pode criar localmente, para conseguir rodar a aplicação, criando o seguinte arquivo:

Direito do mouse:



Digite, estas variáveis no arquivo e depois salve o arquivo:

EXPO_PUBLIC_BASEURL = localhost:3000

EXPO_PUBLIC_GOOGLE_API_KEY =
AlzaSyAY29bRdTYObjae7dMm4xo5uBSMiItmHOI

EXPO_PUBLIC_NODE_ENV = development

```
frontend > .env.development
1 EXPO_PUBLIC_BASEURL = localhost:3000
2 EXPO_PUBLIC_GOOGLE_API_KEY = AIzaSyAY29bRdTYObjae7dMm4xo5uBSMiItmHOI
3 EXPO_PUBLIC_NODE_ENV = development
```

Depois de executar o comando, espere a inicialização da aplicação na sua versão de desenvolvimento:

```
PS C:\Users\lucss\Documents\RecruitRadar\frontend> yarn startDevelopment
yarn run v1.22.22
$ NODE_ENV=development expo start
Starting project at C:\Users\lucss\Documents\RecruitRadar\frontend
env: load .env.development
env: export EXPO_PUBLIC_BASEURL EXPO_PUBLIC_GOOGLE_API_KEY EXPO_PUBLIC_NODE_ENV
Starting Metro Bundler
The following packages should be updated for best compatibility with the installed expo version:
@react-native-async-storage/async-storage@2.0.0 - expected version: 1.23.1
@react-native-community/datetimepicker@8.2.0 - expected version: 8.0.1
@react-native-community/slider@4.5.4 - expected version: 4.5.2
@react-native-picker/picker@2.8.1 - expected version: 2.7.5
react@18.3.1 - expected version: 18.2.0
react-dom@18.3.1 - expected version: 18.2.0
react-native@0.75.4 - expected version: 0.74.5
react-native-gesture-handler@2.20.0 - expected version: ~2.16.1
react-native-safe-area-context@4.11.1 - expected version: 4.10.5
react-native-screens@3.34.0 - expected version: 3.31.1
@types/react@18.3.11 - expected version: ~18.2.79
typescript@5.6.3 - expected version: ~5.3.3
Your project may not work correctly until you install the expected versions of the packages.

> Metro waiting on exp://192.168.100.83:8081
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Web is waiting on http://localhost:8081

> Using Expo Go
> Press s | switch to development build

> Press a | open Android
> Press w | open web
```

Agora existe diversas opções:

Apertar A para rodar em emulador android;

Apertar W para caso, queria ver uma versão previa no navegador;

Ou, de preferência, scanear o código na versão mobile do expo:

[Como instalar e configurar o Expo do React Native | Alura](#)

Tópico: “Programando direto no celular”