



## Preparando o ambiente: Criando o projeto

### Criando o projeto

Com o React Native devidamente instalado em nosso ambiente de produção, começaremos a desenvolver nosso aplicativo. Nos passos anteriores, disponibilizamos orientações traduzidas para instalação em vários sistemas operacionais, as quais podem ser encontradas em inglês também na parte "React Native CLI Quickstart" neste link.

Aqui em nossa aula, usaremos o Windows com emulador do Android.

Depois de termos instalado o Node, Python2 e JDK além do React Native, iniciaremos nosso trabalho de fato. Abriremos o PowerShell com o caminho da pasta onde o projeto será criado. Em seguida na mesma linha, escreveremos React-native com init para iniciarmos o projeto Lighteria na versão 0.60.4 específica, pois é interessante trabalharmos com a mesma para não termos problemas, afinal algumas alterações podem acontecer ao longo das atualizações.

```
React-native init Lighteria - -version 0.60.4
```

[COPIAR CÓDIGO](#)

Ao executarmos, o sistema começará a baixar as dependências. Não usaremos o yarn que acelera esse download por enquanto, mas não é essencial utilizá-lo.

Terminado este processo, o PowerShell fará várias operações e baixará diversos pacotes para nosso projeto. Ao final, apresentará instruções de execução para iOS e Android. Primeiro, ao executarmos o comando ls na linha seguinte e criaremos o diretório Lighteria. Também precisaremos do emulador do Visual Studio Code

aberto, depois escreveremos `cd Lighteria` para entrarmos na pasta, e, depois de limparmos a tela, por fim rodaremos o React-native com `run-android`.

Com isso, começaremos a executar e buildar o nosso aplicativo zerado. Em seguida, o sistema instanciará alguns processos e subirá o servidor Node que fará a comunicação entre nosso código e o emulador.

Uma nova janela se abrirá indicando a execução do Metro Bundler na porta 8081. Desta forma, não precisaremos recompilar o aplicativo sempre que atualizarmos nosso texto direto no emulador, pois o servidor conseguirá assimilar as novas pastas, informações e arquivos.

O deixaremos rodando minimizado por enquanto. Sempre que fizermos o comando `run-android` ou `run-ios`, o sistema subirá o Bundler.

No PowerShell, veremos a mensagem de sucesso do build de nosso aplicativo, e poderemos iniciá-lo no simulador. Inclusive, poderemos ver seu ícone escrito "Lighteria" na tela inicial do emulador.

Abrindo o aplicativo, receberemos uma mensagem de boas vindas ao React com algumas instruções; o primeiro passo é editar o arquivo `App.js` para mudarmos esta página inicial. Para vermos as alterações, bastará apertarmos a letra "R" do teclado duas vezes para darmos reload. Assim, o BUNDLE fará a leitura de `index.js` e pegará as informações do Node.

Se fecharmos a janela do servidor e apertarmos "R + R" novamente, o simulador dará um erro, pois tentará pegar os dados de um local que não existe.

Para conseguirmos dar continuidade, não precisaremos fechar a aplicação e todos os outros programas para buildarmos de novo, bastará apenas escrevermos `React-native` com `start` dentro do diretório do projeto para subirmos o servidor novamente e lermos as dependências.

Com isso, conseguiremos atualizar o emulador apertando duas vezes a tecla "R" do teclado.

Abriremos uma nova instância do PowerShell, visto que o atual está rodando o servidor Node com Metro Bundler e não poderá ser alterado agora.

Na nova janela, escreveremos `cd` apenas. Depois, abriremos o Explorador de Arquivos, encontraremos o diretório "Lighteria", clicaremos sobre a pasta do projeto e a arrastaremos até à nova instância do PowerShell.

Com isso, traremos o caminho exato para a linha de comando. Dentro, digitaremos `code .` para abrirmos o trabalho no Visual Studio Code.

A seguir, começaremos a entender as alterações no servidor.