**React Native**

# Introdução

React Native é uma ferramenta que permite a utilização de JavaScript para criar aplicações mobile. Sua diferença está na produção de interface nativa (android, ios), não tem web view, não tem browser rodando por baixo e interface web sendo criada – CSS, HTML. É uma ponte entre o código nativo (Java, Objective C) e o JS, forma de manipular a interface nativa utilizando JS. Faz com que o dispositivo entenda o JS, ao invés de fazer a conversão, utilizando a dependência JavaScript core.

Continua utilizando o React.

# Instalação

Precisa de algumas dependências do Java e do Android SDK para conseguir rodar o projeto no emulador do android.

react-native init nome\_pasta : inicia um novo projeto

react-native run-android : rodar dentro da pasta do projeto

Para rodar o projeto não precisa fazer tudo de novo, basta abrir o cmd na pasta do arquivo e digitar: react-native start

Metro Bundler: lê todo o código js que criou na aplicação uni tudo em um único arquivo .js e transforma esse arquivo em uma forma que a aplicação nativa consiga entender.

# Pastas do Projeto

* Android: onde ficam as configurações e o código nativo da aplicação android. Utiliza apenas para fazer o deploy (gerar o executável da aplicação).
* IOS: mesma coisa do android.
* Node\_modules: todas as dependências instaladas
* App.json: determina como a nossa aplicação é exibia para o usuário final (display\_name: nome da aplicação, name: nome internamente para desenvolvimento).
* Index.js: arquivo introdutório da aplicação, principal, é carregado assim que a aplicação é montada (appRegistry: é utilizado uma única vez em toda a aplicação, registra um componente que é o arquivo inicial da aplicação além do index.js).
* Package.json: mesmo package.json que temos na web, determina quais são as dependências, dependências de desenvolvimento.
* Criar a pasta src para colocar todo o código da aplicação.
* Services: criar uma interface para consumir/receber dados (conexão com banco de dados, consumo de apis).

# Componentes

Componetização: é uma arquitetura em que separa o código da aplicação em vários componentes, muito utilizada em web e mobile.

É um conjunto de código entre: visualização (mostrado ao usuário final), código de lógica (js), estilização (semelhante ao CSS).

É escrito como uma classe e por padrão, você exporta a classe do seu arquivo (export default class). Sempre possui um componente por arquivo, não é recomendado ter mais de um componente por arquivo.

É recomendável criar componentes sempre quando você vai reusá-lo (ex: lista com itens, itens da lista, header, navegação por abas). TUDO na aplicação é componente.

Possui apenas um método obrigatório: render(). É a parte visual que o componente vai retornar para o usuário.

# Elementos

É necessário importar os componentes do react-native sempre quando for necessário (semelhante as tags HTML) (ex: button, TextInput)

* View: caixa sem estilização, criar caixas (semelhante ao div)

# Estilização

O CSS da aplicação é utilizado a classe StyleSheet do react-native. É feita utilizando o flexboxlayout do CSS3 – **pesquisar no canal do youtube da rocketseat sobre esses layouts pra react-native.**

StyleSheet.creat({objeto com todas as estilizações possíveis}) -> em que é possível acessá-las dentro de componentes.

Utiliza as vírgulas, e não ;, porque é um objeto JS e sem hífen – cC. Aspas duplas para valores Strings no CSS.

# Bibliotecas

## Navegação

Mais de uma rota: biblioteca -> react-navigation: utilizada para a navegação. Com o yarn instalado, execute o comando na pasta do projeto: yarn add react-navigation. Possui vários tipos de navegação. Existe navegação por abas, menu lateral.

O react navigation coloca um header na aplicação, pois ele faz a parte de estilização da parte de navegação automaticamen

O **createStackNavigator** oferece um tipo de navegação em stack, navegação por cliques em botões.

Parâmetro: rotas da navegação, configurações de estilo.

## Axios

Faz o papel de buscar dados da api.

Comando: yarn add axios