

# Workshop de React Native

Computação Móvel 2018/2019



# Organizado por



Ana Falcão



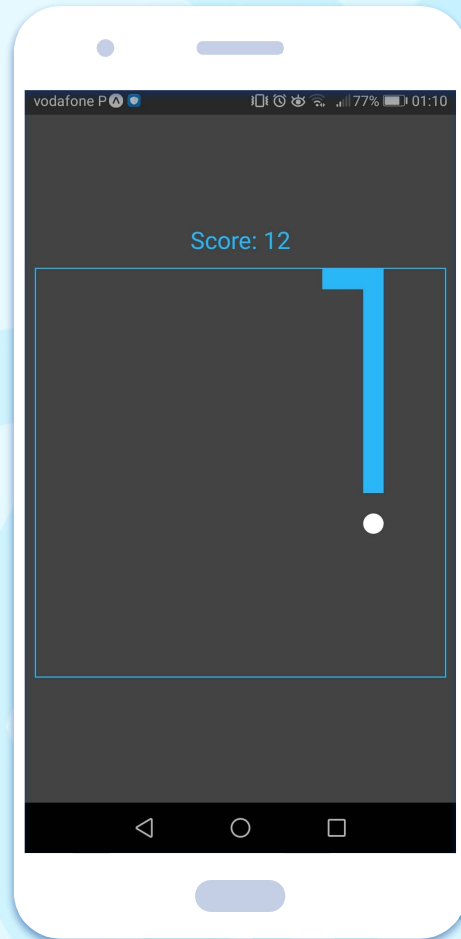
Gonçalo Lopes



João Alves

# O que vamos aprender?

- Introdução ao React Native
- Desenvolver o jogo do Snake



# React Native

- JavaScript framework, usa JavaScript e React
- Permite o desenvolvimento de apps Android e iOS
- “Look and feel native”: Usa os mesmos elementos de interface de iOS e Android



## Vantagens

- Aprendizagem rápida
- Desenvolvimento rápido
- Desenvolvimento Cross-platform
- Hot reload

## Desvantagens

- Recente
- “Constantly Changing Environment”
- Erros, instabilidade e falta de compatibilidade

# Ambiente de desenvolvimento



JavaScript runtime built  
on Chrome's V8  
JavaScript environment



Package Manager for JS;  
default in node.js

# A Framework

Native app

.ipa

.apk

React Native Libraries / ReactJS  
JavaScriptCore

Application Code

“bridge”

Platform UI  
Components

Core libraries

Style &  
Layout

Platform API

Native components

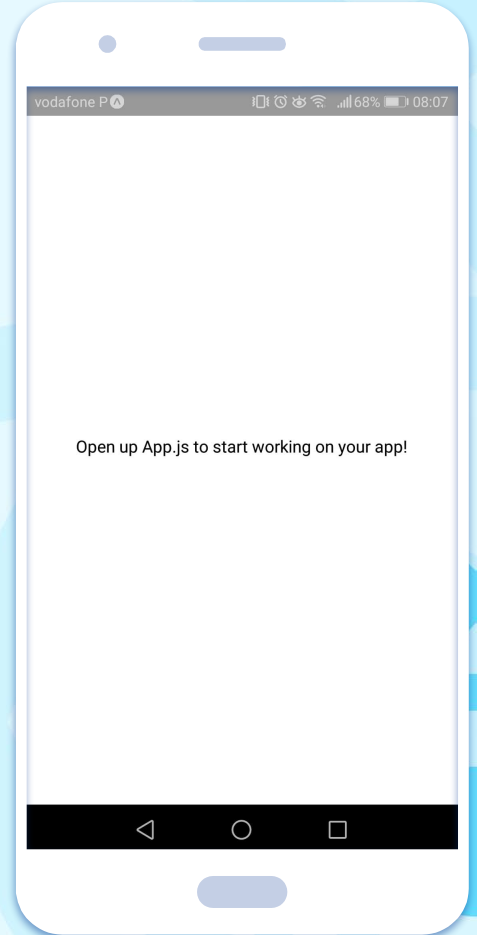
# Como começar?

```
npm install -g expo-cli
```

```
expo init AwesomeProject
```

```
cd AwesomeProject
```

```
npm start
```



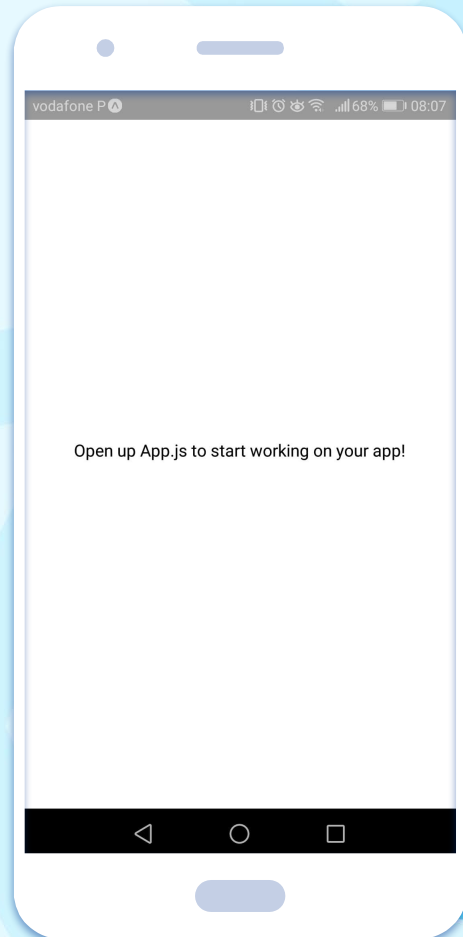


# Como Começar?

Iniciar um projeto !

```
npm install -g expo-cli  
expo init AwesomeProject
```

```
npm start
```



# Step 1:

Executar a aplicação !

```
npm install
```

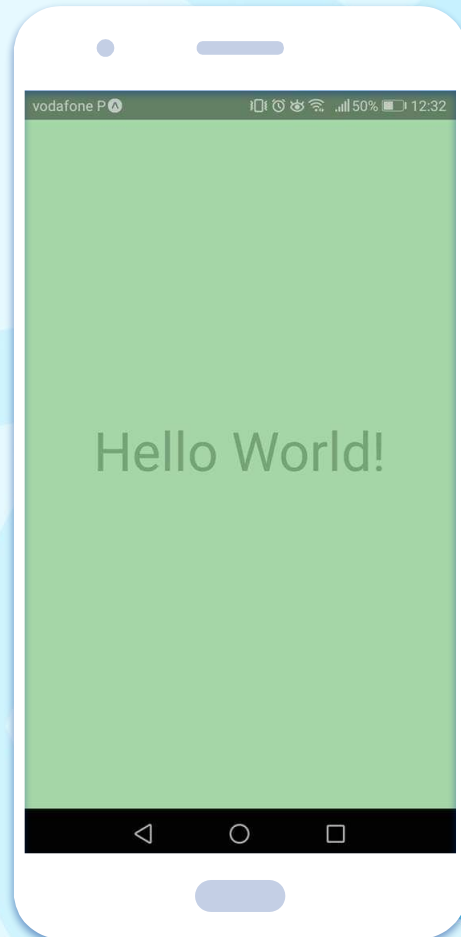
```
cd AwesomeSnake
```

```
npm start
```

Usar a rede do DEI!

Em Iphones usar uma app de QR code reading

Em Windows em vez de 'npm start' usar: expo start

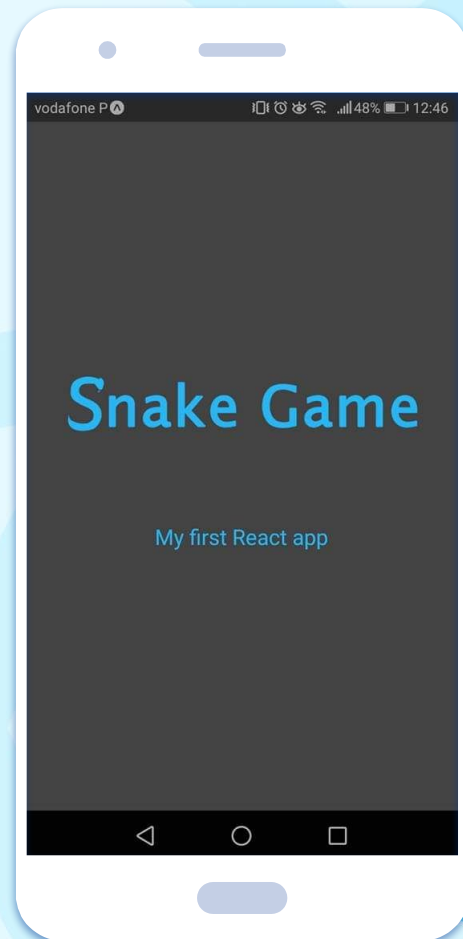


# Step 2:

Realizar o Menu

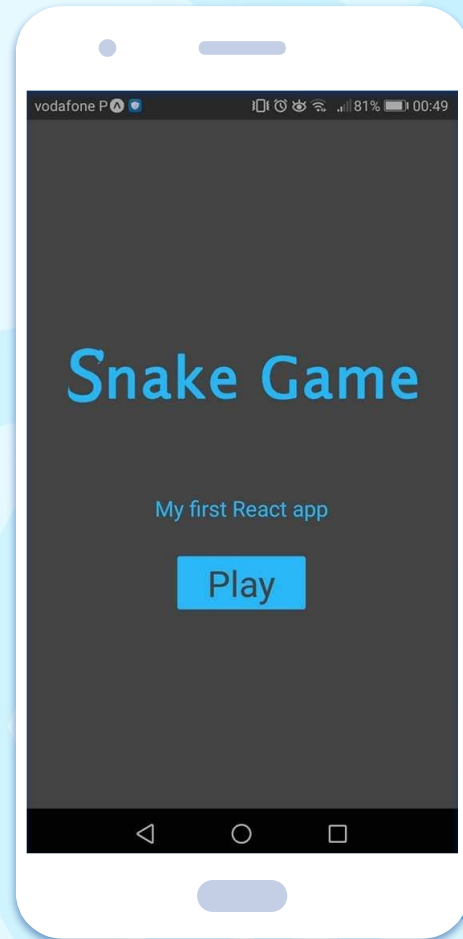
`<Image></Image>`

`<Text></Text>`



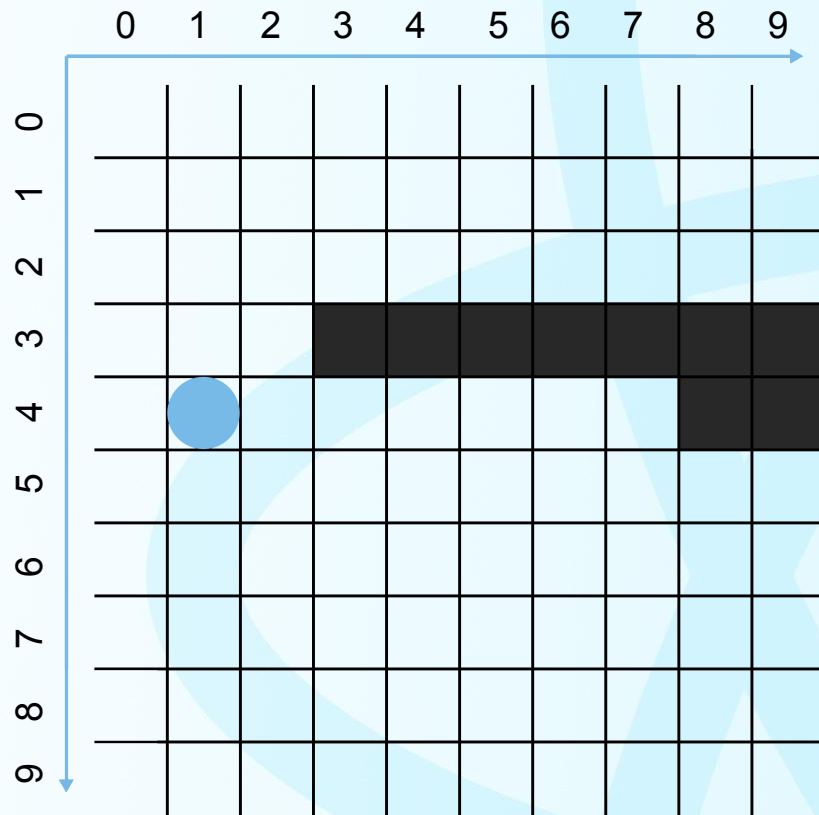
# Step 3:

Usar uma biblioteca:  
'react-router-native'



# Representação do jogo?

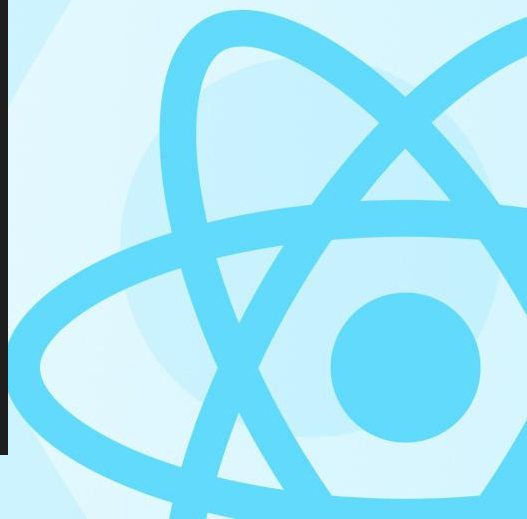




# 1) Props vs. State

- Os componentes podem ser customizados com parâmetros diferentes, sendo acedidos através do **Props**

```
1  export default class Welcome extends React.Component{  
2    render(){  
3      return (  
4        <View>  
5          <Text>Hello {this.props.name}!</Text>  
6        </View>  
7      );  
8    }  
9  }
```



## 1) Props vs. State

- **State** pertence ao componente e vai ser alterado pelo mesmo em tempo real! Por norma, o estado deve ser alterado via **setState()**
- **setState()** - muda o estado do componente e diz que o componente deve ser “re-rendered” com o novo estado

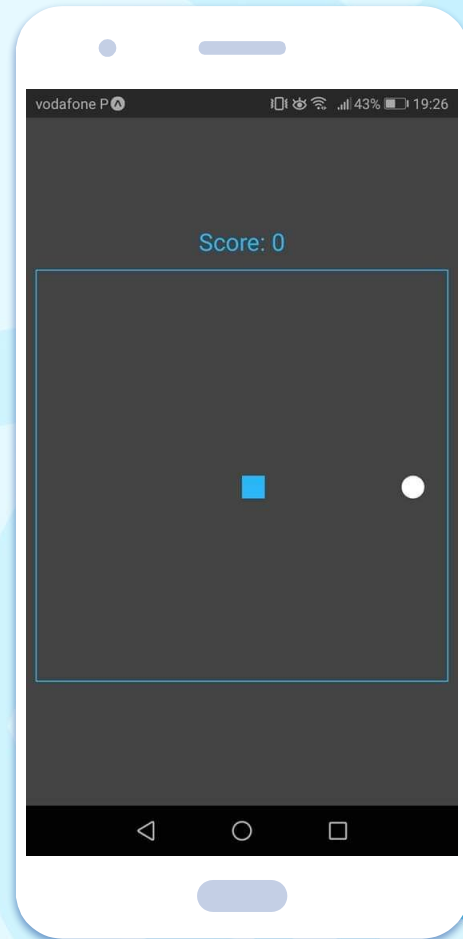


### 3) Props vs. State

```
14 export default class Login extends React.Component{
15   constructor(props) {
16     super(props);
17     this.state = { name: "Ana" };
18   }
19   render(){
20     return (
21       <View>
22         <Welcome name={this.state.name}></Welcome>
23         <Button
24           onPress={() => { this.setState( { name: "Pinguim!" } )} }
25           title="Press Me" />
26       </View>
27     );
28   }
29 }
```

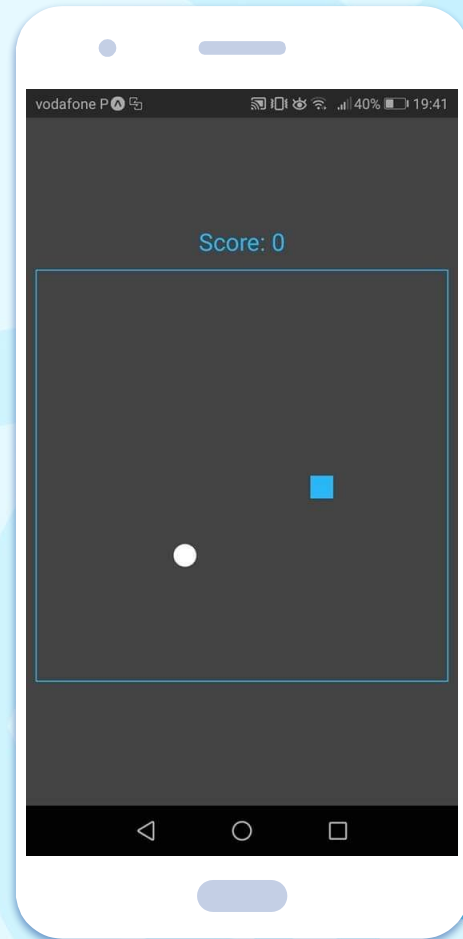
# Step 4:

## Criar área de jogo.



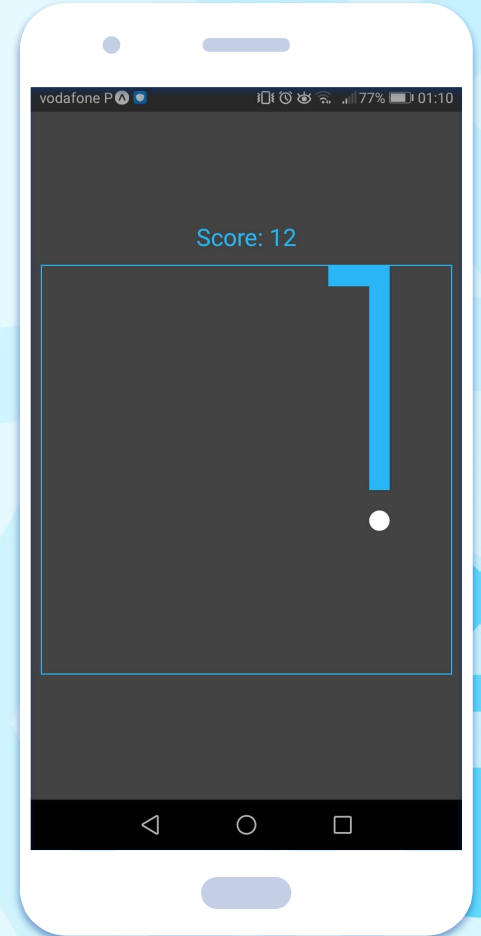
# Step 5:

Colocar o jogo a mover!



# Step 6:

## Handling swipes!



# CHALLENGE TIME !!!!!

Põe o teu nome no quadro!



# Obrigada pelo vosso tempo!

Esperemos que tenham gostado!

