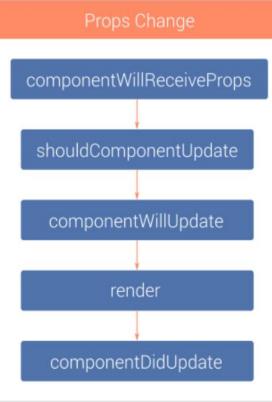@dielduarte

hotmart

farmaki

REACT

# REACT

- Virtual DOM
- Declarative syntax
- JSX

# JSX

```
...
render() {
  return(
    <MyButton color="Blue">
      Hello Facebook Developers Circle!
    </MyButton>
  );
}
...
```

# JSX

```
...
render() {
  return(
    React.createElement(
      MyButton,
      {color: 'blue'},
      'Hello Facebook Developers Circle!'
    )
  );
}
...
```

| First Render | Props Change | State Change |
|---|---|---|
| getDefaultProps | componentWillReceiveProps | shouldComponentUpdate |
| getInitialState | shouldComponentUpdate | componentWillUpdate |
| componentWillMount | componentWillUpdate | render |
| render | render | componentDidUpdate |
| componentDidMount | componentDidUpdate | |

**Unmount**

componentWillUnmount
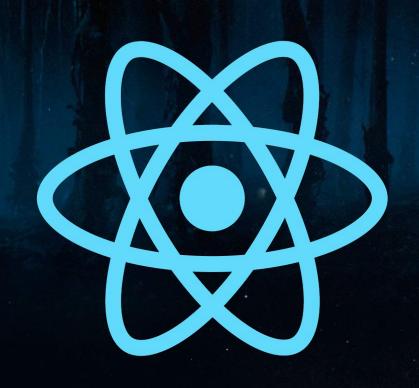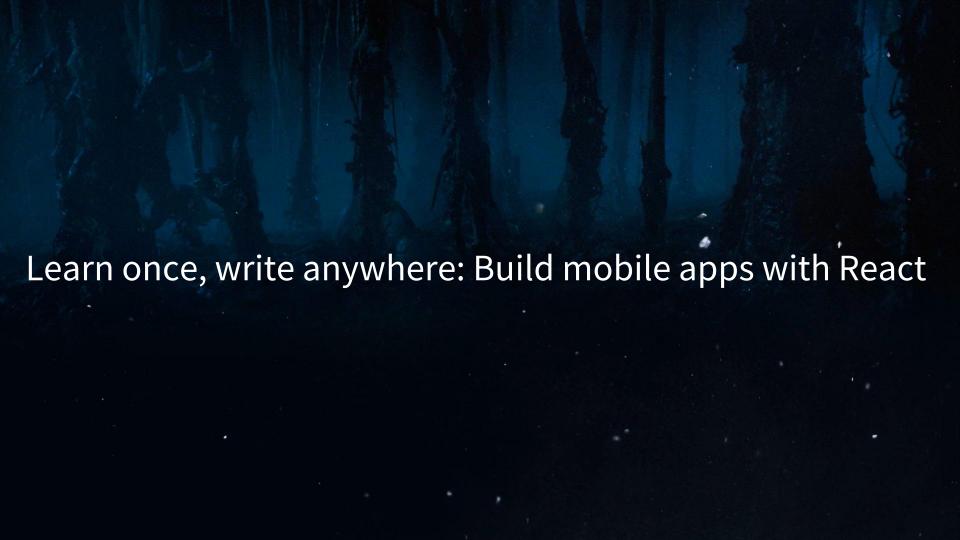
REACT NATIVE

Learn once, write anywhere: Build mobile apps with React
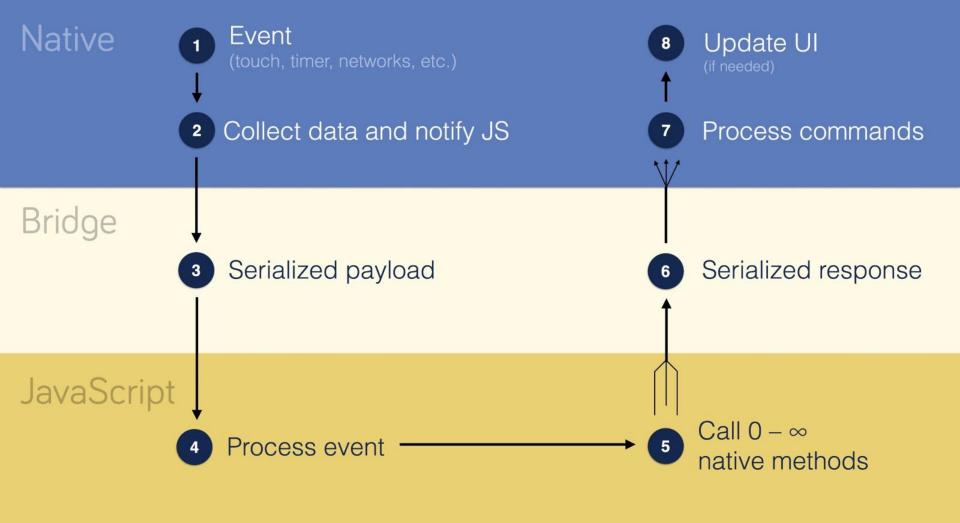
Então você quer dizer que react native
**Compila js para código nativo?**

O que acontece quando
**Abrimos um app react native ?**

**Native**

1 **Event**
(touch, timer, networks, etc.)

2 **Collect data and notify JS**

8 **Update UI**
(if needed)

7 **Process commands**

**Bridge**

3 **Serialized payload**

6 **Serialized response**

**JavaScript**

4 **Process event**

5 **Call 0 – ∞ native methods**

E como é feita
**essa comunicação?**

```objc
#import "CalendarManager.h"
#import <React/RCTLog.h>

@implementation CalendarManager

RCT_EXPORT_MODULE();

RCT_EXPORT_METHOD(addEvent:(NSString *)name location:(NSString
*)location)
{
  RCTLogInfo(@"Pretending to create an event %@ at %@", name,
location);
}
```

```
import { NativeModules } from 'react-native';

var CalendarManager = NativeModules.CalendarManager;
CalendarManager.addEvent('Birthday Party', '4 Privet Drive, Surrey');
```

OK, você me convenceu mas...
**como eu posso começar?**

REACT X REACT NATIVE

# REACT
## ( html markup )

- Block elements
  ( div, section..)

- Text elements
  ( h1, h2, p, span... )

- Links, buttons
  ( a, button.. )

- ...

**X**

# REACT NATIVE
## (react native core components)

- Block elements
  ( View )

- Text elements
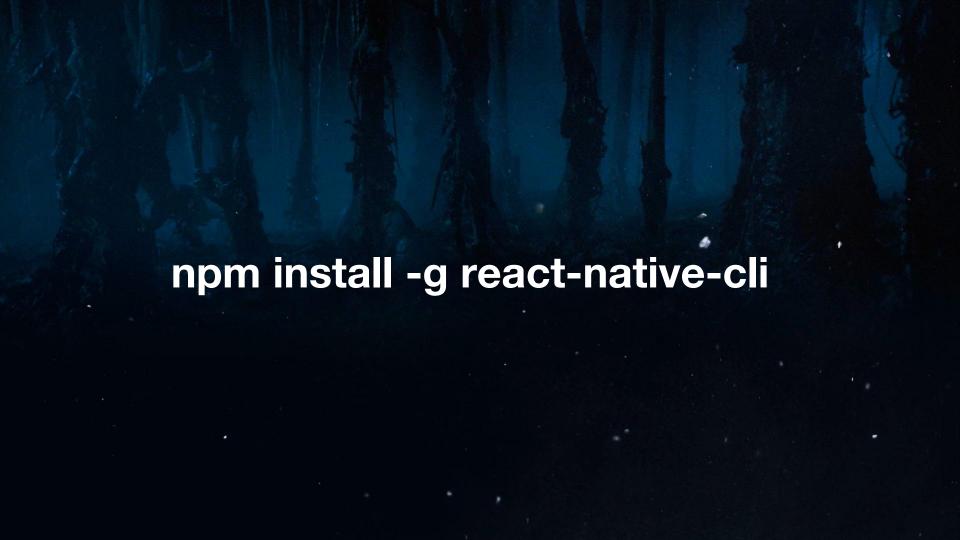  ( Text )

- Links, buttons
  ( Button,  TouchableHighlight )

- ...

# REACT

```jsx
import React, { PureComponent } from 'react';

...
 render() {
   return(
     <div>
       <h1>Hello!</h1>
     </div>
   );
 }
...
```

# REACT NATIVE

```jsx
import React, { PureComponent } from 'react';
import { View, Text } from 'react-native';

...
 render() {
   return(
     <View>
       <Text>Hello!</Text>
     </View>
   );
 }
...
```

npm install -g react-native-cli

# REACT NATIVE CLI

- start [options]
- run-ios [options]
- run-android [options]
- new-library [options]
- bundle [options]
- unbundle [options]
- eject [options]
- link [options] [packageName]
- unlink [options] <packageName>
- install [options] <packageName>
- uninstall [options] <packageName>
- upgrade [options]
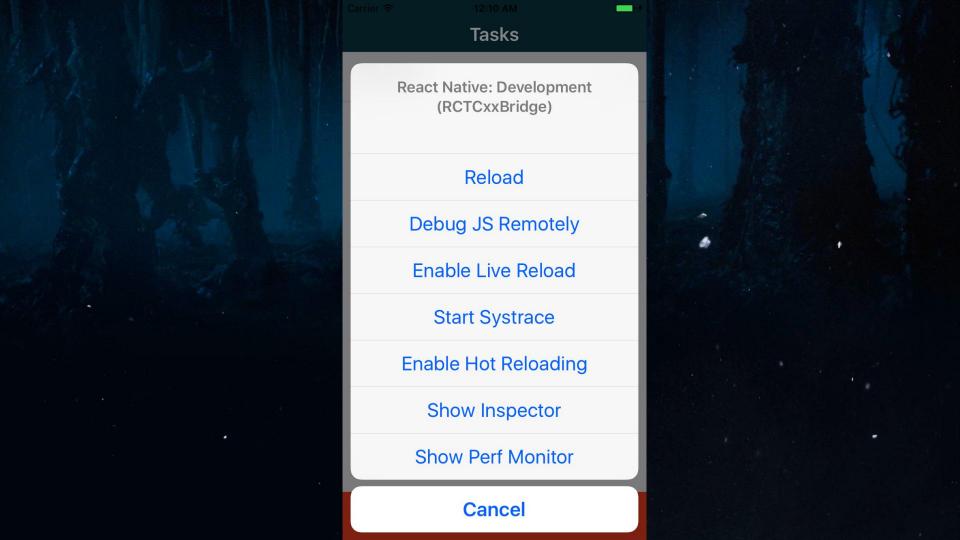- log-android [options]
- log-ios [options]

STYLESHEET

# STYLESHEET

```
import { StyleSheet } from 'react-native';
...

const styles = StyleSheet.create({
  red: {
    color: 'red',
  }
});


...
```

# STYLESHEET

```
import React, { PureComponent } from 'react';
import { View, Text } from 'react-native';

...
 render() {
   return(
     <View>
       <Text style={styles.red}>Hello!</Text>
     </View>
   );
 }
...
```

# DEBUG MENU

React Native: Development
(RCTCxxBridge)

Reload

Debug JS Remotely

Enable Live Reload

Start Systrace

Enable Hot Reloading

Show Inspector

Show Perf Monitor

Cancel

# REACT NATIVE APIs

**ALERT**

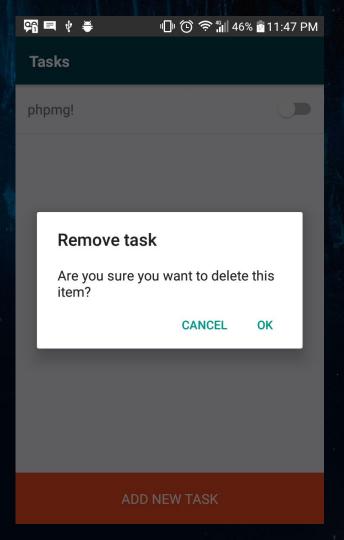APPSTATE

GEOLOCATION

LINKING
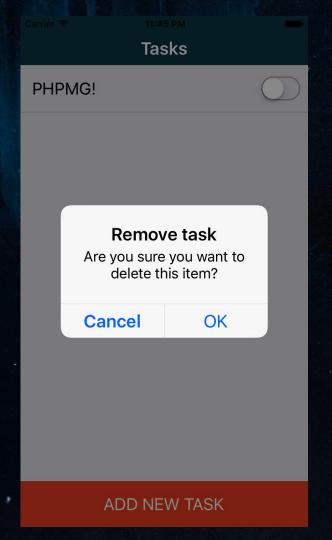
DIMENSIONS

**ASYNC STORAGE**

VIBRATIO
N

ANIMATED

# ALERT API

```
import { Alert } from 'react-native';

...

Alert.alert(
    'Remove task',
    'Are you sure you want to delete this item?',
    [
      {text: 'Cancel', style: 'cancel'},
      {text: 'OK', onPress: () => console.log('ok...')},
    ]
  );

...
```

# ASYNC STORAGE API

```javascript
import { AsyncStorage } from 'react-native';
...
try {
  await AsyncStorage.setItem('key', 'value');
} catch (error) {
 // Error saving data
}
...
```

# ASYNC STORAGE API

```
import { AsyncStorage } from 'react-native';
...
try {
  const value = await AsyncStorage.getItem('key');
  if (value !== null){
    console.log(value);
  }
} catch (error) {
 // Error retrieving data
}
...
```

**Perguntas?**