

React Native



**Tres en Raya con
React Native**

Tres en Raya con React Native

- Implementación de un **Tres en Raya** con **React Native**
- Partiremos de una aplicación de Tres en Raya implementada con **React**:
<https://github.com/ging/TresenrayaReactRedux/tree/TicTacToe>
- Objetivos
 - Crear un Tres en Raya con React Native **adaptando** la aplicación Tres en Raya de React
 - Implementar **funcionalidades específicas de React Native** (ej: navegación)
- Resultado final:
https://github.com/ging/TresenrayaReactRedux/tree/TicTacToe_RN
https://github.com/ging/TresenrayaReactRedux/tree/TicTacToe_RN2


















Tres en Raya: Primeros pasos

1. Instalar React Native
2. Crear una aplicación React Native
3. Ejecutar la aplicación en un emulador o dispositivo físico para comprobar que la instalación de React Native y la creación de la aplicación ha sido exitosa
4. Descargar como fichero ZIP y descomprimir la aplicación Tres en Raya implementada con React:
<https://github.com/ging/TresenrayaReactRedux/tree/TicTacToe>

Tres en Raya: Paso 5

En este paso vamos a copiar los **componentes** de la aplicación React

Para ello creamos una carpeta **app** en el directorio raíz de la aplicación React Native y copiamos dentro de ella el directorio **app/components** de la aplicación React

- ▼  AppReact
 - ▼  app
 - ▼  assets
 - ▼  css
 -  main.css
 - ▼  components
 -  **App.jsx**
 -  **Board.jsx**
 -  **Header.jsx**
 -  **Square.jsx**
 - ▶  vendors
 -  index.html
 -  main.js
 - ▼  styles
 -  app.css
 -  package.json
 -  ...

Tres en Raya: Paso 6

Renombrar los ficheros del directorio **components** de **.jsx a .js**

Renombrar el fichero **app/components/App.js** a **GameScreen.js**

La estructura de ficheros debería quedar acorde a la mostrada a la derecha

App.js debe ser modificado para renderizar GameScreen

En los componentes debemos **reemplazar las etiquetas HTML por componentes React Native**

Vamos a ver paso a paso que modificaciones son necesarias



Tres en Raya: App.js

```
import React from 'react';

import GameScreen from './app/components/GameScreen';

export default class App extends React.Component {
  render() {
    return (
      <GameScreen/>
    );
  }
}
```

React Native: Componentes

- En lugar de utilizar **elementos HTML** como React, React Native utiliza **componentes nativos**

React

```
import React from 'react';

export default class Header
  extends React.Component {
  render() {
    return (
      <header>
        {this.props.text}
      </header>
    );
  }
}
```

React Native

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

export default class Header
  extends React.Component {
  render() {
    return (
      <Text>
        {this.props.text}
      </Text>
    );
  }
}
```

React Native: Componentes con estilo

React

Header.jsx

```
import React from 'react';
import './assets/css/hstyles.css';

export default class Header
  extends React.Component {
  render() {
    return (
      <header className="header">
        {this.props.text}
      </header>
    );
  }
}
```

hstyles.css

```
.header{ color: red; }
```

React Native

Header.js

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

export default class Header
  extends React.Component {
  render() {
    return (
      <Text style={{ color: 'red' }}>
        {this.props.text}
      </Text>
    );
  }
}
```


React Native: Componentes con estilo

React

Header.jsx

```
import React from 'react';
import '../assets/css/hstyles.css';

export default class Header
  extends React.Component {
  render() {
    return (
      <header className="header">
        {this.props.text}
      </header>
    );
  }
}
```

hstyles.css

```
.header{ color: red; }
```

React Native

Header.js

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

export default class Header
  extends React.Component {
  render() {
    return (
      <Text style={styles.header}>
        {this.props.text}
      </Text>
    );
  }
}

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

Tres en Raya: GameScreen.js

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

import Header from './Header.js';
import Board from './Board.js';
[...]

export default class GameScreen extends React.Component {
  [...] //No modifications are required
  render(){
    var text = "Turn of " + this.state.turn;
    return (
      <View style={{flex:1, margin:10, justifyContent:'center'}}>
        <Header text={text}/>
        <Board values={this.state.values} onClick={this._onClick}
          winner={this.state.winner}/>
      </View>
    );
  }
}
```

Tres en Raya: Header.js

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

export default class Header extends React.Component {
  render() {
    return (
      <Text style={styles.header}>
        {this.props.text}
      </Text>
    );
  }
}

const styles = StyleSheet.create({
  header: {
    textAlign: 'center',
    fontSize: 25,
    padding: 10
  }
});
```

Tres en Raya: Board.js

```
import React from 'react';
import { View, StyleSheet } from 'react-native';
import Square from './Square.js';

export default class Board extends React.Component {
  [...] //No modifications are required
  render(){ [...] }
}

const styles = StyleSheet.create({
  board: {
    flex: 1,
    flexDirection: 'column',
    justifyContent: 'space-between'
  },
  boardRow: {
    flex: 1,
    flexDirection: 'row',
    justifyContent: 'space-between'
  }
})
```

Tres en Raya: Board.js

```
render(){
  var board = this.props.values.map((rowValues,rowIndex) => {
    let boardRow = rowValues.map((value,columnIndex) => {
      let squareKey = "" + rowIndex + columnIndex;
      let squareDisabled = (value === "-" ? false : true);
      if(typeof this.props.winner !== "undefined"){ squareDisabled = true; }
      return (
        <Square value={value} key={squareKey} rowIndex={rowIndex}
        columnIndex={columnIndex} onClick={this._onClick}
        squareDisabled={squareDisabled}/>
      );
    });
    let boardRowKey = "boardRow" + rowIndex;
    return (<View key={boardRowKey} style={styles.boardRow}>
      {boardRow}</View>);
  });
  return (<View style={styles.board}>{board}</View>);
}
```

Tres en Raya: Square.js

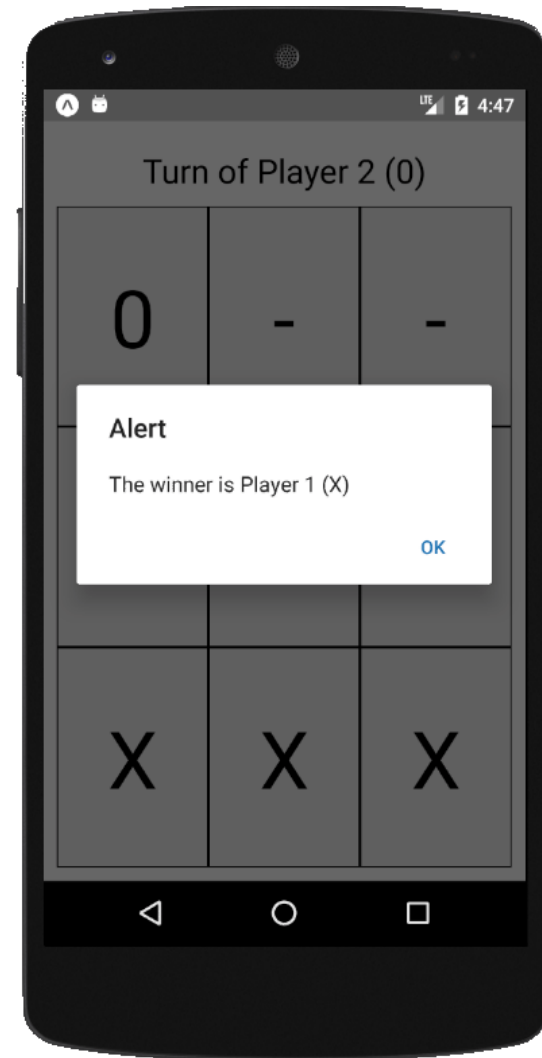
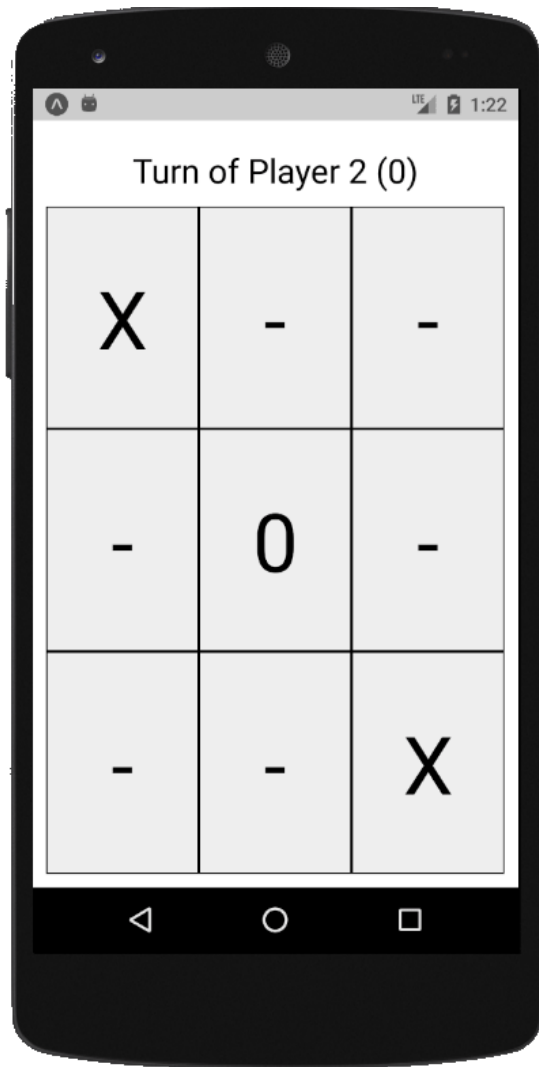
```
import React from 'react';
import { TouchableHighlight, Text, StyleSheet } from 'react-native';

export default class Square extends React.Component {
  [...] //No modifications are required
  render(){
    return(
      <TouchableHighlight style={styles.square}
                        onPress={this._onClick}
                        disabled={this.props.squareDisabled}>
        <Text style={styles.squareText}>{this.props.value}</Text>
      </TouchableHighlight>
    );
  }
}
```

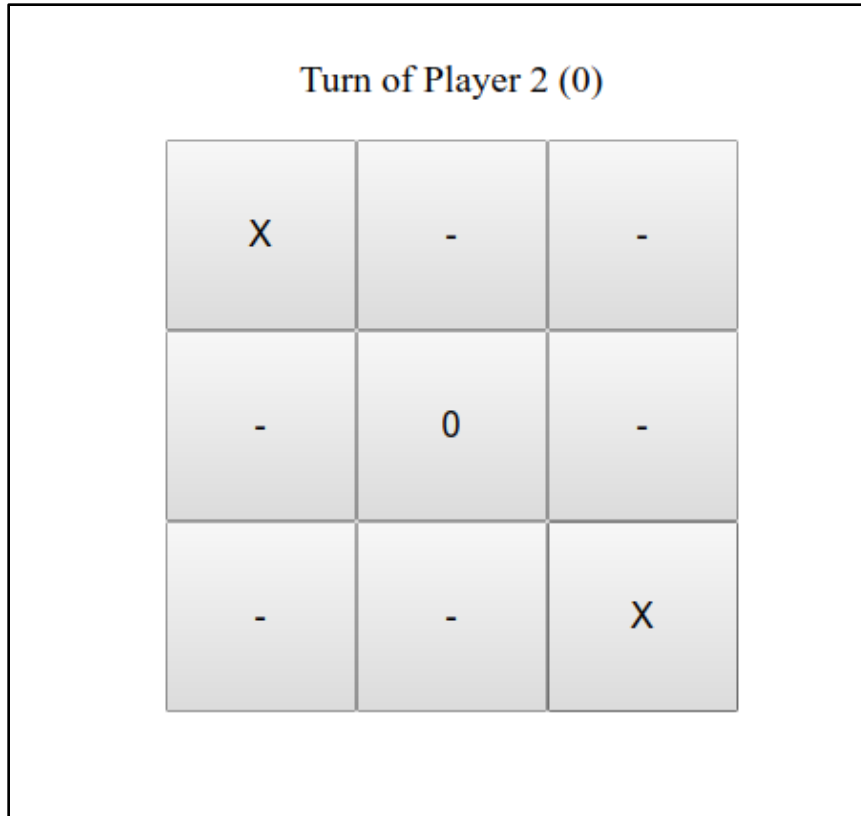
Tres en Raya: Square.js

```
const styles = StyleSheet.create({
  square: {
    flex: 1,
    padding: 10,
    alignItems: 'center',
    justifyContent: 'center',
    borderWidth: 1,
    borderColor: 'black',
    backgroundColor: '#eee'
  },
  squareText: {
    fontSize: 60
  }
})
```

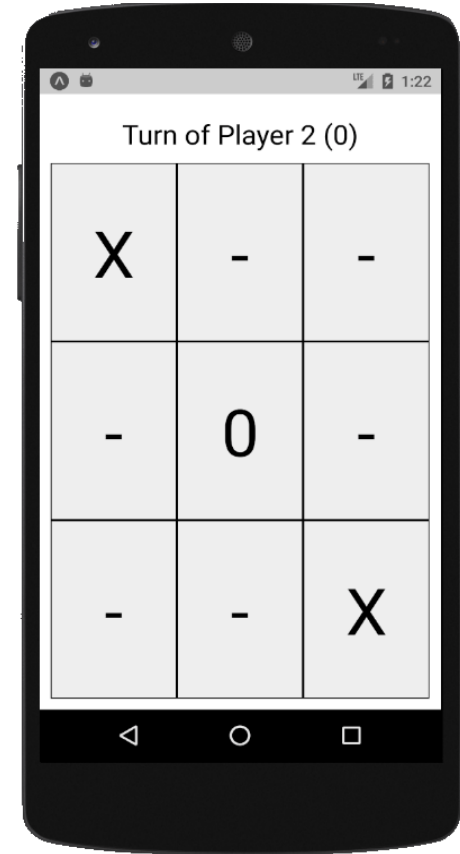
Tres en Raya con React Native: Resultado



Tres en Raya: React y React Native



React
(Aplicación Web)

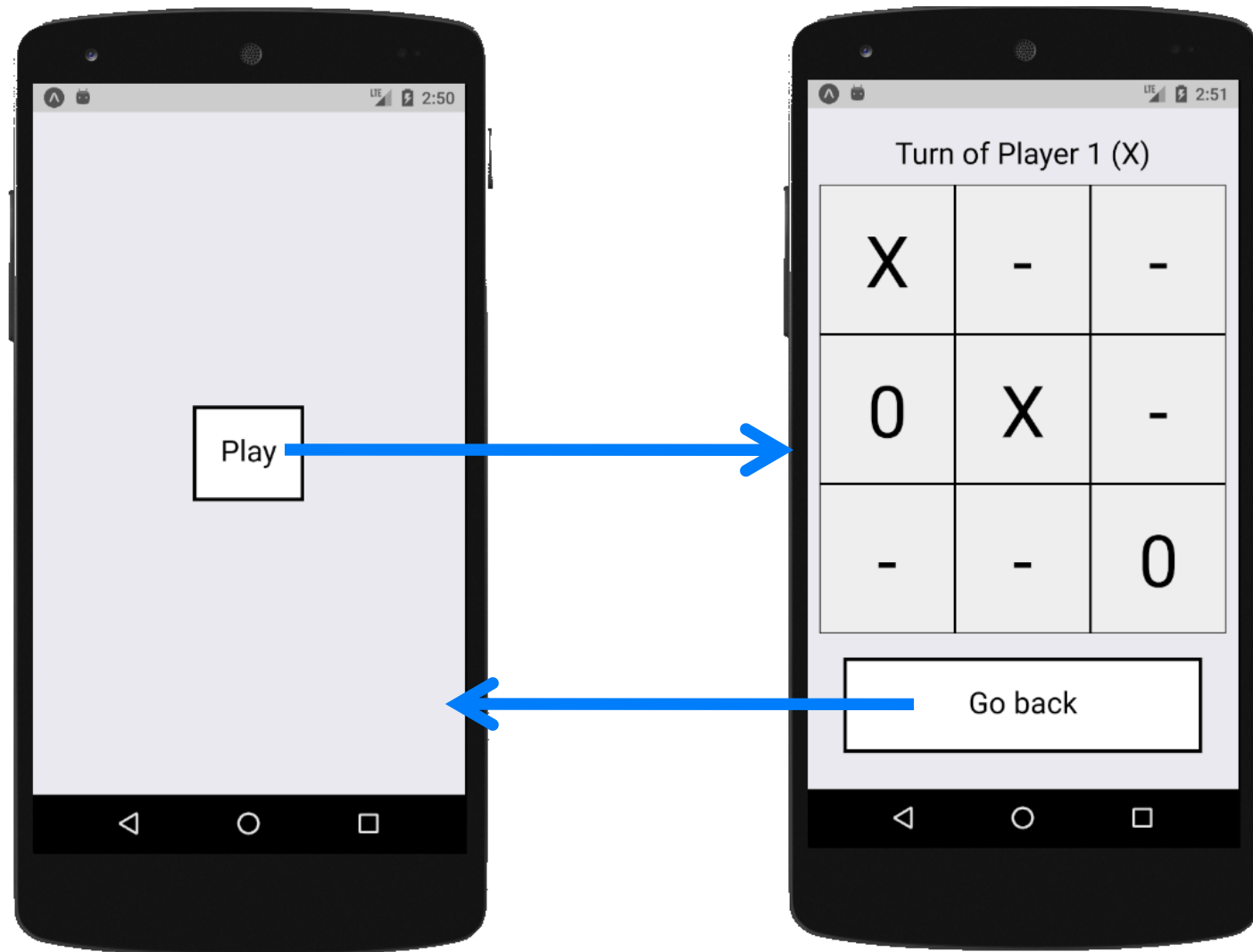


React Native
(Aplicación Nativa)

Tres en Raya con React Native: Navegación

- Vamos a incluir **navegación** en la aplicación
Tres en Raya implementada con React Native
- Crearemos **dos pantallas**:
 - La primera consistirá simplemente en una pantalla con un botón que permitirá ir a la segunda
 - En la segunda pantalla se mostrará el tablero para jugar al tres en raya y además habrá un botón para regresar a la primera pantalla
- La navegación entre las pantallas será implementada usando el componente **StackNavigator** ofrecido por la biblioteca **React Navigation**

Tres en Raya con React Native: Navegación



Tres en Raya con React Native: Navegación

Instalamos **React Navigation**:

```
npm install --save react-navigation
```

Creamos un componente

MyButton en el archivo

app/components/MyButton.js

Creamos la primera pantalla

"IndexScreen" en el archivo

app/components/IndexScreen.js

Creamos la segunda pantalla

"GameScreen" modificando

app/components/GameScreen.js

Importamos ambas pantallas en


App.js y usamos un

StackNavigator para gestionarlas

▼  AppReactNative

▼  app

▼  components

 Board.js

 **GameScreen.js**

 Header.js

 **IndexScreen.js**

 **MyButton.js**

 Square.js

►  node_modules

 **App.js**

 app.json

 package.json

 ...

TER - Navegación: App.js

```
import React from 'react';
import { StackNavigator } from 'react-navigation';

import IndexScreen from './app/components/IndexScreen';
import GameScreen from './app/components/GameScreen';

export default App = StackNavigator({
  IndexScreen: {
    screen: IndexScreen
  },
  GameScreen: {
    screen: GameScreen
  }
},{
  headerMode: 'none'
})
```

TER - Navegación: IndexScreen.js

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

import MyButton from './MyButton.js';

export default class IndexScreen extends React.Component {
  render() {
    return (
      <View style={{ flex:1, alignItems:'center', justifyContent:'center' }}>
        <MyButton
          onPress={() => this.props.navigation.navigate('GameScreen')}
          text={"Play"}/>
      </View>
    )
  }
}
```

TER - Navegación: GameScreen.js

```
import React from 'react';
import { View } from 'react-native';
import Header from './Header.js';
import Board from './Board.js';
import MyButton from './MyButton.js';
[...]
```

```
export default class GameScreen extends React.Component {
  [...]
  render(){
    var text = "Turn of " + this.state.turn;
    return (
      <View style={{flex:1, margin:10, justifyContent:'center'}}>
        <Header text={text}/>
        <Board values={this.state.values} onClick={this._onClick}
          winner={this.state.winner}/>
        <MyButton
          onPress={() => this.props.navigation.goBack(null)} text={"Go back"}/>
      </View>
    );
  }
}
```

TER - Navegación: MyButton.js

```
import React from 'react';
import { StyleSheet, Text, TouchableHighlight } from 'react-native';
export default class MyButton extends React.Component {
  render() {
    return (
      <TouchableHighlight onPress={this.props.onPress}>
        <Text style={styles.text}>{this.props.text}</Text>
      </TouchableHighlight>
    )
  }
}
const styles = StyleSheet.create({
  text: {
    padding: 20,
    margin: 20,
    backgroundColor: 'white',
    color: 'black',
    borderWidth: 3,
    borderColor: 'black',
    fontSize: 25,
    textAlign: 'center'
  }
})
```


¿Preguntas?

Aldo Gordillo

agordillo@dit.upm.es