



Universidad Politécnica de Madrid

React Native

Componentes avanzados

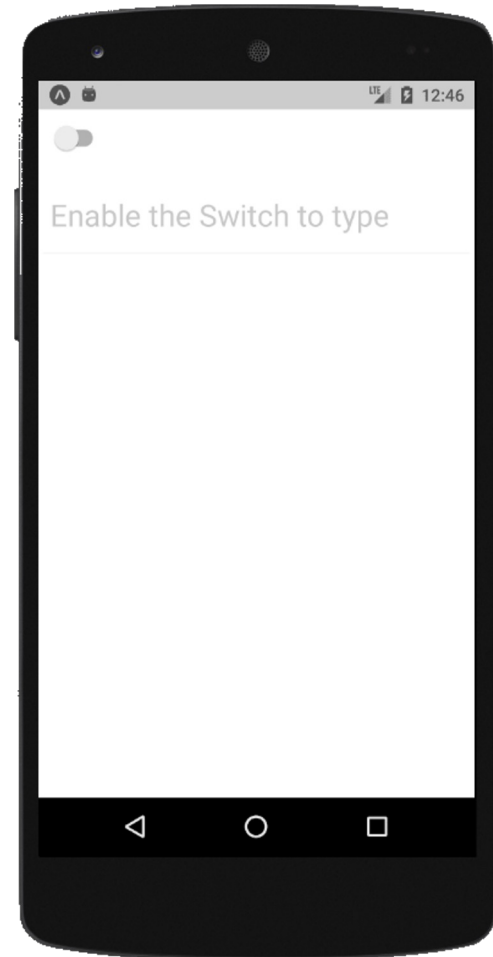
Álvaro Alonso González
Enrique Barra Arias

- El componente **Switch** permite incluir interruptores (*on/off*) en una aplicación
- Permite a los usuarios introducir valores **booleanos** (como los input de tipo *checkbox* en HTML)
- Es un **componente controlado*** de React que requiere un callback **onValueChange** que actualice su propiedad **value**

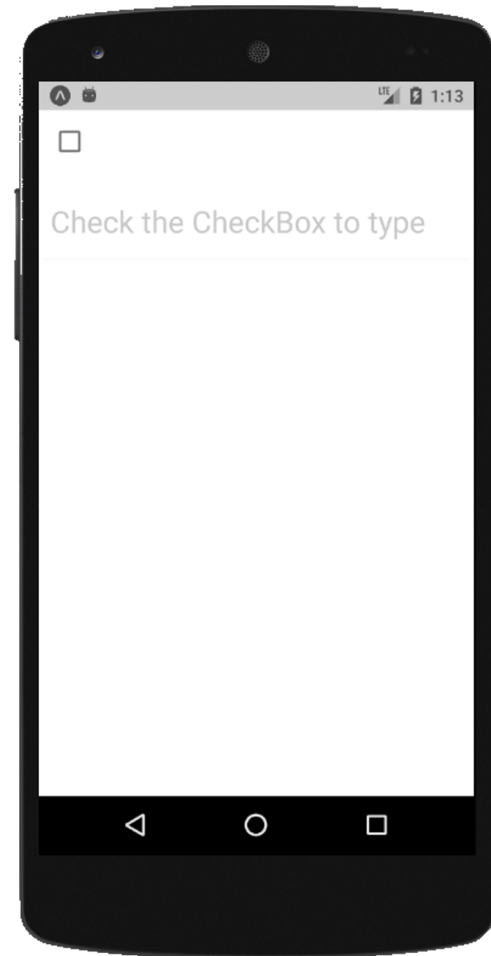
*Un componente controlado de React es un componente de entrada de datos con estado propio dependiente de la entrada de datos y cuyo valor es controlado por React

<https://reactjs.org/docs/forms.html>

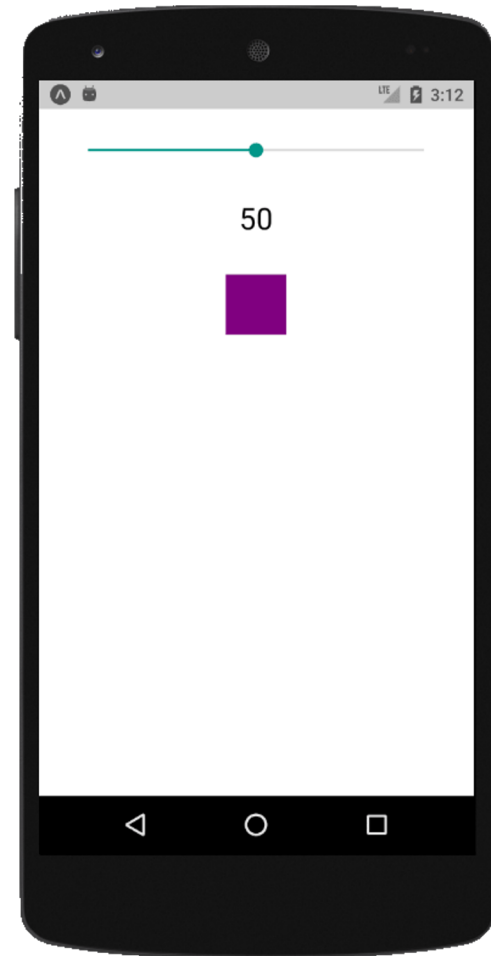
- Documentación:
<https://reactnative.dev/docs/switch>



- El componente **CheckBox** permite incluir casillas en una aplicación para que los usuarios introduzcan valores **booleanos** (como los input de tipo *checkbox* en HTML)
- Muy similar a Switch pero con distinta interfaz gráfica
- Es un **componente controlado** de React que requiere un callback **onValueChange** que actualice su propiedad **value**
- Documentación:
<https://facebook.github.io/react-native/docs/checkbox.html>



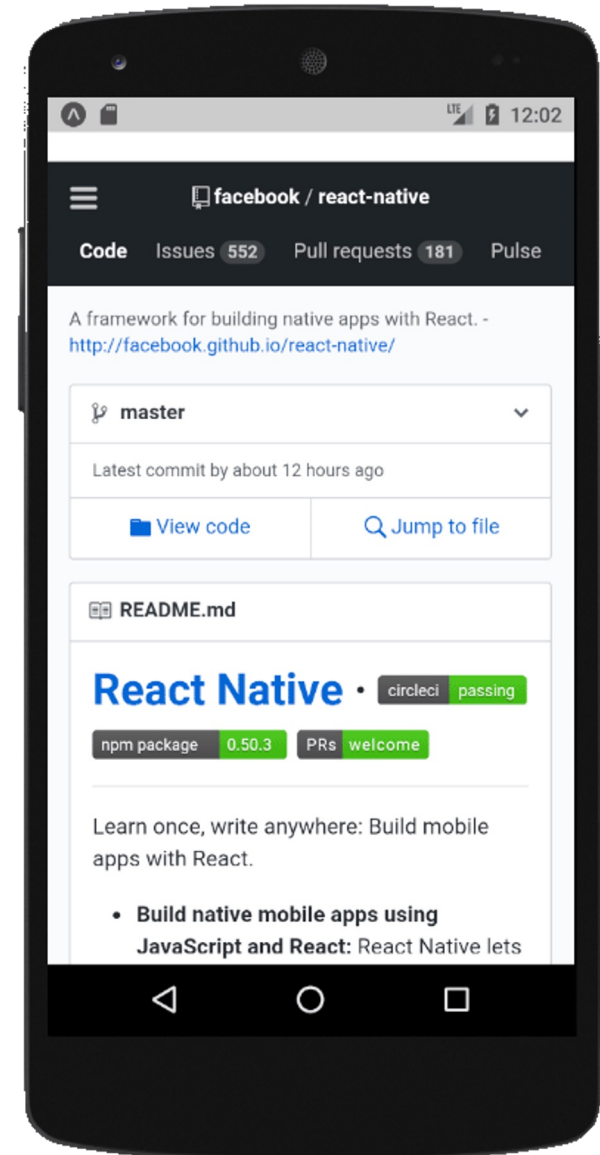
- El componente **Slider** permite a los usuarios seleccionar un único valor de una lista de **valores numéricos**
- Tiene una propiedad **value** y dos callbacks: **onValueChange** (llamado de forma continua mientras el usuario arrastra el slider) y **onSlidingComplete** (llamado cuando el usuario suelta el slider)
- Documentación (Ahora está mantenido por la comunidad): <https://github.com/callstack/react-native-slider>



- El componente **Modal** permite mostrar de una forma sencilla contenido sobre una vista o pantalla
- No sustituye a los **navegadores**
- Se pueden especificar varias **propiedades** (animación para mostrar el modal, estilo, ...) y **callbacks** para cuando el modal se muestre (*onShow*) o para cuando el usuario pulse el botón de retroceso (*onRequestClose*)
- Documentación:
<https://reactnative.dev/docs/modal>

- El componente **WebView** permite renderizar **contenido web** en una vista nativa
- Permite especificar diversas **propiedades** para el contenido web: icono de carga, estilo, habilitación de JavaScript, etc.
- Permite inyectar **código JavaScript**
- Proporciona diversos **callbacks**: *onLoad*, *onError*, *onMessage*, *onNavigationStateChange*, etc.
- Documentación (ahora mantenido por la comunidad):

<https://github.com/react-native-webview/react-native-webview>



- El componente **StatusBar** controla la **barra de estado** de la aplicación
- Podemos definir algunas **propiedades**: color de fondo, estilo, visible/oculta, etc.
- Las propiedades pueden ser establecidas directamente en el componente StatusBar (recomendado) o a través de una **API** ofrecida por él
- Pueden existir varios componentes StatusBar al mismo tiempo
 - Esto permite especificar **barras de estado diferentes** para diferentes pantallas de una misma aplicación
 - En este caso las propiedades se mezclarán siguiendo el orden en el que los componentes fueron montados
- Documentación:
<https://reactnative.dev/docs/statusbar>



Universidad Politécnica de Madrid

React Native

Componentes avanzados

Álvaro Alonso González
Enrique Barra Arias