



# **REACT NATIVE:**

## **Rompiendo Paradigmas**

## **del Desarrollo Móvil**

# Relator



---

Ricardo Cuevas Muhlenbrock  
Ingeniero en  
 [muhlenbrock](https://github.com/muhlenbrock)

Informática

# TABLA DE CONTENIDOS

01

## INTRODUCCIÓN

Plataformas para el desarrollo móvil

02

## MANOS EN LA MASA

- Creación proyecto
- Estructura del proyecto
- Manejo de estilos
- Navegación entre vistas

03

## TIPOS DE COMPONENTES

- Funcionales
- Clases

04

## HERRAMIENTAS PARA TRABAJAR EN EQUIPO

- Slack
- Bitrise
- Sentry

01

## INTRODUCCIÓN

Plataformas para el desarrollo móvil

# PLATAFORMAS PARA EL DESARROLLO MÓVIL

## CÓDIGO MULTIPLATAFORMA

Pros:

- Mantenimiento de un código fuente.
- Equipo de trabajo más fácil de encontrar.
- Costos de desarrollo más bajos.
- Multiplataforma.
- Fácil y rápido para desarrollar un prototipo.

Contras:

- Performance puede ser un poco menor.
- Instalar bibliotecas puede ser todo un desafío
- Acceso a funcionalidades nuevas.

## CÓDIGO NATIVO

Pros:

- Mejor performance.
- Seguridad?
- UI nativa para cada sistema operativo.
- Fácil uso de dispositivos como:
  - Cámaras
  - GPS
  - Sensores

Contras:

- Mantenimiento de múltiples códigos fuente.
- Equipo de trabajo más difícil de encontrar.
- Elevado costo para el desarrollo.

# PLATAFORMAS PARA EL DESARROLLO MÓVIL

## CÓDIGO MULTIPLATAFORMA



React Native



## CÓDIGO NATIVO



Swift

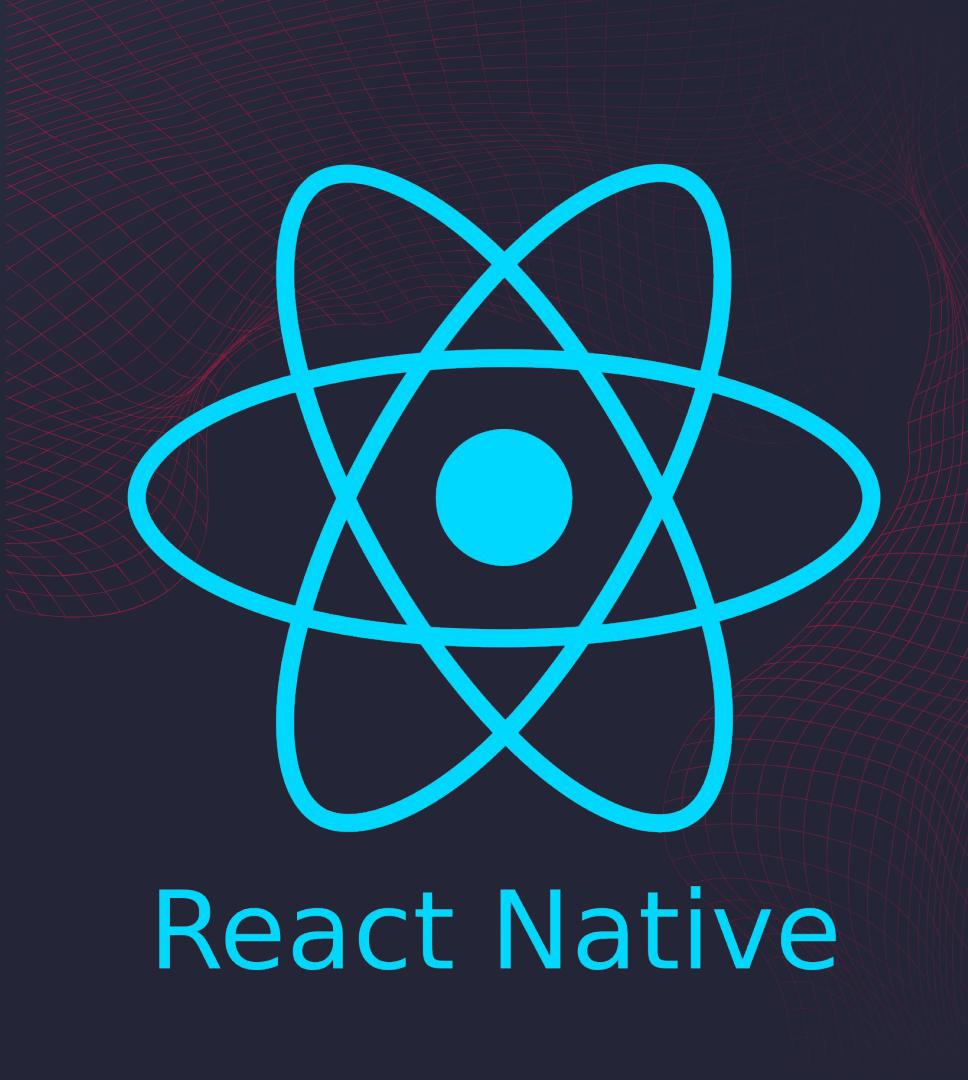


[OBJ-C]

ObjectiveC

## ¿Por qué usar React Native?

- Animaciones 60 fps.
- Componentes reutilizables.
- Se puede desarrollar para Android, iOS, Mac OS y Windows.
- No ocupa el webview como contenedor de la app.
- Comunidad de desarrolladores robusta.
- Pocos issues en su repositorios.



React Native



# APPS Y EMPRESAS QUE USAN REACT NATIVE

# SIGUIENTE CONTENIDO

demasiado bla bla

02

## MANOS EN LA MASA

- Creación proyecto
- Estructura del proyecto
- Manejo de estilos
- Navegación entre vistas

## REQUISITOS PREVIOS:

- Instalar Node
- Instalar Java Development Kit
- Instalar Android Studio

Link Documentación

<https://reactnative.dev/docs/environment-setup>



# CREANDO NUESTRO PRIMER PROYECTO EN REACT NATIVE

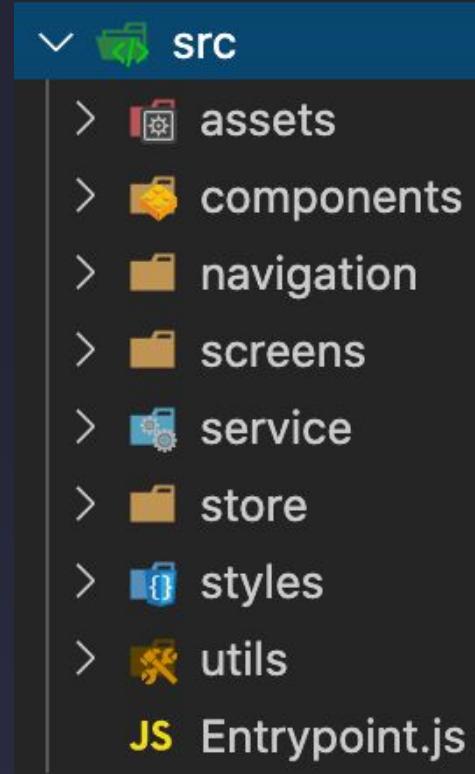
## ESTRUCTURA DEL PROYECTO POR DEFECTO

```
> 📁 _tests_
> 📁 .vscode
> 📁 android
> 📁 ios
> 📁 node_modules
  ⚙ .buckconfig
  ⚙ .eslintrc.js
  ⚙ .flowconfig
  ⚙ .gitattributes
  ⚙ .gitignore
  ⚙ .prettierrc.js
  ⚙ .watchmanconfig
  JS App.js
  ↗ app.json
  ⚙ babel.config.js
  JS index.js
  JS metro.config.js
  ⚙ package.json
  ⚙ yarn.lock
```

# ESTRUCTURA DEL PROYECTO PERSONALIZADA

```
> 📁 _tests_
> 📁 .vscode
> 📁 android
> 📁 ios
> 📁 node_modules
< 📁 src
  > 📁 assets
  > 📁 components
  > 📁 navigation
  > 📁 screens
  > 📁 service
  > 📁 store
  > 📁 styles
  > 🐛 utils
    JS Entrypoint.js
  🚀 .buckconfig
  ⚙ .eslintrc.js
  ⚡ .flowconfig
  🔑 .gitattributes
  🔑 .gitignore
  🎨 .prettierrc.js
  🏃 .watchmanconfig
  ↗ app.json
  BABEL babel.config.js
  📄 eim.keystore
  📁 eim.p12
  JS index.js
  JS metro.config.js
  JSON package.json
  JS react-native.config.js
  📄 README.md
  📁 yarn.lock
```

## ESTRUCTURA DEL PROYECTO PERSONALIZADA



# MANEJO DE ESTILOS



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

class Example extends Component {
  render() {
    return (
      <>
        <View>
          <Text style={{ color: 'red' }}> Red</Text>
          <Text style={{ fontSize: 30 }}>Big</Text>
        </View>
      </>
    );
  }
}
```

## ESTILOS EN LÍNEA

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

const styles = StyleSheet.create({
  red: {
    color: 'red'
  },
  big: {
    fontSize: 30
  }
});

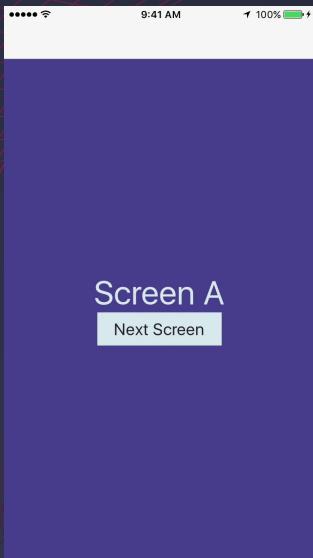
class Example extends Component {
  render() {
    return (
      <>
        <View>
          <Text style={styles.red}>Red</Text>
          <Text style={styles.big}>Big</Text>
        </View>
      </>
    );
  }
}
```

## ESTILOS USANDO UNA HOJA DE ESTILO

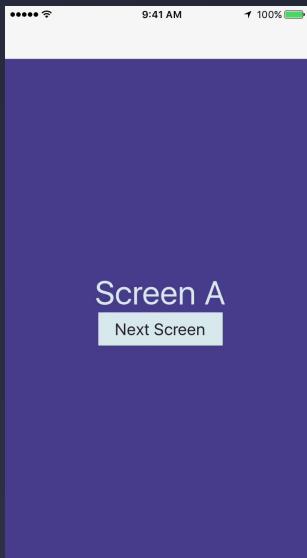
# NAVEGACIÓN ENTRE VISTAS

- React Navigation
- React Navigation Transitions

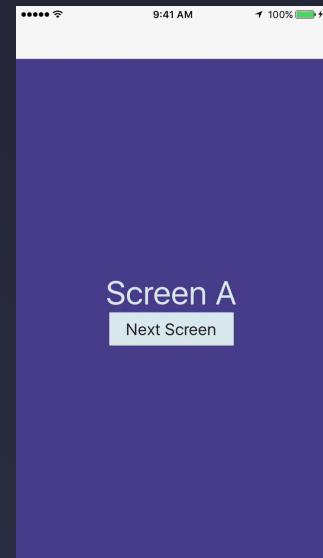
fromLeft



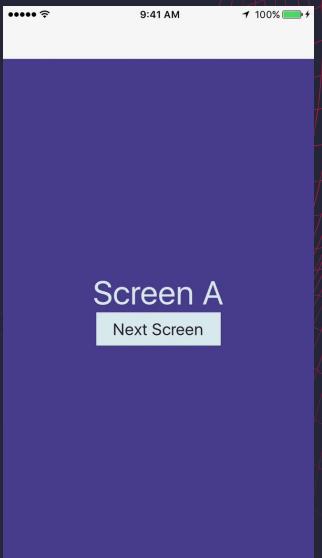
fromTop



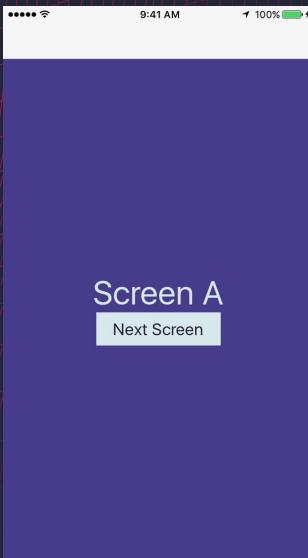
fadeIn



zoomIn



zoomOut

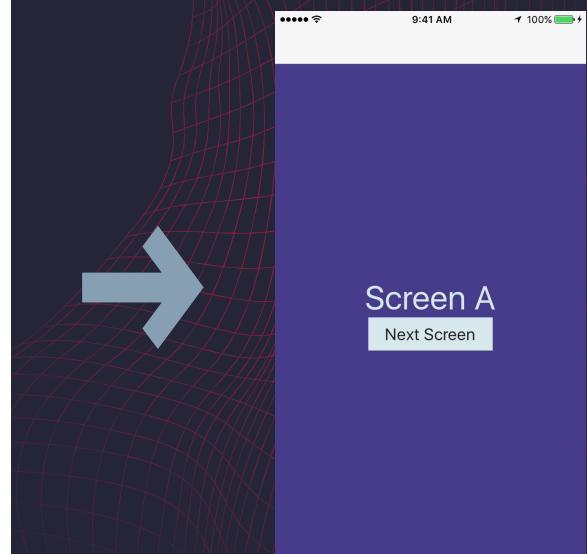
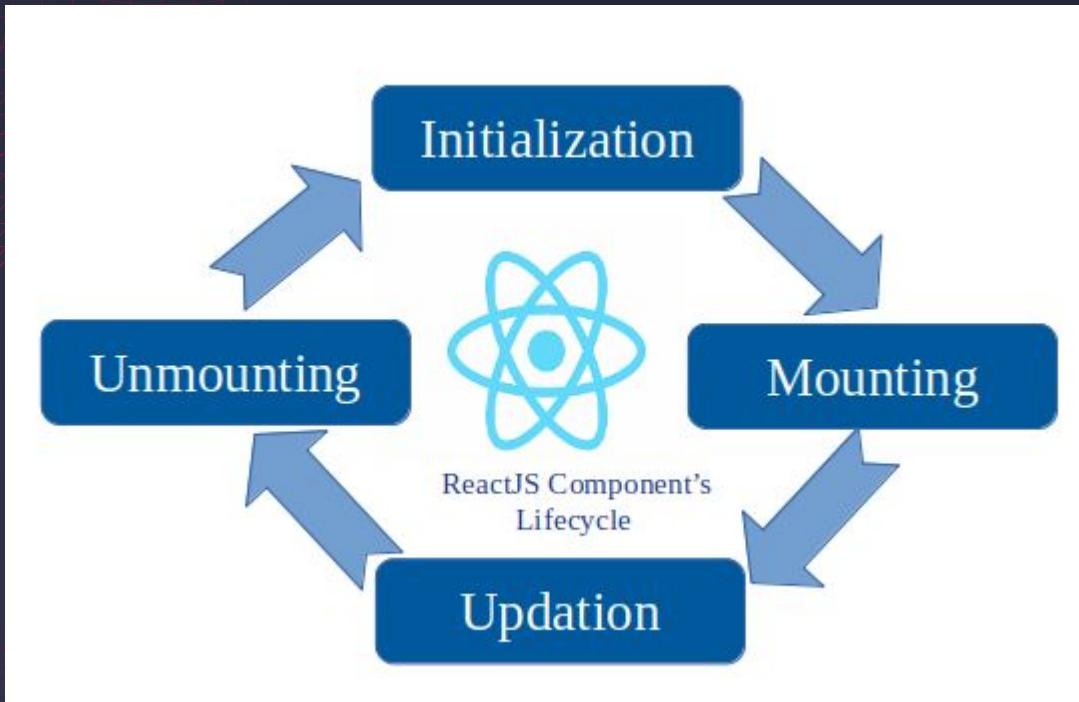


03

## TIPOS DE COMPONENTES

- Funcionales
  - Clase
-

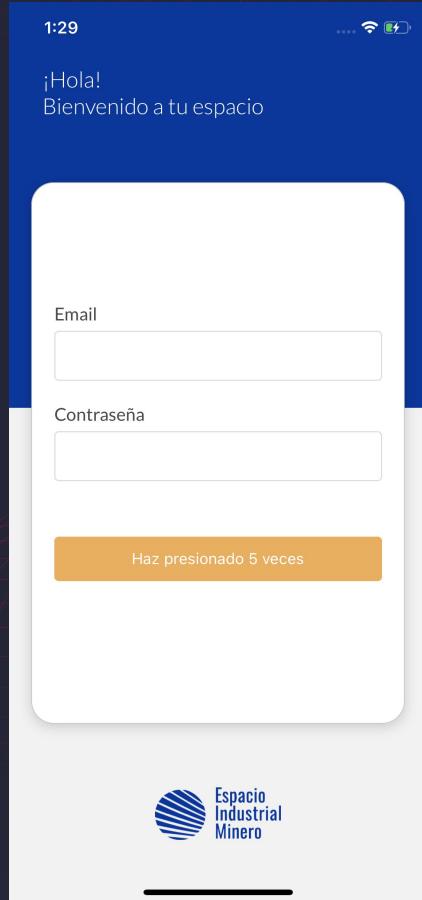
## COMPONENTES Y CICLO DE VIDA



```
import React, {useState, useEffect} from 'react';
import {Text, TouchableOpacity} from 'react-native';
import {s} from './CounterButtonStyle';

export const CounterButton = ({children, styleText}) => {
  const [count, setCount] = useState(0);
  useEffect(() => {
    console.log('');
    console.log('Entrando al componente CounterButton');
    console.log('Ciclo de vida componentDidMount');
    console.log('');
  }, []);
  useEffect(() => {
    console.log('Actualizando estados del state count: ', count);
    console.log('componentDidUpdate');
    console.log('');
  }, [count]);
  useEffect(() => {
    return () => {
      console.log('Saliendo del componente CounterButton');
      console.log('componentWillUnmount');
      console.log('');
    };
  }, []);
  return (
    <TouchableOpacity
      style={s.btn}
      onPress={() => {
        setCount(count + 1);
      }}>
      <Text style={styleText}>Haz presionado {count} veces</Text>
    </TouchableOpacity>
  );
};
```

# COMPONENTES FUNCIONALES

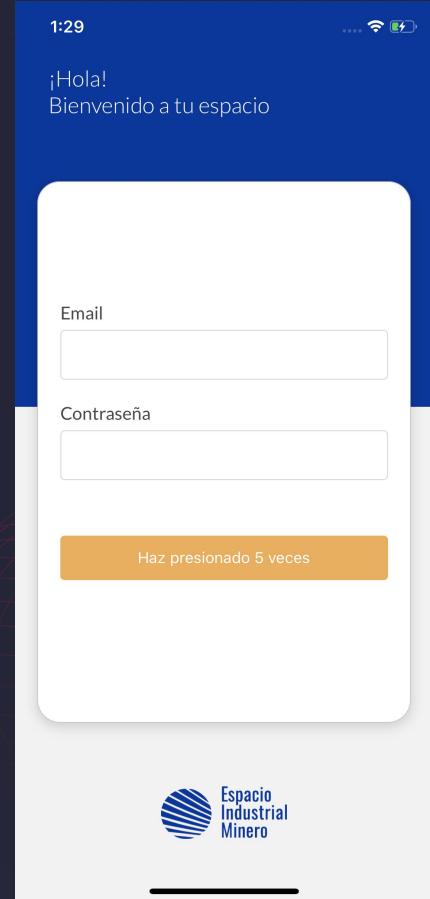


```
import React, {useState, useEffect} from 'react';
import {Text, TouchableOpacity} from 'react-native';
import {s} from './CounterButtonStyle';

export const CounterButton = ({children, styleText}) => {
    const [count, setCount] = useState(0);
    useEffect(() => {
        console.log('');
        console.log('Entrando al componente CounterButton');
        console.log('Ciclo de vida componentDidMount');
        console.log('');
    }, []);
    useEffect(() => {
        console.log('Actualizando estados del state count: ', count);
        console.log('componentDidUpdate');
        console.log('');
    }, [count]);
    useEffect(() => {
        return () => {
            console.log('Saliendo del componente CounterButton');
            console.log('componentWillUnmount');
            console.log('');
        };
    }, []);
    return (
        <TouchableOpacity
            style={s.btn}
            onPress={() => {
                setCount(count + 1);
            }}>
            <Text style={styleText}>Haz presionado {count} veces</Text>
        </TouchableOpacity>
    );
};
```

PROPS

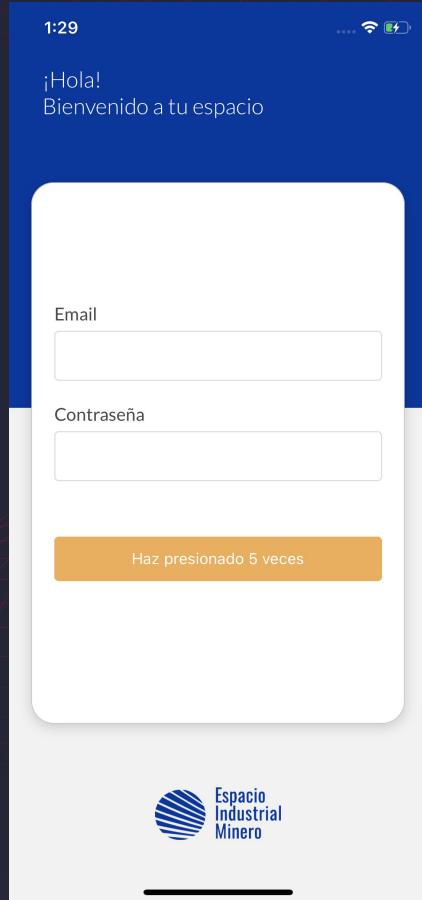
# COMPONENTES FUNCIONALES



```
import React, {useState, useEffect} from 'react';
import {Text, TouchableOpacity} from 'react-native';
import {s} from './CounterButtonStyle';

export const CounterButton = ({children, styleText}) => {
  const [count, setCount] = useState(0);
  STATE
  useEffect(() => {
    console.log('');
    console.log('Entrando al componente CounterButton');
    console.log('Ciclo de vida componentDidMount');
    console.log('');
  }, []);
  useEffect(() => {
    console.log('Actualizando estados del state count: ', count);
    console.log('componentDidUpdate');
    console.log('');
  }, [count]);
  useEffect(() => {
    return () => {
      console.log('Saliendo del componente CounterButton');
      console.log('componentWillUnmount');
      console.log('');
    };
  }, []);
  return (
    <TouchableOpacity
      style={s.btn}
      onPress={() => {
        setCount(count + 1);
      }}>
      <Text style={styleText}>Haz presionado {count} veces</Text>
    </TouchableOpacity>
  );
};
```

# COMPONENTES FUNCIONALES



# COMPONENTES DE CLASE



04

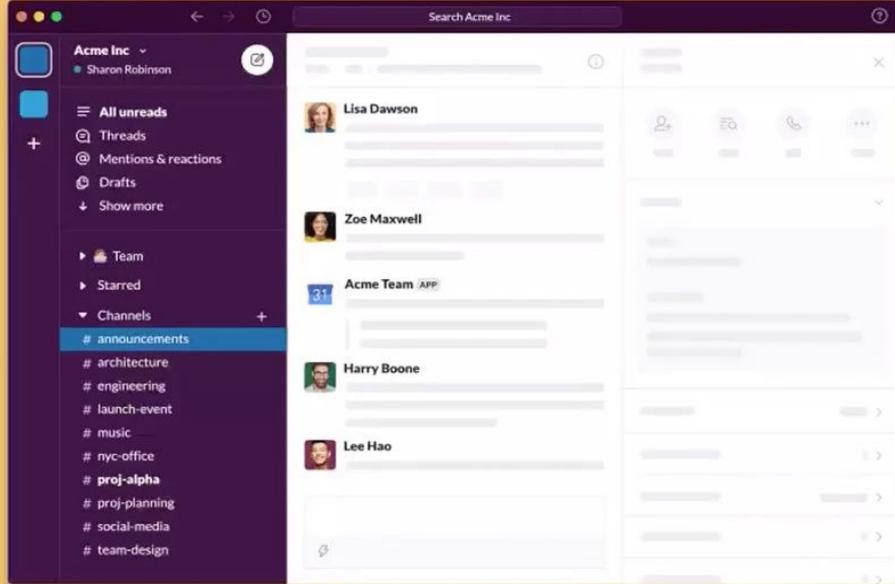
## HERRAMIENTAS PARA TRABAJAR EN EQUIPO

- Slack
- Bitrise
- Sentry



## Slack

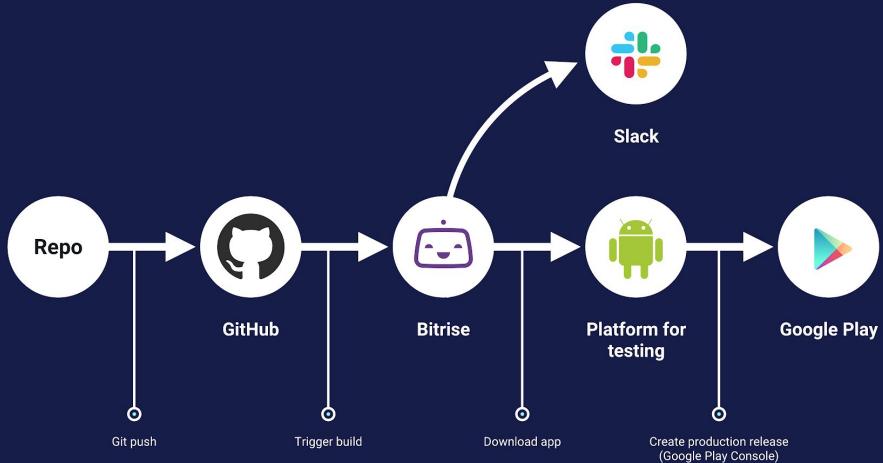
Es una aplicación de mensajería para equipos.





## BITRISE

Herramienta para la  
integración continua y  
distribución continua (CI/CD)





# SETRY

Sentry ayuda a los desarrolladores a diagnosticar, corregir y optimizar el rendimiento de su código

?

200.83.128.91

iOS  
Version: 13.5

iPhone 8  
Arch: arm64

TAGS

app.device aa000fed... device iPhone 8 device.family iOS dist 921 event.origin javascript handled yes level error mechanism generic os iOS 13.5 os.name iOS os.rooted no release 1.9.4 (921) ⓘ user ip:200.83.128.91

EXCEPTION (most recent call first)

TypeError

null is not an object (evaluating 's.trim')

mechanism generic handled yes

js views/profile/EditProfile.js in \_validatePayload at line 136:9

```
131. if (lastName.trim(' ') === '') {  
132.   newLastNameError = 'Tu apellido es requerido';  
133. } else if (lastName.trim(' ').length < 2) {  
134.   newLastNameError = 'Largo insuficiente';  
135. }  
136. if (email.trim(' ') === '') {  
137.   newEmailError = 'Tu email es requerido';  
138. } else if (validateEmail(email.trim(' ')) !== true) {  
139.   newEmailError = validateEmail(email.trim(' '));  
140. }  
141. if (newNameError === '' || newLastNameError === '' || newEmailError === '') {
```

views/profile/EditProfile.js in call at line 189:33

Called from: node\_modules/regenerator-runtime/runtime.js in tryCatch

app:///[native code] in callImmediates

Called from: node\_modules/react-native/Libraries/BatchedBridge/MessageQueue.js in \_\_callImmediates

app:///[native code] in value ⓘ

## LINKS DE INTERÉS

- [Documentación React Native](#)
- [React Navigation](#)
- [Bibliotecas, conferencias, etc en awesome-react-native](#)
- [Plantilla app Superhero](#)
- [Documentación Redux](#)
- [React Navigation Transitions](#)
- [Video React Native EU 2020 - Virtual Edition](#)



# GRACIAS!

---

¿Tienes dudas?

rcuevas@oxidocs.com  
+569 40 25 6609  
[www.oxidocs.com](http://www.oxidocs.com)