

UNIVERSIDAD PRIVADA FRANZ TAMAYO

**FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS**



UNIVERSIDAD PRIVADA FRANZ TAMAYO

DEFENSA HITO 3 - TAREA FINAL

Nombre Completo: **Erick Rodrigo Aranibar Arias**

Asignatura: **PDM**

Carrera: **INGENIERÍA DE SISTEMAS**

Paralelo: **PDM(1)**

Docente: **Lic. William R. Barra Paredes**

fecha: **11/05/2020**

github: **<https://github.com/friki57/PDM.git>**

1. Parte teórica.

Defina que es un componente en React Native y muestre un ejemplo.

Es una estructura de código javascript la cual permite encapsular código HTML, CSS y Javascript en un solo elemento, de modo que el código se realiza para que cada componente sea independiente de los demás y también maximiza la reutilización por su manejo de propiedades y parámetros.

Ejemplo:

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

```
export default class Button extends Component {  
  constructor(props) {  
    super(props);  
  }  
  render() {  
    return (  
      <View>  
        <TouchableOpacity  
          style={stylesButton.container}  
          onPress={this.props.onPress}>  
          <Text  
            style={stylesButton.text}>{this.props.titleButton}  
          </Text>  
        </TouchableOpacity>  
      </View>  
    );  
  }  
}
```

```
const stylesButton = StyleSheet.create({  
  container: {
```

```

width: '100%',
alignItems: 'center',
justifyContent: 'center',
backgroundColor: "#D53F3D"
},
text: {
  color: Colors.white,
  textAlign: 'center',
  height: 20,
},
});

```

Explique cómo se realiza la navegación entre screens en React Native.

A través del objeto navigator el cual almacena cualquier cantidad de componentes, los cuales son las diferentes pantallas de la navegación, y los indexa por medio de nombres. El navigator le hereda a todas estas pantallas una propiedad para manipular la navegación y cambiar a cualquier otra de las pantallas que estén alojadas en el navigator por medio del nombre.

Ejemplo:

```
this.props.navigation.navigate("Inicio");
```

Que significa IaaS, PaaS y SaaS .

IaaS es un modelo arquitectónico de TI para ingerir y recolectar de manera centralizada cualquier tipo de archivo de registro proveniente de cualquier fuente o ubicación; tales como servidores, aplicaciones, dispositivos, etc.

PaaS es un entorno de desarrollo e implementación completo en la nube, con recursos que permiten entregar todo, desde aplicaciones sencillas basadas en la nube hasta aplicaciones empresariales sofisticadas habilitadas para la nube.

SaaS es un modelo de distribución de software en el que tanto el software como los datos manejados son centralizados y alojados en un único servidor externo a la empresa.

Que es Firebase, Firestore y explique a que se refiere cuando se habla de Baas.

Firebase es una plataforma móvil creada por Google, cuya principal función es desarrollar y facilitar la creación de apps de elevada calidad de una forma rápida.

Firestore es una base de datos flexible y escalable para la programación en servidores, dispositivos móviles y la Web desde Firebase y Google Cloud Platform.

Baas es un software el cual provee servicio como base de datos a otros softwares que se conecten a este.

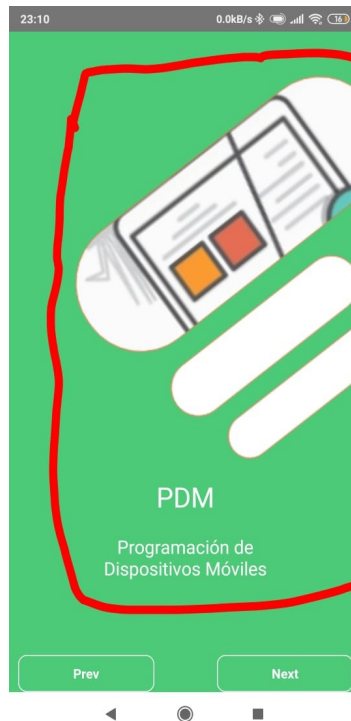
Defina o explique si React es lo mismo que React Native. Si son distintos liste cuales son las diferencias.

Ambos son bastante similares debido al hecho de que React es el núcleo de React Native, entonces React Native hereda todas las propiedades y funcionalidad de React. Pero aún así yo opino que las dos principales diferencias son:

- La plataforma para que se utilizan es distinta, React es para desarrollo web mientras que React Native es para desarrollo móvil.
- React utiliza el HTML puro característico de los sitios web como su base mientras que en React Native consta de sus propias etiquetas únicas.

2.- Parte práctica.

Componente de texto e imagen (Onboarding)



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

```
import Colores from "../Config/Colores";
```

```
export default class Description extends Component {  
  constructor(props){  
    super(props);  
  }  
  render()  
  {  
    console.disableYellowBox = true;  
    return (  

```

```

<View style={{ flex: 1, justifyContent: 'center', alignItems: 'center'}}>
    <Image source={this.props.imagen} style={{width: 400, height: 450, alignSelf: 'flex-end',
marginBottom: 0, marginRight: -100}}/>
    <Text style = {{width: 250, fontSize: 30, margin: 10, textAlign: "center", color : Colores.blanco}}>
        {this.props.Titulo}
    </Text>
    <Text style = {{width: 250, fontSize: 20, margin: 10, textAlign: "center", color : Colores.blanco}}>
        {this.props.Descripcion}
    </Text>
</View>
)
}
}

```

Componente de botones direccionales.



```

import React, { Component } from "react";
import { View, Text, Alert, StyleSheet, Image } from 'react-native';

import Colores from "../Config/Colores";
import Button from "../Button";

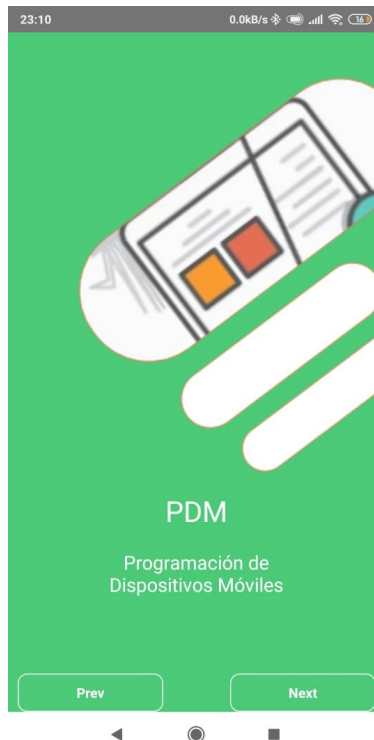
export default class ButtonsFotter extends Component {
  constructor(props){
    super(props);

    this.navegarAtras = this.navegarAtras.bind(this);
    this.navegarAdelante = this.navegarAdelante.bind(this);
  }
  navegarAtras()
  {
    if(this.props.atras)
      this.props.navigation.navigate(this.props.atras)
  }
  navegarAdelante()
  {
    if(this.props.adelante)
      this.props.navigation.navigate(this.props.adelante)
  }
  render()
  {
    console.disableYellowBox = true;
    return (
      <View style = {{width: "100%", flexDirection : "row", alignItems: "stretch", marginTop: 50}}>
        <View style = {{marginLeft: 10, marginRight: 70}}>
          <Button
            texto = "Prev"

```

```
        funcion = {this.navegarAtras}
      />
    </View>
    <View style = {{marginRight: 10, alignItems: "flex-end"}}>
      <Button
        texto = "Next"
        funcion = {this.navegarAdelante}
      />
    </View>
  </View>
);
}
}
```


Acoplamiento de ambos componentes en una pantalla.



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

import Colores from "../Config/Colores";
import Button from "../Componentes/Button";
import Description from "../Description";
import ButtonsFotter from "../Componentes/ButtonsFotter";

export default class OnboardingFactory extends Component {
  render()
  {
    console.disableYellowBox = true;
    return (
```

```

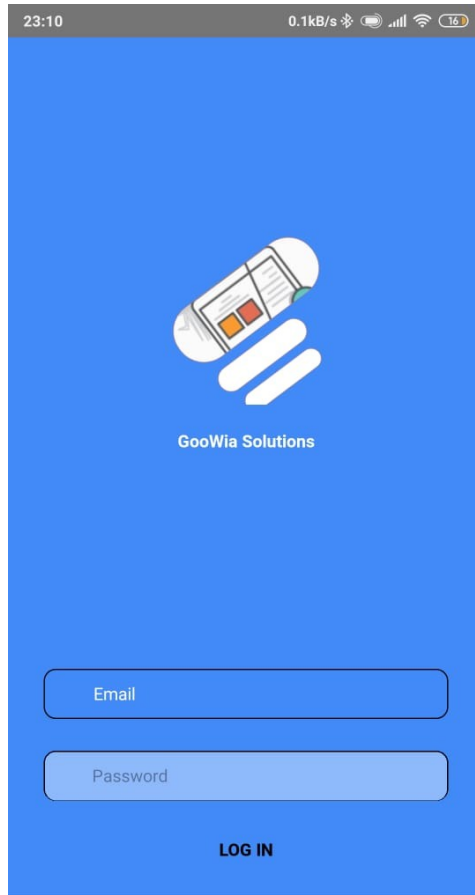
        <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center', backgroundColor:
this.props.color }}>
    <Description
        imagen = {this.props.imagen}
        color = {this.props.color}
        Titulo = {this.props.Titulo}
        Descripcion = {this.props.Descripcion}
    />
    <ButtonsFotter
        navigation = {this.props.navigation}
        atras = {this.props.rutaAtras}
        adelante = {this.props.rutaAdelante}
    ></ButtonsFotter>
</View>
);

}

}

```

Diseño del Login Screen.



```
import React, {useState} from  
'react';
```

```
import {  
  StyleSheet,  
  View,  
  SafeAreaView,  
  KeyboardAvoidingView,  
  Alert,  
} from 'react-native';
```

```
import ButtonLogin from '../Componentes/login/Button';  
import TextInputLogin from '../Componentes/login/TextInput';  
import LogoLogin from '../Componentes/login/Logo';
```

```
import EmailTextField from '../Componentes/login/EmailTextField';
import DismissKeyboard from '../Componentes/login/DismissKeyboard';
import FirebasePlugin from '../Plugins/Firebase';
```

```
import Utils from '../utils/Utils';
import Images from '../Config/Imagenes';
import Constants from '../Config/Constants';
import Colors from '../Config/Colores';
```

```
const LoginScreen = ({navigation}) => {
  const [email, setEmail] = useState("");
  const [errorEmail, setErrorEmail] = useState("");
  const [password, setPassword] = useState("");
  const [errorPassword, setErrorPassword] = useState("");
  const [isLoading, setIsLoading] = useState(false);
```

```
/**
```

```
 * @name _validateEmailAddress
```

```
 * @returns {boolean}
```

```
 * @private
```

```
 */
```

```
const _validateEmailAddress = () => {
  let isValidEmail = Utils.isValidEmail(email);
  isValidEmail
    ? setErrorEmail("")
    : setErrorEmail(Constants.STRING.EMAIL_ERROR);
  return isValidEmail;
};
```

```
/**
```

```

* @name _validatePassword
* @returns {boolean}
* @private
*/
const _validatePassword = () => {
  let isValidPassword = Utils.isValidField(password);
  isValidPassword
    ? setErrorPassword("")
    : setErrorPassword(Constants.STRING.PASSWORD_ERROR);
  return isValidPassword;
};

/**
* @name _onPress
* @desc onPres event
* @private
*/
const _onPress = () => {
  let emailData = _validateEmailAddress();
  let passwordData = _validatePassword();

  if (emailData && passwordData) {
    loginApp(email, password);
  } else {
    Alert.alert(Constants.STRING.EMPTY_TITLE, Constants.STRING.EMPTY_VALUES);
  }
};

/**
* @name loginApp

```

```

* @param {string} email
* @param {string} password
*/
const loginApp = (email, password) => {
  try {
    setIsLoading(true);
    FirebasePlugin.auth()
      .signInWithEmailAndPassword(email, password)
      .then(user => {
        setIsLoading(false);
        navigation.navigate('Register');
      })
      .catch(error => {
        FirebasePlugin.auth()
          .createUserWithEmailAndPassword(email, password)
          .then(user => {
            setIsLoading(false);
            navigation.navigate('Register');
          })
          .catch(error => {
            setIsLoading(false);
            Alert.alert('Invalid Values', error.message);
          });
      });
  });
} catch (error) {
  setIsLoading(true);
  Alert.alert('Firebase Error', error.message);
}
};

```

```

return (
  <DismissKeyboard>
    <KeyboardAvoidingView
      style={stylesLoginScreen.container}
      behavior="height"
      enabled>
        <View style={stylesLoginScreen.container}>
          <SafeAreaView>
            <LogoLogin style={stylesLoginScreen.logo} />
            <View style={stylesLoginScreen.form}>
              <EmailTextField
                onChangeText={email => {
                  setEmail(email);
                }}
                onEndEditing={_validateEmailAddress}
                error={errorEmail}
                source={Images.EMAIL}
                placeholder={Constants.STRING.EMAIL}
                secureTextEntry={false}
                autoCorrect={false}
              />
              <TextInputLogin
                onChangeText={password => {
                  setPassword(password);
                }}
                onEndEditing={_validatePassword}
                error={errorPassword}
                source={Images.USERNAME}
                placeholder={Constants.STRING.PASSWORD}
                secureTextEntry={true}

```

```

        autoCorrect={false}
      />
      <ButtonLogin
        isLoading={isLoading}
        onPress={_onPress}
        titleButton={Constants.STRING.TITLE_BUTTON}
      />
    </View>
  </SafeAreaView>
</View>
</KeyboardAvoidingView>
</DismissKeyboard>
);
};

```

```

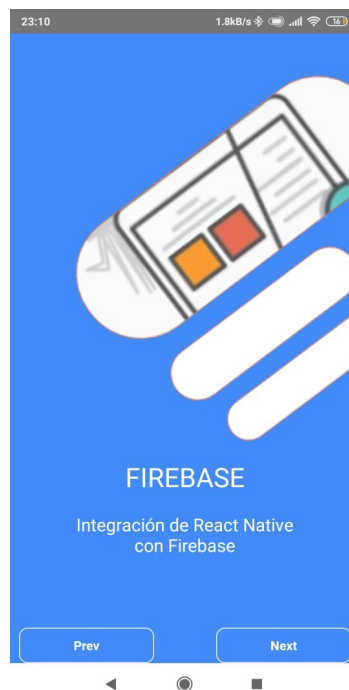
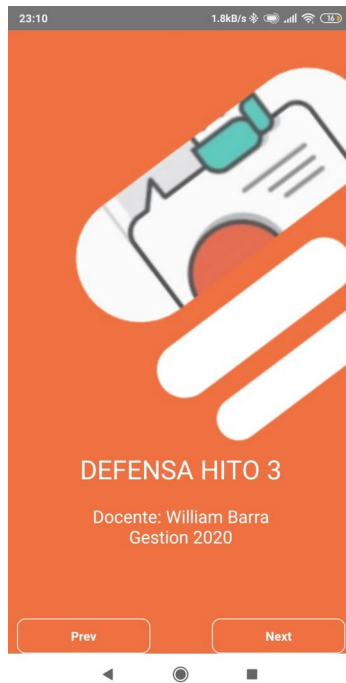
const stylesLoginScreen = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: Colors.azul,
    justifyContent: 'center',
    alignItems: 'center',
  },
  logo: {
    width: '100%',
    resizeMode: 'contain',
    alignSelf: 'center',
  },
  form: {
    justifyContent: 'center',
    width: '80%',
  },
});

```



```
    marginBottom: 20,  
  },  
});  
  
export default LoginScreen;
```

Navegabilidad y creación de todas las pantallas del Onboarding.



```

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

import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';

import imagenes from "../Config/Imagenes"
import Colores from "../Config/Colores"

import OnboardingFactory from "../OnboardingFactory";
import Login from "../Login";

class Pantalla1 extends Component {
  render()
  {
    return (
      <OnboardingFactory
        imagen = {imagenes.img1}
        color = {Colores.verde}
        Titulo = "PDM"
        Descripcion = "Programación de Dispositivos Móviles"
        rutaAdelante = "Pantalla2"
        navigation = {this.props.navigation}
      />
    )
  }
}

class Pantalla2 extends Component {
  render()
  {

```

```

return (
  <OnboardingFactory
    imagen = {imagenes.img2}
    color = {Colores.naranja}
    Titulo = "DEFENSA HITO 3"
    Descripcion = "Docente: William Barra Gestion 2020"
    rutaAtras = "Pantalla1"
    rutaAdelante = "Pantalla3"
    navigation = {this.props.navigation}
  />
)
}
}

class Pantalla3 extends Component {
  render()
  {
    return (
      <OnboardingFactory
        imagen = {imagenes.img1}
        color = {Colores.azul}
        Titulo = "FIREBASE"
        Descripcion = "Integración de React Native con Firebase"
        rutaAtras = "Pantalla2"
        rutaAdelante = "Login"
        navigation = {this.props.navigation}
      />
    )
  }
}

```

```

const Stack = createStackNavigator();
export default class NavegacionPrincipal extends Component {
  render()
  {
    return (
      <NavigationContainer>
        <Stack.Navigator screenOptions={{
          headerShown: false
        }}>
          <Stack.Screen
            name="Pantalla1"
            component={Pantalla1}
            options={{
              title: "Inicio"
            }}
          />
          <Stack.Screen
            name="Pantalla2"
            component={Pantalla2}
            options={{
              title: "Inicio"
            }}
          />
          <Stack.Screen
            name="Pantalla3"
            component={Pantalla3}
            options={{
              title: "Inicio"
            }}
          />
        </Stack.Navigator>
      </NavigationContainer>
    );
  }
}

```

```
<Stack.Screen
  name="Login"
  component={Login}
  options={{
    title: "Login"
  }}
/>
</Stack.Navigator>
</NavigationContainer>
);
}
}
```