	UNIVERSIDAD DON BOSCO **FACULTAD DE INGENIERÍA** **ESCUELA DE COMPUTACIÓN**	**CICLO 02**
	DISEÑO Y PROGRAMACIÓN DE SOFTWARE MULTIPLATAFORMA **CREACIÓN DE COMPONENTES**	**GUIA DE LABORATORIO N° 5**

I. Resultados de aprendizaje

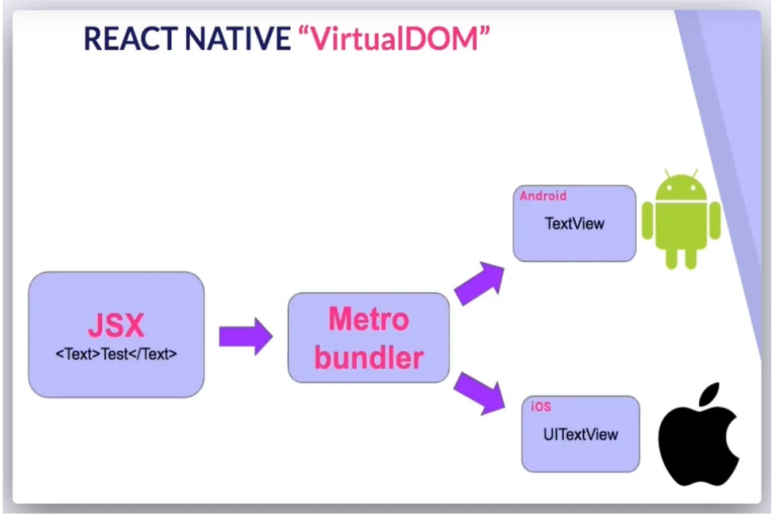
Que el estudiante:

- Cree proyectos de React Native
- Haga diseño de interfaces de usuario y manipule controles.

II. Introducción

¿Qué es React Native?

React Native es un framework de programación de aplicaciones nativas multiplataforma que está basado en JavaScript y ReactJS.



En React existe un **"VirtualDOM"** , en el que tenemos nuestro JSX (JSX es una extensión de la sintaxis de JavaScript), en el cual definimos los documentos HTML, y estos se transforman en componentes del navegador a través de JavaScript.

Con React Native ocurre algo parecido, ya que tenemos nuestros componentes JSX, que van a ser distintos a los componentes HTML y que tendrán otros tags y otros nombres, ya que no estamos utilizando HTML.

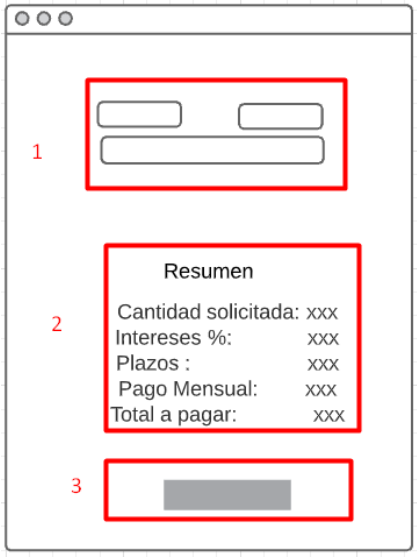
Lo que va a suceder es que el compilador que tiene React Native los va a convertir en elementos nativos de la interfaz para Android y para iOS , lo cual va a permitir que estas aplicaciones tengan un look and feel parecido a aplicaciones nativas, un rendimiento prácticamente igual y una experiencia de navegación y de usuario muy similar a las aplicaciones nativas, ya que lo que se está generando es interfaz nativa.

III. DESARROLLO

Para realizar la práctica se tiene dos alternativas, se puede realizar local con **VSC** u **online**

https://snack.expo.dev/ (**deben de registrarse**)

1. En Visual Studio Code inicie una terminal
2. Sitúese en la carpeta donde desea guardar la aplicación.
3. Procedemos a crear nuestro proyecto "**cotizadorprestamos**"
4. Procederemos a crear la estructura de nuestro proyecto



1.Una cabecera donde tendremos un formulario para extraer los datos para realizar la cotización

2. La sección donde nos mostrará la información sobre el interés, el plazo y pago mensual a realizar del préstamo solicitado.

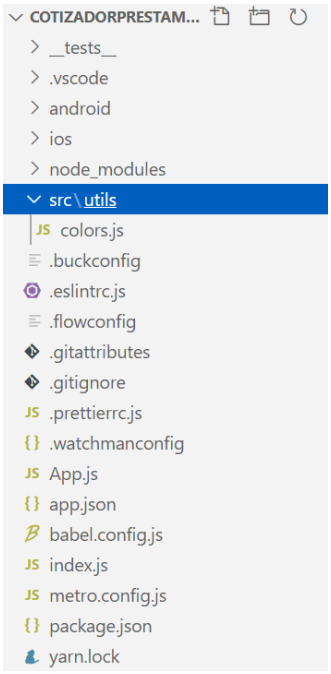
3.Donde tendremos el botón para realizar el cálculo o recálculo de nuestro préstamo

5. Primero vamos a empezar a modificar nuestro archivo App.js

```
/**
 * @format
 * @flow strict-local
 */
import React from 'react';
import {
  SafeAreaView,
  StyleSheet,
  View,
  Text,
} from 'react-native';

export default function App(){
return(
  <>
  <SafeAreaView style={styles.Header}>
    <Text>Header</Text>
  </SafeAreaView>
  <View>
    <Text>Result</Text>
  </View>
  <View>
    <Text>Footer</Text>
  </View>
  </>
);
}

const styles = StyleSheet.create({
  Header:{
    backgroundColor:'blue',
    height:200,
    borderBottomLeftRadius:30,
    borderBottomRightRadius:30,
    alignItems:'center'
  }
})
```

Este código genera la siguiente vista:

6. para trabajar con los mismos colores en toda la aplicación podemos crear un archivo para el manejo de los nombres de los colores:
    a. Primero crearemos una carpeta llamada src, dentro de esta ubicamos una carpeta de nombre utils y dentro de esta crearemos un archivo de nombre colors.js.



    b. Nuestro archivo colors.js tendremos el siguiente código:

```
export default{

  PRIMARY_COLOR:'#0098D3',

  PRIMARY_COLOR_DARk: '#006691',

}
```

7. Ahora procedemos a importar estos colores dentro de nuestra app. Haciendo los siguientes cambios en app.js
    a. primero importar nuestro fichero donde guardamos los colores que utilizaremos dentro de nuestra aplicación.

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

import colors from './src/utils/colors';

export default function App(){
return(
  <>
  <SafeAreaView style={styles.Header}>
    <Text>Header</Text>
  </SafeAreaView>
```

b. Vamos a cambiar el color en nuestras reglas de estilo de la siguiente manera:

```
const styles = StyleSheet.create({
  Header:{
    backgroundColor: colors.PRIMARY_COLOR,
    height:200,
    borderBottomLeftRadius:30,
    borderBottomRightRadius:30,
    alignItems:'center'
  }
})
```

8. Vamos a configurar nuestro Statusbar del telefono, asi cuando nuestro color de la aplicación sea oscuro está siempre pueda tener visibilidad para ello realizaremos los siguientes cambios:

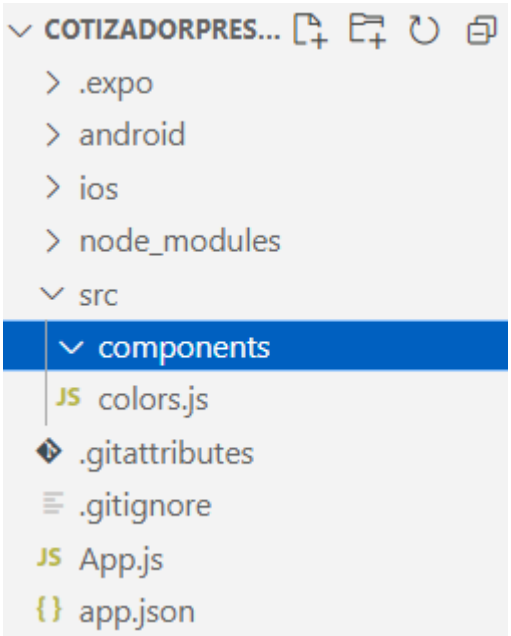
```
import React from 'react';
import {   SafeAreaView, StyleSheet, View, Text, StatusBar,} from 'react-native';

import colors from './src/utils/colors';

export default function App(){
return(
  <>
  <StatusBar barStyle="light-content"/>
  <SafeAreaView style={styles.Header}>
    <Text>Header</Text>
  </SafeAreaView>
```

9. Ahora empezaremos a darle forma a nuestro proyecto primero le pondremos un titulo haciendo los siguientes cambios:

```
<StatusBar barStyle="light-content"/>
<SafeAreaView style={styles.Header}>
  <Text style={styles.HeadApp}>Cotizador de Prestamos</Text>
</SafeAreaView>
<View>
  <Text>Result</Text>
</View>
<View>
  <Text>Footer</Text>
```

```
const styles = StyleSheet.create({
  Header:{
    backgroundColor: colors.PRIMARY_COLOR,
    height:200,
    borderBottomLeftRadius:30,
    borderBottomRightRadius:30,
    alignItems:'center'
  },
  HeadApp:{
    fontSize:25,
    fontWeight:'bold',
    color:'#fff',
    marginTop:15,
  },
})
```

10. Ya con el título podemos ir creando nuestros componentes, para eso vamos a crear una carpeta dentro de nuestro proyecto con el nombre components



11. Dentro de la carpeta components, vamos agregar el archivo Forms.js el cual contendrá el siguiente código:

```jsx
import React from 'react';
import {StyleSheet, TextInput, View} from 'react-native';
import {Picker} from '@react-native-picker/picker';
import colors from '../utils/colors';

export default function Form(props) {
  const {setCapital, setInterest, setMonths} = props;

  return (
    <View style={styles.viewForm}>
      <View style={styles.viewInputs}>
        <TextInput
          placeholder="Cantidad a pedir"
          keyboardType="numeric"
          style={styles.input}
          onChange={(e) => setCapital(e.nativeEvent.text)}
        />
        <TextInput
          placeholder="Interes %"
          keyboardType="numeric"
          style={[styles.input, styles.inputPercentage]}
          onChange={(e) => setInterest(e.nativeEvent.text)}
        />
      </View>
      <Picker
        style={picketSelectStyles}
        onValueChange={(value) => setMonths(value)}
        placeholder={{
          label: 'Seleccióna los plazos...',
          value: null,
        }}>
            <Picker.Item label="3 meses" value="3" />
            <Picker.Item label="6 meses" value="6" />
            <Picker.Item label="12 meses" value="12" />
            <Picker.Item label="24 meses" value="24" />
        </Picker>
    </View>
  );
}


const styles = StyleSheet.create({
  viewForm: {
    position: 'absolute',
    bottom: 0,
    width: '85%',
    paddingHorizontal: 50,
    backgroundColor: colors.PRIMARY_COLOR_DARK,
    borderRadius: 30,
    height: 180,
    justifyContent: 'center',
  },
  viewInputs: {
    flexDirection: 'row',
  },
  input: {
    height: 50,
    backgroundColor: '#fff',
```

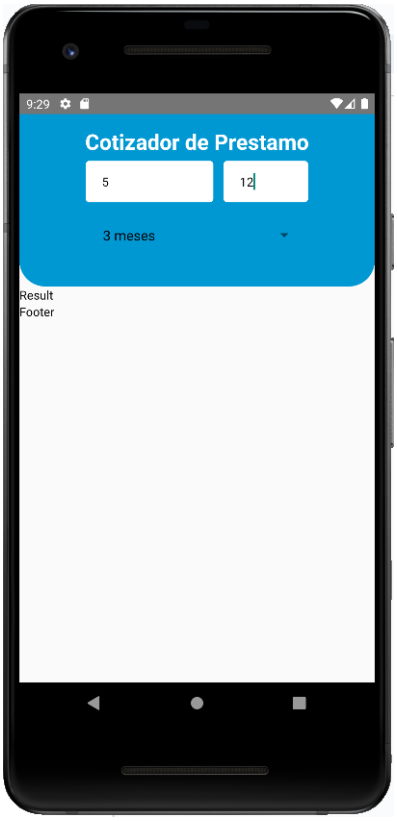
```
    borderWidth: 1,
    borderColor: colors.PRIMARY_COLOR,
    borderRadius: 5,
    width: '60%',
    marginRight: 5,
    marginLeft: -5,
    marginBottom: 10,
    color: '#000',
    paddingHorizontal: 20,
  },
  inputPercentage: {
    width: '40%',
    marginLeft: 5,
  },
});


const picketSelectStyles = StyleSheet.create({
  inputIOS: {
    fontSize: 16,
    paddingVertical: 12,
    paddingHorizontal: 10,
    borderWidth: 1,
    borderColor: 'grey',
    borderRadius: 4,
    color: 'black',
    paddingRight: 30,
    backgroundColor: '#fff',
    marginLeft: -5,
    marginRight: -5,
  },
  inputAndroid: {
    fontSize: 16,
    paddingHorizontal: 10,
    paddingVertical: 8,
    borderWidth: 0.5,
    borderColor: 'grey',
    borderRadius: 8,
    color: 'black',
    paddingRight: 30,
    backgroundColor: '#fff',
  },
});
```

12. debemos instalar el siguiente componente en nuestro proyecto para poder utilizar el picker para los meses:

    npm install react-native-picker-select
    npm install @react-native-picker/picker
    npx pod-install
13. Debemos modificar también nuestro archivo App.js

```
import React, { useState } from 'react';
import {   SafeAreaView, StyleSheet, View, Text, StatusBar,} from 'react-native';
import colors from './src/utils/colors';
import Form from './src/components/Form';

export default function App(){
  const [capital, setCapital] = useState(null);
  const [interest, setInterest] = useState(null);
  const [months, setMonths] = useState(null);

return(
  <>
  <StatusBar barStyle="light-content"/>
  <SafeAreaView style={styles.Header}>
    <Text style={styles.HeadApp}>Cotizador de Prestamos</Text>
    <Form
          setCapital={setCapital}
          setInterest={setInterest}
          setMonths={setMonths}
        />
  </SafeAreaView>
  <View>
    <Text>Result</Text>
  </View>
  <View>
    <Text>Footer</Text>
  </View>
  </>
```

14. Podemos pedir datos y capturarlos:



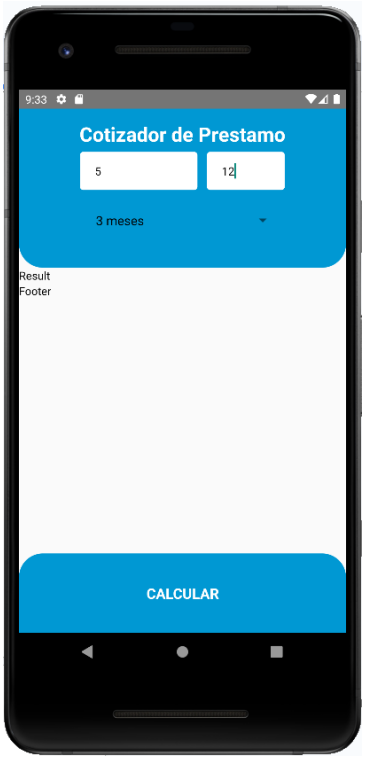
15. Ahora que podemos capturar los datos desde el formulario, procederemos a realizar la segunda parte de nuestra aplicación, vamos a agregar un nuevo archivo a nuestra carpeta components llamado Footer.js

```
import { RefreshControlBase } from "react-native";
import React from 'react';
import {StyleSheet, Text, View, TouchableOpacity} from 'react-native';
import colors from '../utils/colors';
export default function Footer(props) {
  const {calculate} = props;
  return (
    <View style={styles.viewFooter}>
      <TouchableOpacity style={styles.button} onPress={calculate}>
        <Text style={styles.text}>CALCULAR</Text>
      </TouchableOpacity>
    </View>
  );
}
const styles = StyleSheet.create({
  viewFooter: {
    position: 'absolute',
    bottom: 0,
    width: '100%',
    backgroundColor: colors.PRIMARY_COLOR,
    height: 100,
    borderTopLeftRadius: 30,
    borderTopRightRadius: 30,
    alignItems: 'center',
    justifyContent: 'center',
  },
  button: {
    backgroundColor: colors.PRIMARY_COLOR_DARK,
    padding: 16,
    borderRadius: 20,
    width: '75%',
  },
  text: {
    fontWeight: 'bold',
    fontSize: 18,
    color: '#fff',
    textAlign: 'center',
  },
});
```

16. También debemos de modificar el archivo App.js

```
11  import colors from './src/utils/colors';
12  import Form from './src/components/Form';
13  import Footer from './src/components/Footer';
14
15  export default function App(){
16    const [capital, setCapital] = useState(null);
17    const [interest, setInterest] = useState(null);
18    const [months, setMonths] = useState(null);
19
20  return(
21    <>
22    <StatusBar barStyle="light-content"/>
23    <SafeAreaView style={styles.Header}>
24      <Text style={styles.HeadApp}>Cotizador de Prestamos</Text>
25      <Form
26              setCapital={setCapital}
27              setInterest={setInterest}
28              setMonths={setMonths}
29            />
30    </SafeAreaView>
31    <View>
32      <Text>Result</Text>
33    </View>
34    <Footer></Footer>
35    </>
36  );
37  }
```

17. podemos ver la siguiente pantalla



18. Procederemos a mostrar los resultados para ello debemos vamos a crear nuestro tercer componente llamado Result.js que debe contener el siguiente codigo:

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

export default function Result(props) {
  const {capital, interest, months, total, errorMessage} = props;

  return (
    <View style={styles.content}>
      {total && (
        <View style={styles.boxResult}>
          <Text style={styles.title}>RESUMEN</Text>
          <DataResult title="Cantidad solicitada:" value={`${capital}
€`} />
          <DataResult title="Interes %:" value={`${interest} %`} />
          <DataResult title="Plazos:" value={`${months} meses`} />
          <DataResult title="Pago mensual:"
value={`${total.monthlyFee} €`} />
          <DataResult
            title="Total a pagar:"
            value={`${total.totalPayable} €`}
          />
        </View>
      )}
      <View>
        <Text style={styles.error}>{errorMessage}</Text>
      </View>
    </View>
  );
}


function DataResult(props) {
  const {title, value} = props;


  return (
    <View style={styles.value}>
      <Text>{title}</Text>
      <Text>{value}</Text>
    </View>
  );
}


const styles = StyleSheet.create({
  content: {
    marginHorizontal: 40,
  },
  boxResult: {
```

```
    padding: 30,
  },
  title: {
    fontSize: 25,
    textAlign: 'center',
    fontWeight: 'bold',
    marginBottom: 20,
  },
  value: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    marginBottom: 20,
  },
  error: {
    textAlign: 'center',
    color: '#f00',
    fontWeight: 'bold',
    fontSize: 20,
  },
});
```

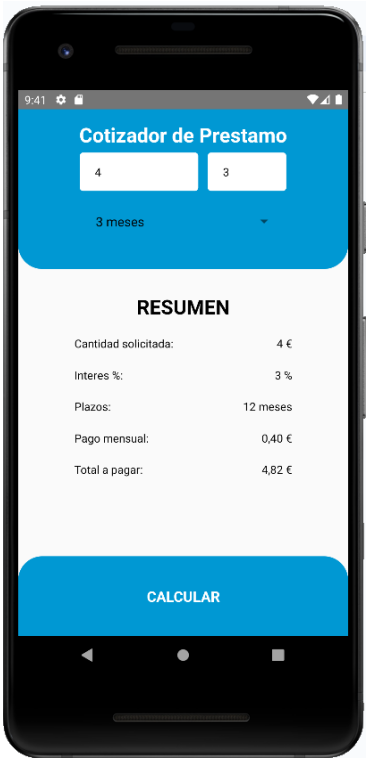
19. y procedemos a modificar nuevamente nuestro App.js, de tal forma que debe quedarnos de la siguiente forma

```
import React, {useState, useEffect} from 'react';
import {
  StyleSheet,
  View,
  Text,
  SafeAreaView,
  StatusBar,
  Button,
} from 'react-native';
import Form from './src/components/Form';
import Footer from './src/components/Footer';
import Result from './src/components/Result';
import colors from './src/utils/colors';



export default function App() {
  const [capital, setCapital] = useState(null);
  const [interest, setInterest] = useState(null);
  const [months, setMonths] = useState(null);
  const [total, setTotal] = useState(null);
  const [errorMessage, setErrorMessage] = useState('');

  useEffect(() => {
    if (capital && interest && months) calculate();
```

```
    else reset();
  }, [capital, interest, months]);

  const calculate = () => {
    reset();
    if (!capital) {
      setErrorMessage('Añade la cantidad que quieres solicitar');
    } else if (!interest) {
      setErrorMessage('Añade el interes del prestamos');
    } else if (!months) {
      setErrorMessage('Seleccióna los meses a pagar');
    } else {
      const i = interest / 100;
      const fee = capital / ((1 - Math.pow(i + 1, -months)) / i);
      setTotal({
        monthlyFee: fee.toFixed(2).replace('.', ','),
        totalPayable: (fee * months).toFixed(2).replace('.', ','),
      });
    }
  };

  const reset = () => {
    setErrorMessage('');
    setTotal(null);
  };

  return (
    <>
      <StatusBar barStyle="light-content" />
      <SafeAreaView style={styles.safeArea}>
        <View style={styles.background} />
        <Text style={styles.titleApp}>Cotizador de Prestamos</Text>
        <Form
          setCapital={setCapital}
          setInterest={setInterest}
          setMonths={setMonths}
        />
      </SafeAreaView>
      <Result
        capital={capital}
        interest={interest}
        months={months}
        total={total}
        errorMessage={errorMessage}
      />
      <Footer calculate={calculate} />
    </>
  );
}
```

```
const styles = StyleSheet.create({
 safeArea: {
  height: 290,
  alignItems: 'center',
 },
 background: {
  backgroundColor: colors.PRIMARY_COLOR,
  height: 200,
  width: '100%',
  borderBottomLeftRadius: 30,
  borderBottomRightRadius: 30,
  position: 'absolute',
  zIndex: -1,
 },
 titleApp: {
  fontSize: 25,
  fontWeight: 'bold',
  color: '#fff',
  marginTop: 15,
 },
});
```

20. Después de estos cambios deberemos poder ver la aplicación:



21. Para finalizar, si no da errores la aplicación creará la apk para android.

III. EJERCICIOS COMPLEMENTARIOS

Calcular el salario neto de un empleado, solicitando nombre y salario base

salario neto = salario base – Deducciones (ISSS- 3%, AFP-4%, RENTA-5%)

Al finalizar la aplicación debe mostrar el salario neto del empleado.

VI. BIBLIOGRAFIA

- React, una biblioteca de JavaScript. (2013-2020). Facebook, Inc. Jordan Walke. Recuperado de https://es.reactjs.org/docs/getting-started.html