	<b>UNIVERSIDAD DON BOSCO</b> <b>FACULTAD DE INGENIERÍA</b> <b>ESCUELA DE COMPUTACIÓN</b>	<b>CICLO 02</b>
	<b>DISEÑO Y PROGRAMACIÓN DE SOFTWARE MULTIPLATAFORMA</b> <b>REACT NATIVE LAYOUT AND FLEXBOX</b>	<b>GUIA DE LABORATORIO</b> <b>Nº 7</b>

## I. OBJETIVOS DEL APRENDIZAJE

Que el estudiante aprenda a:

- Adquiera el dominio para trabajar con FlexDirection.
- Cree proyectos react native utilizando FlexDirection.

## II. INTRODUCCIÓN

**Flexbox** está diseñado para proporcionar un diseño uniforme en diferentes tamaños de pantallas. Es normal que se utilice una combinación de *flexDirection*, *alignItems* y *justifyContent* para lograr una posición correcta de los elementos. Flexbox definirá cómo se “llenarán” sus elementos sobre el espacio disponible a lo largo de su eje principal. El espacio se dividirá de acuerdo con la propiedad **flex** de cada elemento. En el siguiente ejemplo, existe un contenedor principal y tres contenedores secundarios (rojo, amarillo y verde). La propiedad **flexDirection** utilizada en el contenedor principal está definida como “column”, esto significa que los contenedores secundarios serán alineados de arriba hacia abajo con respecto al contenedor principal.

Para el caso de los contenedores secundarios, la propiedad **flex** definida para cada uno de ellos es de la siguiente manera: contenedor rojo -> flex: 1, contenedor amarillo -> flex: 2 y contenedor verde flex: 3. Esto significa que la sumatoria de los flex de cada contenedor nos da un valor de 6, lo que se convierte en el total de espacio del contenedor principal. Dando como resultado final que el contenedor rojo tenga un espacio de 1/6, el contenedor amarillo 2/6 y el contenedor verde 3/6.

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

const Flex = () => {
  return (
    <View style={[[styles.container, {
      flexDirection: "column"
    }]]}>
      <Text>Contenedor principal</Text>

      <View style={{ flex: 1, backgroundColor: "red" }}>
        <Text>Contenedor secundario</Text>
      </View>
      <View style={{ flex: 2, backgroundColor: "darkorange" }} >
        <Text>Contenedor secundario</Text>
      </View>
      <View style={{ flex: 3, backgroundColor: "green" }} >
        <Text>Contenedor secundario</Text>
      </View>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
  },
});

export default Flex;
```



En React Native se pueden utilizar las siguientes propiedades para garantizar un buen diseño de nuestra aplicación:

Propiedad	Valor	Descripción
flexDirection	'column', 'row'	Se utiliza para alinear sus elementos vertical u

		horizontalmente.
justifyContent	'center', 'flex-start', 'flex-end', 'space-around', 'space-between'	Se utiliza para distribuir los elementos dentro del contenedor.
alignItems	'center', 'flex-start', 'flex-end', 'stretched'	Se utiliza para distribuir el elemento dentro del contenedor a lo largo del eje secundario (opuesto a flexDirection).

### III. DESARROLLO DE PRÁCTICA

#### Parte I

1. Cree un proyecto react-native con el nombre de: agencia de viajes.
2. Edite el archivo App.js con las siguientes propiedades Flex.

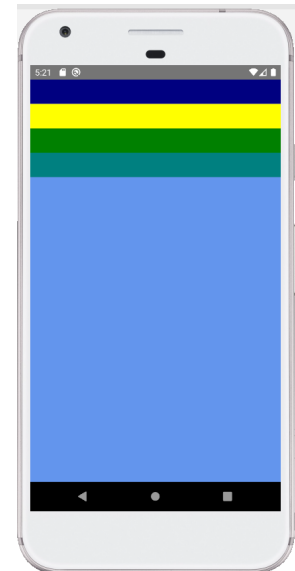
```
import React from 'react';
import {View, StyleSheet} from 'react-native';

const App = () =>{
  return(
    <>
    <View style={styles.contenedor}>
      <View style={styles.caja1}></View>
      <View style={styles.caja2}></View>
      <View style={styles.caja3}></View>
      <View style={styles.caja4}></View>
    </View>
    </>
  );
};

const styles = StyleSheet.create({
  contenedor:{
    backgroundColor: 'cornflowerblue',
    flex:1,
    flexDirection: 'column'
  },
  caja1:{
    padding:20,
    backgroundColor: 'navy'
  },
  caja2:{
    padding:20,
    backgroundColor: 'yellow'
  },
  caja3:{
    padding:20,
    backgroundColor: 'green'
  },
  caja4:{
    padding:20,
    backgroundColor: 'teal'
  },
});

export default App;
```

Resultado



3. Ahora vamos a modificar la siguiente propiedad en nuestra constante estilo para visualizar las cajas de izquierda a derecha

```
const styles = StyleSheet.create({
  contenedor:{
    backgroundColor:'cornflowerblue',
    flex:1,
    flexDirection:'row'
  },
  caja1:{
    padding:20,
    backgroundColor:'navy',
    flex:1,
  },
  caja2:{
    padding:20,
    backgroundColor:'yellow',
    flex:1,
  },
  caja3:{
    padding:20,
    backgroundColor:'green',
    flex:1,
  },
  caja4:{
    padding:20,
    backgroundColor:'teal',
    flex:2,
  },
});
```

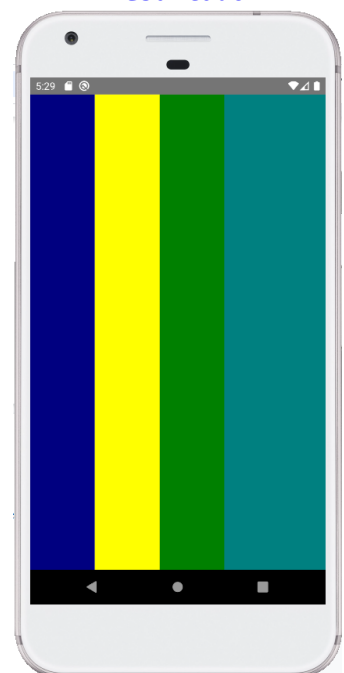
Resultado



4. Ahora vamos a modificar el factor de crecimiento de cada uno de los elementos. Modifique la constante styles de la siguiente forma:

```
const styles = StyleSheet.create({
  contenedor:{
    backgroundColor:'cornflowerblue',
    flex:1,
    flexDirection:'row'
  },
  caja1:{
    padding:20,
    backgroundColor:'navy',
    flex:1,
  },
  caja2:{
    padding:20,
    backgroundColor:'yellow',
    flex:1,
  },
  caja3:{
    padding:20,
    backgroundColor:'green',
    flex:1,
  },
  caja4:{
    padding:20,
    backgroundColor:'teal',
    flex:2,
  },
});
```

Resultado



5. Modifiquemos las propiedades de estilo para probar la alineación de contenido, los valores que puede utilizar para esas propiedades las encuentra en la tabla presentada en la introducción de esta guía. Utilice cada propiedad con sus respectivos valores e identifique los cambios que se producen.

```
const styles = StyleSheet.create({
  contenedor:{
    backgroundColor: 'cornflowerblue',
    flex:1,
    flexDirection:'row',
    justifyContent: 'space-around',
    alignItems: 'flex-end'
  },
});
```

## Parte II

1. Ahora que se han comprendido las propiedades de **flex**, vamos a diseñar nuestra App para una agencia de viaje. Primero descargue las imágenes proporcionadas por su docente, luego cree una carpeta llamada src y guarde sus recursos en dicha carpeta.
2. Modifique su archivo App.js con el siguiente código:

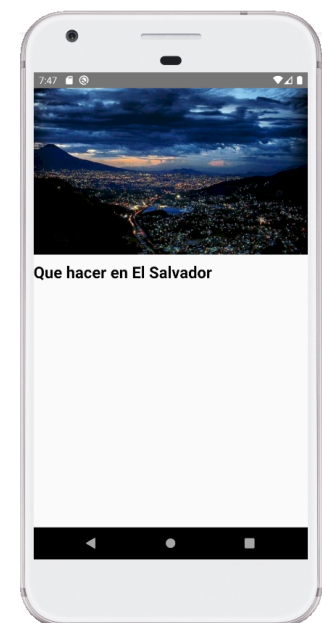
```
import React from 'react';
import {View, StyleSheet, Image} from
'react-native';

const App = () =>{
  return(
    <>
    <View >
      <View style={{flexDirection:'row'}} >
        <Image
          style={styles.banner}
          source={require('./src/img/bg.jpg')} />
        </View>
      </View>
    </>
  );
};

const styles = StyleSheet.create({
  banner:{
    height:250,
    flex:1
  }
});

export default App;
```

Resultado



3. Crearemos un componente ScrollView para mostrar las diferentes actividades en El Salvador.
  - a. Importe las librerías correspondiente al elemento ScrollView

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

b. Agregue el siguiente código

```
<View style={{flexDirection: 'row'}}>
  <Image style={styles.banner} source={require('./src/img/bg.jpg')} />
</View>

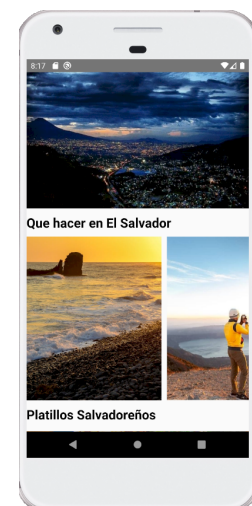
<View style={styles.contenedor}>
  <Text style={styles.titulo}>Que hacer en El Salvador</Text>
  <ScrollView horizontal>
    <View>
      <Image
        style={styles.ciudad}
        source={require('./src/img/actividad1.jpg')}
      />
    </View>
    <View>
      <Image
        style={styles.ciudad}
        source={require('./src/img/actividad2.jpg')}
      />
    </View>
    <View>
      <Image
        style={styles.ciudad}
        source={require('./src/img/actividad3.jpg')}
      />
    </View>
    <View>
      <Image
        style={styles.ciudad}
        source={require('./src/img/actividad4.jpg')}
      />
    </View>
    <View>
      <Image
        style={styles.ciudad}
        source={require('./src/img/actividad5.jpg')}
      />
    </View>
  </ScrollView>
</View>
```

c. Modifique los siguientes estilos

```
const styles = StyleSheet.create({
  banner:{
    height:250,
    flex:1
  },
  titulo:{
    fontWeight:'bold',
    fontSize:24,
    marginVertical:10
  },
  contenedor:{
    marginHorizontal:10,
  },
  ciudad:{
    width:250,
    height:300,
    marginRight:10
  }
});

export default App;
```

Resultado



4. Ahora agregamos una lista de platillos salvadoreños

a. Sustituir el View por un ScrollView

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

const App = () =>{
  return(
    <>
    <ScrollView >
      <View style={{flexDirection:'row'}} >
        <Image
          style={styles.banner}
          source={require('./src/img/bg.jpg')} />
      </View>
    </>
  )
}
```

- b. Agregar el siguiente código después del ScrollView de: “Cosas que hacer en El Salvador”

```
<Text style={styles.titulo}>Platillos Salvadoreños</Text>
<View>
  <View>
    <Image
      style={styles.mejores}
      source={require('./src/img/mejores1.jpg')} />
    </View>

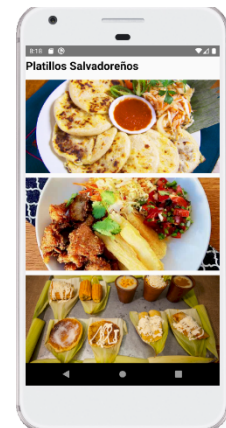
    <View>
      <Image
        style={styles.mejores}
        source={require('./src/img/mejores2.jpg')} />
      </View>

    <View>
      <Image
        style={styles.mejores}
        source={require('./src/img/mejores3.jpg')} />
      </View>
    </View>
  </View>
```

- c. Adicionar la siguiente regla de estilo en la variable “style”

```
mejores:{
  width:'100%',
  height:200,
  marginVertical:5
}
```

Resultado



5. Ahora procedemos a crear la sección de rutas.
- a. Adicionar la siguiente sección de código

```

61 |         style={styles.mejores}
62 |         source={require('./src/img/mejores3.jpg')} />
63 |     </View>
64 | </View>
65 |
66 |     <Text style={styles.titulo}>Rutas Turisticas</Text>
67 |     <View style={styles.listado}>
68 |       <View style={styles.listaItem}>
69 |         <Image
70 |           style={styles.mejores}
71 |           source={require('./src/img/ruta1.jpg')} />
72 |       </View>
73 |
74 |       <View style={styles.listaItem}>
75 |         <Image
76 |           style={styles.mejores}
77 |           source={require('./src/img/ruta2.jpg')} />
78 |       </View>
79 |
80 |       <View style={styles.listaItem}>
81 |         <Image
82 |           style={styles.mejores}
83 |           source={require('./src/img/ruta3.jpg')} />
84 |       </View>
85 |
86 |       <View style={styles.listaItem}>
87 |         <Image
88 |           style={styles.mejores}
89 |           source={require('./src/img/ruta4.jpg')} />
90 |       </View>
91 |     </View>
92 |
93 | </View>

```

b. Adicionar las siguientes reglas de estilo

```

listaItem:{
  flexBasis:'49%'
},
listado:{
  flexDirection:'row',
  flexWrap:'wrap',
  justifyContent:'space-between'
}

```

Resultado



6. Ahora adicionaremos un componente Modal para brindar información de las imágenes, para ello deberemos realizar lo siguiente en código:

a. Agregamos lo siguiente en nuestro código

```
import React,{useState} from 'react';
import {View, StyleSheet,Image, Text, ScrollView, Modal, Button, TouchableHighlight} from 'react-native';

const App = () =>{
  const [modalVisibleplaya, setModalVisibleplaya] = useState(false);
  return(
    <>
    <ScrollView >
      <Modal transparent={true} animationType="slide" visible={modalVisibleplaya} onRequestClose={() => {
        alert('Modal has been closed.')}} >
        <View style={styles.vistaModal}>
          <View style={styles.Modal}>
            <Text style={styles.subtitulo}>Ir a la playa</Text>
            <Text>El Salvador cuenta con hermosas playas a nivel Centroamérica.</Text>
            <Button title="Cerrar" onPress={()=>{setModalVisibleplaya(!modalVisibleplaya)}} ></Button>
          </View>
        </Modal>
      </ScrollView>
    </>
  )
}
```

b. Hacemos cambios en la imagen de la actividad 1

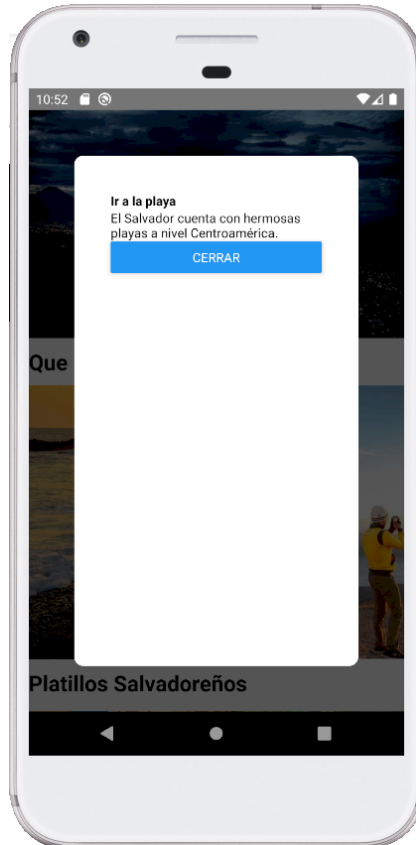
```
<View style={styles.contenedor}>
  <Text style={styles.titulo}>Que hacer en El Salvador</Text>
  <ScrollView horizontal>
    <View>
      <TouchableHighlight
        onPress={()=>{setModalVisibleplaya(!modalVisibleplaya)}}
      >
        <Image
          style={styles.ciudad}
          source={require('./src/img/actividad1.jpg')} />
        </TouchableHighlight>
      </View>
    </ScrollView>
  </View>
```

c. Agregamos los siguientes estilos

```
vistaModal: {
  backgroundColor: '#000000aa',
  flex: 1,
},
Modal: {
  backgroundColor: '#fff',
  margin: 50,
  padding: 40,
  borderRadius: 10,
  flex: 1,
},
subtitulo: {
  fontWeight: 'bold',
  fontSize: 14,
  justifyContent: 'center',
},
```

7. Debemos obtener el siguiente resultado, cuando damos clic en la actividad 1.





8. Proceda a agregar el componente Modal al menos a un elemento de cada sección de la aplicación desarrollada.

#### IV. EJERCICIO COMPLEMENTARIO

Realizar una aplicación para un hotel donde muestre el tipo de habitación, los servicios con que cuenta para los huéspedes y muestre lugares de interés cercanos al hotel.