



PRACTICA 6 – STYLESHEETS

ESTUDIANTE: PÉREZ MELLADO JENNYFER

GRUPO: 211

MAESTRO: IVAN ISAY GUERRA LOPEZ

MATERIA: DESARROLLO DE APPS MÓVILES

FECHA: 09 – 10 – 2025

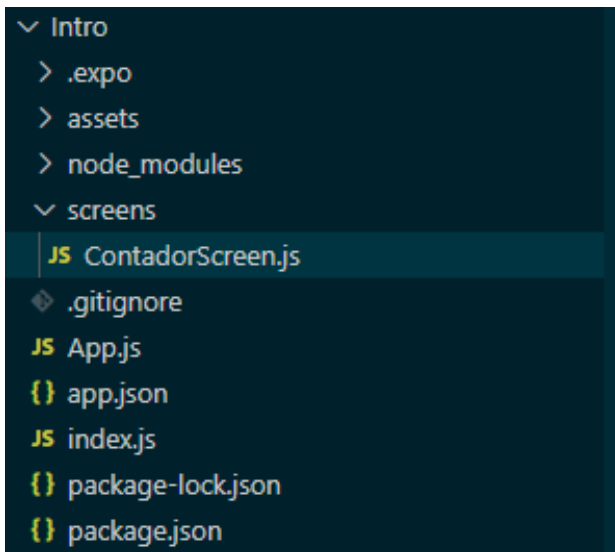
CONTENIDO DE CLASE:

➤ STYLESHEETS

- Permite definir estilos como objetos JavaScript
- Muchas propiedades son parecidas a CSS, pero en camelCase
 - backgroundColor
- Usa Flexbox para distribuir elementos
- Puedes tener un archivo JS para estilos compartidos entre componentes

PRÁCTICA DE CLASE:

Como inicio creamos una nueva carpeta con nuestro proyecto llamada screens, la cual llevara un archivo ContadorScreen del tipo java, a este archivo moveremos todo el contenido de lo que teníamos en el archivo de app.



Y del archivo de App borrar casi todo su contenido, solo dejando la parte principal con el return

```
Intro > JS App.js > ...
1  //1. Imports: Zona de importaciones
2
3  import { StyleSheet, Text, View, Button } from 'react-native';
4
5
6  //2. Main: Zona de componentes
7  export default function App() {
8
9  }
10
11
```

Al archivo app, vamos a agregar la importación de ContadorScreen en el archivo de app, y mandamos a llamar a la función mediante etiquetas para así verlas en la app.

```
1 //1. Imports: Zona de importaciones
2
3 import { StyleSheet, Text, View, Button } from 'react-native';
4 import ContadorScreen from './screens/ContadorScreen';
5
6
7 //2. Main: Zona de componentes
8 export default function App() {
9
10   return (
11     <ContadorScreen />
12   );
13 }
14
15
16
17
```

Ahora, vamos a trabajar solo en el archivo que creamos, arriba en nuestra función, podemos ver la propiedad de estilo, el objeto con ciertas características y que indica las características de estilo que van a afectar la función, o esa parte de la función que estamos viendo en la pantalla de la app, abajo en la zona de estilos, están todas las piezas para modificar.

```
<View style={styles.container}>
  <Text> Contador: {contador}</Text>
  <Button title='Agregar' onPress={()=>setContador(contador + 1)} />
  <Button title='Quitar' onPress={()=>setContador(contador - 1)} />
  <Button title='Reiniciar' onPress={()=>setContador(0)} />
  <StatusBar style="auto" />
</View>

);

//3. Estilos: Zona estética para componentes.
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#d09be0ff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

Las primeras propiedades que podemos observar son `alignItems`, que mueve los objetos de izquierda a derecha, es decir en el eje x; y `justifyContent` que los mueve de arriba abajo, es decir, sobre el eje y; esto mediante las palabras `start`, `center` y `end`.

```
//3. Estilos: Zona estética para componentes.  
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    backgroundColor: '#d09be0ff',  
    alignItems: 'end',  
    justifyContent: 'center',  
  },  
});
```

Las siguientes modificaciones son de texto, color, cambiar tamaño, tipografía, si va a estar en negritas, itálica(cursiva) y para que este tachada, esto mediante `color`, `fontSize`, `fontFamily`, `fontWeight`, `fontStyle` y `textDecorationLine`.

```
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    backgroundColor: '#d09be0ff',  
    alignItems: 'center',  
    justifyContent: 'center',  
  },  
  texto: {  
    color: 'rgba(31, 9, 82, 1)',  
    fontSize: 30,  
    fontFamily: 'Times New Roman',  
    fontWeight: 'bold',  
    fontStyle: 'italic',  
    textDecorationLine: 'line-through'  
  },  
});
```

~~Contador:~~

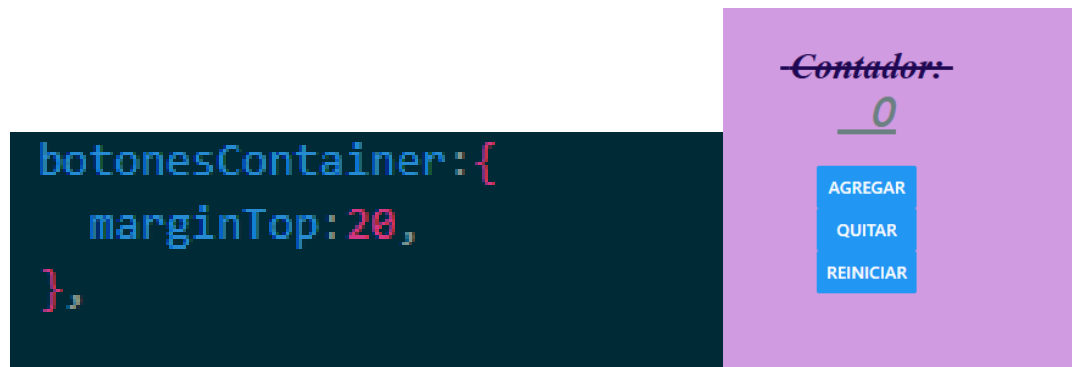
0

AGREGAR

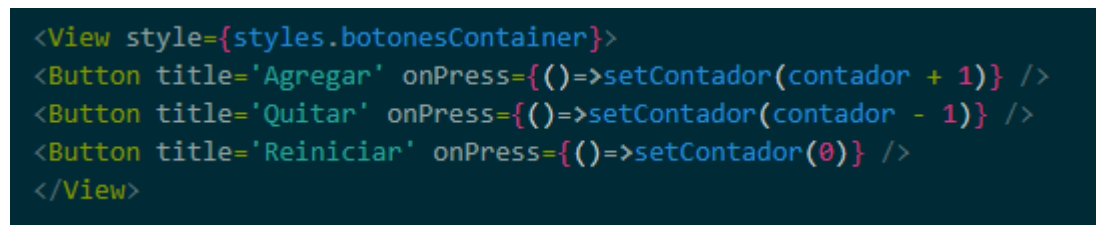
QUITAR

REINICIAR

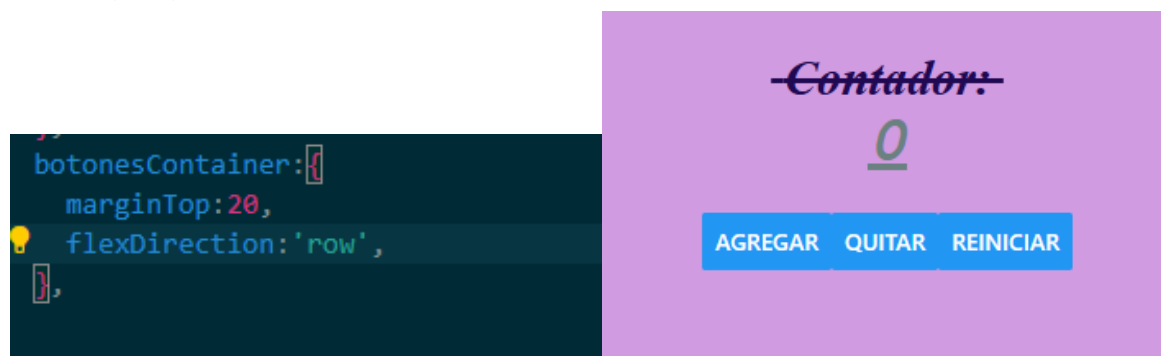
Otras formas de estilo para modificar los botones son el margen que nos va a separar los botones de el ultimo texto, y los va a poner como del mismo tamaño de cuadrado



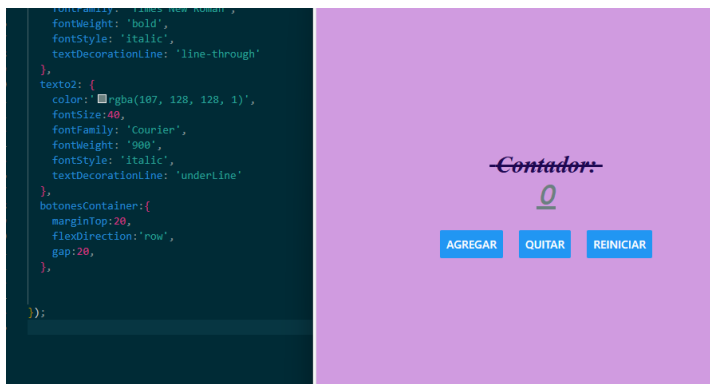
Para que se vea se agrega etiqueta de view y estilo en los botones



La otra propiedad es flexDirection, para acomodar los colores por columna, en vertical, etc,



La última propiedad es para separar cada botón en su espacio, separación entre cada botón



Los botones que usamos son predeterminados, por ello su estilo no puede modificarse mucho, el color es lo único que puede ser modificado en estos mismos, y eso se realiza directamente como una propiedad en el botón.

```
<View style={styles.container}>
  <Text style={styles.text}> Contador: </Text>
  <Text style={styles.text2}>{contador}</Text>

  <View style={styles.botonesContainer}>
    <Button color='yellow' title='Agregar' onPress={()=>setContador(contador + 1)} />
    <Button color='pink' title='Quitar' onPress={()=>setContador(contador - 1)} />
    <Button color='blue' title='Reiniciar' onPress={()=>setContador(0)} />
  </View>

  <StatusBar style="auto" />
</View>
```

Al realizar dicha modificación se ve de esta manera.



Conclusión: En esta práctica aprendí los principales componentes de estilo, para así modificar tanto texto y hasta cierto punto botones dentro de la app.