

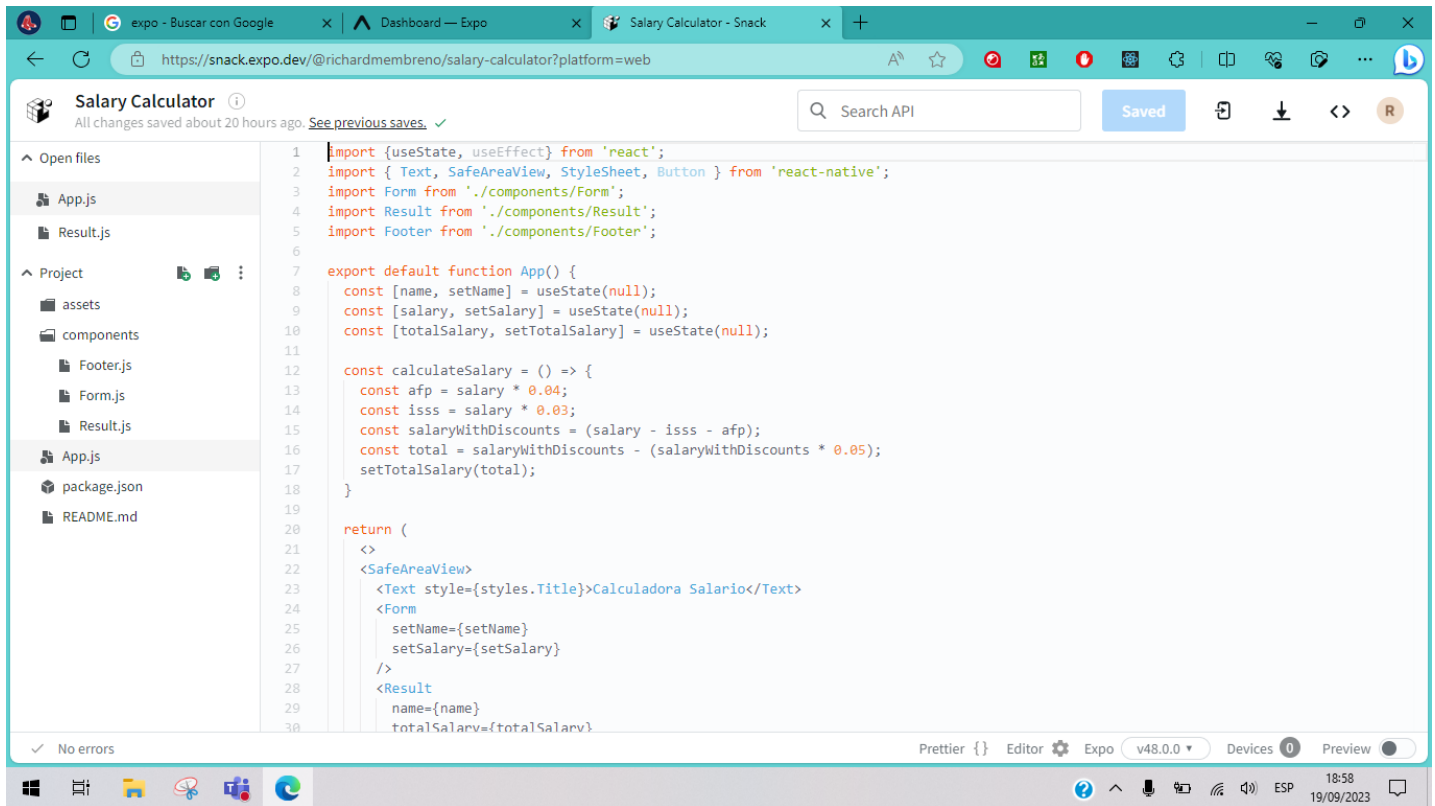
Ricardo Enrique Membreño Rivera

Diseño y Programación de Software Multiplataforma

Guía 5

[https://github.com/richardMembreno/Discusion\\_Guia5.git](https://github.com/richardMembreno/Discusion_Guia5.git)

Archivo App.js



The screenshot displays the Expo Snack web editor interface. The browser address bar shows the URL `https://snack.expo.dev/@richardmembreno/salary-calculator?platform=web`. The editor title is "Salary Calculator" with a note "All changes saved about 20 hours ago". The left sidebar shows the project structure with files like `App.js`, `Result.js`, `Footer.js`, `Form.js`, `package.json`, and `README.md`. The main editor area shows the `App.js` file with the following code:

```
1 import {useState, useEffect} from 'react';
2 import { Text, SafeAreaView, StyleSheet, Button } from 'react-native';
3 import Form from './components/Form';
4 import Result from './components/Result';
5 import Footer from './components/Footer';
6
7 export default function App() {
8   const [name, setName] = useState(null);
9   const [salary, setSalary] = useState(null);
10  const [totalSalary, setTotalSalary] = useState(null);
11
12  const calculateSalary = () => {
13    const afp = salary * 0.04;
14    const isss = salary * 0.03;
15    const salaryWithDiscounts = (salary - isss - afp);
16    const total = salaryWithDiscounts - (salaryWithDiscounts * 0.05);
17    setTotalSalary(total);
18  }
19
20  return (
21    <>
22    <SafeAreaView>
23      <Text style={styles.Title}>Calculadora Salarial</Text>
24      <Form
25        setName={setName}
26        setSalary={setSalary}
27      />
28      <Result
29        name={name}
30        totalSalary={totalSalary}
31      />
22    </SafeAreaView>
23    </>
24  );
25}
```

The bottom status bar indicates "No errors", "Prettier" formatting, and the Expo version "v48.0.0". The system clock shows 18:58 on 19/09/2023.

```
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
return (
  <>
    <SafeAreaView>
      <Text style={styles.Title}>Calculadora Salario</Text>
      <Form
        setName={setName}
        setSalary={setSalary}
      />
      <Result
        name={name}
        totalSalary={totalSalary}
      />
      <Footer calculateSalary={calculateSalary}/>
    </SafeAreaView>
  </>
);

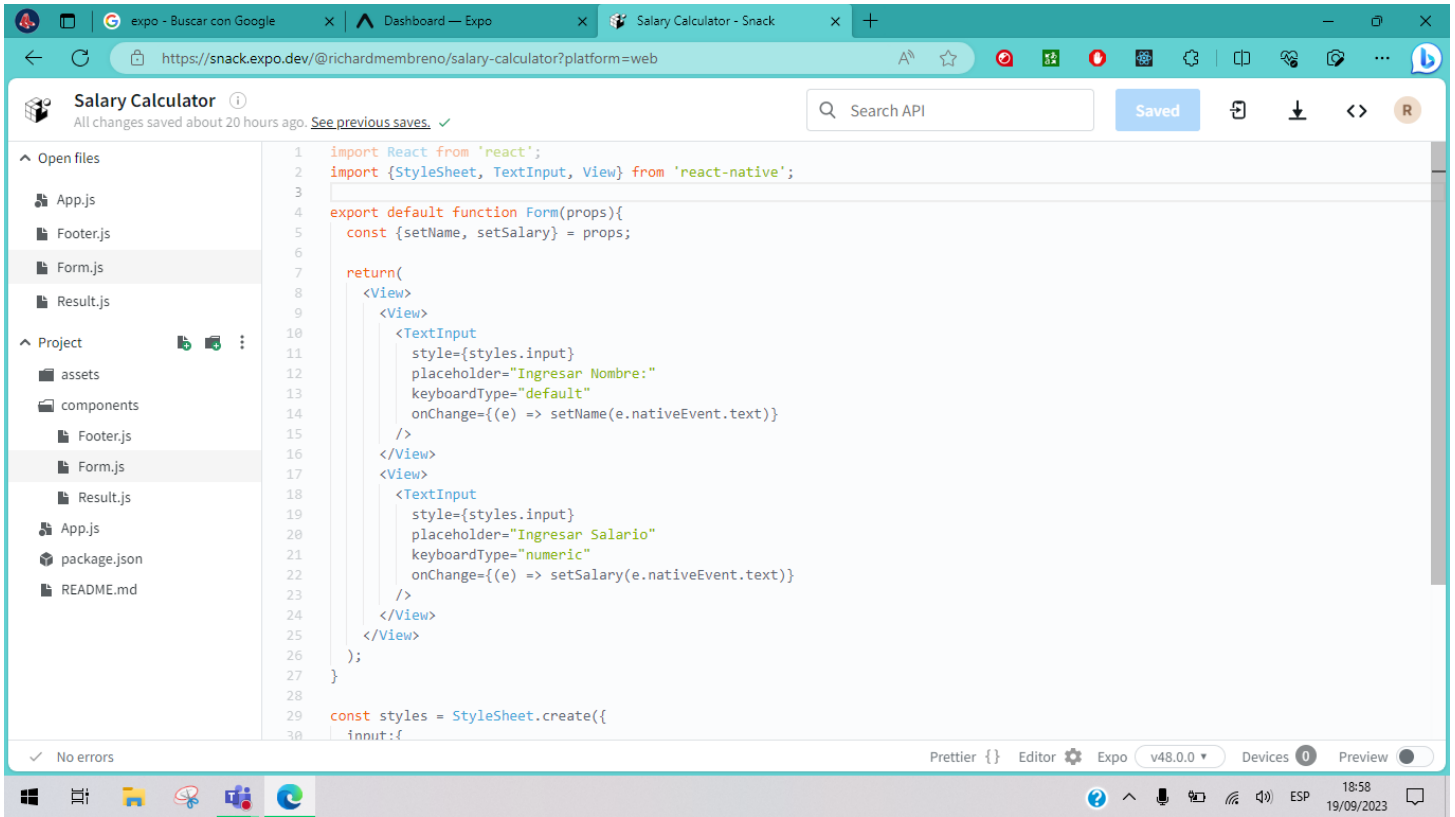
const styles = StyleSheet.create({
  Title: {
    fontSize: 25,
    margin: 'auto',
    padding: 50,
    paddingLeft: 70,
    paddingRight: 70,
    fontFamily: 'Arial',
    fontWeight: 'bold'
  }
});
```

## Archivo Footer.js

```
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
export default function Footer(props) {
  const {calculateSalary} = props;
  return (
    <View>
      <TouchableOpacity onPress={calculateSalary} style={styles.buttonContainer}>
        <Text style={styles.buttonText}>Calcular</Text>
      </TouchableOpacity>
    </View>
  );
}

const styles = StyleSheet.create({
  buttonContainer: {
    elevation: 8,
    backgroundColor: '#5F76EA',
    borderRadius: 10,
    paddingVertical: 10,
    paddingHorizontal: 12,
    width: '85%',
    marginHorizontal: 25
  },
  buttonText: {
    fontSize: 18,
    color: 'white',
    fontWeight: 'bold',
    alignSelf: 'center',
    textTransform: 'uppercase'
  }
});
```

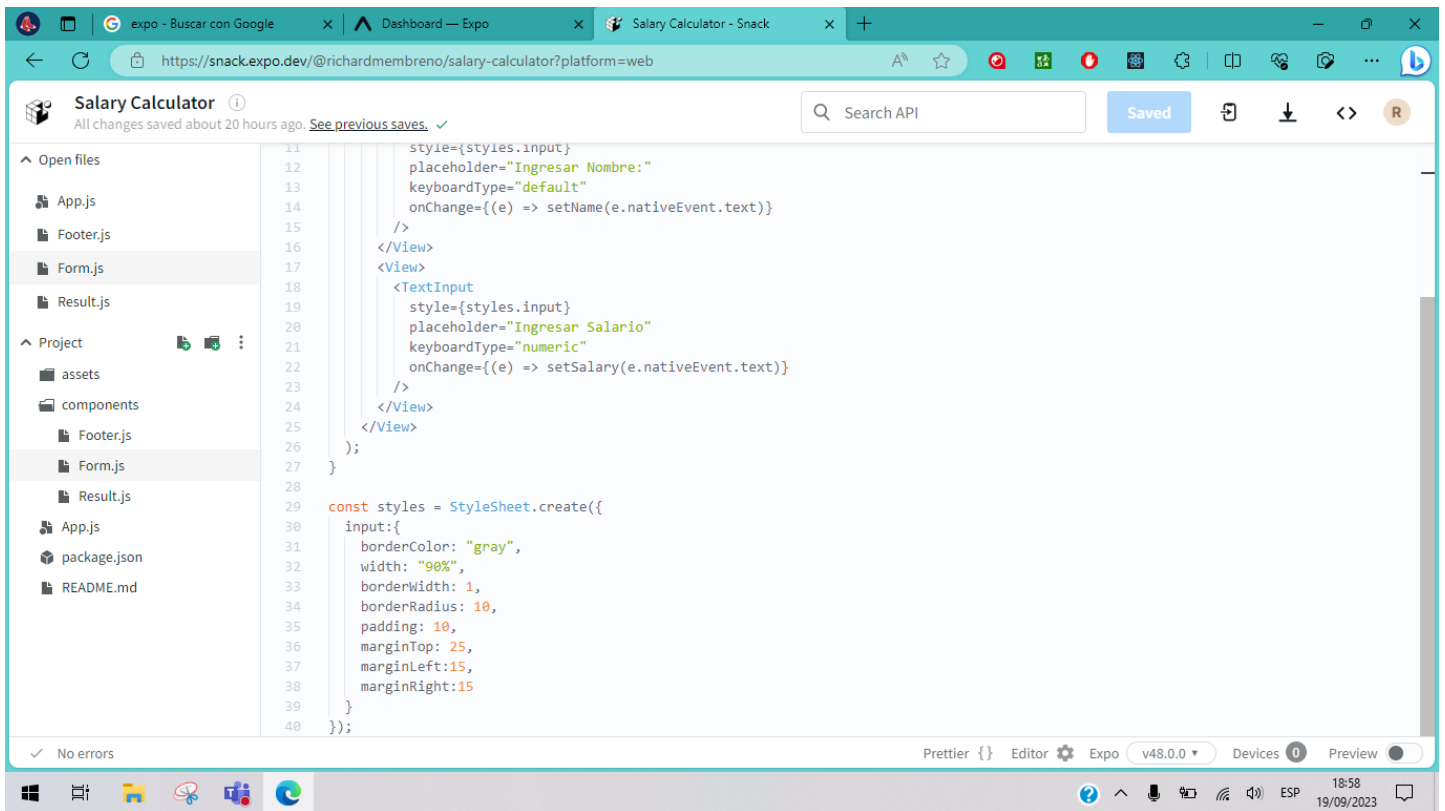
# Archivo Form.js



The screenshot shows the Expo Snack editor interface. The top bar includes the title "Salary Calculator" and a search bar. The left sidebar shows the project structure with files like App.js, Footer.js, Form.js, and Result.js. The main editor area displays the code for Form.js, which is a functional component that takes props and returns a JSX element with two text inputs. The first input is for the name, and the second is for the salary. The code is as follows:

```
1 import React from 'react';
2 import {StyleSheet, TextInput, View} from 'react-native';
3
4 export default function Form(props){
5   const {setName, setSalary} = props;
6
7   return(
8     <View>
9       <View>
10        <TextInput
11          style={styles.input}
12          placeholder="Ingresar Nombre:"
13          keyboardType="default"
14          onChange={(e) => setName(e.nativeEvent.text)}
15        />
16      </View>
17      <View>
18        <TextInput
19          style={styles.input}
20          placeholder="Ingresar Salario"
21          keyboardType="numeric"
22          onChange={(e) => setSalary(e.nativeEvent.text)}
23        />
24      </View>
25    </View>
26  );
27 }
28
29 const styles = StyleSheet.create({
30   input:{
```

The bottom status bar shows "No errors" and the Expo version "v48.0.0".

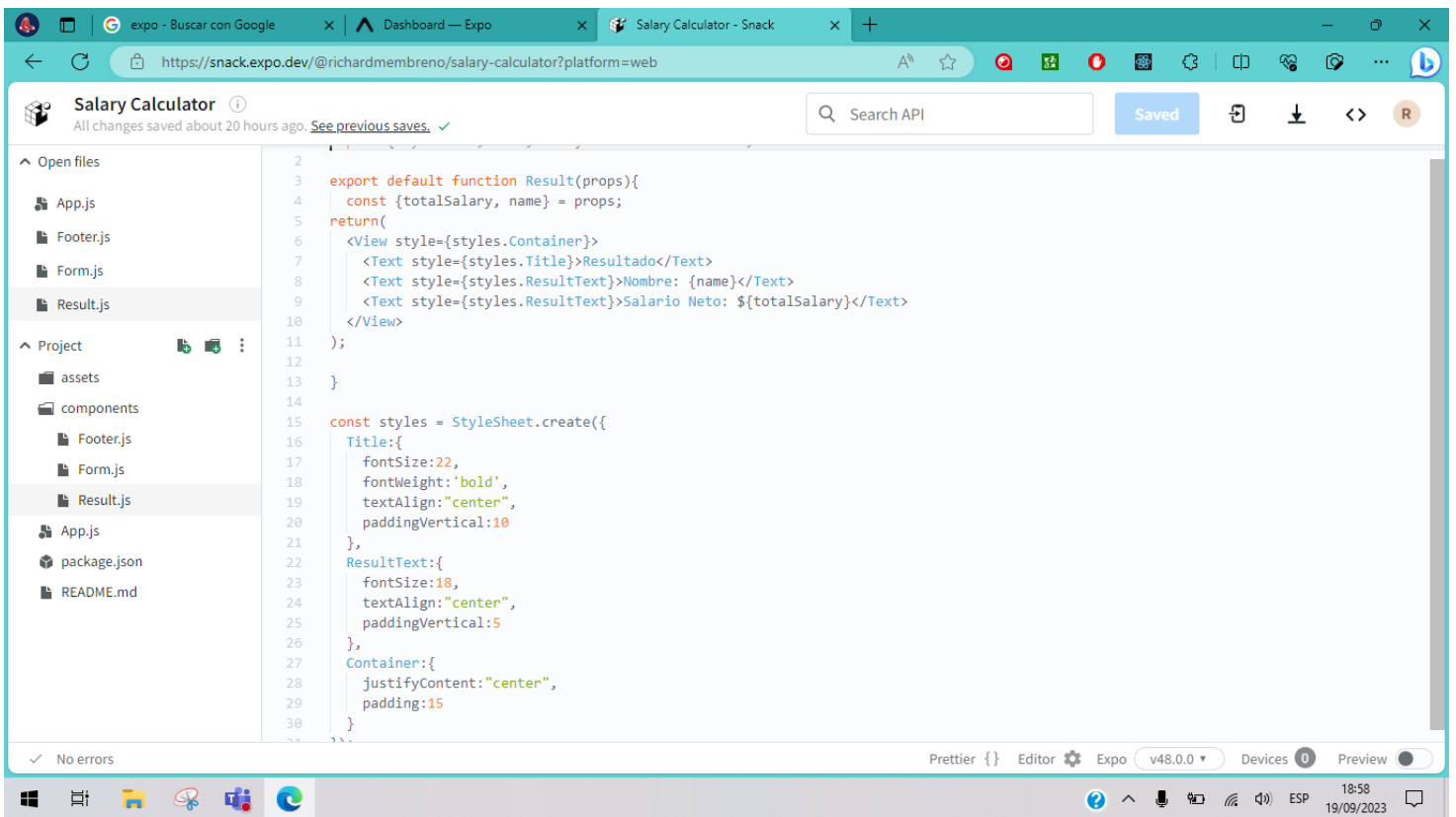


The screenshot shows the Expo Snack editor interface with the same code as the first screenshot, but with additional styles defined for the input field. The code is as follows:

```
11 style={styles.input}
12 placeholder="Ingresar Nombre:"
13 keyboardType="default"
14 onChange={(e) => setName(e.nativeEvent.text)}
15 />
16 </View>
17 <View>
18   <TextInput
19     style={styles.input}
20     placeholder="Ingresar Salario"
21     keyboardType="numeric"
22     onChange={(e) => setSalary(e.nativeEvent.text)}
23   />
24 </View>
25 </View>
26 );
27 }
28
29 const styles = StyleSheet.create({
30   input:{
31     borderColor: "gray",
32     width: "90%",
33     borderWidth: 1,
34     borderRadius: 10,
35     padding: 10,
36     marginTop: 25,
37     marginLeft:15,
38     marginRight:15
39   }
40 });
```

The bottom status bar shows "No errors" and the Expo version "v48.0.0".

## Archivo Result.js



```
2
3 export default function Result(props){
4   const {totalSalary, name} = props;
5   return(
6     <View style={styles.Container}>
7       <Text style={styles.Title}>Resultado</Text>
8       <Text style={styles.ResultText}>Nombre: {name}</Text>
9       <Text style={styles.ResultText}>Salario Neto: ${totalSalary}</Text>
10    </View>
11  );
12
13 }
14
15 const styles = StyleSheet.create({
16   Title:{
17     fontSize:22,
18     fontWeight:'bold',
19     textAlign:"center",
20     paddingVertical:10
21   },
22   ResultText:{
23     fontSize:18,
24     textAlign:"center",
25     paddingVertical:5
26   },
27   Container:{
28     justifyContent:"center",
29     padding:15
30   }
31 })
```

## Funcionamiento



