

---

# UNIVERSIDAD DON BOSCO

---

## Ingeniería en Ciencias de la Computación



Desarrollo de Software Multiplataforma

**Primer Taller Practico**

Catedrático: **Ing. Alexander Sigüenza**

Presentado Por:

**Aguilar Urquilla, Erick Gilberto**

**AU171965**

**Lemus Torres, Diego Alberto**

**LT171997**

Fecha de Presentación: **04/09/2021**

## Ejercicio 03 (Completado 100%)

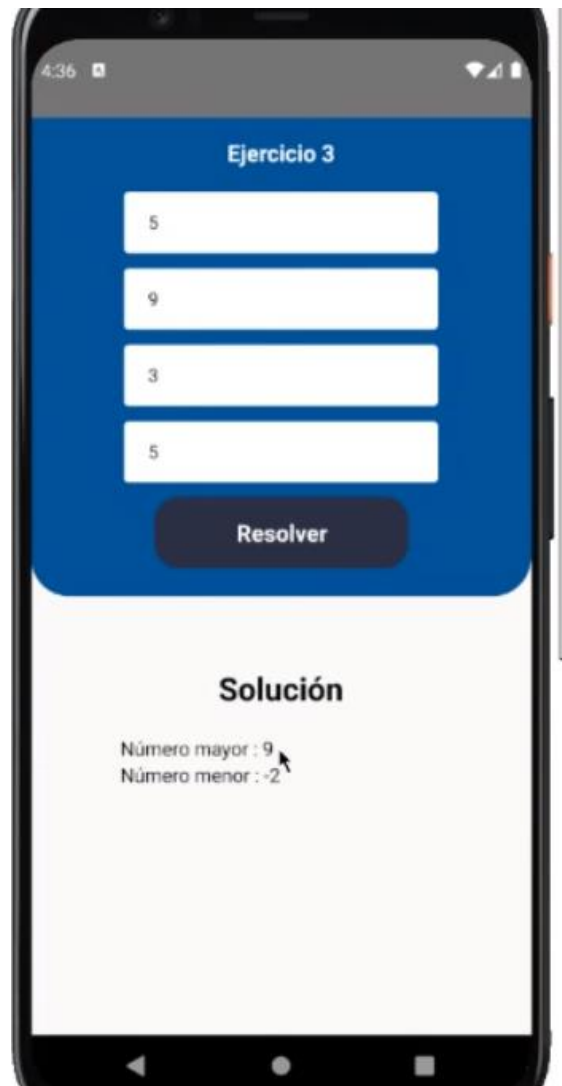
3. (40%) Realizar una aplicación. Solicitar al usuario 4 números enteros y calcular el mayor y menor de ellos.
  1. No se aceptan números negativos ni cero.
  2. El **número mayor** se debe de sumar 10, si el número menor es mayor a 10.
  3. El **número menor** se debe de restar 5, si el número mayor es menor a 50
  4. Las impresiones de los resultados se realizan en pantalla.

A continuación, se muestran las diferentes validaciones que se solicitan en cada literal. El código arrojará números negativos en las respuestas si la resta determinada en esa acción lo realiza. Se utilizarán condicionales de if-elseif para poder mejorar el comportamiento de los arreglos numéricos que se generan al ingresar los datos

```
TallerPractico1DPS > ejercicio3 > JS App.js > [x] App > [x] verify
48  });
49
50  const App = () => {
51    const [num1, setNum1] = useState(0);
52    const [num2, setNum2] = useState(0);
53    const [num3, setNum3] = useState(0);
54    const [num4, setNum4] = useState(0);
55    const [solution, setSolution] = useState(null);
56    const [errorMessage, setErrorMessage] = useState('');
57
58    const verify = () => {
59      reset();
60      if (!num1 || num1 <= 0) {
61        setErrorMessage('El valor del número 1 no es válido');
62      } else if (!num2 || num2 <= 0) {
63        setErrorMessage('El valor del número 2 no es válido');
64      } else if (!num3 || num3 <= 0) {
65        setErrorMessage('El valor del número 3 no es válido');
66      } else if (!num4 || num4 <= 0) {
67        setErrorMessage('El valor del número 4 no es válido');
68      } else {
69        let arrayNums = [num1, num2, num3, num4];
70
71        let maxNum = Math.max(...arrayNums);
72        let minNum = Math.min(...arrayNums);
73
74        let maxNumModified = (minNum > 10 ? maxNum + 10 : maxNum);
75        let minNumModified = (maxNum < 50 ? minNum - 5 : minNum);
76
77        if (isNaN(maxNumModified)) {
78          maxNumModified = 'El número encontrado como máximo no es válido';
79        }
80
81        if (isNaN(minNumModified)) {
82          minNumModified = 'El número encontrado como máximo no es válido';
83        }
84
85        setSolution({
86          maxNum: maxNumModified,
87          minNum: minNumModified
88        });
89      }
90    };
91
92    const reset = () => {
93      setErrorMessage('');
94      setSolution(null);
95    };
96  };
97
98  return (
99    <div>
100      <h3>Ejercicio 3</h3>
101      <div>
102        <input type="text" value={num1} onChange={e => setNum1(e.target.value)} />
103        <input type="text" value={num2} onChange={e => setNum2(e.target.value)} />
104        <input type="text" value={num3} onChange={e => setNum3(e.target.value)} />
105        <input type="text" value={num4} onChange={e => setNum4(e.target.value)} />
106      </div>
107      <div>
108        <button onClick={verify}>Verificar</button>
109        <button onClick={reset}>Resetear</button>
110      </div>
111      <div>
112        <p>Mensaje de error: {errorMessage}</p>
113        <p>Resultado: {solution}</p>
114      </div>
115    </div>
116  );
117}
```



Se muestra que no acepta negativos



Se muestra cómo funciona al de restar 5, si el número mayor es menor a 50

### Ejercicio 3

33

25

14

17

Resolver

### Solución

Número mayor : 43  
Número menor : 9

Se muestra cómo funciona al sumar 10, si el número menor es mayor a 10.

## Ejercicio 2(Completado 100%)

2. **(30%)** Realizar una aplicación. 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.

The screenshot shows a mobile application interface. At the top, the status bar displays the time 4:39 and signal icons. The app title is "Calcular descuentos de salario". Below the title, there are two input fields: "Salario Base" with the value "650" and "Nombre Empleado" with the value "Erick Aguilar". Below these fields, the title "Resumen de Erick Aguilar" is displayed. Underneath, a list of calculations is shown: "Salario base: 650 \$", "Descuento ISSS: 19.5 \$", "Descuento AFP: 26 \$", "Descuento Renta: 32.5 \$", and "Salario Neto: 572 \$". At the bottom of the screen, there is a large blue button labeled "CALCULAR".

Resumen de Erick Aguilar	
Salario base:	650 \$
Descuento ISSS:	19.5 \$
Descuento AFP:	26 \$
Descuento Renta:	32.5 \$
Salario Neto:	572 \$

Se muestra cómo funciona al ingresar un salario de \$650

```

TallerPractico1DPS > ejercicio2 > JS App.js > App
51 const App = () => {
52   const [name, setName] = useState(null);
53   const [salary, setSalary] = useState(null);
54   const [total, setTotal] = useState(null);
55   const [errorMessage, setErrorMessage] = useState('');
56
57   useEffect(() => {
58     if (!name && salary) {
59       reset();
60     }
61   }, [name, salary]);
62
63   const calculate = () => {
64     reset();
65     if (!name) {
66       setErrorMessage('Añade el nombre del empleado');
67     } else if (!salary) {
68       setErrorMessage('Añade el salario base');
69     } else {
70       let isssDiscount = salary * 0.03;
71       let afpDiscount = salary * 0.04;
72       let rentaDiscount = salary * 0.05;
73       let netSalary = salary - (issDiscount + afpDiscount + rentaDiscount);
74
75       setTotal({
76         baseSalary: salary,
77         isssDiscount: isssDiscount,
78         afpDiscount: afpDiscount,
79         rentaDiscount: rentaDiscount,
80         netSalary: netSalary,
81       });
82     }
83   };
84
85   const reset = () => {
86     setErrorMessage('');
87     setTotal(null);
88   };
89 }

```

Se muestra el código utilizado para las validaciones nulas, así como también la declaración y cálculo de las variables a utilizar con respecto al salario ingresado

```

return (
  <>
    <StatusBar barStyle="light-content" />
    <SafeAreaView style={styles.Header}>
      <Text style={styles.HeadApp}>Calcular descuentos de salario</Text>
      <Form setName={setName} setSalary={setSalary} />
    </SafeAreaView>
    <Result
      name={name}
      salary={salary}
      total={total}
      errorMessage={errorMessage}
    />
    <Footer calculate={calculate} />
  </>
);
};

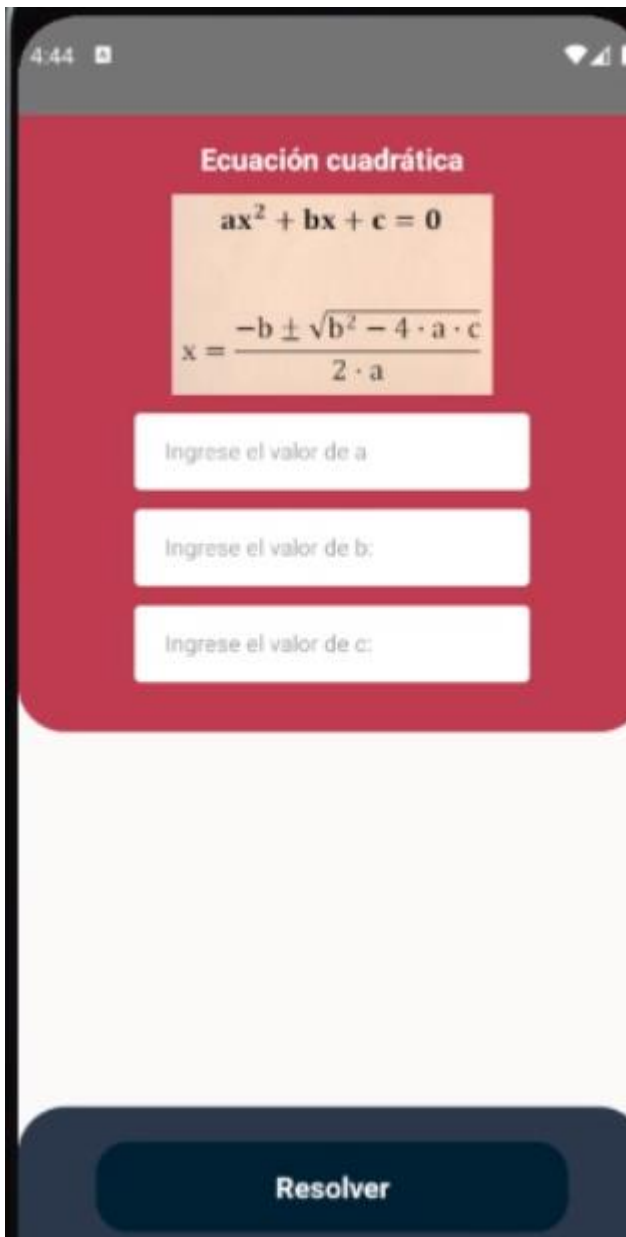
export default App;

```

## Ejercicio 1(Completado 100%)

1. (30%) Realizar una aplicación. Que resuelva la solución de la ecuación cuadrática

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



The screenshot shows a mobile application interface with a red header and a white body. The header contains the title "Ecuación cuadrática" in white text. Below the header, there is a white box containing the quadratic equation  $ax^2 + bx + c = 0$  and the solution formula  $x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$ . Below this box, there are three white input fields with the labels "Ingrese el valor de a", "Ingrese el valor de b:", and "Ingrese el valor de c:". At the bottom of the screen, there is a dark blue button with the text "Resolver" in white.

Vista del diseño del programa

### Ecuación cuadrática

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

1

2

3

### Solución

X1: No se encontro solución

X2: No se encontro solución

Se muestra cómo reacciona la aplicación al intentar obtener raíces imaginarias





Se muestra como funciona la aplicación al obtener raíces reales

```

56
57 const App = () => {
58   const [a, setA] = useState(0);
59   const [b, setB] = useState(0);
60   const [c, setC] = useState(0);
61   const [solution, setSolution] = useState(null);
62   const [errorMessage, setErrorMessage] = useState('');
63
64   const solve = () => {
65     reset();
66     if (!a) {
67       setErrorMessage('Escribe el valor de la letra A');
68     } else if (!b) {
69       setErrorMessage('Escribe el valor de la letra B');
70     } else if (!c) {
71       setErrorMessage('Escribe el valor de la letra C');
72     } else {
73       let x1 = ((-b) + Math.sqrt(Math.pow(b, 2) - 4 * a * c)) / (2 * a);
74       let x2 = ((-b) - Math.sqrt(Math.pow(b, 2) - 4 * a * c)) / (2 * a);
75
76       if (isNaN(x1)) {
77         x1 = 'No se encontro solución';
78       }
79
80       if (isNaN(x2)) {
81         x2 = 'No se encontro solución';
82       }
83
84       setSolution({
85         x1: x1,
86         x2: x2
87       });
88     }
89   };

```

Se muestra la parte principal del código que consiste: en la declaración de las diferentes variables y su procedimiento para obtener las raíces respectivas. Por ultima se muestra la validación ante raíces imaginarias