

# Universidad Don Bosco



**Docente:** Ing. Alexander Alberto Sigüenza Campos

**Materia:** Diseño y programación de software multiplataforma.

## Primer taller práctico

**Presentado por:**

Ruiz Saz, Carlos José RS181977

Pineda Valle, Erick Alberto PV181965

**Ciclo:** 02-2021

Soyapango, sábado 4 de septiembre del 2021

## Contenido

Ejercicio 1 (Porcentaje 100%) .....	3
Ejercicio 2 (Porcentaje 100%) .....	8
Ejercicio 3 (Porcentaje 100%) .....	12

## Ejercicio 1 (Porcentaje 100%)

Desarrollar una aplicación que resuelva la solución de la ecuación cuadrática.

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

Interfaz de la aplicación

Número 1 (A)

Número 2 (B)

Número 3 (C)

**CALCULAR SOLUCIÓN DE  
ECUACIÓN CUADRÁTICA**

La aplicación cuenta con 3 componentes:

- Form: Es la parte superior que contiene 3 TextInput para la entrada de las variables A,B y C.
- Result: Es la parte central de la app en donde se mostrarán las 2 soluciones de la ecuación cuadrática.
- Footer : la parte inferior de la app en la que podemos observar un botón que al presionarlo la app calculará las soluciones.

```

export default function App(){
  const [num1,setNum1]=useState(null);
  const [num2,setNum2]=useState(null);
  const [num3,setNum3]=useState(null);
  const [total,setTotal]=useState(null);
  const [errorMessage,setErrorMessage]=useState("");

  useEffect(() => {
    if (num1 && num2 && num3) calculate();
    else reset();
  }, [num1, num2,num3]);
  const calculate = () => {
    reset();
    if (!num1.trim()) {
      setErrorMessage('Añade el primer valor');
    } else if (!num2.trim()) {
      setErrorMessage('Añade el segundo valor');
    } else if (!num3.trim()) {
      setErrorMessage('Añade el tercer valor');
    } else if ((Math.pow(num2,2)-(4*num1*num3))<0) {
      setErrorMessage('Raíz negativa, intente resolver otra ecuación cuadrática');
    }else {
      const x1=(-num2+Math.sqrt(Math.pow(num2,2)-(4*num1*num3)))/(2*num1);
      const x2=(-num2-Math.sqrt(Math.pow(num2,2)-(4*num1*num3)))/(2*num1);

      setTotal({
        new_x1:x1.toFixed(2).replace('.', ','),
        new_x2:x2.toFixed(2).replace('.', ','),
      });
    }
  }
}

```

En el siguiente fragmento de código podemos observar la declaración de variables en este caso:

Num1=A

Num2=B

Num3=C

Además, podemos observar las validaciones realizadas para evitar campos vacíos, campos llenados con espacios en blanco y una validación especial para este ejercicio que es la de verificar que la raíz en el ejercicio no nos devuelva una raíz negativa, ya que tendríamos que tratar con números imaginarios.

Por último, podemos destacar en la captura las variables x1 y x2, ya que estamos resolviendo una ecuación cuadrática debemos obtener 2 respuestas, una positiva y otra negativa.

A continuación, se muestran captura de las validaciones:



A blue rounded rectangle containing three white rectangular input fields stacked vertically. The top field has a vertical cursor line at the beginning.

**Añade el primer valor**



A blue rounded rectangle containing three white rectangular input fields stacked vertically. The top field contains the number '2'. The middle field has a vertical cursor line at the beginning.

**Añade el segundo valor**



A blue rounded rectangle containing three white rectangular input fields stacked vertically. The top field contains the number '2'. The middle field contains the number '1' followed by a vertical cursor line.

**Añade el tercer valor**

**Raíz negativa, intente resolver otra ecuación cuadrática**

En este último error no es posible resolver el ejercicio que se ingreso debido a que la raíz interna en la fórmula queda negativa y debe tratarse con imaginarios.

Como podemos observar la aplicación cumple con el manejo de errores para que el usuario no se lleve una mala experiencia al momento de utilizarla.

Demostración del funcionamiento:

En seguido se mostrará el correcto funcionamiento de la aplicación, por lo que tomaremos el siguiente ejemplo:

$$x^2 + 4x - 21 = 0$$

$$x = \frac{-4 \pm \sqrt{16 - 4 \cdot (-21)}}{2}$$

$$x = \frac{-4 \pm \sqrt{100}}{2}$$

$$x = \frac{-4 \pm 10}{2}$$

$$x = -2 \pm 5$$

$$x = -2 + 5 \quad x = -2 - 5$$

$$x = 3 \quad x = -7$$

Debemos identificar los datos así:

A=1

B=4

C=-21

Se ingresarán dichos valores a la app y se muestra el resultado a continuación:



A screenshot of a mobile application interface. It features a blue background with three white rectangular input fields stacked vertically. The first field contains the number '1', the second contains '4', and the third contains '-21'.

### Resultados de la ecuación cuadrática

A:	1
B:	4
C:	-21
X1:	3,00
X2:	-7,00

**CALCULAR SOLUCIÓN DE  
ECUACIÓN CUADRÁTICA**

Se observa que se obtuvieron los resultados esperados.

## Ejercicio 2 (Porcentaje 100%)

Desarrollar una aplicación que calcule el salario neto de un empleado, solicitando nombre y salario base.

Tomar en cuenta lo siguiente:

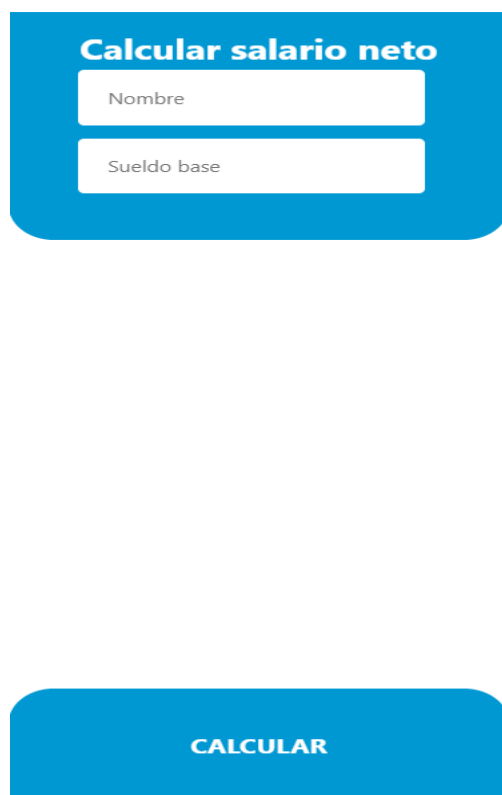
ISSS-3%

AFP-4%

RENTA-5%

Salario neto=salario base-Deducciones.

Interfaz de la aplicación



The image shows a mobile application interface with a blue header and footer. The header contains the title "Calcular salario neto" in white. Below the title are two white input fields with blue borders, labeled "Nombre" and "Sueldo base". The footer is a solid blue bar with the word "CALCULAR" in white capital letters.

La aplicación cuenta con 3 componentes:

- **Form:** Es la parte superior que contiene 2 TextInput para la entrada de las variables nombre del empleado y sueldo base.
- **Result:** Es la parte central de la app en donde se mostrarán el resumen del salario base, deducciones y el salario neto del empleado
- **Footer :** la parte inferior de la app en la que podemos observar un botón que al presionarlo la app calculará el salario neto.



```

export default function App(){
  const [nombre,setNombre]=useState(null);
  const [sueldo,setSueldo]=useState(null);
  const [total,setTotal]=useState(null);
  const [errorMessage,setErrorMessage]=useState("");

  useEffect(() => {
    if (nombre && sueldo) calculate();
    else reset();
  }, [nombre, sueldo]);
  const calculate = () => {
    reset();
    if (!nombre.trim()) {
      setErrorMessage('Añade el nombre ');
    } else if (!sueldo.trim()) {
      setErrorMessage('Añade el sueldo base');
    } else if (sueldo<=0) {
      setErrorMessage('El sueldo debe ser mayor a 0');
    } else {

      const AFP=sueldo*0.04;
      const ISSS=sueldo*0.03;
      const RENTA=sueldo*0.05;
      const sueldoneto=sueldo-AFP-ISSS-RENTA;

      setTotal({
        AFPfee: AFP.toFixed(2).replace('.', ','),
        ISSSfee:ISSS.toFixed(2).replace('.', ','),
        RENTAFee:RENTA.toFixed(2).replace('.', ','),
        SueldoNeto:sueldoneto.toFixed(2).replace('.', ','),
      });
    }
  };
  const reset = () => {
    setErrorMessage('');
    setTotal(null);
  };
}

```

En el siguiente fragmento de código podemos observar la declaración de variables en este caso:

- Nombre del empleado
- Sueldo base del empleado

Además, podemos observar las validaciones realizadas para evitar campos vacíos, campos llenados con espacios en blanco y una validación para que el sueldo base que se ingresé deba ser mayor a 0.

Podemos observar las variables AFP,ISSS,RENTA Y sueldoneto.

En el caso de la variable AFP es el 4% del sueldo base, ISSS es el 3% del sueldo base y RENTA es el 5% del sueldo base, por lo que una vez realizada esas operaciones ya tenemos las deducciones.

Para obtener el salario neto restamos las deducciones calculadas por esas 3 variables (ISSS,AFP Y RENTA) al salario base.

A continuación, se muestran captura de las validaciones:

**Calcular salario neto**

**Añade el nombre**

**Calcular salario neto**

**Añade el sueldo base**

**Calcular salario neto**

**El sueldo debe ser mayor a  
0**

Como podemos observar la aplicación cumple con el manejo de errores para que el usuario no se lleve una mala experiencia al momento de utilizarla.

Demostración del funcionamiento:

En seguido se mostrará el correcto funcionamiento de la aplicación:

### Calcular salario neto

Carlos Saz

1000

### RESUMEN DE SUELDO

Nombre:	Carlos Saz
Sueldo base:	1000 \$
AFP (4%):	40,00 \$
ISSS (3%):	30,00 \$
RENTA (5%):	50,00 \$
Sueldo neto:	880,00 \$

CALCULAR

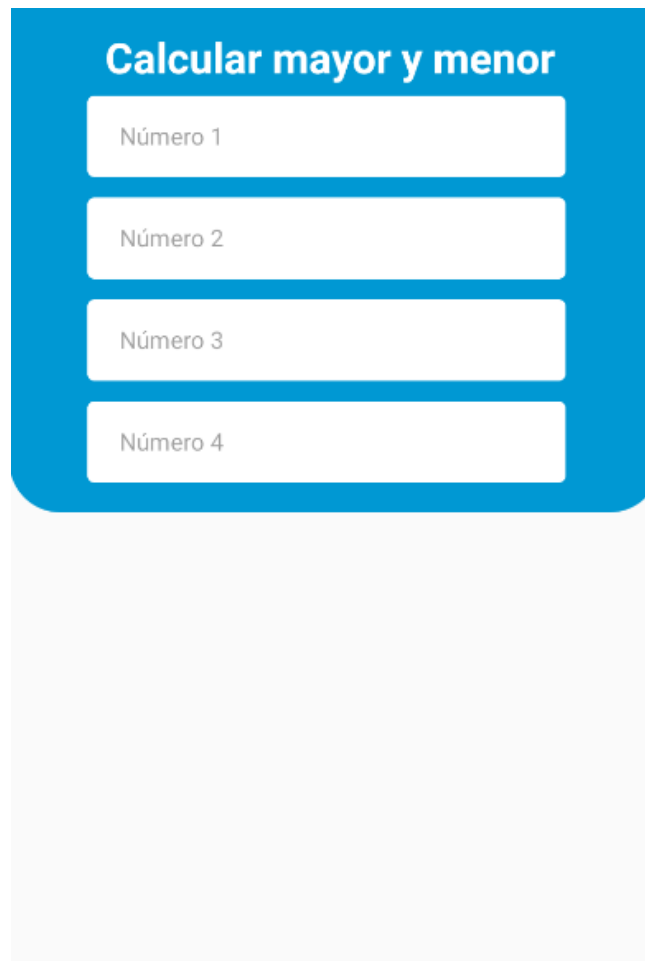
Se observa que se obtuvieron los resultados esperados.

## Ejercicio 3 (Porcentaje 100%)

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.

Interfaz:

La interfaz de usuario se presenta en un contenedor con un fondo azul. En la parte superior, el título "Calcular mayor y menor" está escrito en blanco. Debajo del título, hay cuatro campos de entrada de texto blancos, cada uno con un label gris que dice "Número 1", "Número 2", "Número 3" y "Número 4" respectivamente. Los campos están apilados verticalmente. Debajo de esta sección azul, hay un área gris clara que ocupa la parte inferior de la pantalla, destinada a mostrar los resultados.

Los componentes que contienen el código de funcionamiento son:

- Form: Es la parte superior que contiene 4 TextInput para la entrada de los 4 números.
- Result: Es la parte central de la app en donde se mostrarán el resultado, es decir, el valor máximo y mínimo.

Soyapango, sábado 4 de septiembre del 2021

- Footer : la parte inferior de la app en la que podemos observar un botón que al presionarlo la app calculará el valor máximo y mínimo de los datos ingresados.

```
const [total, setTotal] = useState(null);
const [errorMessage, setErrorMessage] = useState("");

const [numero1, setNumero1] = useState(null);
const [numero2, setNumero2] = useState(null);
const [numero3, setNumero3] = useState(null);
const [numero4, setNumero4] = useState(null);

const [mensajeMax, setMensajeMax] = useState("");
const [mensajeMin, setMensajeMin] = useState("");

useEffect(() => {
  if (numero1 && numero2 && numero3 && numero4) calculate();
  else reset();
}, [numero1, numero2, numero3, numero4]);
```

```
const calculate = () => {
  reset();

  let regg = /^[0-9]+$/;

  if (numero1.trim() <=0 || numero2.trim() <=0 || numero3.trim() <=0 || numero4.trim() <=0) {
    setErrorMessage('No pueden haber números negativos o 0 en los valores que ingresa. ');
  } else if(!regg.test(numero1)){
    setErrorMessage('No puede ingresar números decimales, SOLO ENTEROS. ');
  } else {

    const arregloNumeros =[numero1,numero2,numero3,numero4];

    const maximo = Math.max(...arregloNumeros);
    const minimo = Math.min(...arregloNumeros);

    if(minimo>10){
      const nuevoMax = maximo + 10;
      setMensajeMax('Mínimo mayor a 10, se le suma 10 al máximo, el nuevo maximo es: ' + nuevoMax);
    }
    if(maximo<50){
      const nuevoMin = minimo - 5;
      setMensajeMin('Máximo menor a 50, se le resta 5 al mínimo, el nuevo mínimo es: ' + nuevoMin);
    }

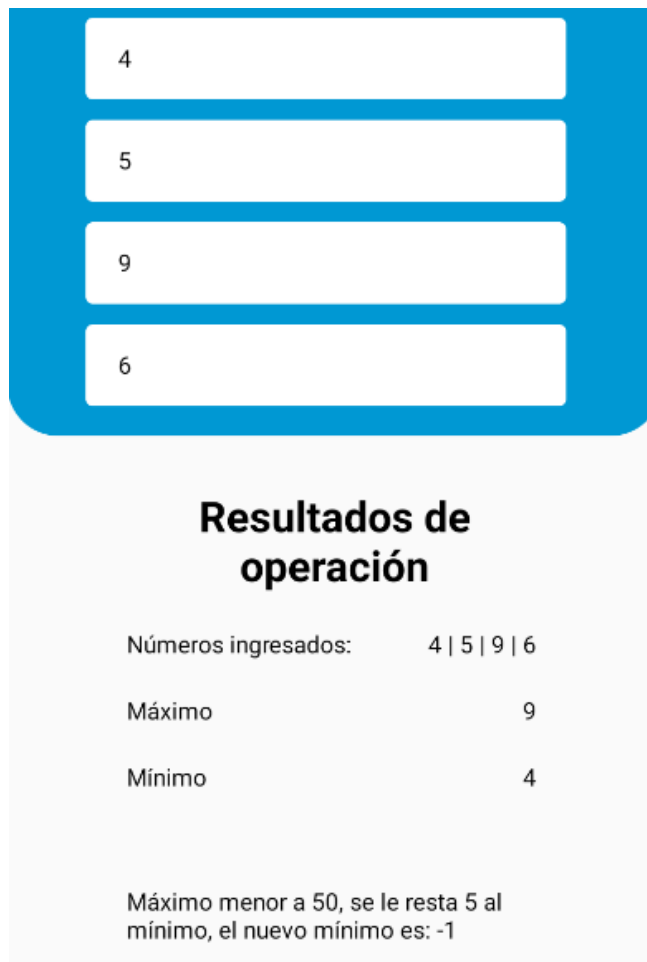
    setTotal({
      maxi:maximo,
      mini:minimo
    });
  }
};

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

En las dos capturas anteriores se puede apreciar parte del código utilizado para resolver el ejercicio. La lógica de dicho código consiste en obtener el valor de los 4 textbox y luego usar una función de JS para obtener el mayor y menor número de los 4 valores.

Además, se realizan validaciones para evitar negativos, cero o valores decimales y estructuras condicionales para cumplir con los requisitos que pide el ejercicio.

Capturas de funcionamiento:



The screenshot shows a web application interface. At the top, there is a blue rounded rectangle containing four white input fields stacked vertically. The first field contains the number '4', the second '5', the third '9', and the fourth '6'. Below this, on a light gray background, is a section titled 'Resultados de operación' in bold black text. Under the title, there is a table with two columns. The first column contains labels: 'Números ingresados:', 'Máximo', and 'Mínimo'. The second column contains the corresponding values: '4 | 5 | 9 | 6', '9', and '4'. At the bottom of the gray section, there is a text block that reads: 'Máximo menor a 50, se le resta 5 al mínimo, el nuevo mínimo es: -1'.

<b>Resultados de operación</b>	
Números ingresados:	4   5   9   6
Máximo	9
Mínimo	4
Máximo menor a 50, se le resta 5 al mínimo, el nuevo mínimo es: -1	

14

51

19

16

## Resultados de operación

Números ingresados: 14 | 51 | 19 | 16

Máximo 51

Mínimo 14

Mínimo mayor a 10, se le suma 10 al  
máximo, el nuevo máximo es: 61

14

45

19

16

### Resultados de operación

Números ingresados: 14 | 45 | 19 | 16

Máximo	45
Mínimo	14

Mínimo mayor a 10, se le suma 10 al máximo, el nuevo máximo es: 55

Máximo menor a 50, se le resta 5 al mínimo, el nuevo mínimo es: 9

Validaciones:



## Calcular mayor y menor

-5

45

19

16

**No pueden haber números negativos o 0 en los valores que ingresa.**

## Calcular mayor y menor

6

45

19.5

16

**No puede ingresar números decimales, SOLO ENTEROS.**