

# Memoria de la Tarea

## Actividad 1: Cálculo del precio total con IVA

### Descripción del problema

Se requiere un programa que permita al usuario ingresar los precios de al menos 5 artículos. La entrada se debe detener cuando el usuario introduzca 0 o deje vacío el campo. El programa debe calcular el IVA (21%) y mostrar el precio total con IVA y sin IVA, redondeado a dos decimales.

### Proceso del desarrollo

Inicialmente, se diseñó un formulario con campos de entrada dinámicos.

Un botón permite agregar nuevos campos para precios, evitando que el usuario esté limitado a un número fijo de artículos.

Usamos la función reduce para calcular la suma de los precios ingresados, y posteriormente, calculamos el total con IVA.

Se añadieron validaciones para asegurarse de que el usuario ingrese al menos 5 valores numéricos antes de realizar los cálculos.

### Funcionamiento

El usuario ingresa los precios en los campos visibles de la página. Al hacer clic en "Calcular", el programa verifica la validez de los datos, calcula el total sin IVA y con IVA, y muestra los resultados en pantalla. Si el usuario no ingresa suficientes valores, aparece un mensaje de error.

## **Actividad 2: Determinar la estación del año según el mes**

### **Descripción del problema**

El programa debe permitir al usuario seleccionar un mes del año y mostrar la estación correspondiente. Los meses se clasifican como:

Invierno: diciembre, enero, febrero.

Primavera: marzo, abril, mayo.

Verano: junio, julio, agosto.

Otoño: septiembre, octubre, noviembre.

### **Proceso del desarrollo**

Se utilizó un elemento select para que el usuario elija un mes de una lista predefinida, reduciendo errores de entrada.

Usamos una estructura switch para asociar cada mes con su estación.

Se añadieron mensajes claros para mostrar los resultados en la página.

### **Funcionamiento**

El usuario selecciona un mes de la lista y presiona el botón "Determinar Estación". El programa evalúa el mes mediante el switch, determina la estación y la muestra en pantalla. Si no se selecciona un mes, aparece un mensaje de error.

## Actividad 3: Suma y Productoria de Números

### Descripción del problema

El usuario ingresa dos números (a y b). Si  $a > b$ , se muestra un error. Si  $a \leq b$ , el programa calcula:

La suma de todos los números entre a y b.

El producto de los números entre a y b, deteniéndose si el producto excede 100 y mostrando el valor parcial.

### Proceso del desarrollo

Se implementaron validaciones para garantizar que  $a \leq b$  y que ambos números sean válidos.

Se usaron bucles for para calcular la suma y la productoria.

Se añadió una condición para interrumpir el cálculo de la productoria si excede 100.

### Funcionamiento

El usuario introduce los números a y b en los campos correspondientes. Al presionar "Calcular", el programa valida las entradas, calcula la suma y la productoria (hasta que se exceda 100) y muestra los resultados en pantalla.

## Actividad 4: Redimensionar pantalla según navegador

### **Descripción del problema**

El programa debe verificar si el navegador del usuario es Google Chrome. Si lo es, redimensiona la ventana del navegador a 500x500. Si no, muestra un mensaje indicando que la operación no está permitida.

### **Proceso del desarrollo**

Se utilizó `navigator.userAgent` para identificar el navegador.

Se agregó una verificación para detectar Google Chrome y excluir navegadores como Microsoft Edge.

La función `window.resizeTo` redimensiona la ventana si se cumple la condición.

### **Funcionamiento**

El usuario presiona el botón "Redimensionar Pantalla". Si el navegador es Chrome, la ventana se ajusta a 500x500. Si no, se muestra un mensaje en pantalla indicando que el navegador no es compatible.

## **Actividad 5: Calculadora con múltiples operaciones**

## **Descripción del problema**

Se requiere un programa que permita realizar operaciones básicas (suma, resta, multiplicación, división y módulo). El usuario debe introducir dos números y una operación. Si se elige un operador no válido o se intenta dividir entre 0, se debe manejar el error.

## **Proceso del desarrollo**

Se creó un objeto calculator con métodos para cada operación básica.

Se añadió un formulario con inputs para los números y un select para el operador.

Se implementaron validaciones para verificar las entradas y manejar divisiones por 0.

## **Funcionamiento**

El usuario introduce los números y selecciona una operación. Al hacer clic en "Calcular", el programa realiza la operación correspondiente y muestra el resultado. Si ocurre un error (como una división entre 0), aparece un mensaje de advertencia.

## **Actividad 6: Suma de números ingresados**

### **Descripción del problema**

El programa debe permitir al usuario ingresar números dinámicamente. La entrada se detiene cuando el usuario no introduce un número válido. Luego, se calcula la suma de todos los números ingresados.

### **Proceso del desarrollo**

Se diseñó un formulario con un campo inicial para ingresar números.

Un botón permite agregar más campos dinámicamente.

La suma se calcula usando reduce y se muestra el resultado en pantalla.

### **Funcionamiento**

El usuario ingresa números en los campos visibles y hace clic en "Calcular Suma". El programa recoge los números válidos, calcula su suma y muestra el resultado en pantalla.

## **Actividad 7: Calculadora Extensible**

### **Descripción del problema**

Se requiere una calculadora que permita realizar operaciones básicas y agregar nuevas operaciones dinámicamente. El usuario introduce una operación en formato número operador número.

### **Proceso del desarrollo**

Se creó un constructor Calculator con un objeto para almacenar operadores.

Se implementó el método addMethod para agregar nuevas operaciones.

Se añadieron inputs para números, operador y funciones de operador.

### **Funcionamiento**

El usuario introduce dos números y un operador para realizar un cálculo.

También puede agregar nuevos operadores mediante un formulario adicional.

Los resultados o errores se muestran en pantalla.

## **Actividad 8: Barajar un array**

### **Descripción del problema**

El programa debe permitir al usuario ingresar números que se almacenarán en un array. Cuando el usuario finalice la entrada, el array se debe barajar aleatoriamente y mostrar el resultado.

### **Proceso del desarrollo**

Se diseñó un formulario para ingresar números y un botón para barajar.

Los números se almacenan en un array dinámico.

Se implementó el algoritmo de Fisher-Yates para barajar el array.

### **Funcionamiento**

El usuario ingresa números en los campos de entrada y hace clic en "Barajar Array". El programa baraja los números y muestra el resultado en pantalla.