

# Ejercicios PSeint.

## Ejercicio 1: Calculadora Simple

**Temática:** Entrada/Salida de datos, operaciones aritméticas, condicionales.

**Descripción:** Este ejercicio crea una calculadora básica que permite al usuario realizar sumas, restas, multiplicaciones o divisiones con dos números. El programa pedirá los dos números y luego le preguntará al usuario qué operación desea realizar.

### Explicación:

1. **Algoritmo CalculadoraSimple:** Define el inicio de nuestro programa.
  2. **Definir num1, num2, resultado Como Real:** Declara tres variables de tipo `Real` (números con decimales) para almacenar los operandos y el resultado.
  3. **Definir operacion Como Caracter:** Declara una variable de tipo `Caracter` para almacenar la letra que representa la operación elegida por el usuario.
  4. **Escribir y Leer:** Estas instrucciones se utilizan para interactuar con el usuario. `Escribir` muestra mensajes en pantalla y `Leer` permite al programa obtener datos introducidos por el usuario desde el teclado.
  5. **Segun operacion Hacer ... FinSegun:** Esta estructura de control (similar a un `switch` en otros lenguajes) evalúa el valor de la variable `operacion`.
    - Para cada caso ("S", "R", "M", "D"), se realiza la operación correspondiente y se muestra el resultado.
    - **Si num2 <> 0 Entonces ... FinSi:** Dentro del caso de la división, se incluye un condicional para evitar divisiones por cero, lo que generaría un error. Si el divisor (`num2`) es distinto de cero, se realiza la división; de lo contrario, se muestra un mensaje de error.
    - **De Otro Modo:** Si el usuario introduce una letra que no corresponde a ninguna operación válida, se ejecuta este bloque y se muestra un mensaje de error.
  6. **FinAlgoritmo:** Indica el final del programa.
-

## Ejercicio 2: Contador de Números Pares e Impares

**Temática:** Bucles (repetitivas), condicionales, contadores.

**Descripción:** Este ejercicio pide al usuario una cantidad de números y luego le solicita que introduzca esa cantidad de números. Al final, el programa cuenta y muestra cuántos de los números introducidos fueron pares y cuántos fueron impares.

---

### Explicación:

1. **Algoritmo ContadorParesImpares:** Inicio del programa.
2. **Definir ... Como Entero:** Se declaran las variables necesarias.
  - o `cantidadNumeros`: Para almacenar cuántos números el usuario quiere procesar.
  - o `i`: Variable de control para el bucle `Para`.
  - o `numero`: Para almacenar cada número introducido por el usuario.
  - o `contadorPares`, `contadorImpares`: Para llevar la cuenta de los números pares e impares, respectivamente.
3. **`contadorPares = 0` y `contadorImpares = 0`:** Es crucial inicializar los contadores a cero antes de usarlos, para que la cuenta comience correctamente.
4. **Escribir y Leer `cantidadNumeros`:** Se le pide al usuario la cantidad de números a introducir.
5. **Para `i = 1` Hasta `cantidadNumeros` Con Paso 1 Hacer ... FinPara:** Este es un bucle `Para` (un `for` en otros lenguajes).
  - o Se repetirá `cantidadNumeros` veces.
  - o En cada iteración, `i` tomará valores desde 1 hasta `cantidadNumeros`.
  - o Dentro del bucle, se pide al usuario que introduzca un `numero`.
6. **Si `numero MOD 2 = 0` Entonces ... Sino ... FinSi:** Este condicional determina si un número es par o impar.
  - o El operador `MOD` (módulo o resto de la división) devuelve el residuo de una división.
  - o Si `numero MOD 2` es igual a 0, significa que el número es divisible por 2, y por lo tanto, es par. En este caso, se incrementa `contadorPares`.
  - o Si no es 0, el número es impar, y se incrementa `contadorImpares`.
7. **Escribir "Cantidad de números pares: ", `contadorPares`:** Al finalizar el bucle, se muestran los resultados finales de los contadores.
8. **FinAlgoritmo:** Final del programa.

## Ejercicio 3: Factorial de un Número

**Temática:** Bucles (repetitivas), funciones matemáticas, validación de entrada.

**Descripción:** Este ejercicio calcula la factorial de un número entero no negativo introducido por el usuario. La factorial de un número  $n$  (denotado como  $n!$ ) es el producto de todos los enteros positivos menores o iguales a  $n$ . ¡Por ejemplo,  $5! = 5 * 4 * 3 * 2 * 1 = 120$ .

### Explicación:

1. **Algoritmo CalcularFactorial:** Inicio del programa.
2. **Definir numero, factorial, i Como Entero:** Se declaran las variables.
  - o **numero:** Para almacenar el número del cual se calculará la factorial.
  - o **factorial:** Para almacenar el resultado de la factorial.
  - o **i:** Variable de control para el bucle **Para**.
3. **factorial = 1:** Se inicializa **factorial** a 1. Esto es importante porque la factorial es un producto, y cualquier multiplicación por 0 daría 0. Además, la factorial de 0 es 1.
4. **Repetir ... Hasta Que numero >= 0:** Este es un bucle **Repetir-Hasta Que** (similar a un **do-while**).
  - o La parte del **Repetir** se ejecuta al menos una vez.
  - o **Si número < 0 Entonces ... FinSi:** Dentro del bucle, se valida que el número introducido no sea negativo. Si lo es, se muestra un mensaje de error y el bucle se repite hasta que el usuario introduzca un número no negativo.
5. **Si número = 0 Entonces ... Sino ... FinSi:** Después de obtener un número válido, se calcula el factorial.
  - o **Caso base:** Si **numero** es 0, el **factorial** se establece en 1, según la definición matemática.
  - o **Para i = 1 Hasta numero Con Paso 1 Hacer ... FinPara:** Si **numero** es mayor que 0, se utiliza un bucle **Para** para calcular el factorial.
    - En cada iteración, **factorial** se multiplica por el valor actual de **i**. Por ejemplo, si **numero** es 5:
      - $i=1: \text{factorial} = 1 * 1 = 1$
      - $i=2: \text{factorial} = 1 * 2 = 2$
      - $i=3: \text{factorial} = 2 * 3 = 6$
      - $i=4: \text{factorial} = 6 * 4 = 24$
      - $i=5: \text{factorial} = 24 * 5 = 120$
6. **Escribir "El factorial de ", numero, " es: ", factorial:** Finalmente, se muestra el resultado al usuario.
7. **FinAlgoritmo:** Final del programa.