

# Proyecto UT1

## Introducción a la programación

# ÍNDICE

- [1. Instrucción:](#)
- [2. Expresión:](#)
- [3. Palabras reservadas:](#)
- [4. Variable:](#)
- [5. Constante:](#)
- [6. Literal:](#)
- [7. Un operador de cada tipo \(Aritméticos, Relacionales, Lógicos, Especiales\)](#)
- [8. Un tipo de dato de: Entero, Real, Cadena.](#)
- [9. Comentario.](#)
- [10. Contador](#)
- [11. Acumulador](#)
- [12. While](#)
- [13. Do While](#)
- [14. For](#)
- [15. Dato de entrada](#)
- [16. Dato de salida](#)
- [17. Cambio de estado](#)
- [18. Aspecto de programación imperativa](#)
- [19. Aspecto de la programación modular](#)
- [20. Diagrama de flujo.](#)
  - [Calculadora](#)
  - [Función acumulador](#)
  - [Función MOD](#)

## 1. Instrucción:

```
Escribir "Selecciona el tipo de operacion: "
```

Realiza una acción.

## 2. Expresión:

```
acumulador + numero
```

Es una combinación de valores, variables, operadores y funciones que se evalúa para producir un resultado.

## 3. Palabras reservadas:

```
FinFuncion
```

Es una palabra reservada porque indica el final de la función

## 4. Variable:

```
resultado
```

La variable muestra el resultado dependiendo de lo que de la operación.

## 5. Constante:

```
acumulador = 0
```

Muestra que acumulador = 0

## 6. Literal:

```
Escribir "Introduce el primer número:"
```

Representa un texto que nunca va a cambiar.

## 7. Un operador de cada tipo (Aritméticos, Relacionales, Lógicos, Especiales)

- Aritmético:

```
num1 / num2
```

- Relacionales:

```
Mientras contar ≥ numerocontar
```

- Lógicos:

```
Mientras numero ≠ 0 Hacer
```

- Especiales:

```
num1 MOD num2
```

## 8. Un tipo de dato de: Entero, Real, Cadena.

- Real:

```
Definir num1 Como Real
```

- Entero:

```
Definir n, num1, num2, resultado Como Entero
```

- Cadena:

```
Definir operacion Como Cadena
```

## 9. Comentario.

```
// Sumar el número al acumulador
```

## 10. Contador

```
Para i Desde 1 Hasta n Hacer
```

Muestra que cuente desde i hasta el número introducido.

## 11. Acumulador

```
acumulador = acumulador + numero
```

Acumula el número introducido y lo va sumando.

## 12. While

```
Mientras numero ≠ 0 Hacer
```

Siempre se va a repetir hasta que sea 0.

## 13. Do While

```
Repetir
```

```
  Escribir "Selecciona el tipo de operacion: "  
  Escribir "Introduzca 1 para la suma"  
  Escribir "Introduzca 2 para la resta"  
  Escribir "Introduzca 3 para la division"  
  Escribir "Introduzca 4 para la multiplicación"  
  Escribir "Introduzca 5 para el acumulador"  
  Escribir "Introduzca 6 para el contador"  
  Escribir "Introduzca 7 para calcular el MOD"  
  Escribir "Introduzca 10 para salir"
```

```
  Leer operacion
```

```
Hasta Que operacion = "10"
```

Todo esto se repite hasta que se introduzca 10 por teclado.

## 14. For

```
Para i Desde 1 Hasta n Hacer
```

Es un bucle que no para hasta que llegue al número introducido.

## 15. Dato de entrada

```
Leer operacion
```

Lee el dato introducido por teclado.

## 16. Dato de salida

```
Escribir "El resultado es: ", resultado
```

Muestra por pantalla el resultado.

## 17. Cambio de estado

```
acumulador = acumulador + numero
```

Cambia de estado porque el número introducido se suma al que ya estaba.

## 18. Aspecto de programación imperativa

La programación imperativa es un paradigma que se centra en describir cómo un programa debe realizar tareas mediante una serie de instrucciones secuenciales que funcionan paso a paso.

## 19. Aspecto de la programación modular

La programación modular es un enfoque que divide un programa en partes más pequeñas y manejables, conocidas como módulos o funciones.

```
Funcion esAcumulador
  Definir numero, acumulador Como Real
  acumulador = 0 // Inicializar el acumulador

  Escribir "Bienvenido al acumulador"

  Escribir "Introduce un número (o 0 para salir):"
  Leer numero

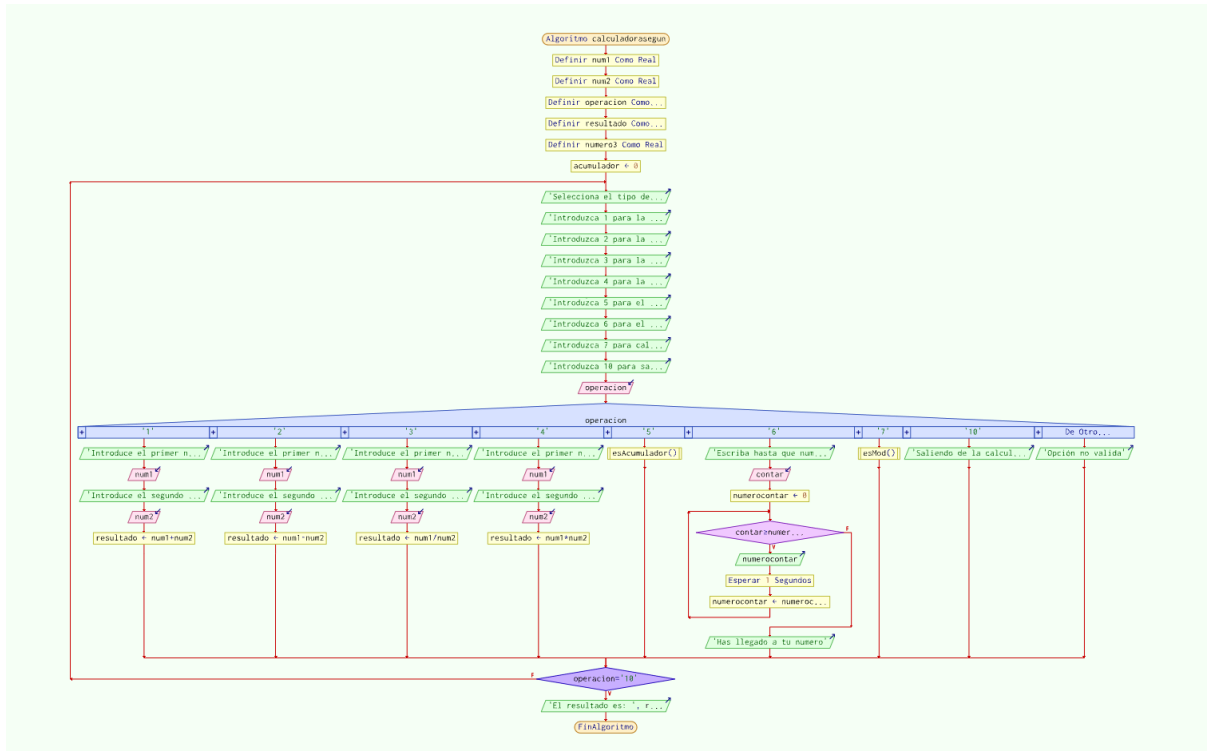
  Mientras numero ≠ 0 Hacer
    acumulador = acumulador + numero // Sumar el número al acumulador
    Escribir "Acumulador actual: ", acumulador

    Escribir "Introduce otro número (o 0 para salir):"
    Leer numero
  Fin Mientras

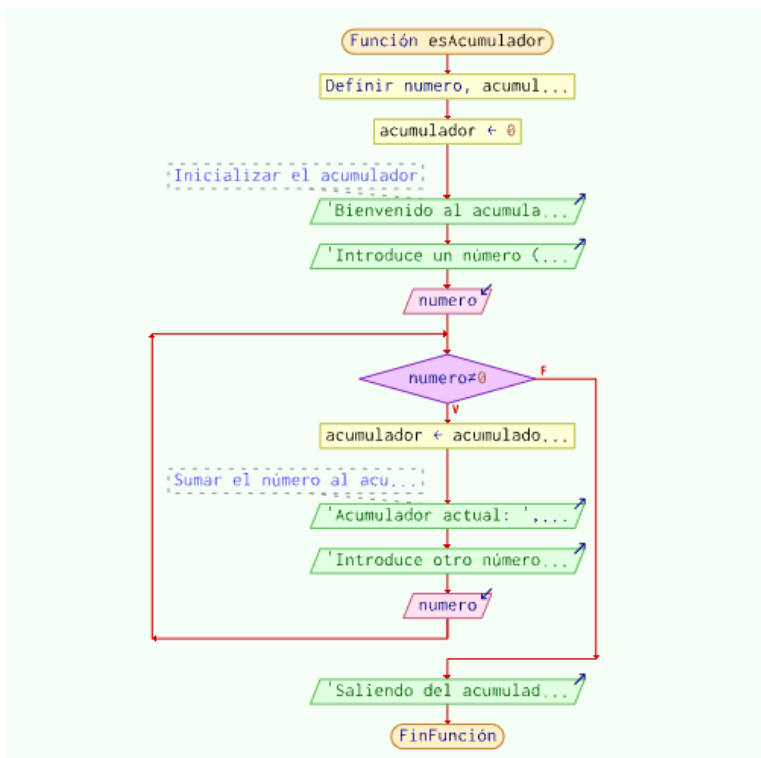
  Escribir "Saliendo del acumulador."
FinFuncion
```

## 20. Diagrama de flujo.

- Calculadora



- Función acumulador



- Función MOD

