

# Lenguaje de programación PLC's

## Lenguaje de programación

### Objeto y campo de aplicación

A Nivel de los lenguajes de programación de uso mas común, las reglas sintácticas y semánticas, el juego de instrucciones fundamentales, las enzayas y la medida de adaptación y adaptación de los equipos. Es el interface entre el programador y el sistema de control.

Definiciones: Tiempo absoluto, via de acceso, acción, argumento, matriz, asignación, bloque funcional, variable, cadena de bits, cuerpo, llamada, cadena de caracteres, comentario, compilar, tipo de datos, declaración, de compilador, doble palabra, plano, ascendente/descendente, función, diagrama de bloques funcionales, dirección, valor, parametro de salida, sentido de corriente, unidad de organización de programa, revisión, tarea, retorno, etapa, secuencia, transición, representación simbólica, etc...

## El estandar IEC-61131

### Elementos comunes

- A- Tipos de datos y variables
- A- Modelo de software
- A- Modelo de comunicación de datos
- A- Unidades de organización del programa
- A- Gráfico funcional secuencia (SFC)
- A- Elementos de configuración

## Lenguajes de programación

- Lista de instrucciones (IL)
- Texto estructurado (ST)
- Diagrama de bloques funcionales (FBD)
- Diagrama de contactos (LD)



## Elementos comunes. Tipos de datos y variables

### \* Tipos de datos:

booleanas, enteros, reales, byte, palabra, cadenas de caracteres, fecha, hora, del tipo, canal analógico de entrada, tipos de datos derivados (creados por el usuario), tipos de datos derivados: BOOL, INT, REAL, BYTE, WORD, STRING, DATE, TIME, ...

### \* Variables

Asignan direcciones del Hardware: E/S, memoria y datos. Locales o globales. Independiente del hardware. Hacen la programación de elementos de los lenguajes de programación de un A.P.

### \* Tipos de datos derivados

### \* Unidades de organización de programa, POU's

Funciones

Bloques

Programas

- Elementos del diagrama secuencial (SFC)

- Elementos globales de configuración

Variables

Recursos

Tareas

Vías de acceso

Bloques funcionales de comunicación

Programa de aplicación

Elementos comunes

- Tipos de datos y variables

- Modelo de software de comunicación, de datos

- Modelo de programación

- Unidades de organización del programa (SFC)

- Grafico funcional secuencial lenguaje de programación (IL)

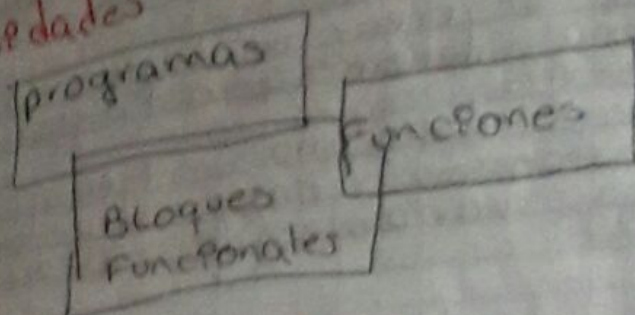
- Lista de instrucciones (Ladder)

- Texto estructurado (ST)

- Diagrama de bloques funcionales (FBD)



- Programa de contactos (LD)  
 Unidades de organización del programa (POU's)



### POU's - Funciones

• Una función se define como una unidad de organización del programa que al ser ejecutada suministra exactamente un elemento de datos y cuya invocación se puede utilizar en lenguajes lógicos como operando en una expresión.

Ejemplo:  $X := \sin(Y) + \cos(Z)$

• Las funciones no deben contener ninguna información de estado interno, es decir, que la invocación de una función con los mismos argumentos (parámetros de entrada) debe suministrar siempre el mismo valor (salida).

• Funciones estándar: ADD, SQRT, SIN, COS, MIN, MAX, AND, OR, etc.

• Funciones definidas por el usuario (ejemplo):

```

FUNCTION simple: REAL
  VAR INPUT
    A, B: REAL;
    C: REAL := 1.0;
  END VAR
  simple := A * B / C;
END FUNCTION
  
```

Nombre

Declaración de variables

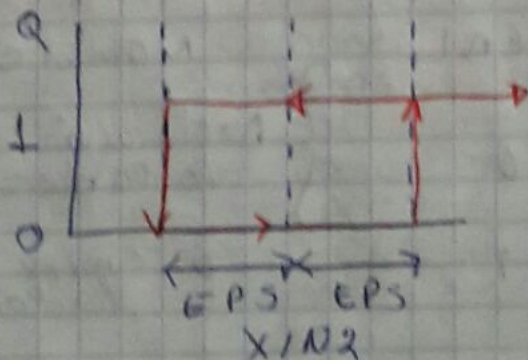
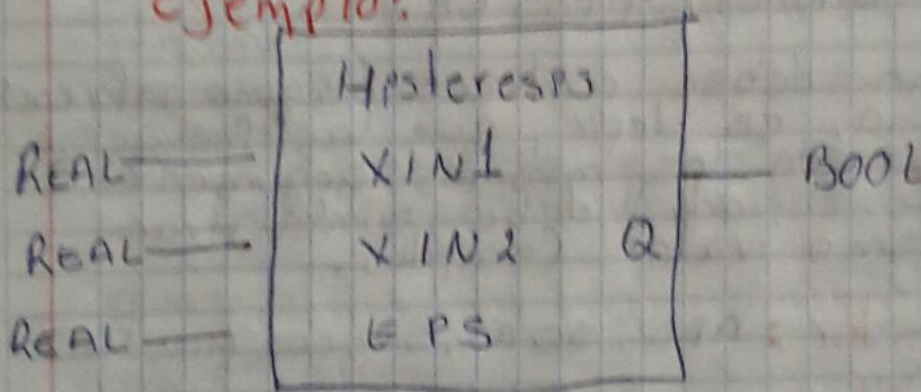
Cuerpo de la función



## POO's . Bloques funcionales

- Un bloque funcional es una unidad de organización del programa que al ser ejecutada suministra uno o más valores.
- Existe la posibilidad de crear múltiples copias de un bloque funcional, denominadas instancias.
- Cada instancia llevará asociado un identificador (el nombre de la instancia) y una estructura de datos que contenga variables de salida e internas.
- Todos los valores de las variables de salida e internas, deberán persistir de una ejecución del bloque funcional al siguiente, por lo que lo llamado de un mismo bloque funcional con los mismos argumentos (parámetros de entrada) no tienen por qué dar siempre los mismos valores de salida.

### Ejemplo:



```

FUNCTION BLOCK Histeresis
  VAR_INPUT
    XIN1, XIN2: REAL
  EPS: REAL; (*Banda Histe*)
  END_VAR
  VAR_OUTPUT
    Q: BOOL := 0
  END_VAR
  IF Q THEN
    IF XIN1 < (XIN2 - EPS) THEN
      Q := 0 (*XIN decreciendo*)
    ELSEIF XIN1 >
      (XIN2 + EPS) THEN
      Q := 1;
      (*XIN creciendo*)
    END_IF
  END_FUNCTION_BLOCK
  
```

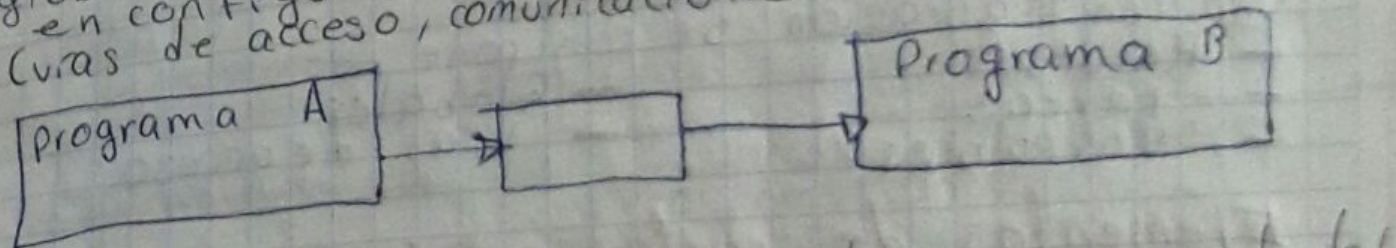


# Unidades de organización del programa POU's

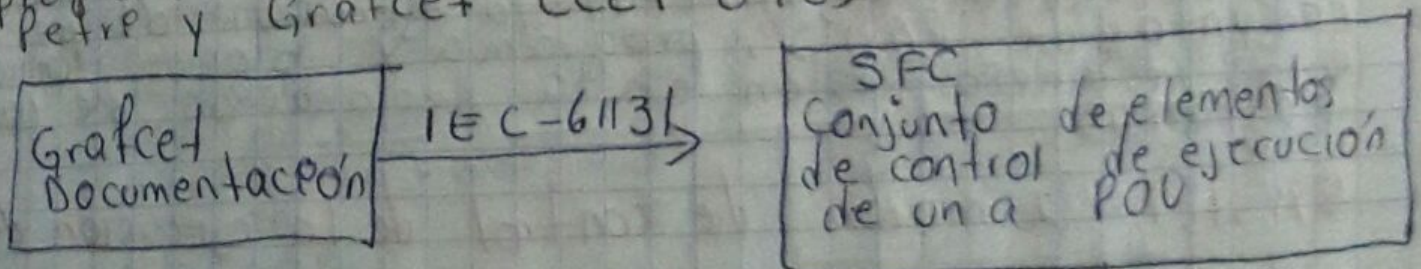
- Los bloques funcionales
- Los programas
- Los programas son un conjunto lógico de los elementos y construcciones del lenguaje de programación que son necesarios para el tratamiento de una tarea que se requiere para el control de un sistema de automatización o proceso mediante un programa.

## POU's. Programas

- Flujo de datos dentro de un programa
- Comunicación entre programas:
  - en la misma configuración (variables globales)
  - en configuraciones distintas (vías de acceso, comunicaciones)



- SFC describe el comportamiento secuencial del programa de control. Deriva de las Redes de Petri y Grafcet (CEI 848)

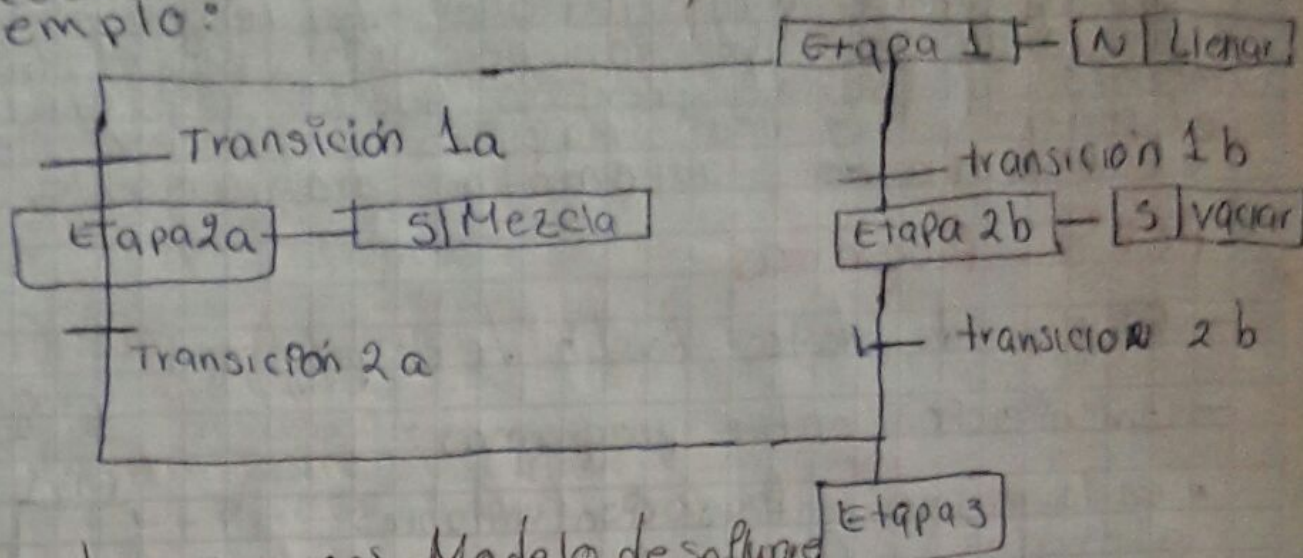


- Permite dividir el problema en partes
- Facilita el rápido diagnóstico del problema y las tareas de mantenimiento.
- Secuencial Function Chart, SFC
- SFC consiste en un conjunto de etapas (STEP's) en las cuales se realizan una serie de



acciones (ACTION BLOCKS) y unas transiciones (TRANSITIONS) que permiten la evolución de una etapa (S) a la (S) siguiente (S).

Sequential Function chart, SFC permite secuencias alternativas y simultáneas.  
Ejemplo:



Elementos comunes. Modelo de software  
Elementos de configuración.

Una configuración se compone de:

- Recursos
  - Globales
- tareas
- variables
  - vías de acceso

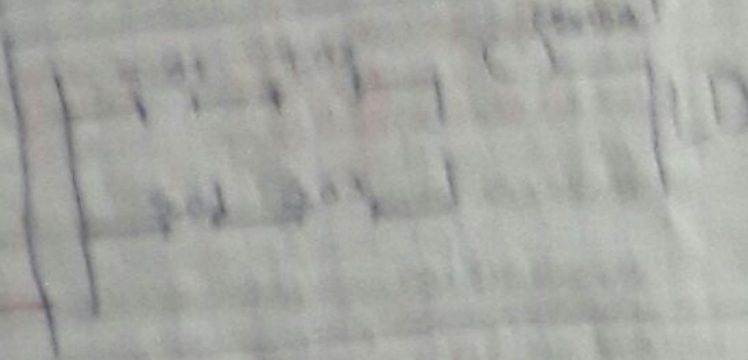
## Elementos de configuración. Tareas

Tarea es un elemento de control de ejecución que es capaz de iniciar la ejecución de un conjunto de POU's, programas y bloques funcionales cuyas instancias están en la declaración de los programas.

Existen 2 modos de control de la ejecución de tareas.

- Por evento: ejecución por flanco ascendente de la entrada SINGLE
  - Por tiempo: ejecución periódica según la entrada INTERVAL
- PRIORITY establece la prioridad de las tareas (0 - máxima)

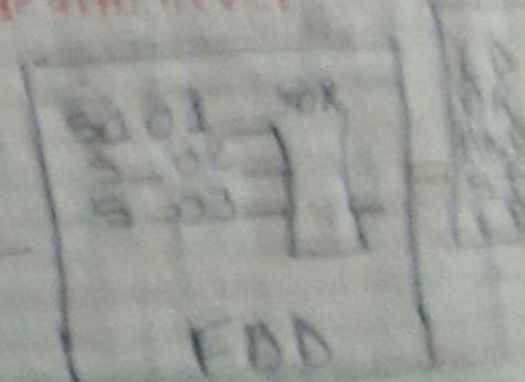




```

IF DATA = EOF THEN A
FOR index = 1 TO 121 DO
  Y = Read_data
  Load(Fid(index))
IF X > 500 THEN A = A + 100
END IF
END FOR
END IF

```





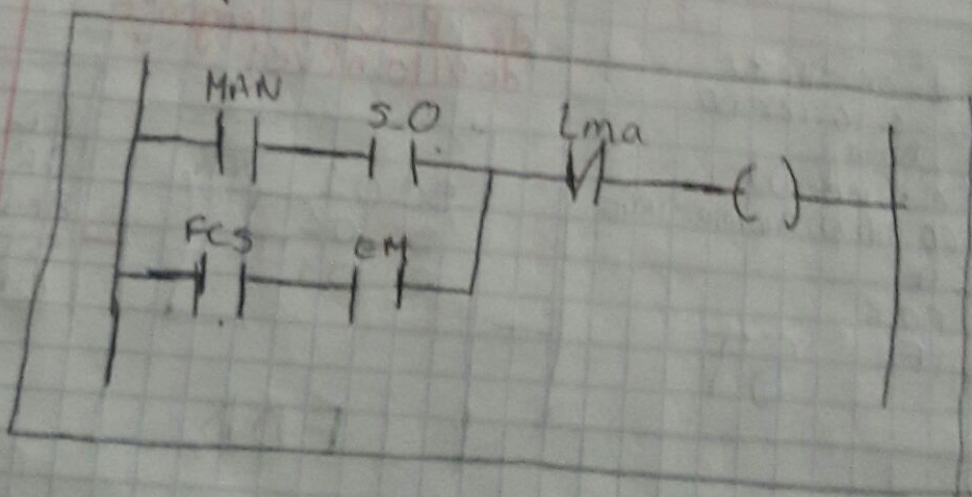
## Lenguajes Gráficos (Ladder y FBD)

- \* Los elementos del lenguaje se representan mediante símbolos o semigráficos
- \* Una red es un conjunto de elementos gráficos interconectados, que debe ser identificada (con número, p.e. debe ser)
- \* Reglas de evaluación de redes, vías de implementación
- \* Posibilidad de saltos y retornos (condicionales e incondicionales)

## Diagrama de Escalera (Ladder Diagram) LD

- Barra de alimentación
- Elementos de enlace y estados = flujo de energía
- Contactos, bobinas y bloques funcionales
- Orden de evaluación de las redes

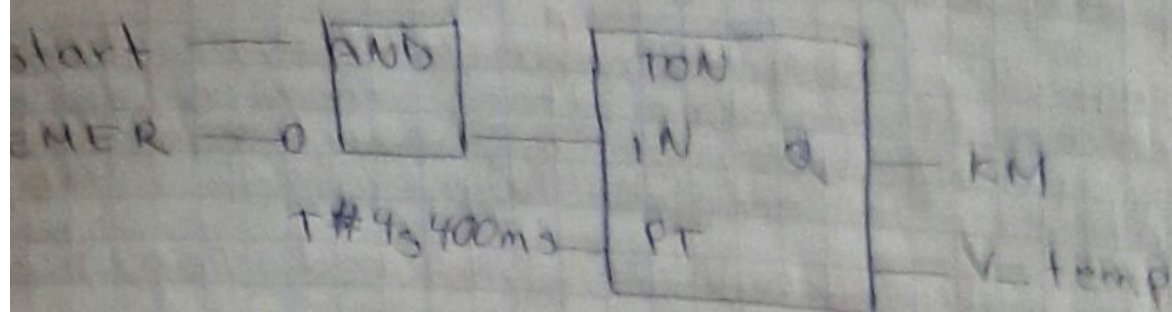
Ejemplo:





## Programa de bloques funcionales (FBD)

Las salidas de los bloques funcionales no se declaran entre si (se presiona el bloque ON)  
 La representación es coherente con la forma CIL-617  
 La evaluación de una red estará terminada antes  
 de la siguiente.



## Diagramas literales (IL y ST)

La representación de literales  
 Los siguientes elementos de estructuración de los POU's

```

... END _ TYPE
... END _ VAR
R _ INPUT ... END _ VAR
R _ OUTPUT ... END _ VAR
R _ IN _ OUT ... END _ VAR
R _ EXTERNAL ... END _ VAR
FUNCTION ... END _ FUNCTION
FUNCTION _ BLOCK ... END _ FUNCTION BLOCK
PROGRAM ... END _ PROGRAM
STEP ... END _ STEP
TRANSITION ... END _ TRANSITION
ACTION ... END _ ACTION
    
```

## Lista de instrucciones ("instruction list")

Características



## INSTRUCCIONES:

Etiqueta	Operador	Operando	Comentarios
START:	LD ANDN ST	Arranque % MXS SALIDA	(*) Pulsar botón (*) NO inhibidor (*) conecta calculador

## Operadores:

LD, ST, S, R, AND, OR, XOR, ADD, SUB, MUL, DIV, GT, GE, EQ, NE, LT, JMP, CAL, RET.

## Funciones y bloques

Se coloca el nombre en el campo del operando y los parámetros, si los lleva, en entre parentesis. Se emplea la operación CAL.

## funcionales:

## Texto estructurado (Structured Text) características

### semejante al pascal

Alto nivel, estructurado en bloques

## Expresiones

Es una construcción sintáctica que al ser evaluada proporciona un valor. Esta compuesta por operadores y operandos. La evaluación se efectúa según una tabla de prioridades.

## Sentencias:

De asignación (C<sub>1</sub> = C<sub>2</sub>)  
De control del flujo del programa:

llamada a bloque funcional, RETURN

De selección: IF, CASE

De iteración: FOR, WHILE, REPEAT, EXIT



Texto estructurado ("structured text"), ST

Ejemplo:

```

IF Dato = "EOF" THEN 128 DA
FOR indice := 1 TO Ccanal [indice]:
  X := Read_dato
  IF X > 2500 THEN Alarma := TRUE;
END IF;
END FOR;
END IF;

```

entornos de programación Muchos de ellos ofrecen:

- Pantallas de programación gráfica
- Soporte para múltiples ventanas
- Operaciones con el ratón
- Menús desplegables
- Ayuda en línea
- Verificación del software durante el diseño.

Características claves de la calidad IEC 6113-3

- \* Software Estructurado — a través de configuración, Recursos y Unidades de Organización de Programa (POUs).
- \* Tipado de Datos — a través de lenguajes que restringen las operaciones a los tipos de datos adecuados.
- \* Control de la ejecución — a través del uso de Tareas.
- \* Comportamiento Secuencial Complejo — a través de los Esquemas Funcional Secuencial.
- \* Encapsulación del Software — a través del uso de Pous, estructuras y tipos de datos complejos.

Beneficios del estándar para los usuarios

- Reduce el gasto en Recursos humanos: formación, depuración, mantenimiento, ...



- Evita fuentes habituales de problemas por su flexibilidad y reusabilidad.
- Técnicas de programación adecuadas a todos los sectores.
- Combinación de elementos y modos de representación.
- Incrementa la conectividad y comunicación entre departamentos y compañías.

### Programación de PLC según IEC 61131-3:WBT

Los controles lógicos programables desempeñan un papel central en el ámbito de la automatización. Estos aparatos se utilizan para controlar máquinas e instalaciones.

El programa de un control lógico programable se puede adaptar flexiblemente a todas las tareas. Para elaborar un programa de mando conforme con la normativa, existen diferentes lenguajes de pasos secuenciales y de programación todos ellos basados en la norma internacional IEC 61131.

Este programa didáctico ofrece la posibilidad de adentrarse en cinco lenguajes de programación: diagramas de funciones, diagramas de contactos, listas de instrucciones, lenguajes de pasos secuenciales y textos estructurados.

Mediante numerosos ejemplos prácticos, los lenguajes de programación son presentados paso por paso.

### Contenido:

- Controles lógicos programables
- Organización de proyectos
- Lenguajes de programación según IEC 61131
- Lenguaje de programación orientado a bloques
- Lenguaje de pasos secuenciales



- texto estructurada
- Programación del desarrollo de un proyecto

Para los principiantes, el programa de ladder es una manera ideal de prepararse en la programación conforme con la norma IEC.

Además de a aprendices, alumnos y estudiantes está dirigido a trabajadores especializados técnicos e ingenieros que, ahora solamente han programado en AWL, KOP o FUP. Los lenguajes superiores conformes con la norma IEC ofrecen numerosas ventajas que es importante conocer y utilizar.

## Controles lógicos programables

- ▲ Uso de controles lógicos programables (PLC)
- ▲ Ventajas de un PLC con respecto a un control lógico cableado (VPS)
- ▲ Estructura básica de un PLC
- ▲ Funcionamiento de un PLC
- ▲ Pormenores y aspectos de fondo relativos a la norma IEC 61131
- ▲ Conocimiento del entorno de programación CoDeSys

## Organización de proyectos

- † Conocimiento de la estación "Distribución"
- † Configuración del hardware
- † Los módulos como elementos centrales de un programa de mando
- † Tipos de módulos
- † Tareas y programa principal
- † Variables como identificadores simbólicos
- † Tipos de datos
- † Declaración de variables, tomando como ejemplo la estación "distribución"

## Lenguajes de programación según IEC 61131

- Funciones básicas
- Lenguajes de programación orientados a enlaces



- Lenguaje de módulos funcionales
- Programa de contactos
- Lista de instrucciones
- Texto estructurado
- Lenguaje de pasos secuenciales

## Lenguajes de programación orientados a enlaces

La función de almacenamiento implementada en

- El programa de contactos
- El lenguaje de módulos funcionales
- La lista de instrucciones.

## Lenguaje de pasos secuenciales

- Elaboración de un plan GRAFCET
- Implementación del mando en el lenguaje de pasos secuenciales tomando como base el plan GRAFCET

## Texto estructurado

- \* Instrucción de selección IF ... THEN ... ELSE
- \* Programación de un módulo funcional
- \* Integración del módulo funcional en un programa existente.

## Programación del desarrollo de un proyecto

- Realización de un ciclo completo de la estación "Distribución" con declaración de variables y la implementación de programa automático.
- Acoplamiento de programa de reset
- Comunicación entre diferentes programas por medio de variables
- Funcionamiento y uso del módulo funcional principal
- Organización de varios programas con una configuración de tareas sencilla