

UT01. INTRODUCCIÓN A LA PROGRAMACIÓN. CONCEPTOS GENERALES.

Índice

1.1. Fases del diseño y puesta a punto de un programa

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones

1.5. Concepto y definición de algoritmo

1.5.1. Diseño e implementación de algoritmos

1.5.2. Estructura general de un programa

1.5.3. Programación estructurada. Estructuras básicas

1.5.4. Variables auxiliares de un programa

1.6. Pseudocódigo

1.7. Pselnt

Fases en la resolución de un problema

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

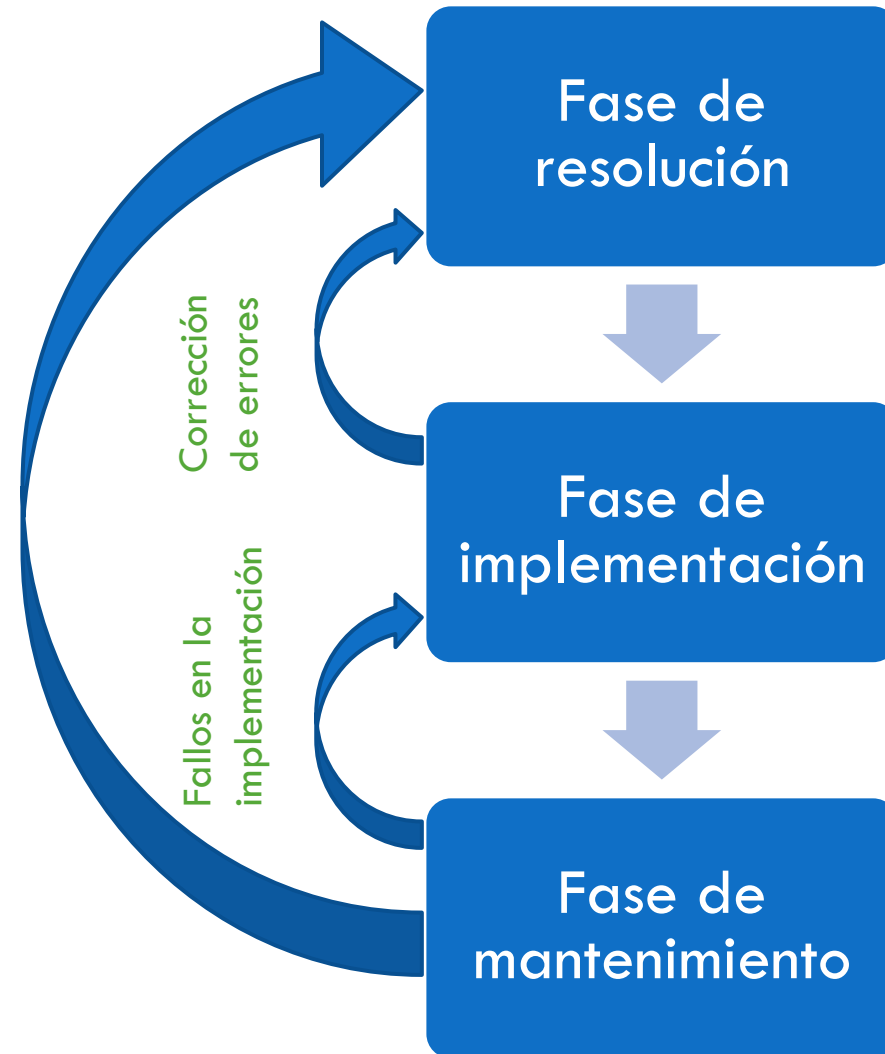
1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PselInt.

Reajustes tras la detección de nuevas necesidades



Fase de resolución

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. Pselnt.

Análisis del problema

1ª aproximación. Estudio del problema.
Definición más exacta posible del mismo.

Resultado:
Conjunto de especificaciones de E y S



Diseño del algoritmo

Representación de la solución del problema

Diseño descendente (top-down)



Verificación manual del algoritmo

Conjunto de datos que incluyan todos los casos posibles

Fallo: volver a cualquiera de las fases anteriores.

Fase de implementación

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7.- PseInt.

- ☐ Solución correcta al problema.
- ☐ Traducir a un lenguaje de implementación concreto.
- Codificación.**
- ☐ Corregir errores escritos.
- ☐ Probar hasta verificar que funciona.
- ☐ **Traducción a lenguaje máquina.**



Fase implementación. Traducción

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

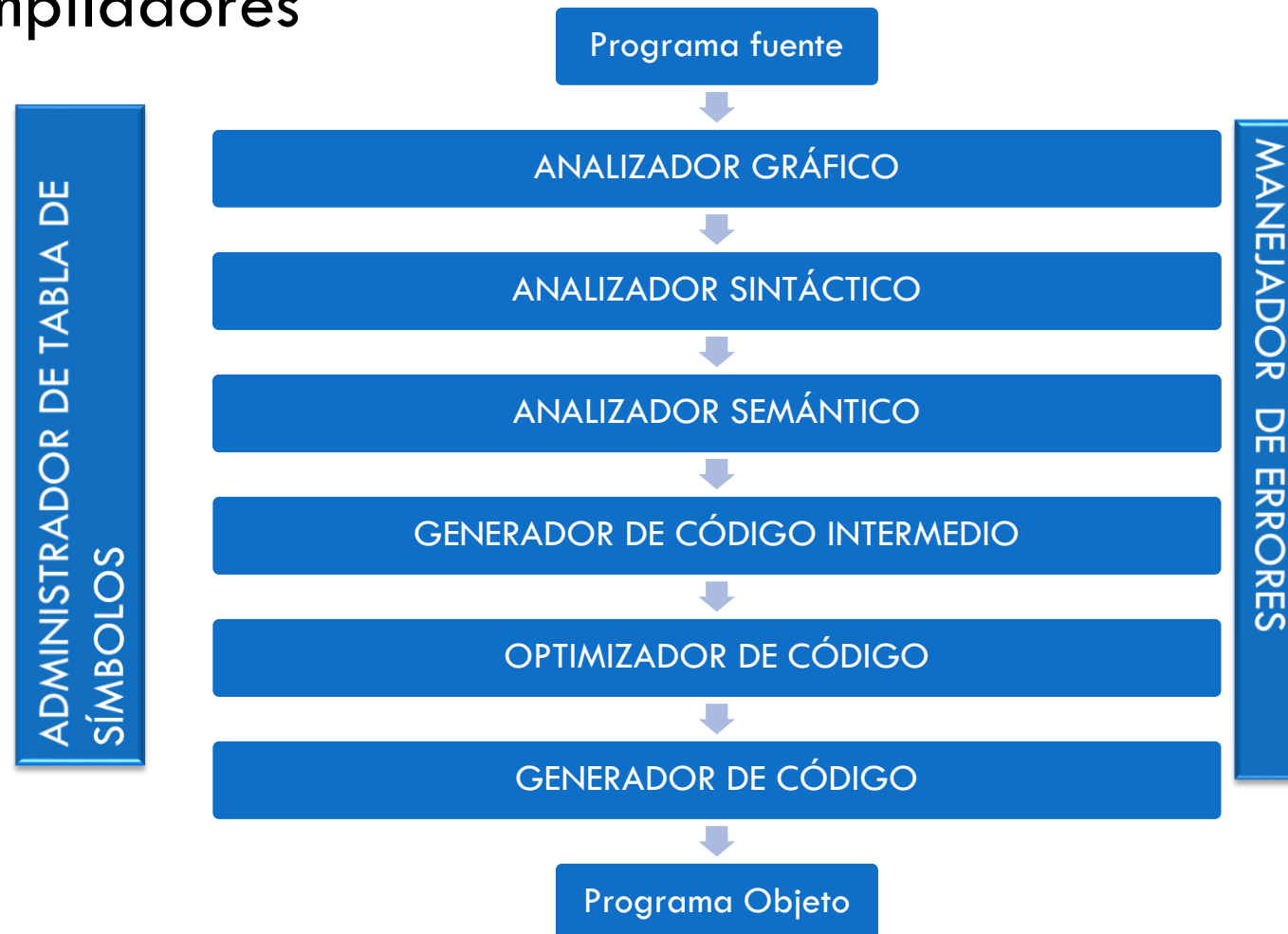
1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PselInt.

□ Compiladores



Fase implementación. Traducción

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

- Dato de entrada: archivo de texto escrito en lenguaje de alto nivel.
- Obtiene listado de errores (depuración de errores).
- Traduce a código máquina (código objeto) ejecutable.
- Suele incluir falsos errores.

Fase implementación. Traducción

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PselInt.

□ Intérpretes.

- ▣ Lee una a una las instrucciones, analiza y traduce a código máquina (si no hay error)
- ▣ Volver a ejecutar \Rightarrow volver a traducir.
- ▣ Ejecución más lenta.
- ▣ T. ejecución menor (mensajes de error más precisos y más localizados).

Fase implementación. Depuración

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

□ Depuración.

- ▣ De errores para su ejecución

□ Pruebas.

- ▣ Ejecución del programa.

- ▣ Pruebas de todos los caminos y todos los grupos de datos.

- ▣ Errores => vuelta a etapas anteriores.

- ▣ Probar por personal distinto del desarrollador.

Fase de explotación y mantenimiento.

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

- Uso de la documentación generada en las fases anteriores.
- Explotación => puesta en marcha.
- Mantenimiento => corregir errores, introducir mejoras, adaptaciones...

Concepto de dato

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

- Todo aquello que puede ser almacenado de forma independiente.
- Conjunto de símbolos que representan valores, hechos, objetos o ideas de forma adecuada para ser tratados.
- Tipos básicos de datos.
- Estructuras de datos.

Constantes y variables

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

□ Constantes y variables.

- Zonas de la memoria del ordenador a las que se asigna un nombre (**identificador**), un tipo de datos, de modo que sólo se podrá almacenar en esa zona de memoria (o posición de memoria) datos de ese tipo.

□ Diferencia:

- **Constantes.** Reciben el valor una vez y no pueden modificarse.
- **Variables.** Pueden cambiar el valor tantas veces como se quiera.

Tipos de datos

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. Pselnt.

- Simples o básicos vs. Estructurados.
- Rango de valores + operadores asociados.
- Implica un tamaño asignado en memoria.
- Tipos simples.
 - ▣ Numérico entero.
 - ▣ Numérico real.
 - ▣ Carácter (ASCII, UNICODE)
 - ▣ Booleano.

- Estructurados.
 - ▣ Cadenas de caracteres.
 - ▣ Vectores y matrices.
 - ▣ Registros.
 - ▣ Archivos o ficheros.
 - ▣ Punteros.
 - ▣ Listas.
 - ▣ Árboles.
 - ▣ Grafos.
- Definidos por el usuario.
 - ▣ Enumerados.
 - ▣ Subrango
 - ▣ Objetos
- Bases de datos.

Expresiones

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7- PseInt.

- Es la representación de un cálculo necesario para la obtención de un resultado.
- Combinación de:
 - ▣ Valor: 1.25, “Juan”
 - ▣ Constante o variable: Pl, num, suma
 - ▣ Función: sqrt(25)
 - ▣ Combinación de todo lo anterior con operadores que cumplen unas determinadas reglas de construcción: $\text{suma} = \text{num} + \text{sqrt}(25)$

Expresiones

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PselInt.

□ Expresiones aritméticas:

□ Suma	+
□ Resta	-
□ Producto	*
□ Potencia	^
□ División entera	\ o div
□ División real	/
□ Módulo	mod o %

Expresiones

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. Pselnt.

□ Relacionales.

- | | |
|------------------------|----------------------|
| □ Igual
asignación) | = (diferencia con la |
| □ Menor | < |
| □ Menor o igual | <= |
| □ Mayor | > |
| □ Mayor o igual | >= |
| □ Distinto | != o <> |

Expresiones

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PselInt.

□ Lógicos o booleanos.

▣ Y o AND

▣ O u OR

▣ NO o NOT

Tablas de verdad de los operadores lógicos.

A	no A
V	F
F	V

A	B	A AND B
F	F	F
F	V	F
V	F	F
V	V	V

A	B	A OR B
F	F	F
F	V	V
V	F	V
V	V	V

Expresiones

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5.- Concepto y definición de algoritmo.

1.5.1.- Diseño e implementación de algoritmos.

1.5.2.- Estructura general de un programa.

1.5.3.- Programación estructurada. Estructuras básicas.

1.5.4.- Variables auxiliares de un programa.

1.6.- Pseudocódigo.

1.7.- PseInt.

□ Evaluación de las expresiones.

- ▣ De izquierda a derecha. Excepciones en determinados lenguajes.

- ▣ Orden de evaluación

- Paréntesis

- Signo

- Potencias

- Productos y divisiones

- Sumas y restas

- Concatenación

- Relacionales

- Negación-NOT

- Conjunción-AND

- Disyunción-OR

Ejercicios

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4.- Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. Pselnt.

Dadas las siguientes variables y constantes:

$X=1$, $Y=4$, $Z=10$, $PI=3.14$, $E=2.71$

evaluar las expresiones:

a) $2 * X + 0.5 * Y - 1 / 5 * Z$

b) $PI * X^2 > Y \text{ OR } 2 * PI * X \leq Z$

c) $E^{(X - 1) / (X * Z) / (X / Z)}$

d) $\text{"DON"} + \text{"JUAN"} = \text{"DON JUAN"} \text{ OR } \text{"A"} = \text{"a"}$

e) $((3 + 2)^2 - 15) / 2 * 5$

f) $5 - 2 > 4 \text{ AND NOT } 0.5 = 1 / 2$

Ejercicios

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. Pselnt.

Construir las expresiones correctas para:

a) $ax^2 + bx + c \geq 0$

b) $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$

c) $\frac{2a}{X}$

d) $\frac{A - B}{X} + \frac{C - D}{Y} > \frac{Y}{X}$

e) A es igual que B, pero no es igual a C.

f) DN, MN, AN (día, mes y año de nacimiento de una persona) y DA, MA, AA (día, mes y año actual). Escribir una expresión para conocer si una persona es o no mayor de edad.

Concepto y definición de algoritmo.

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

□ Algoritmo.

- ▣ Proviene del matemático persa (S. IX) Mohammed Al-Khwarizmi.
→ latín algorismus → algoritmo.

□ Definición:

- ▣ Conjunto de acciones o secuencia de operaciones que ejecutadas en un determinado orden resuelven un problema.
- ▣ Método para resolver un problema mediante una serie finita de pasos precisos y bien definidos.

Concepto y definición de algoritmo

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PselInt.

□ Características:

- ▣ Preciso, sin ambigüedades, pasos ordenados.
- ▣ Bien definido.
- ▣ Finito.
- ▣ Independiente.
- ▣ 1 problema → varios algoritmos → seleccionar el más eficiente.

Diseño y representación de algoritmos.

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

Diagramas de flujo

Símbolos de soporte de información

Teclado



Pantalla



Impresora



Tarjeta perforada



Cinta de papel



Disco magnético



Cinta magnética



Símbolos de proceso

Intercalación



Clasificación



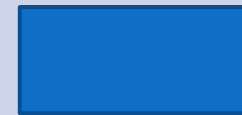
Fusión o mezcla



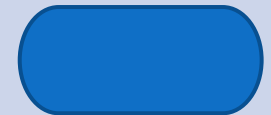
Partición



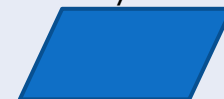
Proceso



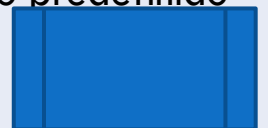
Terminador



Operación E/S



Proceso predefinido



Diseño y representación de algoritmos

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

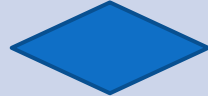
1.6. Pseudocódigo.

1.7. PseInt.

Diagramas de flujo

Símbolos de decisión

Decisión



Bucle

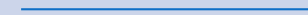


Líneas de flujo

Flechas



Línea de conexión



Símbolos de conexión

Conector



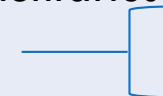
Conector misma pág.



Conector distinta pág.



Comentarios



Diseño y representación de algoritmos.

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

□ Pseudocódigo

- ▣ Lenguaje cercano al lenguaje natural.
- ▣ Lenguaje de especificación de algoritmos.
- ▣ Paso de pseudocódigo a lenguaje de programación sencillo.
- ▣ Sin normas
- ▣ No ambiguo y conciso.

□ Diagramas N-S (Nassi-Shneiderman)

- ▣ Pequeños algoritmos.
- ▣ Sistemas de cajas.
- ▣ Difícil la modificación.

Estructura general de un programa

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

□ Programa.

- ▣ Secuencia lógica de acciones (instrucciones) que manipulan un conjunto de objetos (datos) para obtener unos resultados que serán la solución al problema que resuelve dicho programa.

□ Partes principales.

- ▣ Entrada de datos
- ▣ Proceso o algoritmo
- ▣ Salida de resultados

Estructura general de un programa

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PselInt.

□ Clasificación de las instrucciones.

■ De declaración

■ Primitivas

- de entrada.

- De asignación

- De salida

■ De control.

- Alternativas

- Repetitivas

- Ruptura de secuencia.

■ Compuestas.

Estructura general de un programa

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

- De declaración
 - Se indica el identificador y su tipo de datos
- De entrada.
 - Toma uno o varios datos de un dispositivo de entrada y los almacena en memoria asignándoles un identificador.
- De asignación.
 - Evalúa una expresión y almacena su resultado en un objeto.
- De salida.
 - Envía datos a un dispositivo externo.

Estructura general de un programa

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

□ Pseudocódigo

Programa <nombre_programa>

<Parte declarativa>

Constantes

cte_1=valor_1;

.....

Variables

id_1: tipo_1;

.....

<Parte ejecutable>

Inicio

sentencia 1;

.....

sentencia n;

Fin

□ Ordinograma

- Todo diagrama debe tener un elemento de inicio y otro de fin
- Todos los símbolos deben estar unidos por líneas de flujo.
- Debemos evitar el cruce de las líneas. Para eso utilizamos el elemento conector
- **Proceso:** pueden llegarle varias líneas de flujo pero solo puede salir una de él.
- **Inicio:** no puede llegarle ninguna línea de flujo.
- **Fin:** no puede salir ninguna línea de flujo pero si le pueden llegar varias.

Estructura de la programación estructurada

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

- Secuenciales
- Alternativas
- Cíclicas o repetitivas

Estructuras de la programación estructurada

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

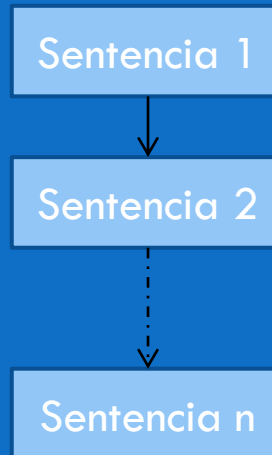
1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PselInt.

Secuenciales

Sentencia 1;
Sentencia 2;
.....;
Sentencia n;



Ejercicios

1. Programa que visualice la suma de dos números introducidos a través de teclado.
2. Programa que visualice el producto de dos números introducidos a través de teclado.
3. Programa que visualice el cubo de un número introducido a través de teclado.
4. Dados dos números introducidos a través de teclado, visualizar la suma, resta, producto y división de dichos números.
5. Visualizar la longitud y el área de un círculo dada la longitud del radio introducido a través de teclado.
6. Visualizar el precio total de un pedido, dado el precio por unidad, el número de unidades y el tanto por ciento de descuento.

Estructuras de la programación estructurada

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

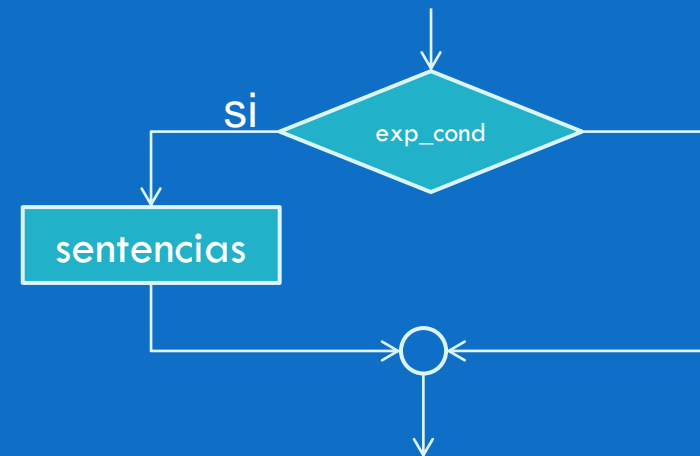
1.6. Pseudocódigo.

1.7. PselInt.

□ Alternativa

▣ Simple

```
si <expresión_condicional> entonces  
    sentencias;  
finSi;
```



Estructuras de la programación estructurada

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

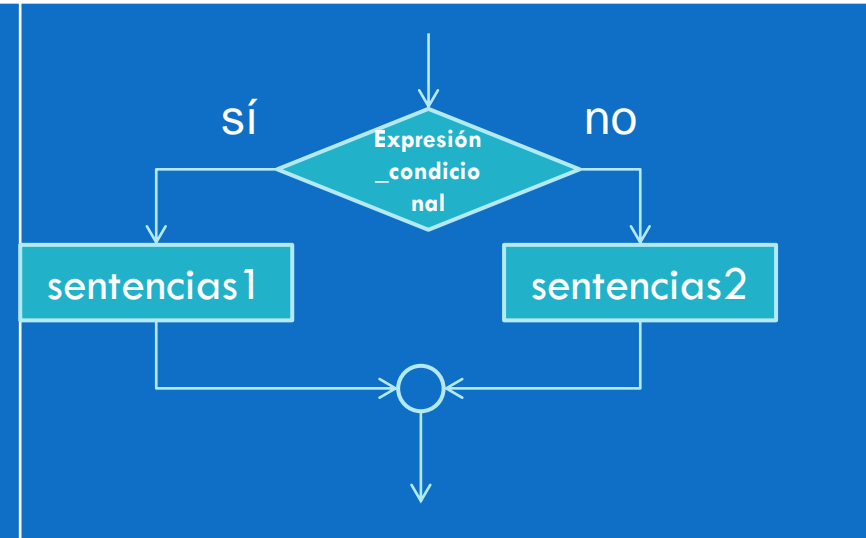
1.6. Pseudocódigo.

1.7. Pselnt.

□ Alternativa

▣ Doble

```
si <expresión_condicional> entonces  
    sentencias1;  
si no  
    sentencias2;  
finSi;
```



Ejercicios

7. Modificar el ejercicio 4, para evitar el problema de la división.
8. Dado un número visualizar si es un cero.
9. Dado un número, visualizar si es positivo o negativo.
10. Dado un número, visualizar si par o impar.
11. Introduce el nombre y el sexo (H o M) de una persona. Visualiza el nombre únicamente en el caso de que se trate de una mujer.

Estructuras de la programación estructurada

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. Pselnt.

□ Alternativa

▣ Múltiple

Según <variable_numérica> hacer

<opción1>:

<setencias1>;

<opción2>:

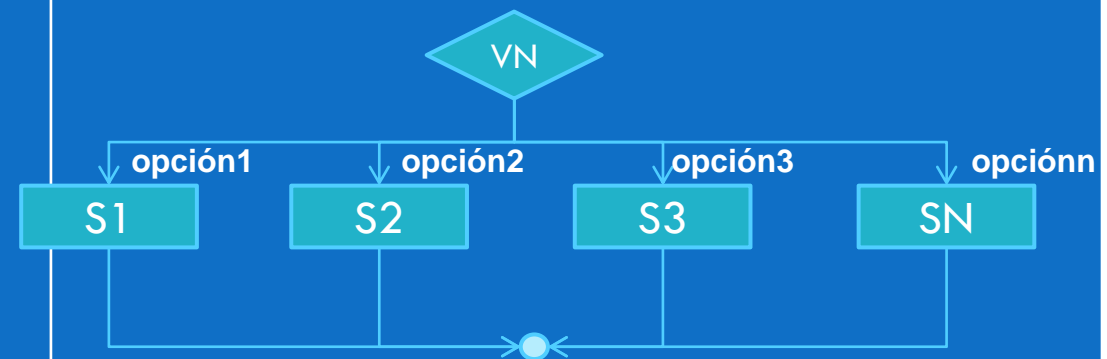
<setencias2>;

.....

de otro modo:

<setenciasn>;

finSegun



Estructuras de la programación estructurada

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5.- oncepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PselInt.

□ Alternativa.

■ Anidamiento de estructuras alternativas.

- Cada elemento “si no” en una estructura alternativa se corresponde con la estructura “si” inmediatamente superior.
- Supone tener completamente una estructura dentro de otra.

Ejercicios

12. Introducir dos número a través de teclado. Visualizar cuál es el mayor y cuál es el menor.
13. Introducir una nota numérica y visualiza su calificación correspondiente alfanumérica, teniendo en cuenta que no existe el 0.
14. Resolver una ecuación de 2º grado $ax^2+bx+c=0$. Los coeficientes a,b,c se introducirán por teclado y los resultados de la ecuación se visualizarán por pantalla:
 - Dos resultados: x1, x2
 - Un resultado: x
 - Soluciones no reales.
15. Introduce por teclado el nombre y la provincia de un gallego. Visualizar su gentilicio.

Estructuras de la programación estructurada

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

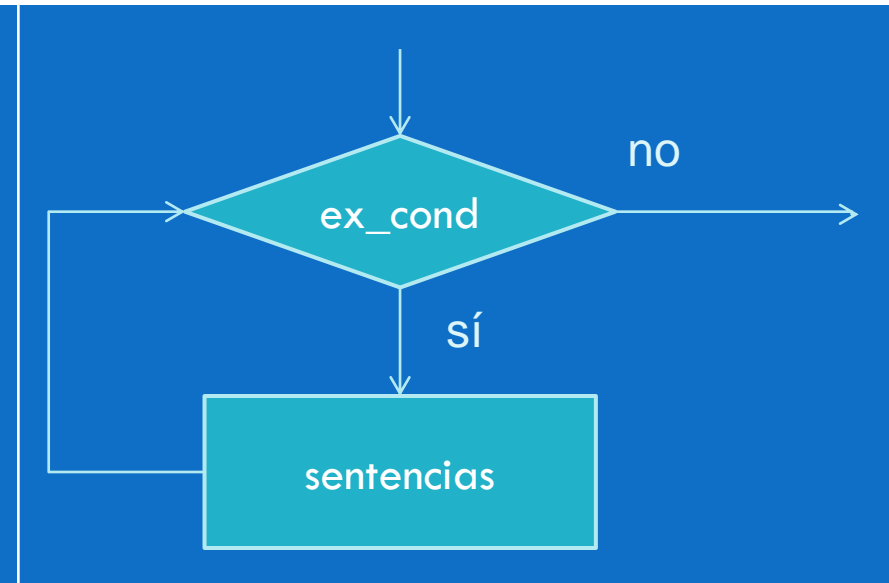
1.6. Pseudocódigo.

1.7. PselInt.

□ Repetitivas o cíclicas.

▣ Mientras

**Mientras <expresión-condicional> hacer
<sentencias>
FinMientras;**



Estructuras de la programación estructurada

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PselInt.

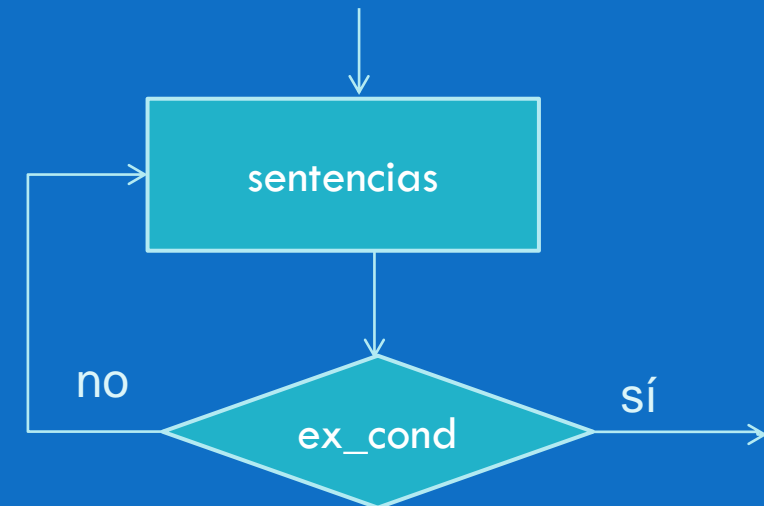
□ Repetitivas o cíclicas.

▣ Hasta.

Repetir

<sentencias>

Hasta <expresión-condicional>;



Estructuras de la programación estructurada

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

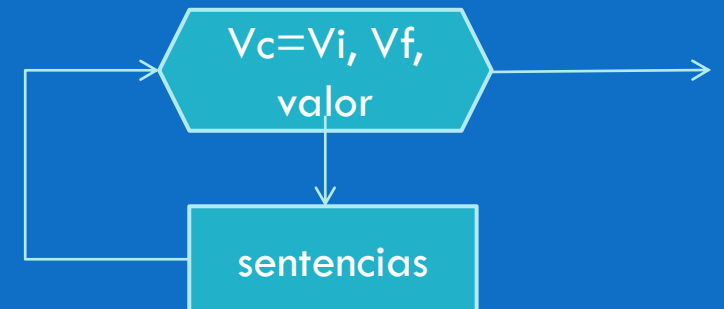
1.6. Pseudocódigo.

1.7. PseInt.

□ Repetitivas o cíclicas.

■ Para.

Para <Vc> = <Vi> hasta <Vf> incremento <valor>
 <sentencias>
FinPara;



Ejercicios

16. Introduce el nombre, lugar de nacimiento y la edad de una serie de personas (finaliza cuando el nombre es “fin”). Sabiendo que la suscripción a un seguro depende los siguientes factores:
 - Si la edad está entre 1 y 50, paga 50 euros.
 - Si es mayor de 50, paga 70 euros.
 - Si es de Cáceres tiene un incremento de 20 euros.
 - Visualizar el nombre junto con el importe a pagar.
17. Leer 20 números y visualizar para cada uno de ellos el doble del valor que se ha tecleado.

Ejercicios

18. ¿Se podría implementar una estructura mientras con una estructura hasta? ¿Y una estructura hasta con una estructura mientras? En caso afirmativo, implementa el/los código(s) correspondiente(s)

Variables auxiliares de un programa

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

- ☐ Contadores.
- ☐ Acumuladores.
- ☐ Swtiches.

Variables auxiliares de un programa

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

□ Contadores.

- Cuenta cualquier evento que pueda ocurrir en un programa.

- Operaciones básicas:

 - Inicialización: $\text{contador} \leftarrow 0;$

 - Contabilización o incremento: $\text{contador} \leftarrow \text{contador} + 1;$

□ Acumuladores.

- Uso: calcular sumas y productos (totales).

- Operaciones básicas:

 - Inicialización: $\text{suma} \leftarrow 0;$

 - Acumulación: $\text{suma} \leftarrow \text{suma} + \text{elemento};$

Variables auxiliares de un programa

1.1. Fases del diseño y puesta a punto de un programa.

1.2. Traducción e interpretación

1.3. Concepto de dato. Constantes y variables

1.4. Tipos de datos. Operadores y Expresiones.

1.5. Concepto y definición de algoritmo.

1.5.1. Diseño e implementación de algoritmos.

1.5.2. Estructura general de un programa.

1.5.3. Programación estructurada. Estructuras básicas.

1.5.4. Variables auxiliares de un programa.

1.6. Pseudocódigo.

1.7. PseInt.

- Interruptores o conmutadores (switches).
 - ▣ Variables de tipo lógico (toma dos valores).
 - ▣ Transmiten información de un punto a otro del programa.
 - ▣ Inicialización a uno de los dos valores
 - ▣ Conmuta su valor en los puntos correspondientes del programa.

Ejercicios

19. Leer 10 notas numéricas y visualizar cuántos suspensos y aprobados hay.
20. Leer 15 números y visualizar cuántos son múltiplos de 5.
21. Leer 7 números y visualizar cuántos son múltiplos de 2, cuántos son de 3 y cuántos de 6.
22. Leer 20 números de 2 en 2 y visualizar si el primero es múltiplo del segundo.
23. Leer 15 letras y visualizar cuántas son vocales.
24. Visualizar los pares que hay entre 1 y 200.
25. Visualizar los 30 primeros impares.
26. Leer 30 números y visualizar si hay algún número 8.
27. Implementa la estructura *para* con una estructura *mientras* y con una estructura *hasta* en dos ejercicios independientes.