CAPÍTULO 3: Algoritmos

Departamento de Sistemas
Universidad de la Frontera

Algoritmos

- Definiciones
- Características
- Algoritmos cotidianos
- Elementos de un Algoritmo
- Representación de un Algoritmo
- Diagrama de Flujo
- Seudocódigo

Algoritmos

Objetivo

Diseñar y construir algoritmos.

La principal razón para que las personas aprendan a diseñar algoritmos (para luego crear programas) es utilizar el computador como una herramienta para resolver problemas.

Algoritmos

Recordar una definición:

- **Definición 3**: Es un conjunto de instrucciones que, combinadas de forma adecuada, resuelven un determinado problema en una cantidad finita de tiempo. Cada instrucción es una indicación sencilla y no ambigua.

Algoritmos

Resolución de problemas.

Abordar la solución de un problema requiere de tres etapas importantes:

- Análisis del problema
- Diseño/desarrollo de algoritmos
- Representación/escritura de algoritmos
  - Representación gráfica
  - Representación escrita (lenguajes de programación)
  - Etc.

Algoritmos

Análisis del problema

- comprender la naturaleza del problema
- definir claramente el problema
- analizar con detalle
- especificar las entradas y salidas

Algoritmos

Diseño/desarrollo de Algoritmos

- descubrir cómo se resuelve el problema
- encontrar las instrucciones que resuelven el problema
- establecer la secuencias de instrucciones que conducen a la solución del problema.
- esforzarse por lograr soluciones eficientes

Algoritmos

Técnica de diseños de algoritmos

- Top-Down
- Bottom-Up

Top-Down

Conocida como Diseño Descendente, consiste en dividir el problema general en sub problemas, a continuación éstos se descomponen en sub problemas más simples, y así sucesivamente hasta que los problemas más pequeños sean fáciles de resolver.

A los algoritmos que resuelven los sub problemas se les llama sub algoritmos

Algoritmos

Refinamiento por pasos sucesivos

- es la estrategia del Top-Down
- inicialmente sólo se indicarán las tareas más importantes que se deben realizar para resolver el problema
- generalmente esta primera descripción es incompleta
- algunas de estas se pueden detallar en forma más precisa los pasos a seguir
- algunos de estos pasos pueden requerir un refinamiento adicional que nos llevará a un nivel más de descripción del algoritmo.

Algoritmos

Ventajas del diseño descendente:

- es más fácil comprender un problema al dividirlo en partes más simple
- la modificación de un sub problema es más fácil
- se puede verificar si la solución es correcta de manera más fácil.

Algoritmos

Bottom-Up

El diseño ascendente se refiere a la identificación de procesos que necesitan computarizarse a medida que vayan apareciendo.

Desventaja del diseño Bottom –Up

- es difícil integrar los subsistemas para tener un desempeño global.
- La integración entre los subsistemas es muy costosa y muchas veces no se consigue y los problemas no se solucionan.

Algoritmos

Representación/escritura de algoritmos

- escribir un algoritmo consiste en realizar una descripción paso a paso en un lenguaje natural.
- debe respetar el conjunto de reglas del lenguaje.
- la secuencias definidas de paso debe conducir a un resultado coherente.
- sólo puede ejecutarse una operación a la vez.
- el flujo de control usual de un algoritmo es secuencial.

Algoritmos

Análisis de Algoritmos

Tarea	Algoritmo
Preparar un pastel	Receta
Ejecutar una sonata	Partitura
Construcción de una caseta	Planos de construcción
Hacer un vestido	Patrón

Algoritmos

Tarea	Entrada
Preparar un pastel	Harina, azúcar, etc
Ejecutar una sonata	
Construcción de una caseta	Ladrillos, arena
Hacer un vestido	Tela, hilos

Algoritmos

Tarea	Salida
Preparar un pastel	Pastel
Ejecutar una sonata	Sonido
Construcción de una caseta	Casa
Hacer un vestido	Vestido

Algoritmos

Diseño/desarrollo de algoritmos

Problema: ¿Qué hacer para ver la película Titanic?

La respuesta es sencilla y puede describirse en forma de algoritmo:

Inicio

Ubicar el cine apropiado

Ir al cine

Comprar una entrada (ticket)

Ver la película

Regresar a casa

Fin

El algoritmo consta de 5 acciones básicas secuenciales

Algoritmos

Refinamiento en pasos sucesivos:

Ubicar el cine apropiado

Tomar el diario
Ir a las páginas sociales
Revisar la cartelera
Tomar nota del cine

Algoritmos

Refinamiento en pasos sucesivos:

Ir al cine

Si ”no hay tiempo para llegar” entonces
decidir otra actividad
olvidar ir al cine
si no
salir rumbo al cine
Finsi

Algoritmos

Comprar una entrada

Ponerse a la cola y avanzar
Si ”no hay entradas” entonces
decidir otra actividad
olvidar ir al cine
si no
pagar la entrada
comprar “palomitas y coca cola”
pasar a la sala
localizar la butaca
Ver la película
abandonar el cine
Regresar a casa
Finsi

Algoritmos

Condición “sine qua non” para el análisis de Algoritmos

- Entrada
- Proceso
- Salida

Algoritmos

Ejercicio 1:

Construir un algoritmo que lea dos números enteros, calcule la suma y la imprima.

Entradas: dos números enteros

Salida: la suma de los dos números enteros

Proceso:

Leer los dos números enteros

Realizar la suma de los dos números enteros

Guardar la suma en una variable de nombre **sum**

Imprimir la suma, es decir, **sum**

Algoritmos

Refinamiento 1:

Declarar dos variables para los números num1 y num2

Declarar una variable para la suma: sum

Inicio

Leer el primer número, num1

Leer el segundo número, num2

sum ← num1 + num2

Escribir sum

Fin

Algoritmos

Refinamiento 2:

Var num1, num2, sum: numéricas

Inicio

Escribir "Ingrese el primer número"

Leer num1

Escribir "Ingrese el segundo número"

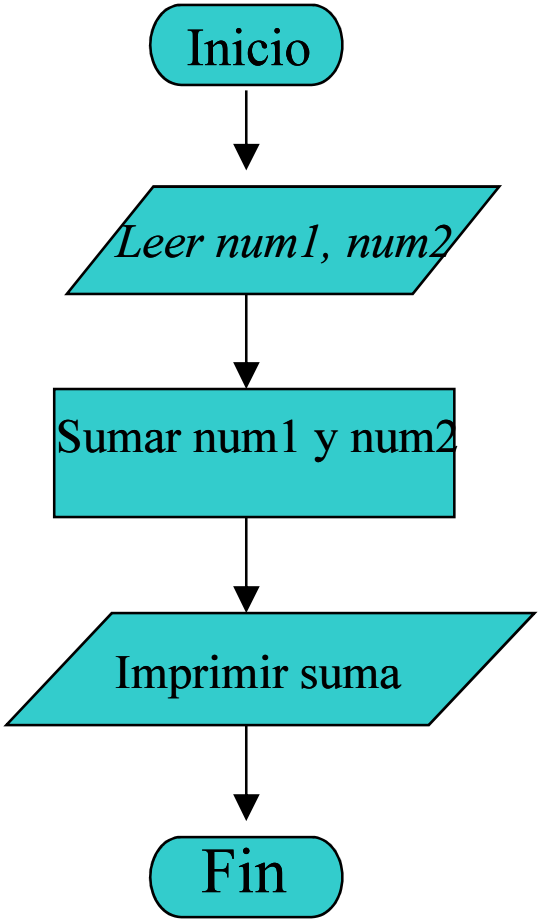
Leer num2

sum ← num1 + num2

Escribir sum

Fin

Algoritmos



Algoritmos

Ejercicio 2:

Construir un algoritmo que lea dos números enteros, calcule la suma y la multiplicación e imprima ambos resultados.

Determinar: **Entradas**, **Salida** y **Proceso**.

Algoritmos

Solución Ejercicio 2:

Entradas: dos números enteros

Salida: la suma y la multiplicación de los dos números enteros

Proceso:

Leer los dos números enteros

Realizar la suma de los dos números enteros

Guardar la suma en una variable de nombre **sum**

Realizar la multiplicación de los dos números enteros

Guardar la multiplicación en una variable de nombre **mult**

Escribir la suma, es decir imprimir **sum**

Escribir la multiplicación, es decir **mult**

Algoritmos

Refinamiento 1:

Declarar dos variables para los números num1 y num2

Declarar dos variable para la suma y multiplicación: sum, mult

Inicio

Leer el primer número, num1

Leer el segundo número, num2

sum ← num1 + num2

mul ← num1 * num2

Escribir sum

Escribir mult

Fin

Algoritmos

Refinamiento 2:

Var num1, num2, sum, mult: numéricas

Inicio

Escribir "Ingrese el primer número"

Leer num1

Escribir "Ingrese el segundo número"

Leer num2

sum ← num1 + num2

mult ← num1 * num2

Escribir sum

Escribir mult

Fin

Algoritmos - PSeInt

PSeInt-PSeudo Interprete

http://pseint.sourceforge.net

- PSeInt está pensado para asistir a los estudiantes que se inician en la construcción de programas o algoritmos computacionales.
- El seudocódigo se utiliza como primer contacto para introducir conceptos básicos como el uso de estructuras de control, expresiones, variables, etc.
- Este software facilita al principiante la tarea de escribir algoritmos.

Algoritmos - PSeInt

PSeInt-PSeudo Interprete

- Permite generar y editar el diagrama de flujo del algoritmo.
- Permite la edición simultánea de múltiples algoritmos.
- Determina y marca claramente errores de sintaxis (mientras escribe) y en tiempo de ejecución.
- Es multiplataforma (probado en Microsoft Windows, GNU/Linux y Mac OS X).
- Es totalmente libre y gratuito (licencia GPL)

Algoritmos - PSeInt

PSeInt técnicamente:

- Es un software que interpreta seudocódigo
- Es de sintaxis sencilla
- Maneja las estructuras de control básicas
- Maneja solo 3 tipos de datos básicos: numérico, alfanumérico y lógico (verdadero-falso).
- Maneja estructuras de datos: arreglos

Algoritmos - PSeInt

Estructura del algoritmo en seudocódigo

```
Proceso SinTitulo
    accion 1;
    accion 1;
        .
        .
        .
    accion n;
FinProceso
```

- La sección “Proceso SinTitulo” es la cabecera del algoritmo
- La sección “acción 1, acción 1,…” es el cuerpo del algoritmo
- La sección de declaraciones se omite dado que el software (PSeInt) se encarga de asignarle el tipo de dato a cada variable según el uso.

Algoritmos - PSeInt

Estructura del algoritmo en seudocódigo

```
Proceso SinTitulo
    accion 1;
    accion 1;
        .
        .
        .
    accion n;
FinProceso
```

- Comienza con la palabra clave *Proceso* seguida del nombre del programa (algoritmo),
- Le sigue una secuencia de acciones (instrucciones) cada una terminada en punto y coma.
- Finaliza con la palabra clave *FinProceso*.

Algoritmos - PSeInt

Estructura del algoritmo en seudocódigo

```
Proceso SinTitulo
     accion 1;
     accion 1;
          .
          .
          .
     accion n;
FinProceso
```

- Las acciones incluyen operaciones de entrada y salida, asignaciones de variables, condicionales si-entonces o de selección múltiple y/o ciclos mientras, repetir o para.

Algoritmos - PSeInt

Estructura del algoritmo seudocódigo, ejemplo

```
1  Proceso suma
2      Escribir 'INGRESE PRIMER NUMERO';
3      Leer a;
4      Escribir 'INGRESE SEGUNDO NUMERO';
5      Leer b;
6      c<-a+b;
7      Escribir 'LA SUMA ES:',c;
8  FinProceso
```

Estructura del algoritmo en Java

```
public int suma(int a, int b){
    int c;
    c = a + b;
    return c;
}
```

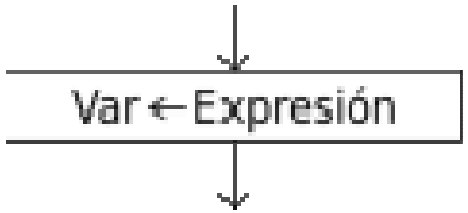
Algoritmos - PSeInt

Asignación

La instrucción de asignación permite almacenar una valor en una variable.

<variable> ← <expresión> ;

- Al ejecutarse la asignación, primero se evalúa la expresión de la derecha y luego se asigna el resultado a la variable de la izquierda.
- El tipo de la variable y el de la expresión deben coincidir.



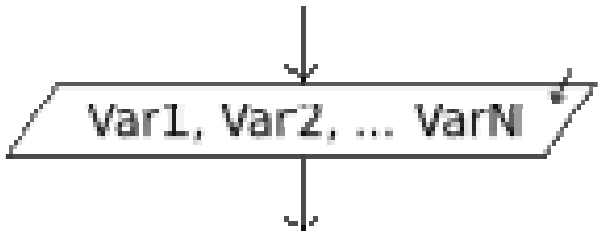
Algoritmos - PSeInt

Entradas

La instrucción Leer permite ingresar información desde el ambiente.

Leer <variablel> , <variable2> , ... ,<variableN> ;

- Esta instrucción lee N valores desde el ambiente (en este caso el teclado) y los asigna a las N variables mencionadas.
- Pueden incluirse una o más variables, por lo tanto el comando leerá uno o más valores.



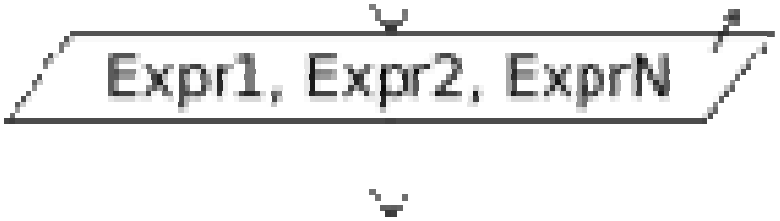
Algoritmos - PSeInt

Salidas

La instrucción Escribir permite mostrar valores al ambiente.

Escribir <exprl> , <expr2> , ... , <exprN> ;

- Esta instrucción imprime al ambiente (en este caso en la pantalla) los valores obtenidos de evaluar N expresiones.
- Dado que puede incluir una o más expresiones, mostrará uno o más valores.



Algoritmos

Ejercicio 3:

Construir un algoritmo que lea dos números enteros y diga cual de ellos es mayor.

Determinar: **Entradas**, **Salida** y **Proceso**.

Algoritmos

Solución Ejercicio 3:

Entradas: dos números enteros

Salida: el mayor de los dos números

Proceso:

Leer los dos números enteros

Comparar los dos números enteros:

Si el primero es mayor o igual al segundo entonces

    Escribir “el primero es el mayor”

si no

    Escribir “el segundo es el mayor”

Finsi

Algoritmos

Refinamiento 1:

Declarar dos variables para los números num1 y num2
Leer el primer número, num1
Leer el segundo número, num2
Si num1 >= num2 entonces
    Escribir num1 es el mayor
si no
    Escribir num2 es el mayor
Finsi

Algoritmos

Refinamiento 2:

Var num1, num2: numéricas

Inicio

Escribir "Ingrese el primer número"

Leer num1

Escribir "Ingrese el segundo número"

Si num1 >= num2 entonces

Escribir num1 es el mayor

si no

Escribir num2 es el mayor

Finsi

Fin

Algoritmos - PSeInt

Condicional Si-Entonces-Sino

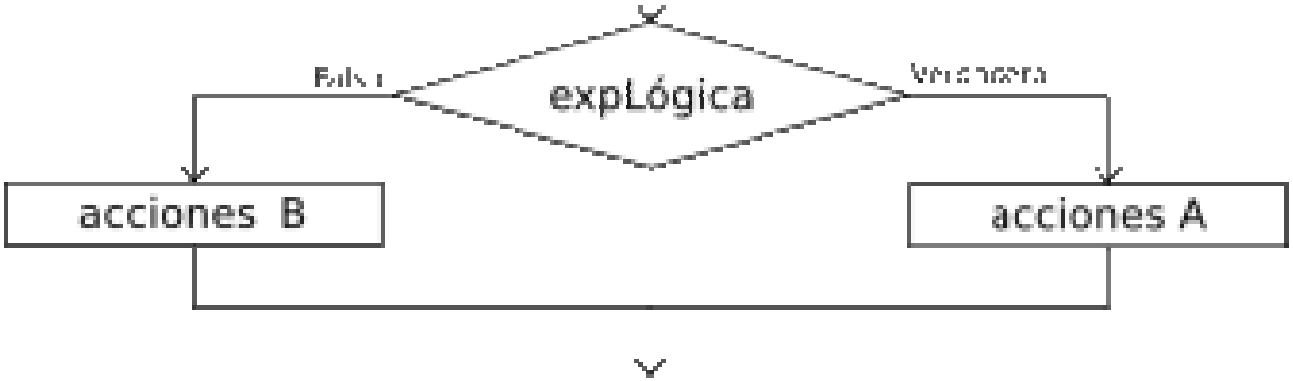
La secuencia de instrucciones ejecutadas por la instrucción Si-Entonces-Sino depende del valor de una condición lógica.

```
Si <condición>   Entonces
        <instrucciones>
Sino
        <instrucciones>
FinSi
```

Se evalúa la condición y se ejecutan las instrucciones que le siguen al Entonces si la condición es verdadera, o las instrucciones que le siguen al *Sino* si la condición es falsa. La condición debe ser una expresión lógica, que al ser evaluada retorna *Verdadero* o *Falso*.

Algoritmos - PSeInt

Condicional Si-Entonces-Sino

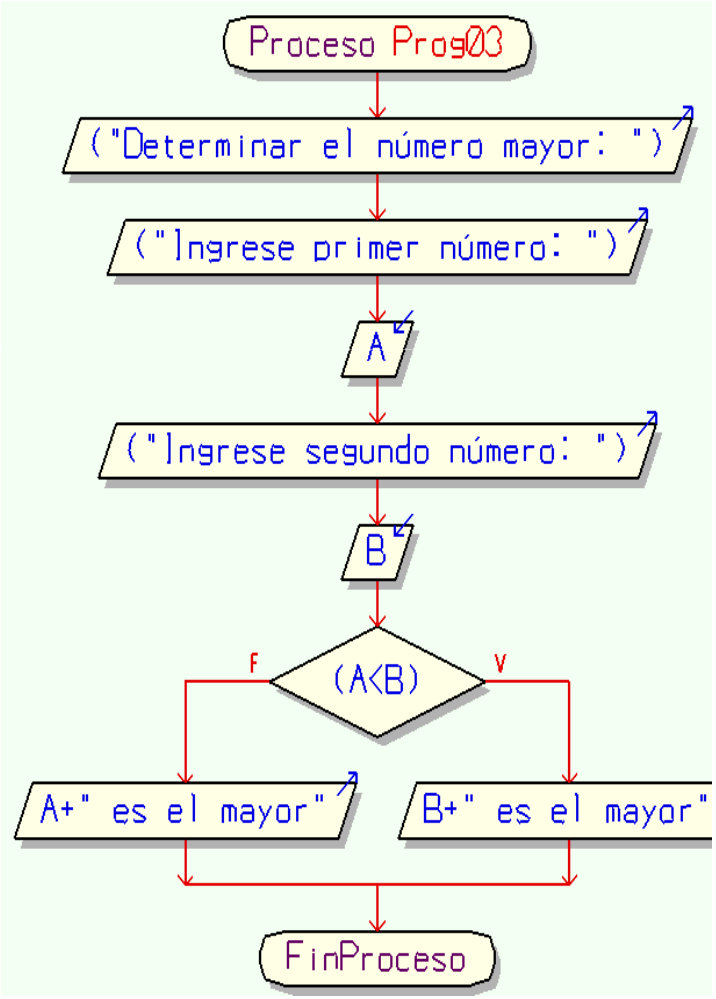


Algoritmos - PSeInt

```
1   Proceso Prog03
2       Escribir ("Determinar el número mayor: ");
3       Escribir ("Ingrese primer número: ");
4       Leer A
5       Escribir ("Ingrese segundo número: ");
6       Leer B
7       Si (A < B) Entonces
8           Escribir B+" es el mayor";
9       sino
10          Escribir A+" es el mayor";
11      FinSi
12  FinProceso
```

```
*** Ejecución Iniciada. ***
Determinar el número mayor:
Ingrese primer número:
> 40
Ingrese segundo número:
> 20
40 es el mayor

*** Ejecución Finalizada. ***
```



Algoritmos - PSeInt

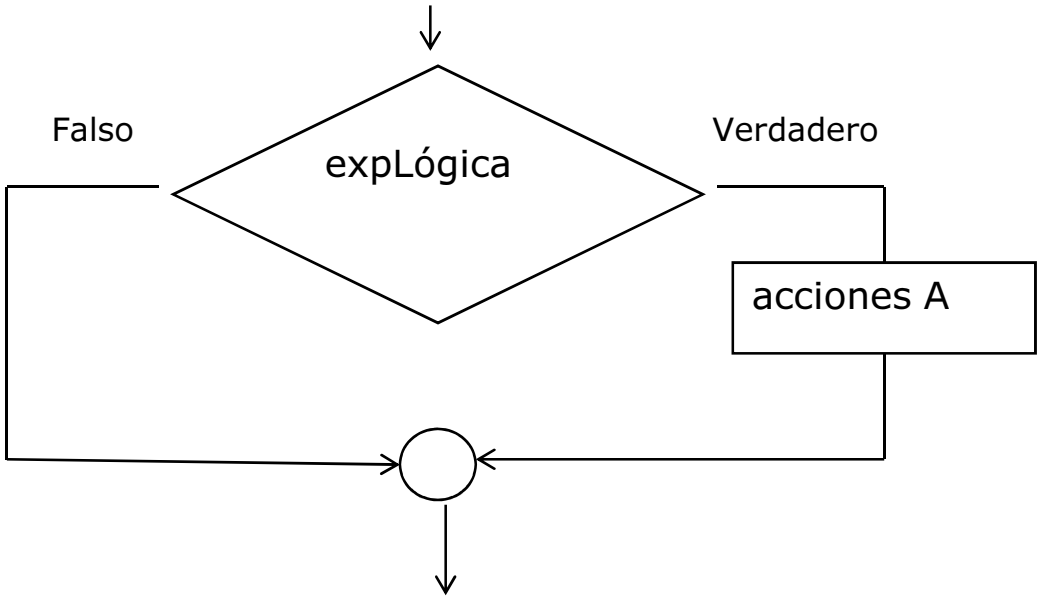
Condicional Si-Entonces

La cláusula Entonces es obligatoria, pero la cláusula Sino puede no estar. En ese caso, si la condición es falsa no se ejecuta ninguna instrucción y la ejecución del programa continúa con la instrucción siguiente.

```
Si <condición>  Entonces
    <instrucciones>
FinSi
```

Algoritmos - PSeInt

Condicional Si-Entonces



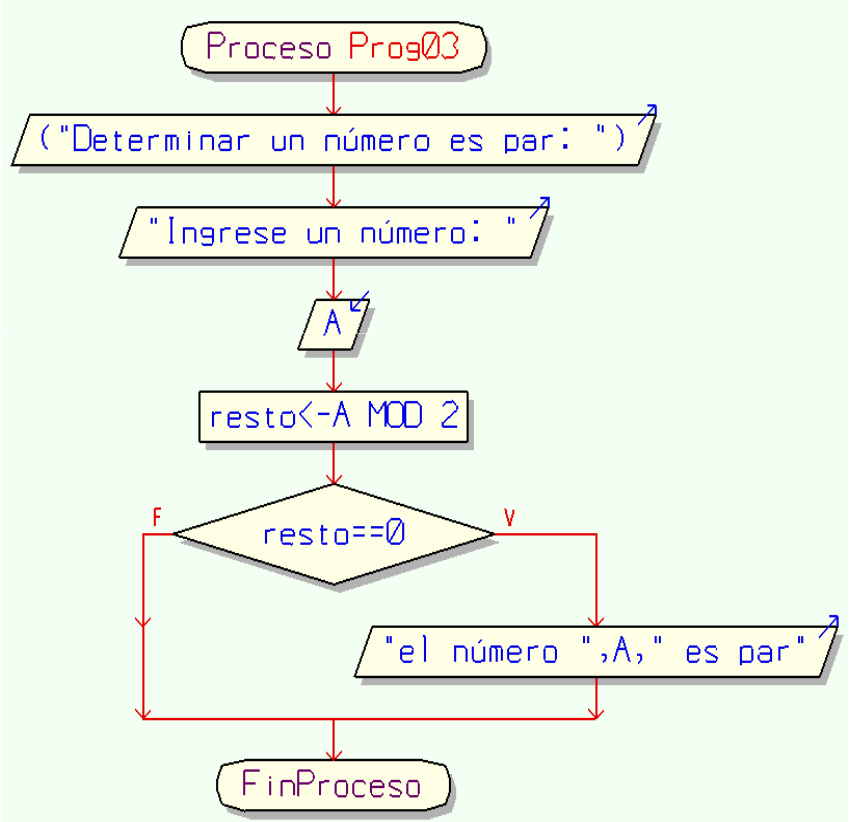
Algoritmos - PSeInt

Ejemplo Prog03

```
1  Proceso Prog03
2      Escribir ("Determinar un número es par: ");
3      Escribir "Ingrese un número: ";
4      Leer A;
5      resto = A MOD 2;
6      Si resto == 0 Entonces
7          Escribir "el número ",A," es par";
8      FinSi
9  FinProceso
```

```
*** Ejecución Iniciada. ***
Determinar un número es par:
Ingrese un número:
> 1204
el número 1204 es par

*** Ejecución Finalizada. ***
```



Algoritmos

Ejercicio 4:

Construir un algoritmo que lea dos números enteros y diga cual de ellos es mayor o si son iguales.

Determinar: **Entradas**, **Salida** y **Proceso**.

Algoritmos

Solución Ejercicio 4:

Entradas: dos números enteros

Salida: el mayor de los dos números o un mensaje indicando que son iguales

Algoritmos

Proceso:

Leer los dos números enteros
Si el primero es mayor o igual al segundo entonces
Si el primero es igual al segundo entonces
Escribir “los números son iguales”
si no
Escribir “el primero es el mayor”
Finsi
si no
Escribir “el segundo es el mayor”
Finsi

Algoritmos

Refinamiento 1:

Declarar dos variables para los números num1 y num2
Leer el primer número, num1
Leer el segundo número, num2
Si num1 >= num2 entonces
    Si num1 = num2 entonces
        Escribir “Los números son iguales”
    si no
        Escribir num1 “ es el mayor”
    Finsi
si no
        Escribir num2 “ es el mayor”
Finsi

Algoritmos

Refinamiento 2:

```
Var num1, num2: numéricas
Inicio
    Escribir "Ingrese el primer número"
    Leer num1
    Escribir "Ingrese el segundo número"
    Si num1 >= num2    entonces
        Si num1 = num2 entonces
            Escribir "los números son iguales"
        si no
            Escribir num1 " es el mayor"
        Finsi
    si no
        Escribir num2 " es el mayor"
    Finsi
Fin
```

Algoritmos

Ejercicio 5

Calcular el área y perímetro de un círculo

a) Análisis del problema

- Definición del problema
- Especificaciones de entrada
- Especificaciones de salida

b) Diseño y desarrollo del algoritmo

c) Escritura del algoritmo

Algoritmos

Ejercicio 6

Calcular el perímetro de un triángulo rectángulo

a) Análisis del problema

- Definición del problema
- Especificaciones de entrada
- Especificaciones de salida

b) Diseño y desarrollo del algoritmo

c) Escritura del algoritmo

Algoritmos

- Definiciones
- Características
- Algoritmos cotidianos
- Elementos de un algoritmo
- Representación de un Algoritmo
- Diagrama de Flujo
- Seudocódigo