



Capítulo 2

Diseño y codificación de algoritmos

2.1. Resultados de aprendizaje

1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.

a) Se han identificado los bloques que componen la estructura de un programa informático.

2.2. Definición

Un algoritmo se puede definir como una secuencia de instrucciones que representan un modelo de solución para un problema. O bien como un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema.

http://ing.unne.edu.ar/pub/informatica/Alg_diag.pdf

En matemáticas un algoritmo se define como un conjunto de instrucciones o reglas bien definidas, ordenadas y finitas que permite llevar a cabo una actividad mediante pasos sucesivos que no generen dudas a quien deba hacer dicha actividad.

Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución.

<https://es.wikipedia.org/wiki/Algoritmo>

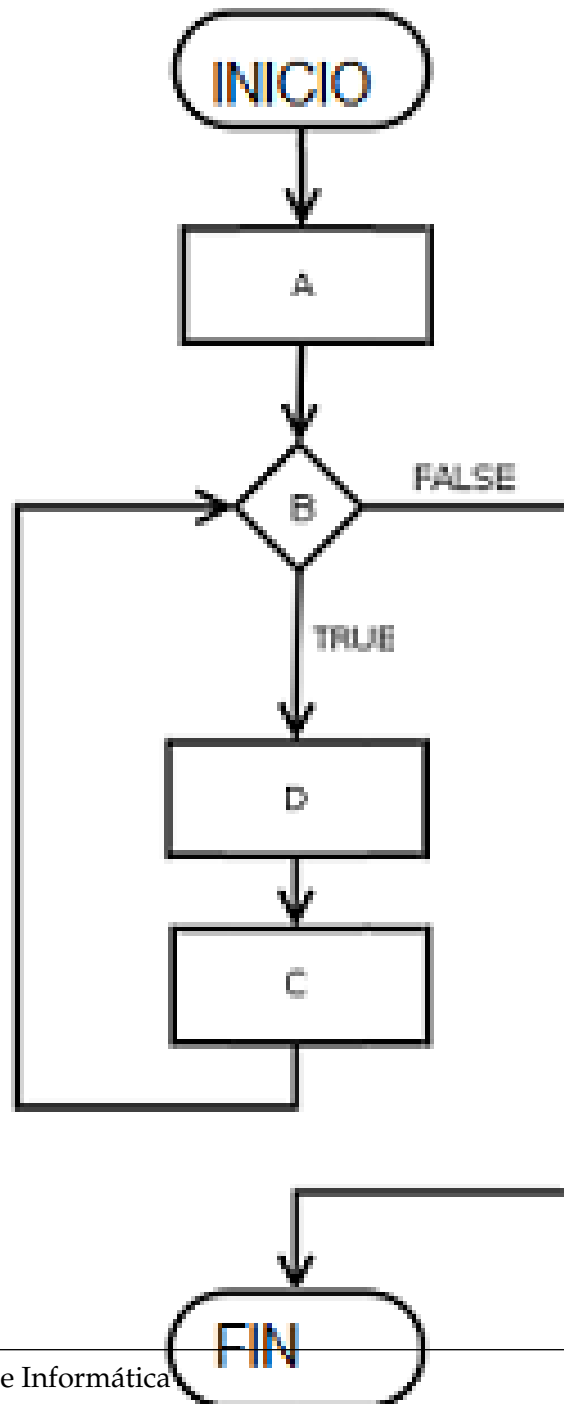
2.3. Diagramas de flujo

El diagrama de flujo o diagrama de actividades es la representación gráfica del algoritmo.

https://es.wikipedia.org/wiki/Diagrama_de_flujo

Estos diagramas utilizan símbolos con significados definidos que representan los pasos del algoritmo, y representan el flujo de ejecución mediante flechas que conectan los puntos de inicio y de fin del proceso. Un diagrama de flujo presenta un único punto de inicio y un único punto de cierre.

for(A;B;C)
D;



2.3.1. Elementos de un diagrama

- Inicio/final



- Instrucción o proceso



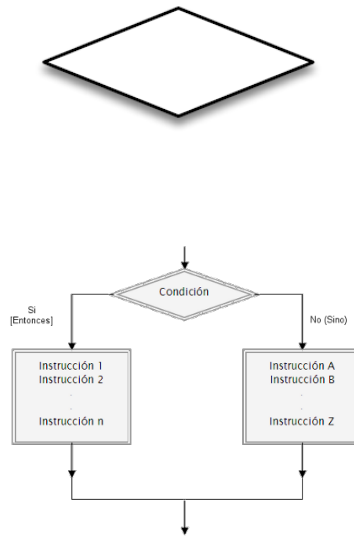
Representa un paso u operación en el proceso.

Cualquier proceso interno realizado por el ordenador como asignación de valor a variables, operaciones matemáticas, etc.

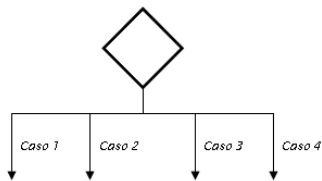
- **Flechas o líneas de flujo** que sirven para representar el flujo o secuencia del programa. Indica qué proceso viene a continuación, es decir, cual es el siguiente paso.
- **Conectores.** Sirven para representar conexiones con otra parte del diagrama o con otro diagrama en otra página.



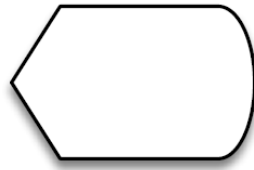
- **Alternativa o decisión.** La condición planteada presenta dos alternativas (casos), siguiendo el programa distinta vía en función del caso.



- **Alternativa o decisión múltiple.** La condición planteada presenta distintas alternativas (casos), siguiendo el programa distinta vía en función del caso.



- **Pantalla.** Permite mostrar mensajes al usuario.



- **Impresora.** Permite imprimir mensajes.



- **Teclado.** Permite recibir valores de entrada tecleados por el usuario. Sólo permite leer un valor cada vez.



- En ocasiones la **entrada y salida de datos**, independientemente de que se lleve a cabo por impresora, por pantalla o por teclado se representa con el siguiente símbolo.

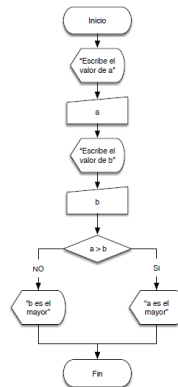


- **Módulo independiente.** Recibe distintos nombres como subprograma, subrutina, proceso, procedimiento, etc. Al llegar a esta llamada el programa pasa a ejecutar todas las instrucciones contenidas en la subrutina para una vez terminadas continuar el flujo.

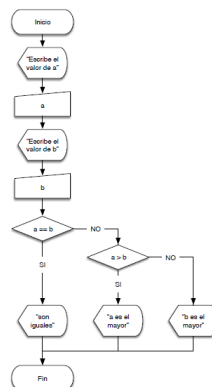


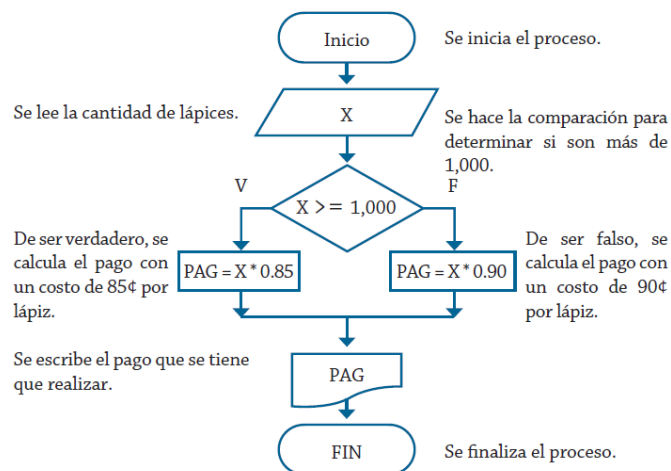
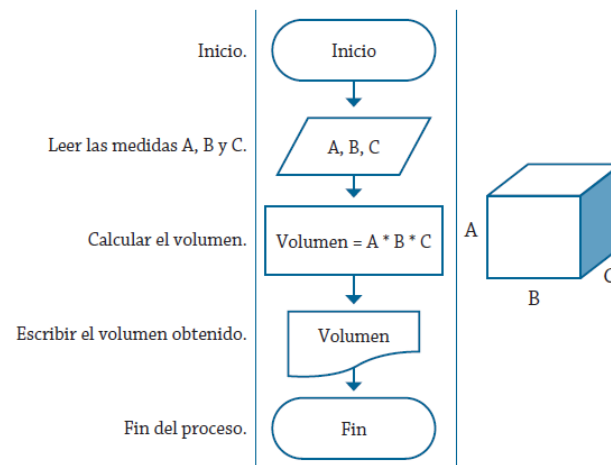
2.4. Ejemplos

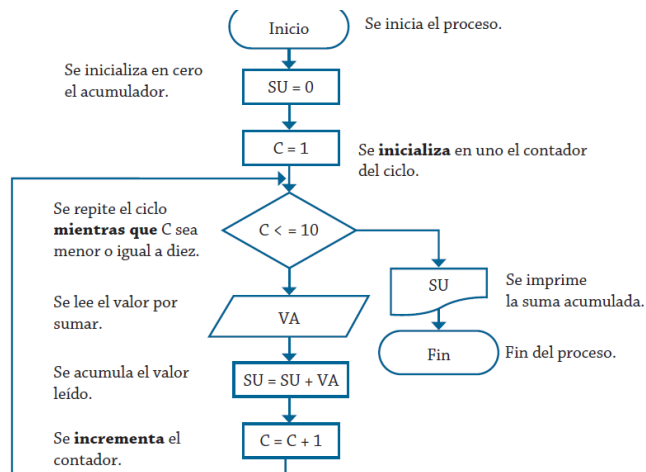
Escribe un programa que, dados dos números, muestre en pantalla cual es mayor.



Mejora el diagrama anterior para que contemple que los dos números puedan ser iguales.



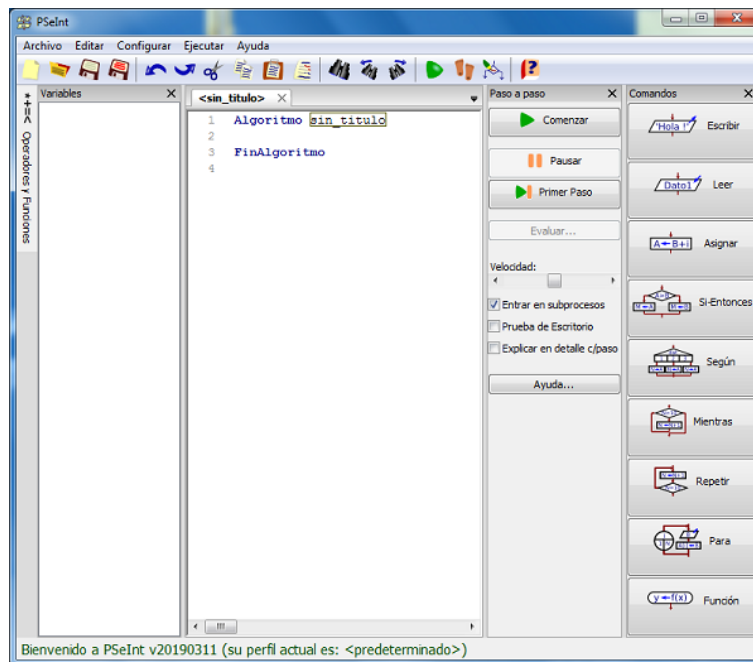




2.5. PSeInt

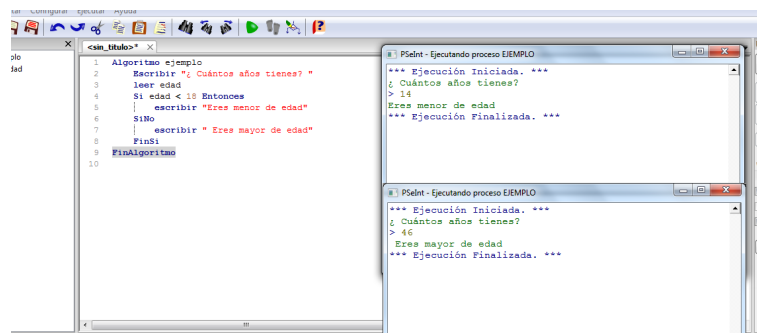
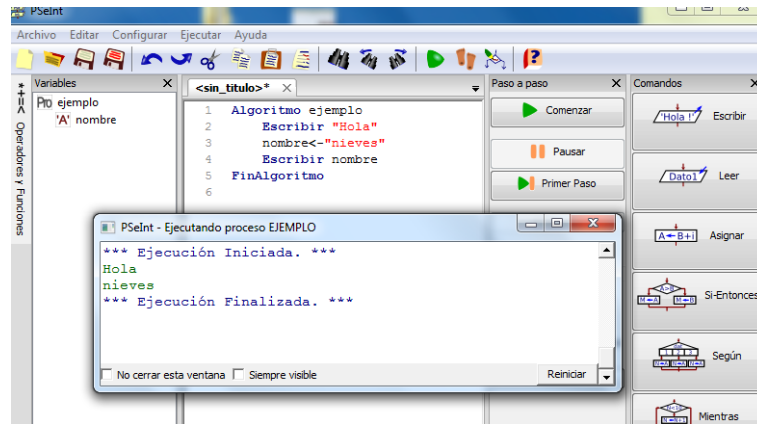
PSeInt, es la abreviatura de Pseudocode Interpreter, Intérprete de Pseudocódigo. Este programa fue creado como proyecto final para la asignatura de Programación 1 de la carrera Ingeniería en Informática de la Facultad de Ingeniería y Ciencias Hídricas de la Universidad Nacional del Litoral.

El pseudocódigo describe un algoritmo utilizando una mezcla de frases en lenguaje común, instrucciones de programación y palabras clave que definen las estructuras básicas. Su objetivo es permitir que el programador se centre en los aspectos lógicos de la solución a un problema.



Los elementos que forman este intérprete son:

- Botón **Escribir** Nos permite mostrar en pantalla texto y/o el contenido de variables.
- Botón **Leer** Nos permite recibir valores a través del teclado y guardarlos en variables.
Leer a; Recibe el valor y lo almacena en a.
Leer a, b, c; Recibe tres valores y los guarda en las variables a, b y c respectivamente
- Botón **Asignar** Nos permite guardar un valor en una variable.
c <- 2; la variable c almacenará el valor 2.
- Botón **Si** Nos permite evaluar el contenido de una variable, y en función de este valor, realizar acciones.
- Botón **Según** Nos permite comparar el contenido de una variable con múltiples valores, y después de comparar una a una las salidas con la evaluación, realizar la acción determinada.



```

1 Segun num_dia_sem Hacer
2   1: Escribir "Lunes";
3   2: Escribir "Martes";
4   3: Escribir "Miércoles";
5   4: Escribir "Jueves";
6   5: Escribir "Viernes";
7   6: Escribir "Sábado";
8   7: Escribir "Domingo";
9 De Otro Modo:
10   Escribir "No es un día de la semana";
11 FinSegun

```

- Botón **Mientras** Permite realizar ciertas acciones de manera repetida mientras se cumpla una condición.

```

1 Mientras num != 0 Hacer
2   Leer num;
3   cant_num<-cant_num+1;
4 FinMientras

```

- Botón **Repetir** Repite una serie de acciones hasta que se cumpla cierta condición.

```

1 Repetir
2   Leer num;
3   cant_num<-cant_num+1;
4 Hasta Que num = 0;

```

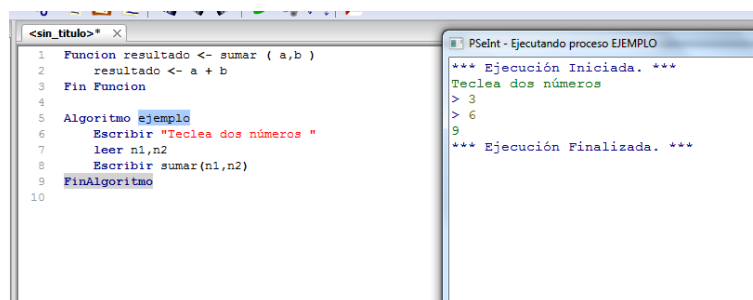
- Botón **Para** Presenta un cierto rango de valores, y para ellos realiza una determinada acción

```

1 Para i <- 1 Hasta 10 Con Paso 1 Hacer
2   Escribir i;
3 FinPara

```

- **Funciones** permite añadir funciones o módulos independientes al programa



2.6. Ejemplos

```
1 Algoritmo Ejemplo1
2   Escribir "Introduce el valor de a"
3   Leer a
4   Escribir "Introduce el valor de b"
5   Leer b
6   Si a > b Entonces
7     Escribir " a es mayor"
8   SiNo
9     Escribir " b es mayor"
10  Fin Si
11 FinAlgoritmo
```

```
1 Algoritmo Ejemplo2
2   Escribir "Introduce el valor de a"
3   Leer a
4   Escribir "Introduce el valor de b"
5   Leer b
6   Si a = b Entonces
7     Escribir " son iguales"
8   SiNo
9     Si a > b Entonces
10      Escribir " a es mayor"
11    SiNo
12      Escribir " b es mayor"
13    Fin Si
14  FinSi
15 FinAlgoritmo
```

```
1 Algoritmo Ejemplo3
2   Leer a,b,c
3   volumen<-a * b * c
4   Escribir volumen
5 FinAlgoritmo
```

```
1 Algoritmo Ejemplo4
2   Leer x
3   Si x >= 1000 Entonces
4     pag <- x * 8.85
5   SiNo
6     pag<- x * 0.9
7   FinSi
8   Escribir pag
9 FinAlgoritmo
```

```
1 Algoritmo Ejemplo5
2   su <- 0
3   c <- 1
4   Mientras c <= 10 Hacer
5     Leer va
6     su <- su + va
7     c <- c + 1
8   Fin Mientras
9   Escribir su
10 FinAlgoritmo
```
