

Técnicas de Programación

TRAMO I - SEMANA 1 CLASE 2





Técnicas de Programación

Ejemplo algoritmo: Comer uvas

Acciones del procesador

- ☐ Tomar uva.
- ☐ Poner uva en la boca.
- ☐ Masticar una uva.
- ☐ Desechar una uva.
- ☐ Informar que no hay uvas.

Condiciones posibles

- ☐ Hay uvas.
- ☐ No hay uvas.
- ☐ Uva en buen estado.
- ☐ Uva en mal estado.



Técnicas de Programación

Ejemplo algoritmo: Comer uvas

Procesos a realizar

1. Comer una uva si está en buen estado.
2. Tomar dos uvas y comerlas de estar en buen estado.
3. Comer un racimo de uvas, desechando las que estén en mal estado.

Técnicas de Programación

Proceso 1: comer una uva si está en buen estado.

Versión 1 (siempre hay uvas)

Comienzo →

Tomar una uva

Si la uva está en buen estado

Entonces

Poner la uva en la boca

Masticar la uva

Tragar la uva

Si no

Desechar la uva

Fin

Técnicas de Programación

Proceso 1: comer una uva si está en buen estado.

Versión 2 tiene la mejora de evaluar si hay uvas.

Comienzo →

Si hay uvas

Entonces

Tomar una uva

Si la uva está en buen estado

Entonces

Poner la uva en la boca

Masticar la uva

Tragar la uva

Si no

Desechar la uva

Si no

Informar no hay uvas

Fin

Técnicas de Programación

Proceso 2: Tomar 2 uvas y comerlas en caso de estar en buen estado.

vamos a crear un subalgoritmo que vamos a reutilizar.
(Así evitamos escribir varias veces lo mismo)

Sub Algoritmo **comer_uva**

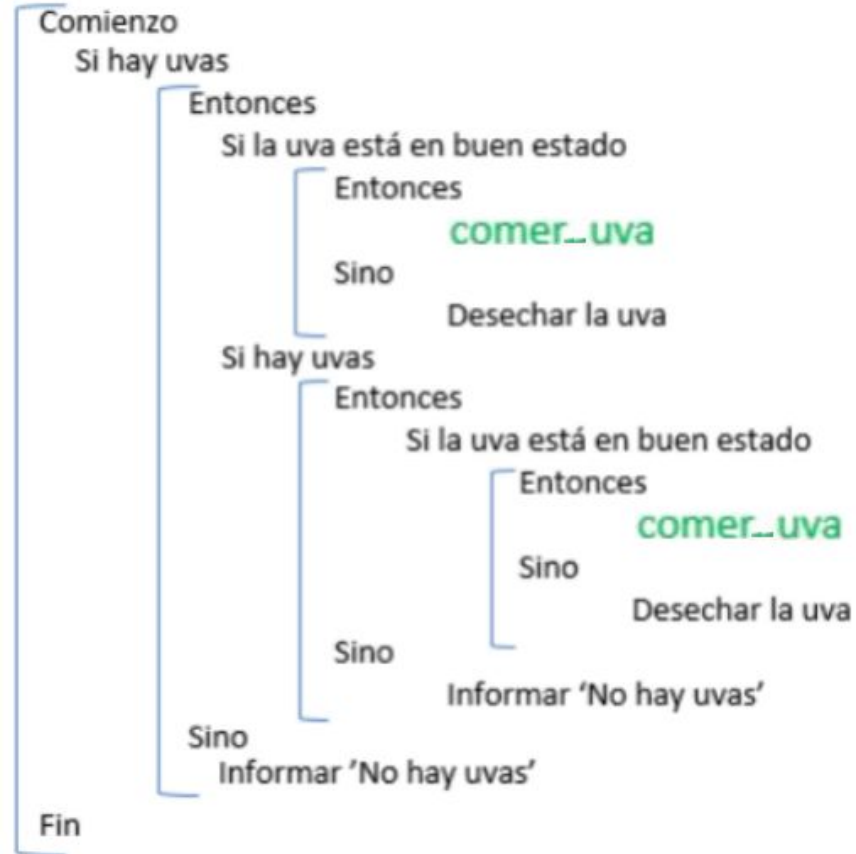
Comienzo

Poner la uva en la boca
Masticar la uva
Tragar la uva

Fin

Técnicas de Programación

Programa principal



Técnicas de Programación

Proceso 3: Comer un racimo de uvas, desechando las que estén en mal estado.

Versión 1

Comienzo

Mientras hay uvas hacer

Si la uva está en buen estado

Entonces

comer...uva

Sino

Desechar la uva

Fin

Técnicas de Programación

Proceso 3: Comer un racimo de uvas, desechando las que estén en mal estado.

Versión 2

Comienzo

Si la uva está en buen estado

Entonces

comer_uva

Sino

Desechar la uva

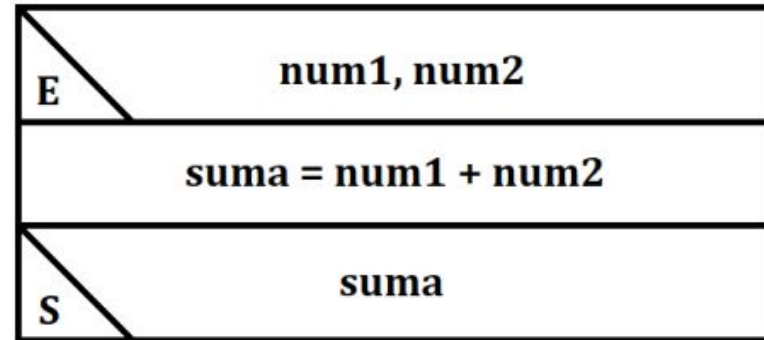
Mientras que haya uvas

Fin

Técnicas de Programación

Diagramación

- ★ Son estructuras que permiten representar de forma gráfica las sentencias que deberá ejecutar un algoritmo.
- ★ Cuando se emprenda el desarrollo de un programa, con un cierto nivel de dificultad, siempre es recomendable la realización de un diagrama.



Técnicas de Programación

Diagramación

Representación de una Entrada

Siempre que se necesite obtener información del usuario para poder desarrollar nuestro programa, existirá lo que se conoce como ingreso de datos o entradas.

Para representar la solicitud de una entrada, se utiliza:



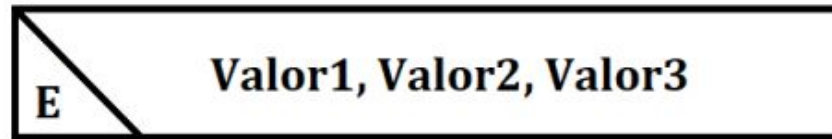
Dentro de este, ponemos solamente el nombre de la variable donde se almacenará el dato que ingrese el usuario.

Técnicas de Programación

Diagramación

Representación de una Entrada

Puedo representar el ingreso de más de un valor en un mismo cuadro de entrada. Separando con comas los nombres de las variables.



No se escribe el mensaje que se muestra al usuario para que realice el ingreso. Solamente se indica el nombre de la variable.



Técnicas de Programación

Diagramación

Representación de un Proceso

Para realizar cálculos y conseguir los resultados necesarios para resolver el problema, existirá lo que conocemos como proceso.



Dentro de este, podemos realizar cualquier operación matemática y asignar el resultado de la misma a una variable.

Técnicas de Programación

Diagramación

Representación de un Proceso

Para guardar el resultado de una operación, la variable que almacenará el mismo va a la izquierda del igual y a la derecha la operación.

RESULTADO = Valor1 + Valor2

Aquí no se declaran las variables que utilizamos, directamente se emplean.



Técnicas de Programación

Diagramación

Representación de una Salida

Siempre que se necesite mostrar en pantalla los valores obtenidos luego haber procesado las entradas, existirá lo que se conoce como egreso de datos o salidas.

Para representar la solicitud de una salida, se utiliza:



Dentro de este, ponemos solamente el nombre de la variable que se desea mostrar al usuario en pantalla.

Técnicas de Programación

Diagramación

Representación de una Salida

Puedo representar el egreso de más de un valor en un mismo cuadro de salida. Separando con comas los nombres de las variables.



No se escribe el mensaje que se muestra al usuario para completar la salida. Solamente se indica el nombre de la variable.



Técnicas de Programación

Pseudocódigo

Permite escribir las distintas instrucciones que se deben llevar a cabo para que un programa realice una determinada función o tarea, de una forma parecida a como lo haríamos en un lenguaje de programación real, pero sin las convenciones propias del tipo de lenguaje en el cual estamos trabajando.

```
1  Proceso Soy_un_Ejemplo
2      Escribir " Hola Mundo!!!" ;
3  FinProceso
4
```



Técnicas de Programación

Pseudocódigo

Declarar las variables que utilizaremos en la resolución del problema:

Definir var1 como

Como Caracter
Como Entero
Como Logico
Como Real

Definir var1, var2, resultado como entero;

var1 es el nombre de la variable y podemos declararla como:

- ☐ Caracter. (cadena / string)
- ☐ Entero.
- ☐ Lógico (boolean -> true/false).
- ☐ Real. (Números con decimales)



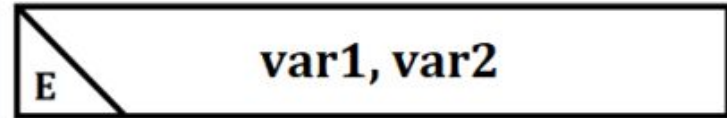
Técnicas de Programación

Pseudocódigo

Para indicar al usuario que debe ingresar un valor, mostraremos en pantalla un mensaje así entiende nuestro objetivo:

var1 y **var2** son las variables ya definidas en donde se almacenarán los valores que ingrese el usuario.

```
Escribir "Ingrese un numero";  
Leer var1;  
Escribir "Ingrese otro número";  
Leer var2;
```



Si no estuviera, por ejemplo, el mensaje “Ingrese un número”, el usuario no sabría que hacer y es muy probable que no ingrese un valor para trabajar.

Técnicas de Programación

Pseudocódigo

Para realizar, por ejemplo, la suma de los dos valores ingresados debo tener definida una variable que almacene el resultado de la operación:

```
resultado = var1 + var2;
```

resultado = var1 + var2

Se realizará la suma de los valores **var1** y **var2** y se almacenará el resultado en la variable **resultado**.

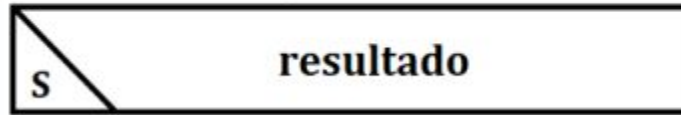


Técnicas de Programación

Pseudocódigo

Mostrar el valor obtenido de la operación al usuario que espera la respuesta del programa.

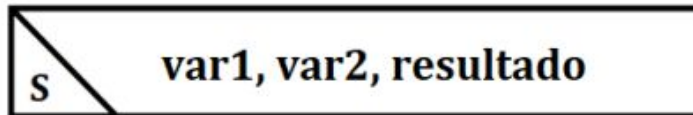
```
Escribir "El resultado de la suma es: ", resultado;
```



Utilizamos Escribir, pero detrás del mensaje a imprimir se debe indicar luego de una coma el nombre de la variable a mostrar.

Si se desea mostrar más de un valor, se indica de la siguiente manera:

```
Escribir "La suma de ", var1, "y ", var2, "es: ", resultado;
```



Técnicas de Programación

Codificación

- Secuencia de sentencias.
(instrucciones que se dan al programa)
- Bifurcación condicional simple y múltiple.
(Decisión/Selección)
- Ciclo, repetición o iteración



Estas estructuras pueden combinarse y anidarse.

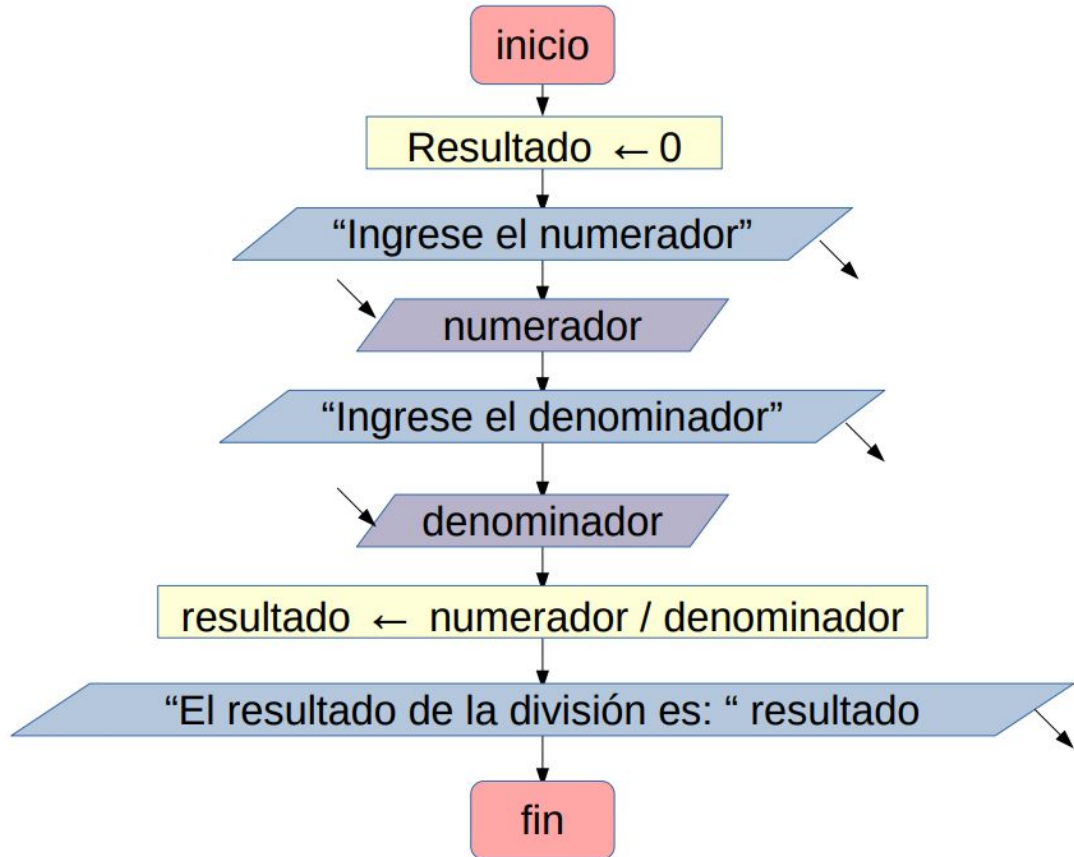
Técnicas de Programación

Diagrama de flujo

SÍMBOLO	NOMBRE	FUNCIÓN
	Inicio / final	Representa el inicio y el final de un proceso
	Línea de flujo	Indica el orden de la ejecución de las operaciones
	Proceso	Representa cualquier tipo de operación
	Decisión	Permite analizar una situación según si su respuesta es verdadera o falsa, sí o no
	Entrada / salida	Representa la lectura de datos en la entrada y la impresión
	Entrada	Representa la lectura de datos en la entrada y la impresión
	salida	Representa la lectura de datos en la entrada y la impresión

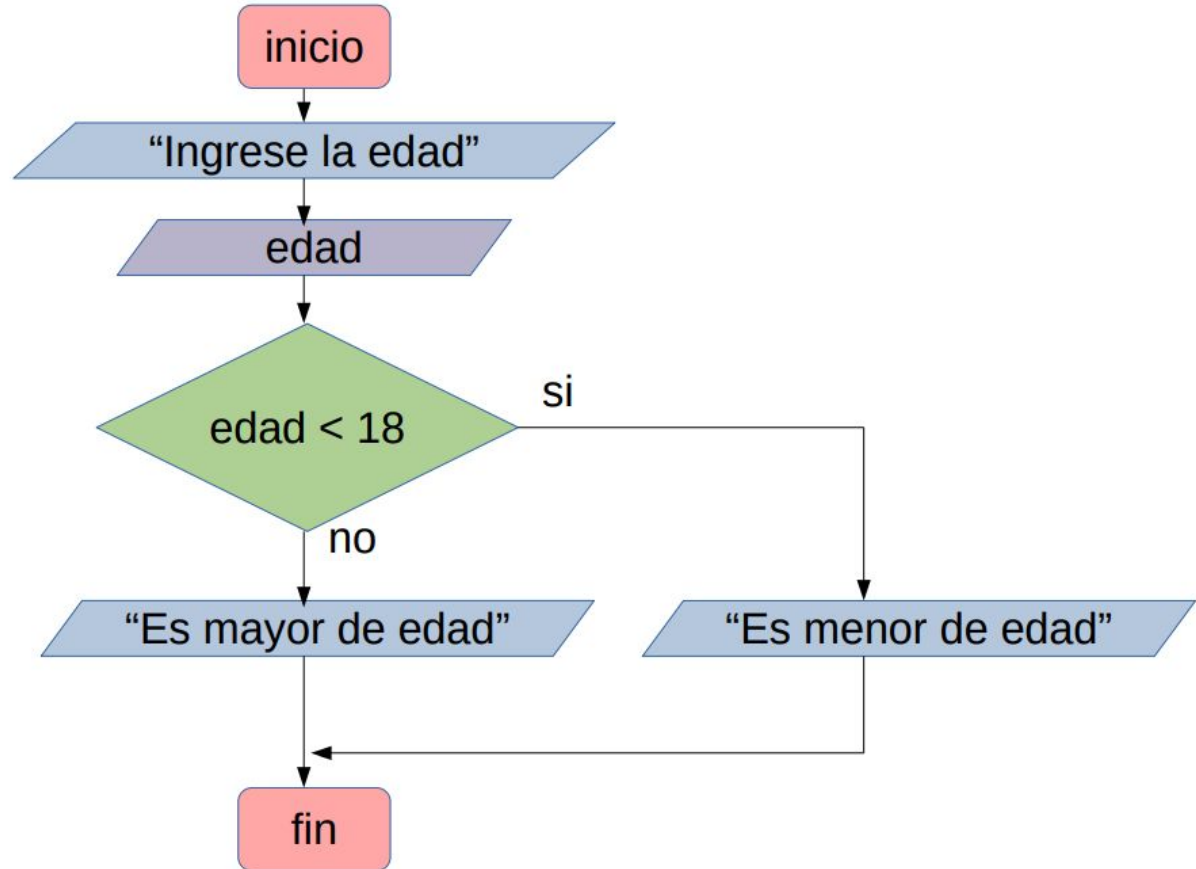
Técnicas de Programación

Secuencia de sentencias



Técnicas de Programación

Decisión / Selección



Técnicas de Programación

Ciclo,
repetición o
iteración

