http://www.robertsallent.com
@robertsallent

v.21.11

JS04: flujo de programa

Introducción a las estructuras de control

Robert Sallent
www.robertsallent.com

SOC Servei d'Ocupació de Catalunya
CIFO Sabadell- Mancomunitat (Vallès)
Generalitat de Catalunya
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

Índice

- Flujo de programa
- Problema de ejemplo
- Diagramas de flujo
- Elementos del diagrama de flujo.
- Ejemplos
- Tres clases de estructuras

- Ejercicios

Flujo de programa

- El flujo de programa hace referencia al **orden en el que se ejecutan las instrucciones**.

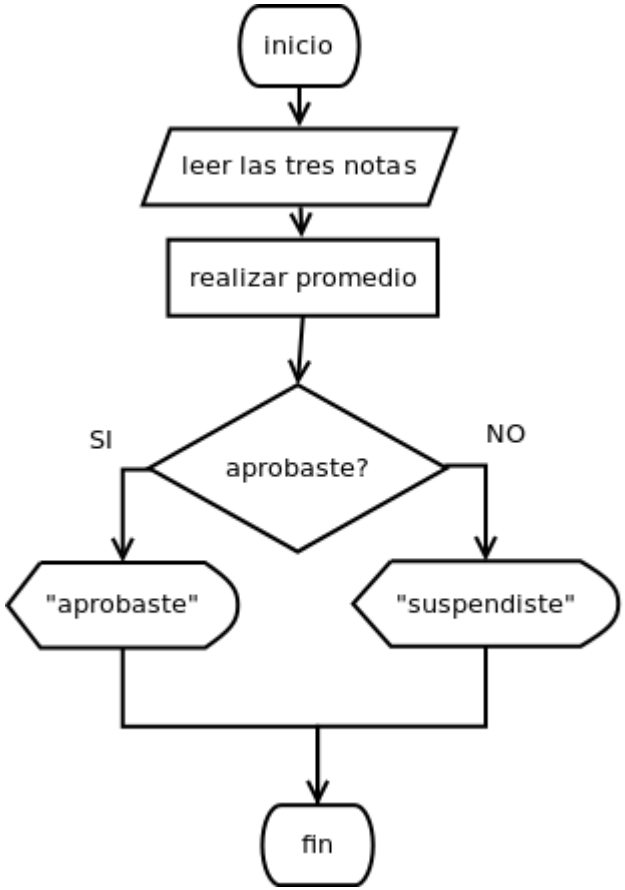
- Por defecto, el flujo es **secuencial**, es decir, se ejecutan una tras otra desde arriba hasta abajo.

- Como veremos estos días, el flujo de programa puede alterarse para permitir :

  - Ejecutar un grupo de instrucciones u otro (selección).
  - Ejecutar repetidamente un grupo de instrucciones (iteración).

Flujo de programa

- En el diagrama de flujo mostrado a la derecha, la evolución del programa se muestra mediante flechas.
- Cada operación se representa mediante un bloque, que tiene una forma distinta en función del tipo de operación.
- Este diagrama concreto tiene una parte secuencial y una bifurcación.



Ejemplo

Análisis, diseño, implementación y prueba de un pequeño programa de ejemplo

Robert Sallent
www.robertsallent.com

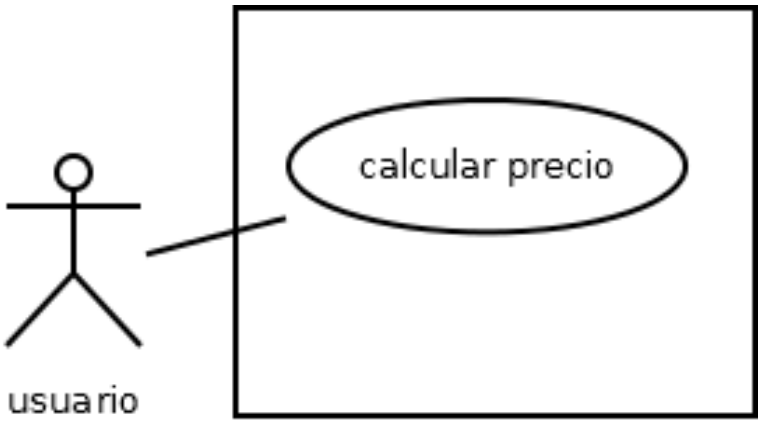
SOC Servei d'Ocupació de Catalunya
CIFO Sabadell- Mancomunitat (Vallès)
Generalitat de Catalunya
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

Problema de ejemplo

- Se nos plantea el siguiente problema:
  - Crear una aplicación web que solicite el precio de un producto y el tipo de *IVA* a aplicar, mostrando el precio final resultante.

Análisis y diseño

- En la fase de análisis, hemos determinada los siguientes usuarios y casos de uso (diagrama hecho con DIA):



Análisis y diseño

- También hemos determinado que la aplicación se ejecutará en el lado del cliente, así que las **tecnologías** a usar serán *HTML* y *CSS* para la presentación y *JavaScript* para el cálculo.

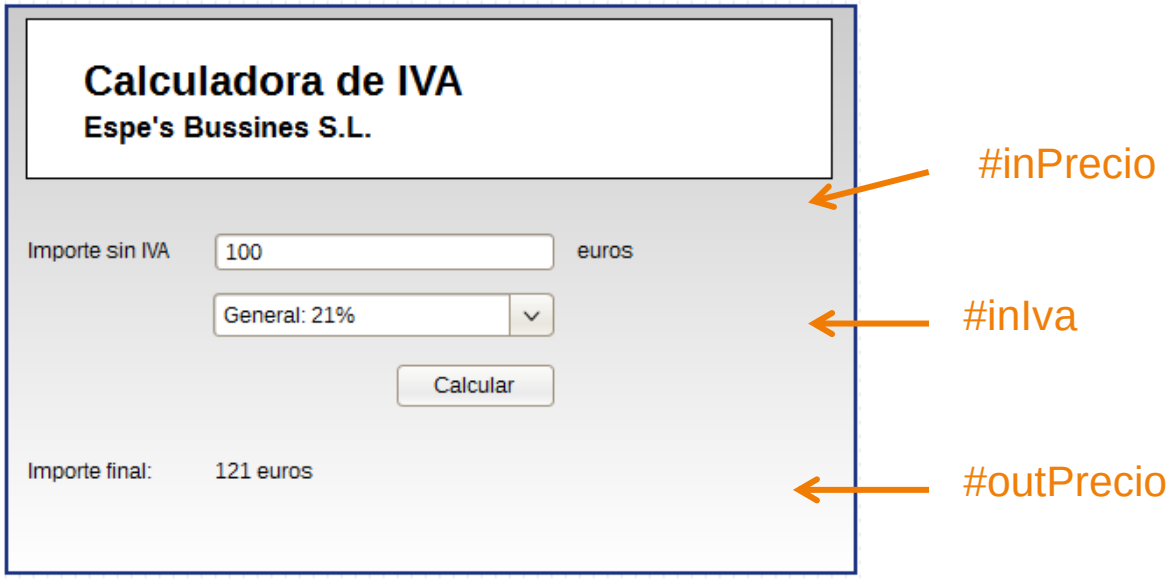
- La operación “calcular precio” consistirá en:

  - El usuario introduce el precio inicial y el tipo de IVA y presiona el botón.
  - El sistema le muestra el precio final.

- Por tanto la interfaz gráfica de usuario debe disponer de dos entradas (*input*) y una salida (*output*).

Diseño de la *GUI*

- La interfaz gráfica de usuario (*GUI*) puede ser así (el esquema se ha dibujado con la herramienta Pencil).



Diseño del programa

- Ahora pensemos en las operaciones que nuestro programa debe realizar para resolver el problema.

- El proceso es simple:

  - Tomar el precio inicial y el tipo de *IVA* desde el formulario.
  - Realizar los cálculos.
  - Mostrar el resultado.

- Esto que acabamos de describir en texto llano, en realidad es el **algoritmo** de resolución del problema que se nos plantea.

Pseudocódigo

- Un algoritmo se puede representar en ***pseudócódigo***, que es una descripción de alto nivel, compacta e informal del principio operativo de un programa:

```
inicio
    obtener el precio y el IVA
    calcular el nuevo precio
    mostrar el precio resultante
fin
```

Pseudocódigo

- El pseudocódigo permite describir algoritmos en un lenguaje humano simplificado que no es dependiente de ningún lenguaje de programación.

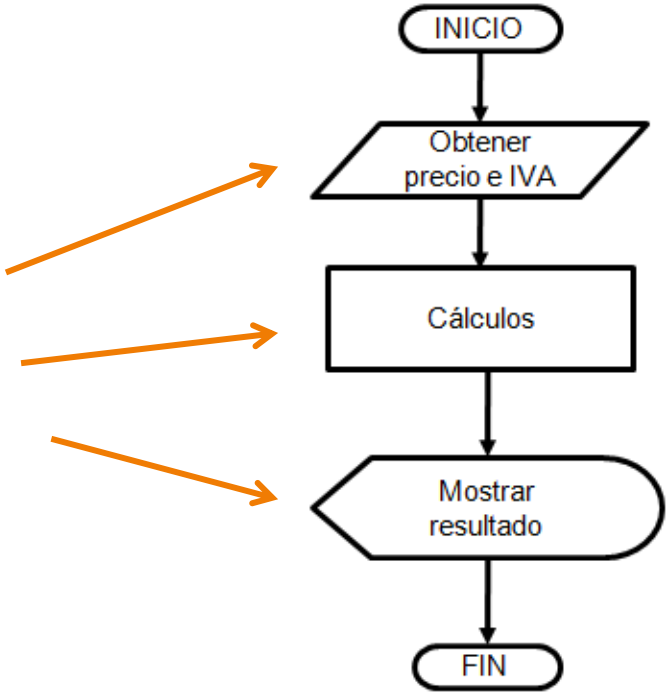
- Puede ser implementado en cualquier lenguaje por cualquier programador que lo utilice.

- Encontraréis mucha más información en el enlace: https://es.wikipedia.org/wiki/Pseudoc%C3%B3digo

Diagrama de flujo

- Pero el algoritmo anterior también se puede representar mediante un **diagrama de flujo**:

```
inicio
  obtener el precio y el IVA
  calcular el nuevo precio
  mostrar el precio resultante
fin
```

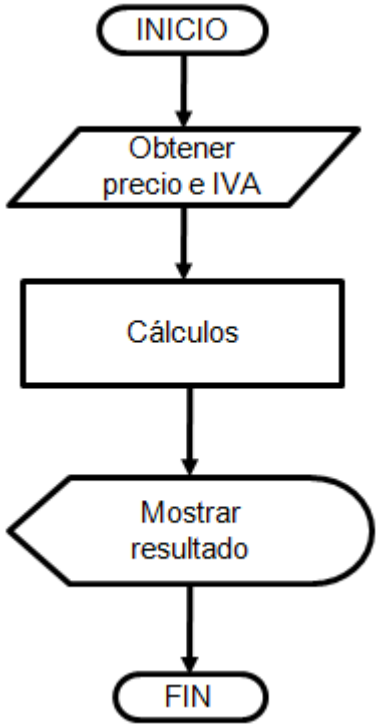


Estructura de bloques

- Observa que en el diagrama de bloques hemos dibujado tres bloques, cada uno para un objetivo distinto.

- Cada una de las tres tareas implicará seguro una o más instrucciones de programación.

- Cuando lo implementemos, separaré las instrucciones de cada uno de estos **bloques lógicos** con un doble salto de línea para hacer el programa más legible.



Estructura de bloques

- Que un programa tenga una **estructura de bloques** significa que está formado por un conjunto de grupos de instrucciones, en lugar de una lista continua de instrucciones.

- Todos los bloques tienen un único punto de entrada, al comienzo de los mismos y un único punto de salida al final.

- Cuando el computador finaliza la ejecución de un bloque, continúa con otro o finaliza.

Estructura de bloques

- Es importante no utilizar instrucciones de salto tipo `GOTO` al programar, puesto que rompen por completo la organización de los bloques y el concepto de ámbitos.

- Esto se hacía hace muchos años en *scripts* o lenguajes de bajo nivel. Hacen que sea más probable la aparición de errores y dificultan la depuración.

Implementación

- Vamos a implementar ahora el programa que hemos analizado y diseñado anteriormente.

  - La interfaz gráfica de usuario será implementada con *HTML* y *CSS* siguiendo el diseño que se marcó y que ha debido ser aceptado por el cliente.
  - La programación se hará con *JavaScript*, implementado el algoritmo que hemos descrito.

Ejemplo interfaz gráfica de usuario (*GUI*)

```
<h1>Calular precio</h1>

<span>Precio:</span>
<input type="number" min="0" step="0.01" id="inPrecio">
<br>
<span>IVA:</span>
<select id="inIva">
    <option value="1.21" selected>Normal (21%)</option>
    <option value="1.1">Reducido (10%)</option>
    <option value="1.04">Super reducido (4%)</option>
    <option value="1">Sin IVA (0%)</option>
</select>
<br>
<button type="button" onclick="calcular()">Calcular</button>
<span>Precio final:</span>
<output id="outPrecio">--</output>
<span> euros</span>
```

Ejemplo programación

```
<script>
    function calcular(){

        //leer el precio e IVA introducidos por el usuario
        var precio = parseFloat(inPrecio.value);
        var iva = parseFloat(inIva.value);

        //realzar los cálculos
        precio *= iva;

        //mostrar el resultado (aprox hasta los céntimos)
        outPrecio.innerHTML = precio.toFixed(2);
    }
</script>
```

Ejemplo resultado

- Ya tenemos una primera versión de la aplicación, pero antes de ponerla en producción, debemos testear que todo funcione correctamente.

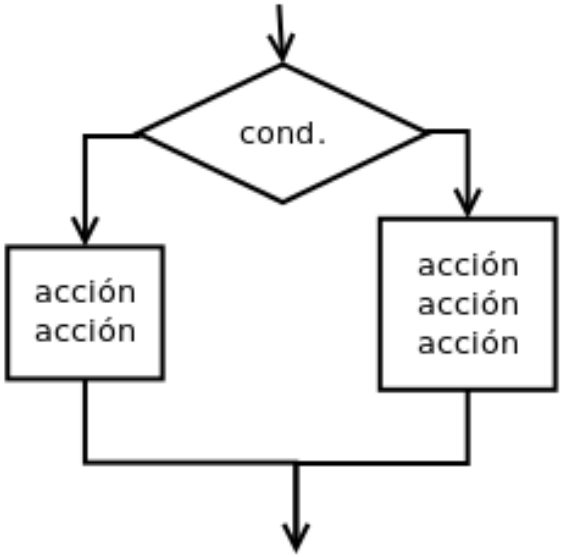
Consideraciones

- En el ejemplo anterior teníamos un flujo de programa **secuencial**, sin bifurcaciones ni iteraciones.

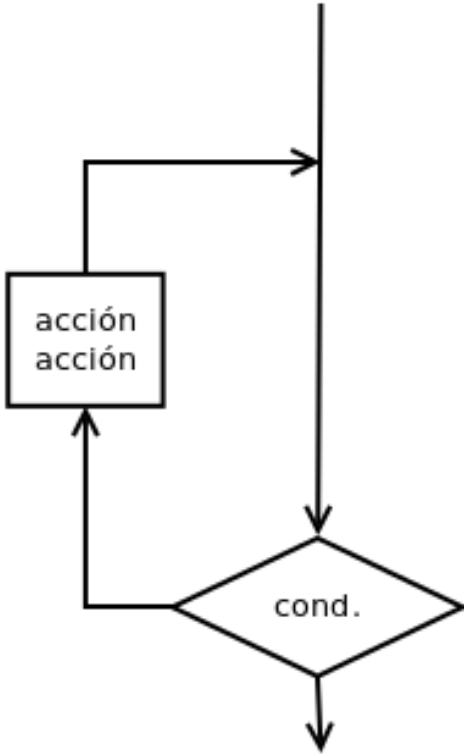
- En las próximas presentaciones veremos ejemplos de algoritmos más complejos y diagramas que incluirán estas estructuras.

Estructuras de control

Secuencial



Selección o bifurcación



Bucle o iteración

Diagrama de flujo

Características del diagrama de flujo, ejemplos y usos

Robert Sallent
www.robertsallent.com

SOC Servei d'Ocupació de Catalunya
CIFO Sabadell- Mancomunitat (Vallès)
Generalitat de Catalunya
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

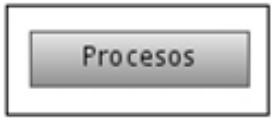
Diagrama de flujo

- Un **diagrama de flujo** es una representación gráfica de un algoritmo.

  - Se utiliza en disciplinas como la programación, la economía, los procesos industriales y la psicología cognitiva.
  - Utiliza símbolos con significados bien definidos que representan los pasos del algoritmo.
  - El flujo de ejecución se representa mediante flechas que conectan los puntos de inicio y de término.

Diagrama de flujo

Inicio o fin del programa



Pasos, procesos o líneas de instruccion de programa de computo

Operaciones de entrada y salida

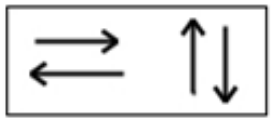
Toma de desiciónes y Ramificación

Conector para unir el flujo a otra parte del diagrama

Diagrama de flujo

Cinta magnética

Disco magnético



Líneas de flujo

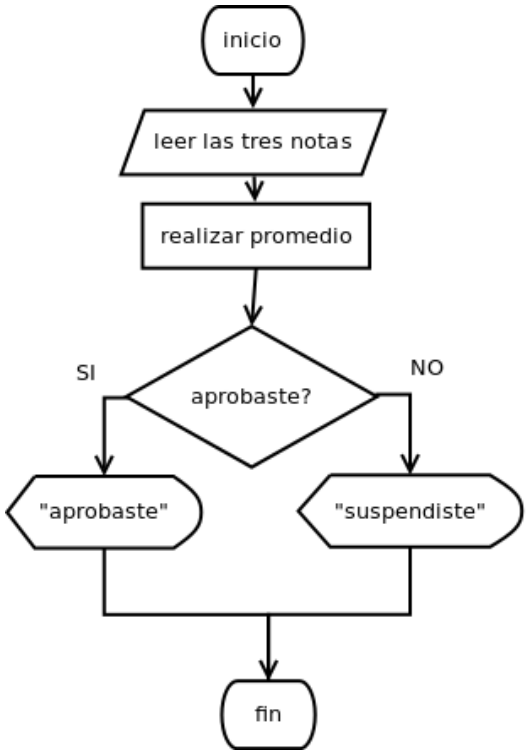
Anotación

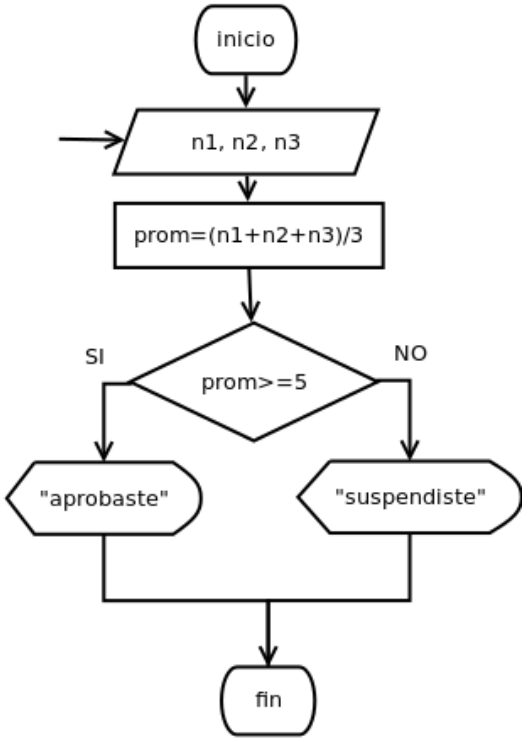
Display, para mostrar datos

Envía datos a la impresora

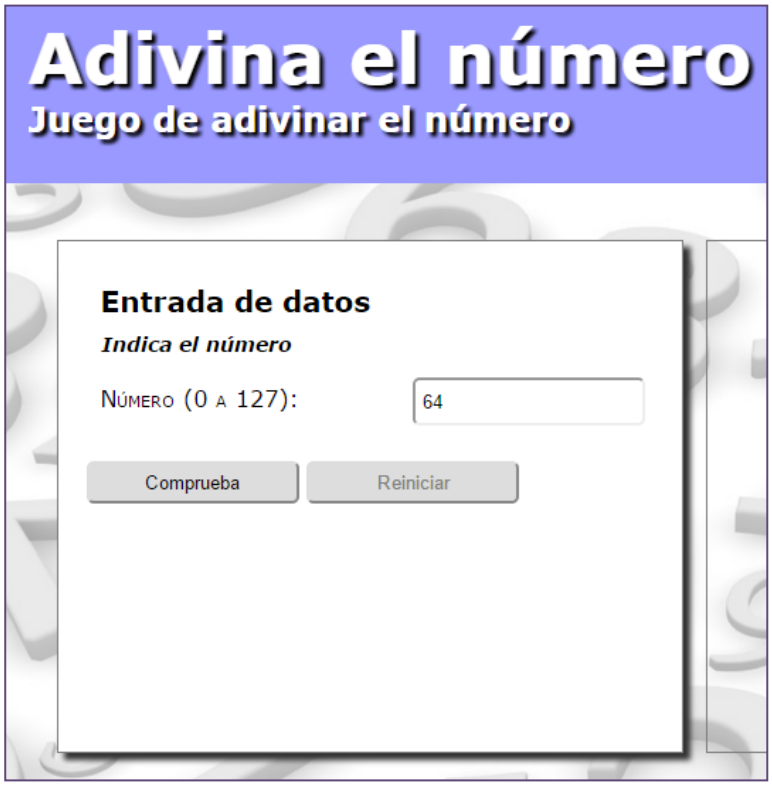
Diagrama de flujo

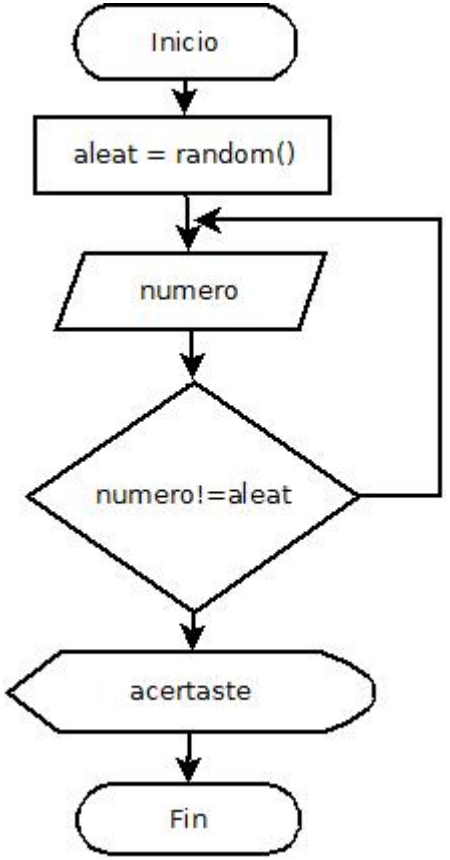
- Los diagramas de flujo se pueden representar a varios niveles:



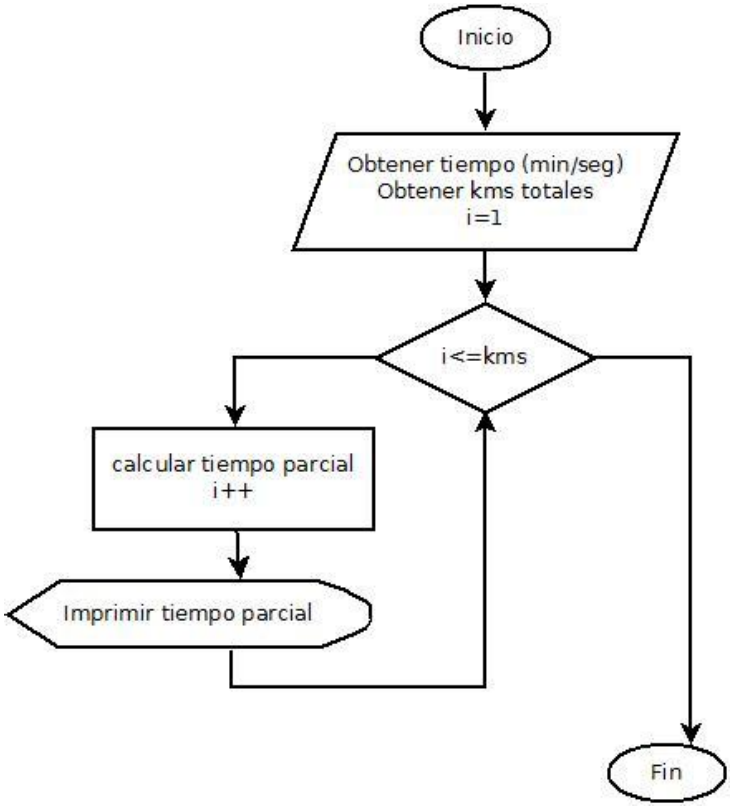


Ejemplos


Adivina el número
Juego de adivinar el número
Entrada de datos
Indica el número
Número (0 a 127):
64
Comprueba
Reiniciar


Inicio
aleat = random()
numero
numero!=aleat
acertaste
Fin

Ejemplos


Inicio
Obtener tiempo (min/seg)
Obtener kms totales
i=1
i<=kms
calcular tiempo parcial
i++
Imprimir tiempo parcial
Fin

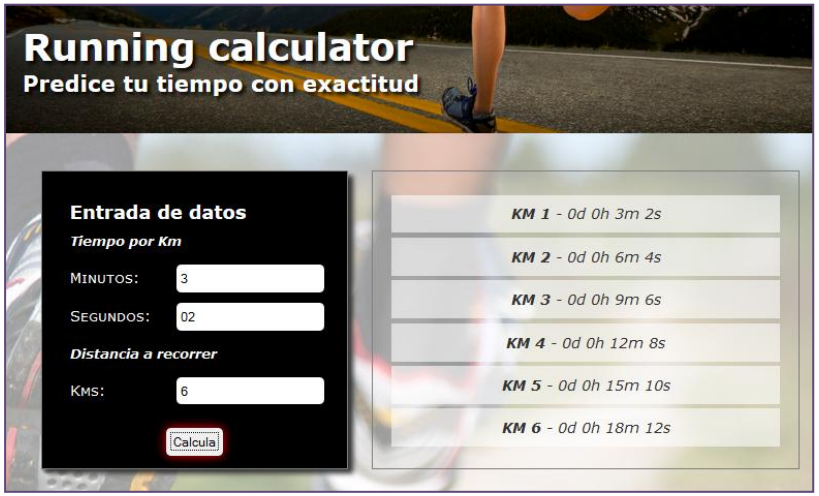

Running calculator
Predice tu tiempo con exactitud
Entrada de datos
Tiempo por Km
MINUTOS: 3
SEGUNDOS: 02
Distancia a recorrer
KMS: 6
Calcula
KM 1 - 0d 0h 3m 2s
KM 2 - 0d 0h 6m 4s
KM 3 - 0d 0h 9m 6s
KM 4 - 0d 0h 12m 8s
KM 5 - 0d 0h 15m 10s
KM 6 - 0d 0h 18m 12s

Diagrama de flujo: DIA

- Una herramienta gratuita para crear diagramas de flujo (entre otros diagramas útiles) y que está disponible para Windows y Linux es el programa **DIA**.

- Es totalmente gratuita y la podéis descargar para Windows desde: http://sourceforge.net/projects/dia-installer/ (en Ubuntu la encontraréis en el centro de software).

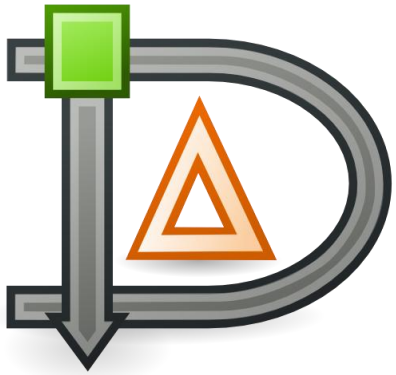
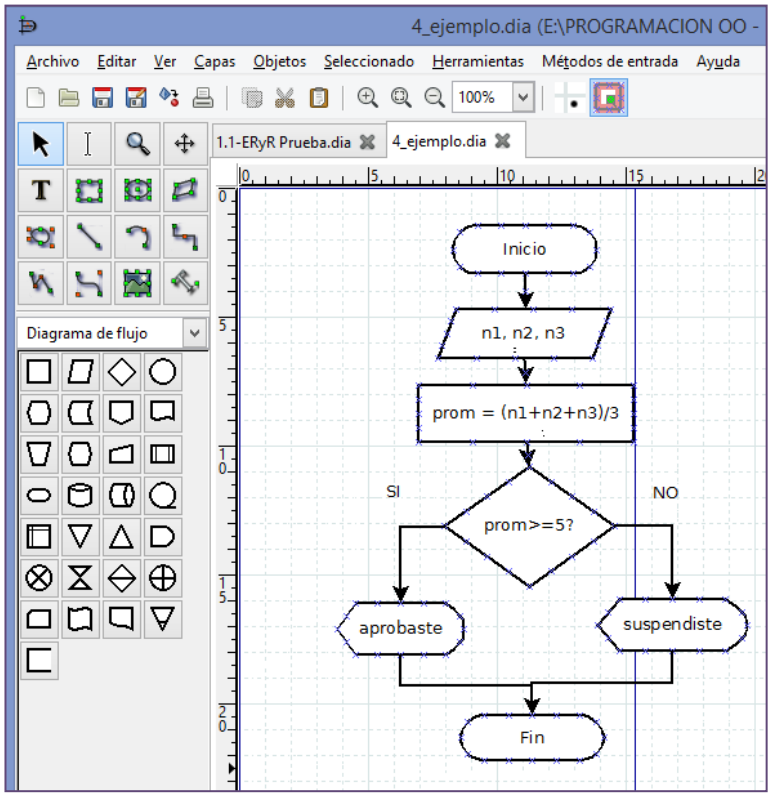
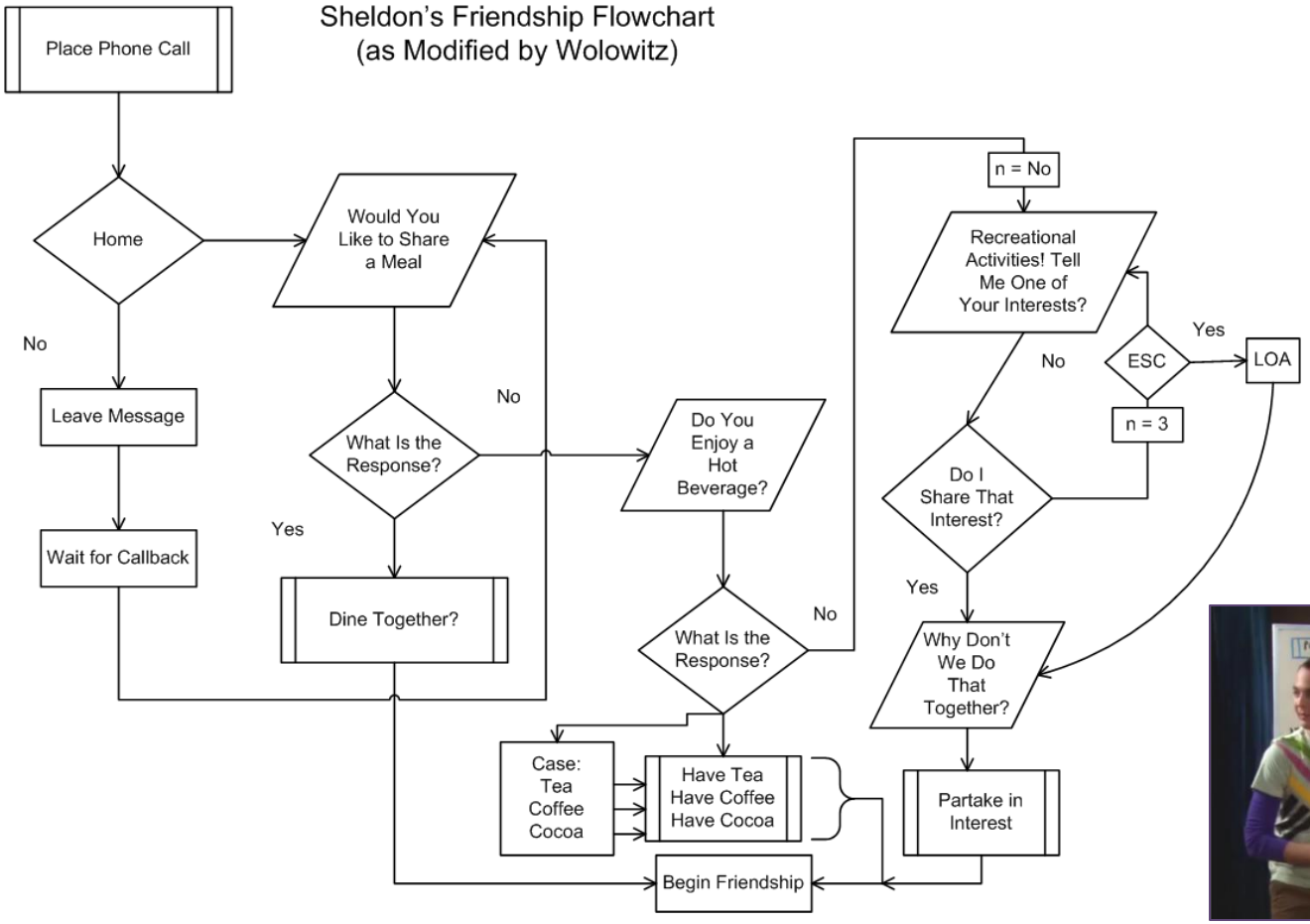


Diagrama de flujo: DIA



Ejemplo


Sheldon's Friendship Flowchart
(as Modified by Wolowitz)
Place Phone Call
Home
No
Leave Message
Wait for Callback
Would You Like to Share a Meal
What Is the Response?
No
Yes
Dine Together?
Do You Enjoy a Hot Beverage?
What Is the Response?
No
Yes
Case: Tea Coffee Cocoa
Have Tea Have Coffee Have Cocoa
Begin Friendship
n = No
Recreational Activities! Tell Me One of Your Interests?
No
Yes
ESC
LOA
n = 3
Do I Share That Interest?
Yes
Why Don't We Do That Together?
Partake in Interest

Ejercicios

Diagramas de flujo

Robert Sallent
www.robertsallent.com

SOC Servei d'Ocupació de Catalunya
CIFO Sabadell- Mancomunitat (Vallès)
Generalitat de Catalunya
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

Ejercicios

1. Dibuja el diagrama de flujo para un programa que te diga el número de años que han pasado entre dos años introducidos por teclado. No debe aparecer en ningún caso la diferencia con signo negativo.

2. Dibuja el diagrama de flujo para un programa que te diga el número de meses que han pasado desde dos fechas (mes y año) que se introducirán por teclado.

3. En parejas, debatir los posibles fallos de cada uno de los diseños. ¿Creéis que salió perfecto a la primera? ☺

Ejercicios

4. Con la solución dada en clase, realizad pruebas sobre papel, teniendo en cuenta que los meses irán de 0 para enero hasta 11 para diciembre. ¿Funciona para todos los casos, incluidos valores negativos para los años?

5. Implementa la aplicación y realiza de nuevo las pruebas que hiciste en papel. ¿Coinciden los resultados obtenidos con los esperados?.

6. Depura el programa como se explicará en clase, comprueba que las variables tienen en todo momento los valores esperados.