

Índice de Contenidos

- 1 Resolución de Problemas con Ordenadores
- 2 Lenguajes de Programación
 - Lenguajes de Programación: tipos
 - Paradigmas de programación
 - Elementos de nuestro lenguaje
- 3 Introducción a Processing
 - Primeras nociones de programación en Processing
 - Expresiones y su impresión en la consola
 - Concepto de Función
 - Ventana gráfica

① Resolución de Problemas con Ordenadores

② Lenguajes de Programación

Lenguajes de Programación: tipos

Paradigmas de programación

Elementos de nuestro lenguaje

③ Introducción a Processing

Primeras nociones de programación en Processing

Expresiones y su impresión en la consola

Concepto de Función

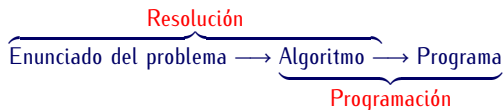
Ventana gráfica

Solución de problemas con programas

RECORDATORIO

Las fases del proceso de resolución de problemas con ordenadores son:

- **Fase de resolución del problema**
 - Fase cuyo objetivo es encontrar un **algoritmo**.
 - Enunciado → Análisis → Diseño → Verificación.
- **Fase de implementación/programación**
Fase en la que dado un **algoritmo** se programa éste en el ordenador.



- **Algoritmo**. Solución constructiva de un problema.

En estas transparencias nos centramos en la segunda fase.

Fase de implementación

- 1 **Codificación.** Escribir el **algoritmo** en un lenguaje de programación.
- 2 **Compilación**, para comprobar que no haya errores en el código fuente, y **ejecución**, para ver su comportamiento.
- 3 **Prueba y verificación.** A partir de una matriz de entrada establecidas y sus salidas esperadas se determina la validez de la solución propuesta al problema.
- 4 **Depuración**, para corregir los errores encontrados.
- 5 **Documentación.** Referenciar desde que se dio el problema hasta la solución encontrada.

A esta segunda fase también se le conoce como **Programación de la Solución**:

Definición (Programación)

La **programación** es un **proceso** mediante el cual creamos **programas** que son **ejecutados** por un sistema computacional, todo ello con el fin de **resolver** una tarea.

① Resolución de Problemas con Ordenadores

② Lenguajes de Programación

Lenguajes de Programación: tipos

Paradigmas de programación

Elementos de nuestro lenguaje

③ Introducción a Processing

Primeras nociones de programación en Processing

Expresiones y su impresión en la consola

Concepto de Función

Ventana gráfica

Programas

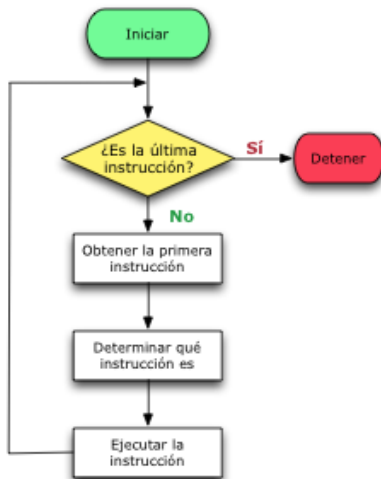
Programa

Un **programa** es el conjunto de **instrucciones** que se ejecutan en un **orden secuencial** y se expresan en un **lenguaje de programación** (que permite comunicar el ordenador con el humano).

Ciclo de instrucción

Un **ciclo de instrucción** es el período que tarda la **unidad central de proceso** (CPU) en ejecutar una instrucción de **lenguaje máquina**.

Ciclo de ejecución de un programa



Profundizamos en ...

① Resolución de Problemas con Ordenadores

② Lenguajes de Programación

Lenguajes de Programación: tipos

Paradigmas de programación

Elementos de nuestro lenguaje

③ Introducción a Processing

Primeras nociones de programación en Processing

Expresiones y su impresión en la consola

Concepto de Función

Ventana gráfica

Lenguajes de programación

Lenguaje de programación

Es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por computadoras.

- Los procesos se expresan mediante **instrucciones y sentencias** que deben de ser realizadas por el ordenador.
 - Están compuestos por una serie de **símbolos, reglas** sintácticas y semánticas que definen el lenguaje.
-
- Existen multitud de lenguajes de programación.
 - Tipos de lenguajes
 - Según su nivel de abstracción.
 - Según su traducción al "lenguaje del ordenador".
 - Según su tipado.

Lenguajes de programación

Según su nivel de abstracción

Lenguaje máquina o código máquina. Consta de 0's y 1's que representan los distintos voltajes eléctricos. Depende de la arquitectura de la máquina. Todos los programas se convierten en lenguaje máquina antes de ser ejecutados. Son ejecutables.

```
1 48 6F 6C 61 2C 20 65 73 74 65 20 65 ...
```

Lenguaje ensamblador. No son ejecutables. Se deben convertir a lenguaje máquina. Consta de instrucciones nemotécnicas que dependen de la arquitectura.

```
1 DW      64 DUP (?)
2 DB      "Hola mundo!!",13,10,"$"
3 ASSUME  CS:CODE, DS:DATA, SS:STACK
4 MOV     AX,DATA
5 MOV     AH,09H
6 INT     21H                                (Apariencia de las instrucciones)
```

Lenguaje de alto nivel. No son ejecutables y se deben convertir a lenguaje máquina. Son independientes de la arquitectura hardware donde se programe.

El **índice TIBOE** indica cuáles son los lenguajes más utilizados cada año.

```
1 echo "Hola Mundo"                        (Bash)
2 print "Hola Mundo";                      (Basic)
3 WriteLn('¡Hola, Mundo!');                (Pascal)
4 System.out.println("Hola Mundo");        (Java)
5 ...
```

Lenguajes de programación

Según su traducción a lenguaje máquina

Conceptos necesarios:

Código fuente : programa expresado con instrucciones de un lenguaje de alto nivel.

Código objeto : programa expresado con las instrucciones y datos en un lenguaje diferente al inicial: código máquina(binario) o bytecode.

Código intermedio o bytecode : es un programa expresado con un código más abstracto que el código máquina (para reducir la dependencia respecto del hardware).

Código ejecutable : es un archivo que al ser invocado por el usuario realiza un proceso (un algoritmo).

Todo lenguaje tiene que ser:

- o **compilado** si es traducido a lenguaje máquina antes de ser ejecutado. Compiladores, enlazadores y ensambladores.
- o **interpretado** si es ejecutado paso a paso su código fuente. Es usual utilizar código intermedio (Java, Lua, C#, Python, Perl, Lisp, ...).

Lenguajes de programación

Según su tipado

Las **variables** permiten almacenar datos en la memoria del ordenador. Todo lenguaje usa **sistema de tipos**: clasifica los tipos de datos y cómo se manipulan.

Lenguajes sin tipo. La mayoría de los ensambladores (permiten que cualquier operación se aplique a cualquier dato, pues son secuencias de bits).

Lenguajes tipados. Si para poder realizar una operación con datos se ha tenido que definir previamente los tipos de datos para los cuales es aplicable.

Ej: para hacer $a + b$, a y b , ¿son reales?, ¿enteros?, ¿texto?

- Los **fuertemente tipados**, Java, son los que no permiten violaciones sobre las operaciones con un tipo de dato.
- Los de **tipado débil**, JavaScript, son los que permiten que un valor de un tipo pueda ser tratado como de otro tipo.
- En los **estáticamente tipado**, C#, cada variable que se quiera utilizar deberá ser declarada con un tipo.
- En los **dinámicamente tipado**, Python, las variables empiezan a existir cuando se les da valor y en tiempo de ejecución pueden cambiar su tipo.

Profundizamos en ...

① Resolución de Problemas con Ordenadores

② Lenguajes de Programación

Lenguajes de Programación: tipos

Paradigmas de programación

Elementos de nuestro lenguaje

③ Introducción a Processing

Primeras nociones de programación en Processing

Expresiones y su impresión en la consola

Concepto de Función

Ventana gráfica

Paradigmas de programación

Un **paradigma de programación** es un estilo de desarrollo de programas. Es decir, un modelo para resolver problemas computacionales.

Imperativo* o procedimental (por procedimientos): es el más usado en general. El que más se ajusta a definición de algoritmo. Se basa en dar instrucciones.

Declarativo. Opuesto al imperativo. Los programas describen los resultados esperados, declarando propiedades y reglas que deben cumplirse. No se indican explícitamente los pasos que deben de realizar para llegar a la solución. Ejemplos: los lenguajes lógicos, los lenguajes algebraicos, y los lenguajes funcionales.

Estructurada.* Orientada a trabajar con subrutinas (o funciones) y tres estructuras de control: secuencias, condicional e iterativa.

Modular*. Divide un programa en módulos. Es una evolución de la programación estructurada. Se basa en el análisis descendente (top-down) y refinamiento sucesivo.

Orientada a Objetos.** El concepto básico es el de objeto. Un objeto es un ente con propiedades y métodos. Se programa cada objeto y sus métodos indican cómo se comunican entre ellos.

Dirigido por eventos.? El flujo del programa está determinado por sucesos que ocurran tanto en el interior como en el exterior del sistema. Por ejemplo, un videojuego.

*, ISCyP; ?, complementario; **, POO.

Profundizamos en ...

① Resolución de Problemas con Ordenadores

② **Lenguajes de Programación**

Lenguajes de Programación: tipos

Paradigmas de programación

Elementos de nuestro lenguaje

③ Introducción a Processing

Primeras nociones de programación en Processing

Expresiones y su impresión en la consola

Concepto de Función

Ventana gráfica

Cómo se Expresarán Nuestros Programas

Instrucciones. Funciones. Librerías.

Si esto no lo entiendes ahora ... tranqui. Es el objetivo de la asignatura.

Instrucción. Responde a una acción que puede entender y ejecutar el lenguaje de programación:

- Expresiones. P.e. las aritméticas. `2 + 3 * 5;`
- Declaración y uso de variables. `int a=2;`

Estructuras. Determina qué instrucciones o estructuras deben realizarse.

- Sentencias de control.
`if (cierto) then { print("hola"); }`

Función o procedimiento. Es un conjunto de instrucciones (y funciones) que se agrupan bajo un nombre para realizar una tarea concreta.

- Funciones del lenguaje.
`print("hola");`
- Funciones de librería (o biblioteca).
`import Stat; mean([2,3,4]);`
- Funciones de usuario.
`real miSuma (a,b) { return a+b; }`

1 Resolución de Problemas con Ordenadores

2 Lenguajes de Programación

Lenguajes de Programación: tipos

Paradigmas de programación

Elementos de nuestro lenguaje

3 Introducción a Processing

Primeras nociones de programación en Processing

Expresiones y su impresión en la consola

Concepto de Función

Ventana gráfica

Processing

- Web: <https://www.processing.org/>
- Descarga: <https://www.processing.org/download/>
- Instalación: Descomprime el .zip y guardar donde se desee (p.e. escritorio).
- Lugar de trabajo: Preferencias > Sketchbook location
- Menú *File* para cargar, guardar y ejemplos.
- Menú *Sketch* para ejecutar y parar los programas.
- Algunas opciones de *File* y *Sketch* en barra de herramientas.

Profundizamos en ...

1 Resolución de Problemas con Ordenadores

2 Lenguajes de Programación

Lenguajes de Programación: tipos

Paradigmas de programación

Elementos de nuestros lenguaje

3 Introducción a Processing

Primeras nociones de programación en Processing

Expresiones y su impresión en la consola

Concepto de Función

Ventana gráfica

Expresiones y algunas Funciones

RECUERDA

Instrucción. Responde a una acción que puede entender y ejecutar el lenguaje de programación:

- **Expresiones.** P.e. las aritméticas. `2 + 3 * 5;`
- Declaración y uso de variables. `int a=2;`

Estructuras. Determina qué instrucciones o estructuras deben realizarse.

- Sentencias de control.
`if (cierto) then { print("hola"); }`

Función o procedimiento. Es un conjunto de instrucciones (y funciones) que se agrupan bajo un nombre para realizar una tarea concreta.

- **Funciones del lenguaje.**
`print("hola");`
- Funciones de librería (o biblioteca).
`import Stat; mean([2,3,4]);`
- Funciones de usuario.
`real miSuma (a,b) { return a+b; }`

Profundizamos en ...

1 Resolución de Problemas con Ordenadores

2 Lenguajes de Programación

Lenguajes de Programación: tipos

Paradigmas de programación

Elementos de nuestro lenguaje

3 Introducción a Processing

Primeras nociones de programación en Processing

Expresiones y su impresión en la consola

Concepto de Función

Ventana gráfica

Expresiones aritméticas

A Processing hay que decirle todo. Necesitas la instrucción `println()`;

Operación	Aritmética	Programación
Suma	+	+
Resta	-	-
Multiplicación	×	*
División	÷	/
Resto	<i>mod</i>	%

Números	Aritmética	Programación
Enteros	..., -2, -1, 0, 1, 2,, -2, -1, 0, 1, 2, ...
Decimales	, (coma)	. (punto)
Reales	π	3.1415 Una aproximación. Punto flotante.

Ejemplo.

```
1 2+3 ✗    2+3; ✗    println(2+3); ✓
2 println(2,5); ✗    println(2.5); ✓
3 println(2/3); ✓    println(2.0/3.0); ✓
4 // Es diferente a 2/3 que 2.0/3.0
```



Funciones matemáticas

Processing	Matemáticas	Significado
<i>abs(x)</i>	$ x $	Valor absoluto
<i>ceil(x)</i>	$\lceil x \rceil$	Entero más cercano superior
<i>exp(x)</i>	e^x	Exponencial en base e
<i>floor(x)</i>	$\lfloor x \rfloor$	Entero más cercano inferior
<i>log(x)</i>	$\ln(x)$	Logaritmo en base e
<i>pow(x, y)</i>	x^y	Potencia y en base x
<i>round(x)</i>	$x \approx x'$	Redondeo de x es x'
<i>sq(x)</i>	x^2	Cuadrado de x
<i>sqrt(x)</i>	\sqrt{x}	Raíz cuadrada de x

Invocación	Matemáticas	Expresión
Sencilla	$\text{sen}(3)$	<code>sin(3);</code>
Composición	$\ln(\sqrt{ -3 })$	<code>log(sqrt(abs(-3)));</code>

Funciones trigonométricas y asociadas

Processing	Matemáticas	Significado
<i>acos(x)</i>	$\cos^{-1}(x)$	Arco coseno de x
<i>asin(x)</i>	$\sen^{-1}(x)$	Arco seno de x
<i>atan(x)</i>	$tg^{-1}(x)$	Arco tangente de x
<i>atan2(y, x)</i>	No existe	Arco tangente de (x, y)
<i>cos(x)</i>	$\cos(x)$	Coseno de x
<i>degrees(x)</i>	No existe	Convierte radianes a grados sexagesimales.
<i>radians(x)</i>	No existe	Convierte grados sexagesimales a radianes.
<i>sin(x)</i>	$\sen(x)$	Seno de x
<i>tan(x)</i>	$tg(x)$	Tangente de x

Ejemplo.

```
1 // Calculamos el seno de 90
2 println(sin(90)); ✘          println(sin(radians(90))); ✔
3 println(sin(3.1415/2)); ✔    println(sin(atan2(1,0))); ✔
```



Evaluación de expresiones matemáticas

Precedencia de operadores

- 1 Paréntesis: (expresión)
- 2 Funciones matemáticas.
- 3 Productos y divisiones: *,/,%.
- 4 Suma y resta. +,-.

En caso de empate precede el situado más a la izquierda.

EJERCICIO 1

Evalúa las siguientes expresiones

- $2 * 4 + 3 * 5$
- $2 + 4 * 3 + 5$
- $2 * 4/3 * 5$
- $1/2.0 + (abs(2 - 4) + 5) * (sin(3.1415) + 2)$



Expresiones alfanuméricas

- Expresiones alfanuméricas es una secuencia de caracteres imprimibles (visibles o no) delimitados por el símbolo `"`.
- Los símbolos delimitadores `"` no forman parte de la cadena.
- Incluye al alfabeto en mayúscula o minúscula.
- Incluye los caracteres de los números: 0, 1, ..., 9.
- No confundir `"2 + 2"` vs `2 + 2`
- Las secuencias de escape son caracteres especiales que constan de dos caracteres. El primero siempre es `\`.

Ejemplo: `"Hola:\nbip\t... saludos \"humanos\" ... bip"`, da como resultado

```
1 Hola:
2 bip      ... saludos "humanos" ... bip
```

Profundizamos en ...

1 Resolución de Problemas con Ordenadores

2 Lenguajes de Programación

Lenguajes de Programación: tipos

Paradigmas de programación

Elementos de nuestro lenguaje

3 Introducción a Processing

Primeras nociones de programación en Processing

Expresiones y su impresión en la consola

Concepto de Función

Ventana gráfica

Función y conceptos asociados

Función: Conjunto de instrucciones para realizar una tarea.

Parámetros: Conjunto de valores necesarios para realizar una función.

Argumentos: Conjunto de valores concretos para realizar una función.
Pueden ser funciones **anidadas**.

Valor de retorno: Valor devuelto por la función.

Ejemplo

- `print()` y `println()` son funciones.
- Admiten un parámetro, que puede ser numérico o alfanumérico.
- En `print("2+2=")`; el argumento es "2+2=" .
- En `print(2+2)`; el argumento es el resultado de la expresión numérica `2+2`.
- Estas funciones de impresión no retornan valores.
- `println(sin(2.1))` es un anidamiento de funciones.



Función destacada

```
1 void setup() {  
2   Instrucciones que tú quieras de Processing  
3 }
```

Profundizamos en ...

1 Resolución de Problemas con Ordenadores

2 Lenguajes de Programación

Lenguajes de Programación: tipos

Paradigmas de programación

Elementos de nuestro lenguaje

3 Introducción a Processing

Primeras nociones de programación en Processing

Expresiones y su impresión en la consola

Concepto de Función

Ventana gráfica

Ventana gráfica

`size(w,h);`

- Sus argumentos representan el ancho, `w`, y el alto, `h`, de la ventana gráfica medido en píxeles.
- Aparece **una sola vez** en todo el sketch y debe ser **la primera línea** del programa.
- El color del fondo viene dado por las instrucciones `background(ggris);` y `background(r, g, b);` cuyos argumentos toman valores entre 0 y 255 si el modo de color es RGB.

Ejemplo.

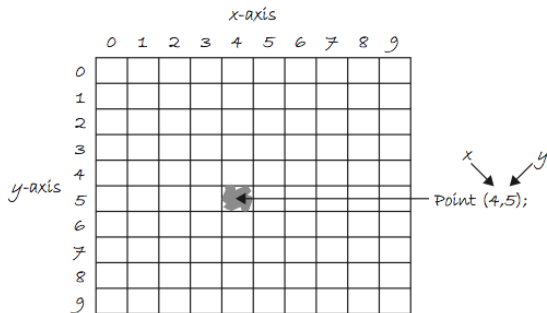
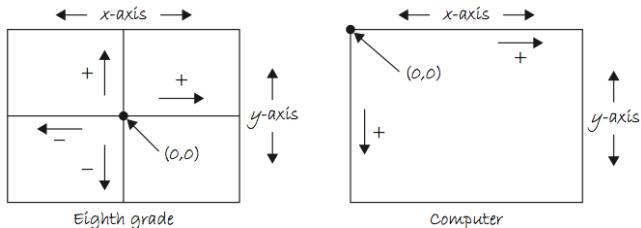
```
1 size(300,400);  
2 background(random(255), random(255), random(255));
```

Función `random(Max)`:

- Consta de un argumento, `Max`.
- Retorna un número aleatorio en `[0, Max)`.



Sistema de Coordenadas



Listado de funciones 2D

Figuras geométricas.

- `point(x,y);`
- `line(x1, y1, x2, y2);`
- `triangle(x1, y1, x2, y2, x3, y3);`
- `rect(a, b, c, d);`
- `quad(x1, y1, x2, y2, x3, y3, x4, y4);`
- `ellipse(a, b, c, d);`
- `arc(a, b, c, d, inicio, final);`
- `text(expresión, x, y);`

“Efectos”

- `fill (r, g, b, alpha);`
- `noFill();`
- `stroke(r, g, b, alpha);`
- `strokeWeight(grosor);`
- `noStroke();`

Cuidado: Aplicar `noFill();` y `noStroke();` no dibujará nada.

Otros

- `save("fichero");`

Ejemplos

Escribe el código necesario para hacer los siguientes dibujos. Después modifica el código para que contenga la función principal.

Ejemplo.

Dibuja un triángulo equilátero “perfecto” con el vértice superior en el punto $(200, 10)$ y su lado opuesto de 180 pixeles de largo.

