



Capítulo 1

¿ Qué es un programa?

1.1. 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.
- b) Se han creado proyectos de desarrollo de aplicaciones.
- c) Se han utilizado entornos integrados de desarrollo.

1.2. Definición

Un programa informático es un conjunto de instrucciones que una vez ejecutadas realizarán una o varias tareas en una computadora.

https://es.wikipedia.org/wiki/Programa_inform%C3%A1tico

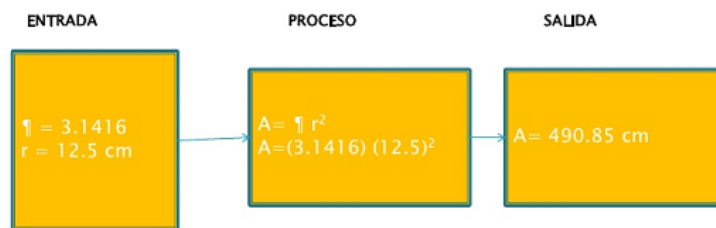
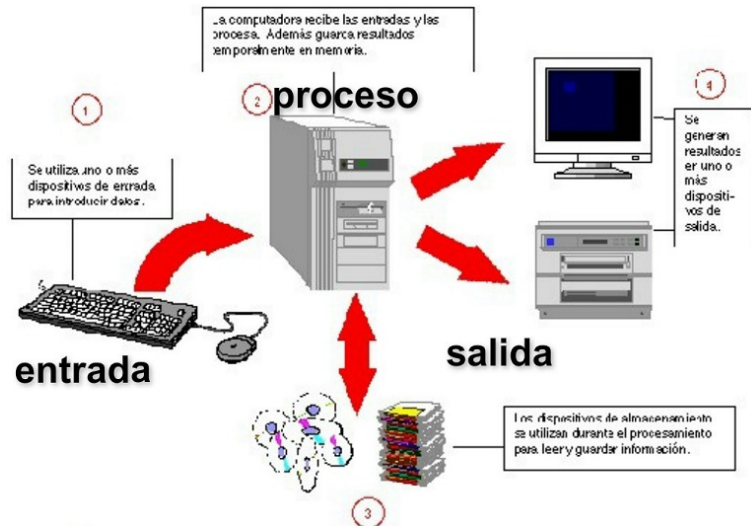


```
public class Mi_Clase_1
{
    public static void main(String args[])
    {
        for(int i = 0; i <= 12; i++)
        {
            System.out.print("12 * "+ i + " = " + 12 * i + "\n");
        }
    }
}
```

run:

```
12 * 0 = 0
12 * 1 = 12
12 * 2 = 24
12 * 3 = 36
12 * 4 = 48
12 * 5 = 60
12 * 6 = 72
12 * 7 = 84
12 * 8 = 96
12 * 9 = 108
12 * 10 = 120
12 * 11 = 132
12 * 12 = 144
GENERACION CORRECTA (total time: 1 second)
```

Un ordenador se podría definir como una máquina capaz de recoger información, procesarla y devolver unos resultados. A partir de una serie de instrucciones, el ordenador procesa automáticamente la información que le es suministrada.



Para ejecutar programas necesitamos **hardware** (componentes físicos del ordenador) y **software** (conjunto de programas que le indican que debe hacer).

En la parte lógica (software) podemos hablar del software de sistema (sistemas operativos, drivers, compiladores, ..) y de software de aplicación (programas de contabilidad, de tratamiento de textos, de gestión de bases de datos, juegos, ...).

Los pasos para crear software, que se verán con más detalle en la asignatura de entornos de desarrollo, son:

- **Planificación** (Conocer por encima la actividad que se quiere informatizar y plantear varias soluciones informáticas).
- **Análisis** (Consiste en conocer a fondo la actividad a informatizar).
- **Diseño del programa** (Diseñar todas las entradas y salidas que va a tener el programa e indicar todos los pasos que tiene que realizar un programa con el fin de realizar una determinada tarea (algoritmo).
- Transformación del algoritmo en un programa (**Codificación**).
- **Compilar, enlazar y depurar**.
- **Ejecución** y validación del programa.

Como ya hemos comentado, un programa es un conjunto de instrucciones que describen, definen o caracterizan la realización de una acción en un ordenador. Hoy en día, los programas se escriben usando "lenguajes de programación", que son a su vez otros programas que simplifican la tarea del programador ya que traen construidas secuencias y utilidades.

Un **lenguaje de programación** es un conjunto de reglas, símbolos y palabras reservadas utilizadas para construir un programa.

Las **instrucciones** son los componentes y la **secuencia** corresponde a la relación que hay entre ellos. A esto se le conoce como la visión algorítmica de un programa.

La **programación** es la acción de determinar la secuencia de instrucciones de un programa informático, con el fin de resolver un problema.

Las instrucciones de un programa para que sean comprendidas por la máquina deben ser sometidas a un proceso de traducción. Hay que pasar las instrucciones escritas en un lenguaje de programación cualquiera,

a una versión o código máquina que es lo que la computadora realmente entiende. Un **lenguaje compilado** es aquel cuyo código fuente, escrito en un lenguaje de alto nivel, es traducido por un compilador a un archivo ejecutable entendible para la máquina en determinada plataforma. Con ese archivo se puede ejecutar el programa cuantas veces sea necesario sin tener que repetir el proceso por lo que el tiempo de espera entre ejecución y ejecución es ínfimo. Un **lenguaje interpretado** es aquel en el cual sus instrucciones o más bien el código fuente, escrito por el programador en un lenguaje de alto nivel, es traducido por el interprete a un lenguaje entendible para la máquina paso a paso, instrucción por instrucción. El proceso se repite cada vez que se ejecuta el programa en cuestión.

El software en general, debe cumplir ciertos requisitos para que sea bien considerado. En primer lugar verificar si el programa hace lo que verdaderamente debe realizar. Luego debe ser claro y legible y de esta manera permitir un fácil mantenimiento y actualización. Por último, debe ser económico en el sentido de consumir la menor cantidad de recursos posibles, es decir, tiempo y memoria.

Un **entorno de desarrollo integrado** o entorno de desarrollo interactivo, en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software. Normalmente, un IDE suele estar formado por un editor de código fuente, herramientas de construcción automáticas y un depurador.

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

En clase utilizaremos el lenguaje de programación compilado **JAVA** y el entorno de desarrollo **NETBEANS** que se puede descargar gratuitamente desde <https://netbeans.org/downloads/>.

1.3. Ejemplos

Inicio

Visualizar en pantalla HOLA MUNDO

Fin



Inicio

Visualizar un mensaje pidiendo un número.

Guardar el número.

Visualizar otro mensaje pidiendo un segundo número.

Capturar el segundo número.

Sumar los dos números y guardar el resultado.

Visualizar el resultado de la suma.

Fin

```
1
2 // Programa que muestra en pantalla Hola Mundo
3 package ejemplounot1;
4
5 public class EjemploUnoT1 {
6
7     public static void main(String[] args) {
8
9         // Sentencia que permite la salida de datos en la consola
10        System.out.println("Hola mundo");
11    }
12 }
```

```
1
2 package ejemplodost1;
3
4 import javax.swing.*;
5
6 public class EjemploDosT1 {
7
8     public static void main(String[] args) {
9
10        // Salida de datos con un cuadro de diálogo
11        JOptionPane.showMessageDialog(null, "Hola mundo");
12    }
13 }
```

```
1 import javax.swing.JOptionPane;
2
3 public class EjemploTresT1 {
4
5     public static void main(String[] args) {
6         try
7         {
8             //Entrada
9             System.out.println("Teclea un carácter");
10            char c = (char) System.in.read();
11
12            //Salida
13            System.out.println("Has tecleado el carácter " + c);
14        }
15        catch(Exception e){
16            //Errores
17            JOptionPane.showMessageDialog(null, "Problemas");
18        }
19    }
20 }
```

```
1 package ejemplounot1;
2
3 import java.io.*;
4
5 public class EjemploUnoT1 {
6
7     public static void main(String[] args) {
8         try{
9
```

```
10      // Variables para guardar datos
11      String nombre;
12      int horas;
13      double pagoPorHora, pagoTotal;
14
15      // Entrada de datos
16      Scanner teclado = new Scanner(System.in);
17
18      System.println.out("¿ Cómo te llamas?");
19      nombre = teclado.nextLine();
20      System.println.out(" ¿ Cuántas horas has trabajado esta
      semana?");
21      horas = teclado.nextInt();
22      System.println.out(" ¿ Cuánto te pagan por hora?");
23      pagoPorHora = teclado.nextDouble();
24
25      // Proceso
26      pagoTotal = horas * pagoPorHora;
27
28      // Salida
29      System.out.println("Hola " + nombre);
30      System.out.println("Tu sueldo es " + pagoTotal);
31  }
32  catch(Exception e){
33
34      // Errores
35      System.out.println("Problemas");
36  }
37  }
38 }
```

```
1  package ejemplocincotl;
2
3  import javax.swing.*;
4
5  public class EjemploCincoTl {
6
7      public static void main(String[] args) {
8
9          String nombre;
10         nombre=JOptionPane.showInputDialog("¿ Quién eres?");
11
12         JOptionPane.showMessageDialog(null, "Bienvenid@ " + nombre);
13     }
14 }
```

1.4. Java

Java es un lenguaje de programación para desarrollar aplicaciones.

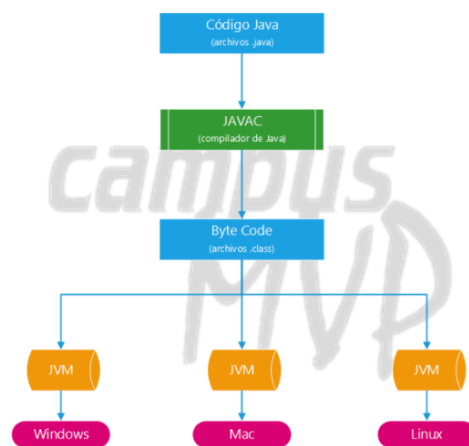
Java es un conjunto (muy grande) de código disponible para su uso.

Java es una plataforma donde ejecutar aplicaciones.

Java es un lenguaje de programación orientado a objetos creado en 1991 y publicado en 1995 por Sun Microsystem (adquirida por Oracle en 2010), con la intención de que los programadores escribieran el código solo una vez y lo ejecutarán en cualquier dispositivo. Esto es posible gracias a que Java cuenta con una JVM o Java Virtual Machine que brinda portabilidad al lenguaje, ya que hoy existen JVMs para diferentes arquitecturas para todas las plataformas.

WORA - write once, run anywhere

La máquina virtual de Java es un entorno virtual de ejecución de aplicaciones, que está disponible para múltiples arquitecturas, dispositivos y sistemas operativos



El JDK (java development kit) es el entorno para crear y ejecutar aplicaciones java. Este JDK posee un compilador que toma nuestro código Java y valida la sintaxis. Si el compilador encuentra algún error en nuestro código nos mostrará un mensaje (errores de compilación) y si todo está bien el compilador nos creará un archivo con código bytecode. Este archivo será ejecutado por la máquina virtual java. Eljdk se puede descargar desde <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

JRE (Java runtime environment) entorno para ejecutar aplicaciones java.

El hecho de que el código java pueda ser ejecutado en cualquier sistema gracias a la máquina virtual hace que se esté utilizando en millones de dispositi-

vos <https://www.campusmvp.es/recursos/post/es-cierto-que-3-mil-millones-de-dispositivos-usan-java>
aspx

El 17 de septiembre de 2019 se publicará la que será la versión 13 de Java siguiendo el nuevo calendario de publicar una nueva versión cada seis meses. <http://openjdk.java.net/projects/jdk/>

1.5. Entornos de desarrollo



<https://www.campusmvp.es/recursos/post/Los-mejores-entornos-de-desarrollo-para-Java>
aspx