

# Programación

## Introducción a la programación

# Programación

## Índice

- Introducción.
- Formas de resolver problemas. Algoritmos y heurísticas.
- El software de programación. Los lenguajes de programación.
- Clasificación de los lenguajes de programación.
  - Lenguajes de bajo nivel.
    - Lenguaje máquina.
    - Lenguaje ensamblador.
  - Lenguajes de alto nivel.
- Compiladores e intérpretes.
- Los entornos de desarrollo integrados (IDE).

# Programación

## Introducción

# Programación

## Introducción

La **información** es todo aquello que permite adquirir cualquier tipo de conocimiento.

Una información consta de una serie de datos. En ocasiones, estos datos, pueden ser utilizados directamente, y en otras, hay que realizar con ellos una serie de operaciones para obtener un resultado. Al conjunto de operaciones que se realizan sobre una cierta información con objeto de conseguir determinados resultados, se le denomina **tratamiento de la información**.

La **informática** es el conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores.

# Programación

## Introducción

Se puede definir el **ordenador** como una máquina universal capaz de interpretar y ejecutar una serie de operaciones elementales (instrucciones), relativas al tratamiento de la información, y resolver cualquier tipo de aplicación o tarea gracias a la utilización automática de programas registrados en ella

Un ordenador o computadora electrónica es una máquina que, mediante determinados programas, permite almacenar y tratar información, y resolver problemas de diversa índole.

Para que un ordenador pueda desarrollar determinadas tareas (mover el ratón, poner un texto en negrita, sumar dos números, etc.) hay que decirle lo que tiene que hacer. Al conjunto de instrucciones que dirigen a una computadora para que realice tareas específicas se conoce como programa.

# Programación

## Introducción

**Programa:** Conjunto unitario de instrucciones que permite a una computadora realizar funciones diversas, como el tratamiento de textos, el diseño de gráficos, la resolución de problemas matemáticos, el manejo de bancos de datos, etc.

Un programa sería una secuencia de instrucciones comprensibles por el ordenador que permiten realizar una determinada tarea con la finalidad de resolver un problema concreto

# Programación

## Introducción

Aunque es capaz de llevar a cabo tareas muy complejas, un ordenador sólo puede realizar una serie de operaciones simples y obedecer instrucciones en lenguaje máquina.

Una instrucción en lenguaje máquina no es más que una cadena de ceros y unos que expresan, de forma codificada, las únicas órdenes que un ordenador es capaz de entender. La potencia de un ordenador no radica tanto en la complejidad de las instrucciones que es capaz de ejecutar, que más bien son muy sencillas, sino en la gran velocidad de ejecución de las mismas.

Como trabajar en lenguaje máquina es muy engorroso para los humanos, se desarrollaron los lenguajes de programación para simplificar el proceso de escritura de instrucciones que las computadoras pueden utilizar.

Los lenguajes de programación presentan características comunes a todos ellos. Para que el ordenador entienda lo que se le dice en un lenguaje de programación hacen falta unos programas (compiladores e intérpretes) que traduzcan las instrucciones escritas en un determinado lenguaje de programación a lenguaje máquina.

# Programación

Formas de resolver problemas.  
Algoritmos y heurísticas



# Programación

## Formas de resolver problemas. Algoritmos y heurísticas

Formas de resolver problemas:

- Algoritmos.
- Heurísticas.

# Programación

## Formas de resolver problemas. Algoritmos y heurísticas

**Algoritmo:** secuencia ordenada y finita de acciones, exenta de ambigüedades y que puede realizar un ordenador y que lleva a la solución de un problema dado.

# Programación

## Formas de resolver problemas. Algoritmos y heurísticas

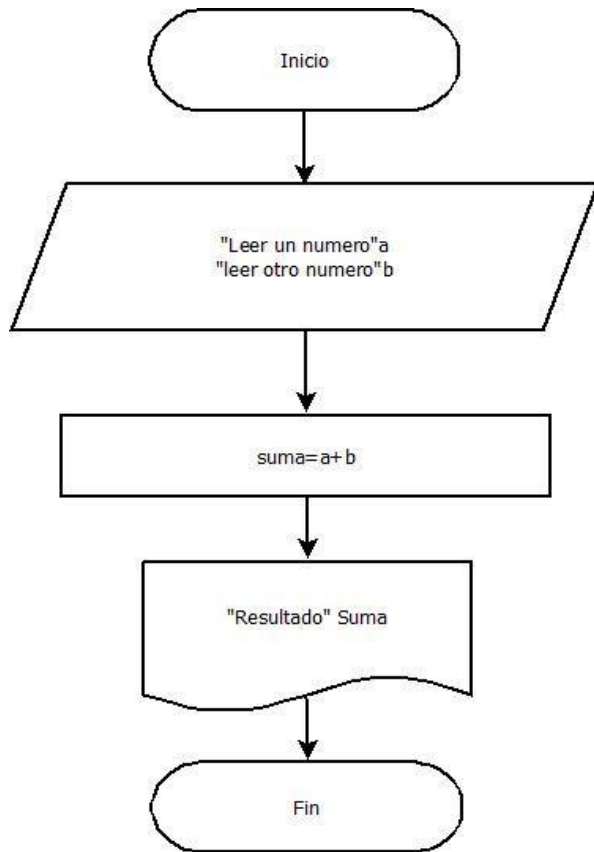


Diagrama de flujo y pseudocódigo correspondiente a un programa que calcula la suma de dos números introducidos por el usuario y muestra el resultado por pantalla.

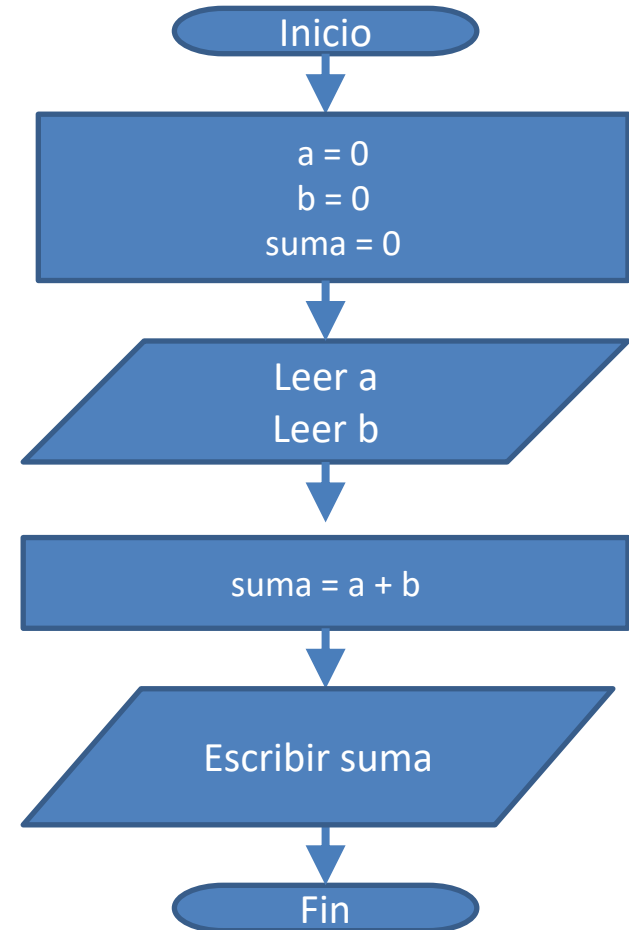
```
Inicio
    leer (a)
    leer (b)
    suma = a + b
    escribir (suma)
Fin
```

# Programación

## Diagramas de flujo y pseudocódigo (Ejercicio)

### Ejercicio:

Diseña un algoritmo que calcule la suma de dos números introducidos por el usuario

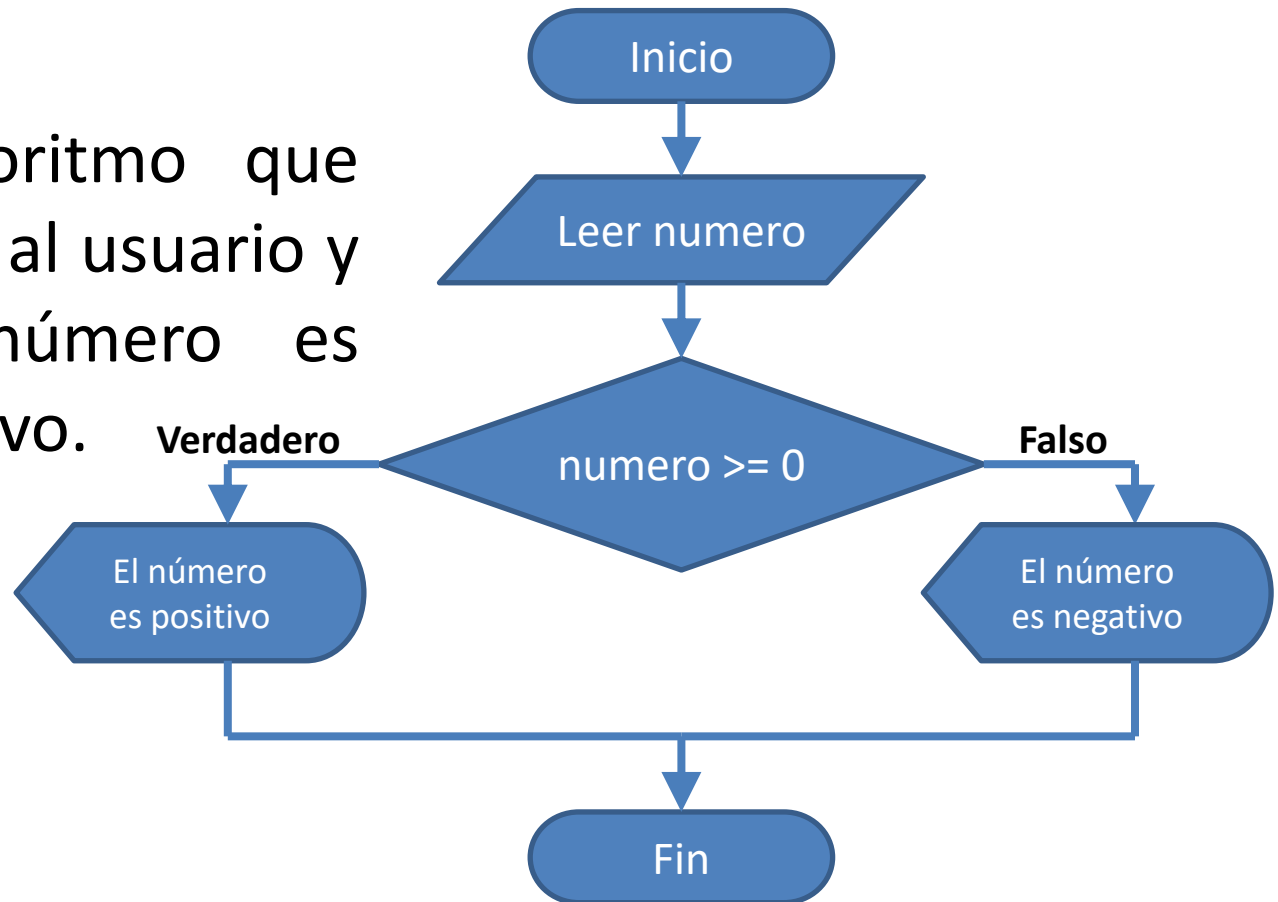


# Programación

## Diagramas de flujo y pseudocódigo (Ejercicio)

### Ejercicio:

Diseña un algoritmo que pida un número al usuario y diga si ese número es positivo o negativo.

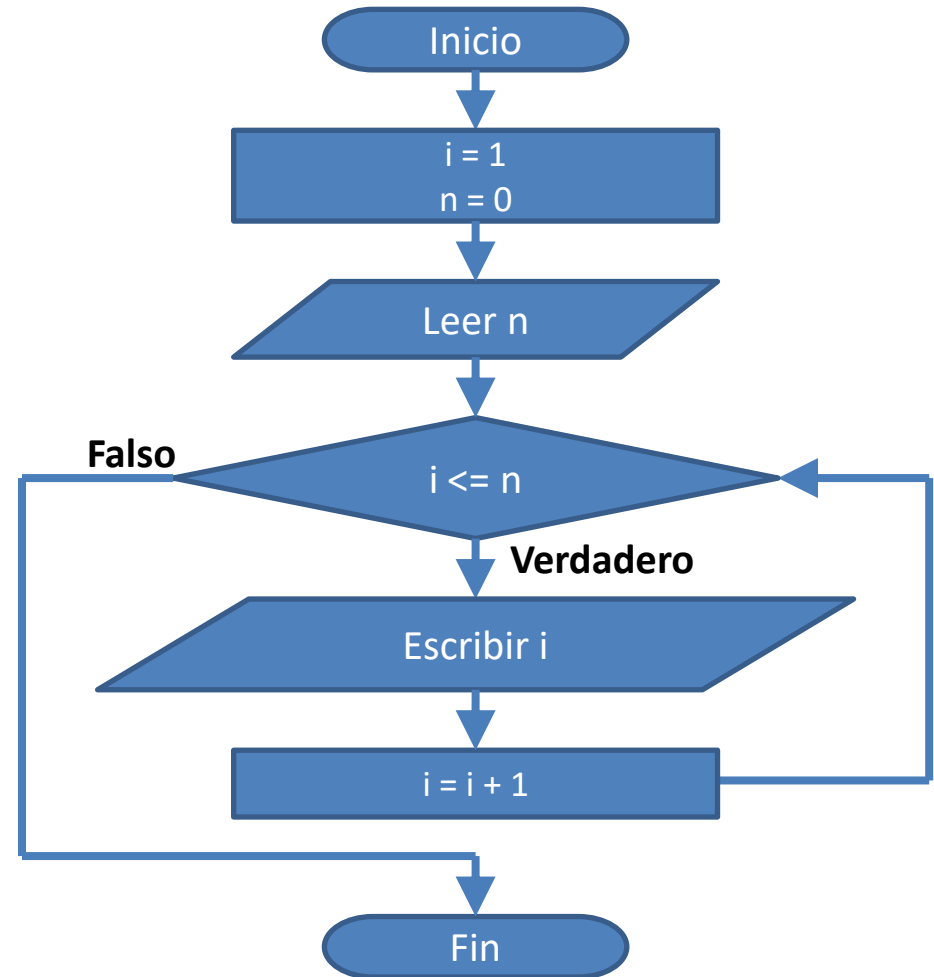


# Programación

## Diagramas de flujo y pseudocódigo (Ejercicio)

### Ejercicio:

Diseña un algoritmo que muestre los  $n$  primeros números naturales.



# Programación

El software de programación. Los  
lenguajes de programación

# Programación

## El software de programación. Los lenguajes de programación

El software se puede clasificar de acuerdo a las siguientes categorías:

- **Software del sistema:** el software del sistema incluye cualquier programa que controle el hardware de la computadora o que se puede utilizar para dar mantenimiento a la computadora de alguna forma que la haga funcionar con mayor eficiencia (sistemas operativos, herramientas, drivers, etc.).
- **Software de programación:** el software de programación incluye los programas que utilizan los programadores para crear nuevos programas (C, C++, Java, etc.).
- **Software de aplicación:** el software de aplicación resuelve problemas específicos de los usuarios. El software de aplicación le dice a la computadora la forma en que debe llevar a cabo tareas específicas para el usuario (procesadores de textos, hojas de cálculo, programas de retoque fotográfico, etc.).



# Programación

## El software de programación. Los lenguajes de programación

El **software de programación** incluye los programas que utilizan los programadores para crear nuevos programas. Los programas se crean utilizando un lenguaje de programación.

Un **lenguaje de programación** es un conjunto de palabras clave o instrucciones y unas reglas sintácticas que indican como construir los programas.

Un **programa** es una secuencia de instrucciones comprensibles por el ordenador que le permiten realizar una determinada tarea con la finalidad de resolver un problema concreto

# Programación

## El software de programación. Los lenguajes de programación

El objetivo de la programación (**programar**) consiste en crear un conjunto de instrucciones para el ordenador. El **programador** es quien crea ese conjunto o lista de instrucciones. Esta lista se conoce como **código** y el proceso de escritura de la lista se conoce como **codificación**. La programación es un proceso que permite al programador crear algo nuevo y resolver problemas.

**Código:** El término código se refiere a las declaraciones escritas en cualquier lenguaje de programación, ya sea código máquina o un lenguaje de alto nivel.

# Programación

## El software de programación. Los lenguajes de programación

Un **lenguaje de programación** es un conjunto de palabras clave o instrucciones y unas reglas sintácticas que indican como construir los programas.

Los programadores utilizan los lenguajes de programación para crear el código fuente de los programas. Posteriormente, convierten el código fuente en código máquina (o código objeto) para la computadora.



# Programación

## El software de programación. Los lenguajes de programación

Los lenguajes de programación comparten una serie de características comunes:

- Existe un conjunto de reglas obligatorias que el programador debe seguir.
- Se utilizan símbolos especiales.
- No se permiten errores léxicos, sintácticos ni semánticos.
- Hay que proporcionar la información en un orden y estructura específicos.

Estas reglas constituyen la sintaxis del lenguaje de programación y pueden variar mucho de un lenguaje a otro. Cuando no se cumplen estas reglas, el compilador o intérprete emitirá errores y advertencias, no podrá traducir el código fuente y fallará en la creación del código objeto.

# Programación

## Clasificación de los lenguajes de programación

# Programación

## Clasificación de los lenguajes de programación

Según su nivel de abstracción, los lenguajes de programación se clasifican en:

- **Lenguajes de alto nivel** (lenguajes orientados al problema).
- **Lenguajes de bajo nivel** (lenguajes orientados al ordenador).

# Programación

## Clasificación de los lenguajes de programación

Clasificación de los lenguajes de programación:

- Lenguajes de bajo nivel.
  - Lenguaje máquina.
  - Lenguaje ensamblador.
- Lenguajes de alto nivel.

# Programación

## Clasificación de los lenguajes de programación

### Lenguajes de bajo nivel

Los lenguajes de bajo nivel se encuentran totalmente vinculados al hardware, es decir, a la estructura interna del ordenador. Están diseñados para poder sacar partido directamente a la máquina y no para hacer frente a la solución de ningún tipo de problema específico.

Bajo esta denominación se engloban tanto el lenguaje máquina como el lenguaje ensamblador.



# Programación

## Clasificación de los lenguajes de programación

### **Lenguajes de bajo nivel. Lenguaje máquina**

Los lenguajes máquina son los lenguajes más básicos. Los lenguajes máquina son muy dependientes del diseño del hardware (Pentium, Macintosh, etc.) y las instrucciones se crean en forma de código máquina (ceros y unos).

Una instrucción en lenguaje máquina no es más que una cadena de ceros y unos que expresan, de forma codificada, las únicas órdenes que un ordenador es capaz de entender.

Una computadora sólo entiende su lenguaje máquina nativo, los comandos de su conjunto de instrucciones. Estos comandos le indican a la computadora que realice operaciones elementales como cargar, almacenar, sumar y restar.

# Programación

## Clasificación de los lenguajes de programación

### Lenguajes de bajo nivel. Lenguaje máquina

Cualquier programa, para ser ejecutado en un ordenador, debe ser traducido a lenguaje máquina. Si un programa ha sido creado en otro lenguaje de programación, deberá sufrir un proceso de traducción antes de ser ejecutado



# Programación

## Clasificación de los lenguajes de programación

### Lenguajes de bajo nivel. Lenguaje ensamblador

El lenguaje ensamblador surgió con la idea de evitar las dificultades que presenta trabajar en lenguaje máquina. En términos generales, la misión de un lenguaje ensamblador es la de simplificar la programación de un determinado ordenador, manteniendo un control directo sobre el hardware del mismo.

Las simplificaciones aportadas por los lenguajes ensambladores provienen fundamentalmente de dos aspectos:

- El empleo de códigos nemotécnicos (abreviaciones cortas del idioma inglés) para representar las instrucciones.
- El empleo de nombres simbólicos para designar los datos y las referencias a memoria.

# Programación

## Clasificación de los lenguajes de programación

### Lenguajes de bajo nivel. Lenguaje ensamblador

Para desarrollar software con un lenguaje ensamblador hay que utilizar un editor de texto para crear los archivos de código. Para convertir el código fuente en código objeto se utiliza un programa de traducción (traductor) especial llamado ensamblador.



Los programas sólo se escriben en lenguaje ensamblador cuando se necesita un código eficiente y rápido (por ejemplo, en juegos de acción), donde la velocidad del programa es muy importante.

# Programación

## Clasificación de los lenguajes de programación

### Lenguajes de alto nivel

El código máquina (ceros y unos) es el único lenguaje que una computadora entiende. Como las personas tienen dificultad para entender el código máquina, se han desarrollado lenguajes de programación cada vez más sofisticados para hacer que el desarrollo de software sea más sencillo. Se les llama lenguajes de alto nivel porque su sintaxis es más parecida al idioma humano que el código de los lenguajes ensamblador o máquina.

Los lenguajes de programación de alto nivel utilizan palabras familiares en lugar de las cadenas de dígitos que forman las instrucciones del lenguaje máquina y permiten que los programadores puedan pensar sobre sus programas en términos humanos en lugar de en términos de computadoras.

Un lenguaje de programación de alto nivel es un lenguaje orientado hacia la resolución de una determinada clase de problemas.

# Programación

## Clasificación de los lenguajes de programación

### Lenguajes de alto nivel

Un lenguaje de alto nivel es independiente de la arquitectura del ordenador que lo soporta. Esto presenta dos ventajas principales:

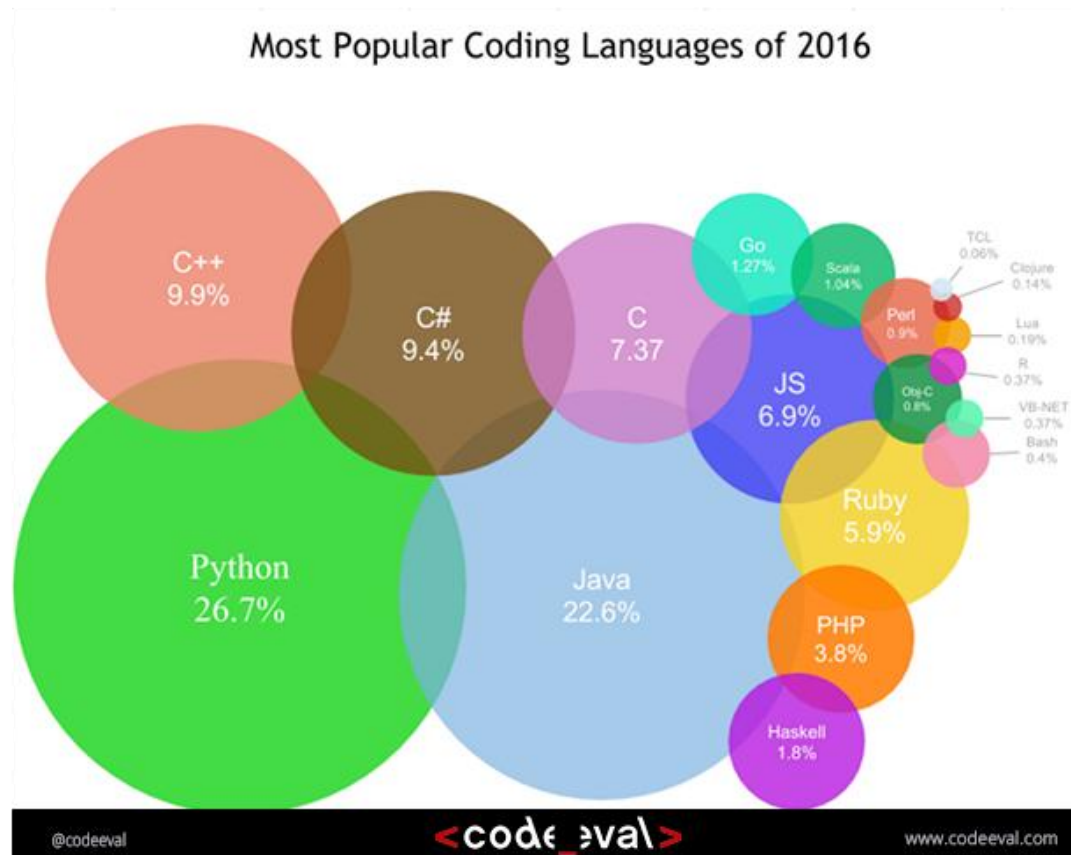
- El programador no tiene que saber nada acerca del ordenador en el que se ejecutará el programa.
- Los programas son transportables, es decir, el mismo programa puede (en teoría) ejecutarse en diferentes tipos de ordenadores.

Otra característica de los lenguajes de alto nivel es que, generalmente, son más cortos que sus equivalentes en bajo nivel.

Entre los lenguajes de programación más conocidos se pueden citar: C, C++, Ada, Java, Pascal, Prolog, Lisp, Visual Basic, Python, etc.

# Programación

## Clasificación de los lenguajes de programación



# Programación

## Compiladores e intérpretes



# Programación

## Compiladores e intérpretes

Las computadoras utilizan el lenguaje máquina.

Los programadores utilizan lenguajes de programación en lugar de lenguajes binarios. Los lenguajes de programación permiten al programador escribir un programa utilizando una variante del idioma inglés clásico. Los resultados se guardan en un archivo y se llaman código fuente.

Después de crear un fragmento de código fuente, hay que convertirlo en código máquina para que pueda ejecutarse en una computadora.

El trabajo de convertir el código fuente en código máquina se puede hacer con dos tipos de programas:

- Compiladores.
- Intérpretes.

# Programación

## Compiladores e intérpretes

**Compiladores:** (C++, Pascal, etc.): un compilador convierte todo el código fuente en código máquina y crea un archivo ejecutable. El resultado del compilador se conoce como código objeto. Una vez compilado el programa, se convierte en un archivo ejecutable independiente que no necesita el compilador para ejecutarse



# Programación

## Compiladores e intérpretes

**Intérpretes:** (LISP, BASIC, Visual Basic, etc.): un intérprete también convierte el código fuente en código máquina. Sin embargo, en lugar de crear un archivo de código objeto ejecutable, lo traduce y luego ejecuta cada línea del programa de una en una. Al traducir el código sobre la marcha, los intérpretes tienen una flexibilidad que no tienen los compiladores. Por otro lado, el código interpretado se ejecuta de manera más lenta que el código compilado.

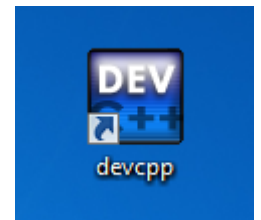
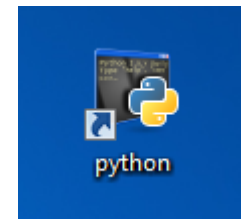
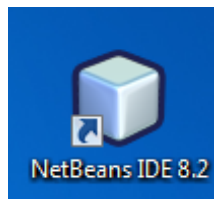
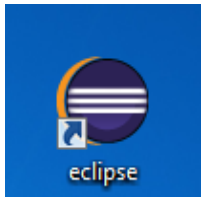
# Programación

## Los entornos de desarrollo integrados (IDE)

# Programación

## Los entornos de desarrollo integrado (IDE)

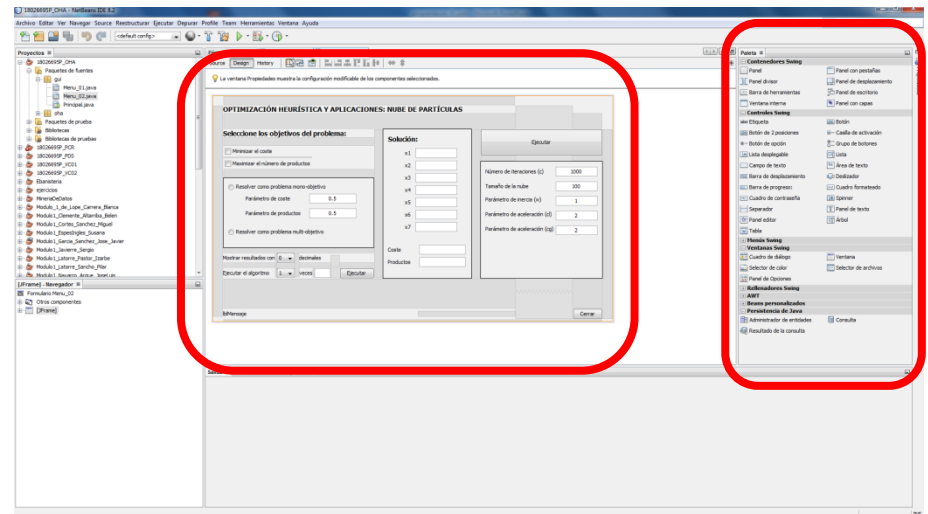
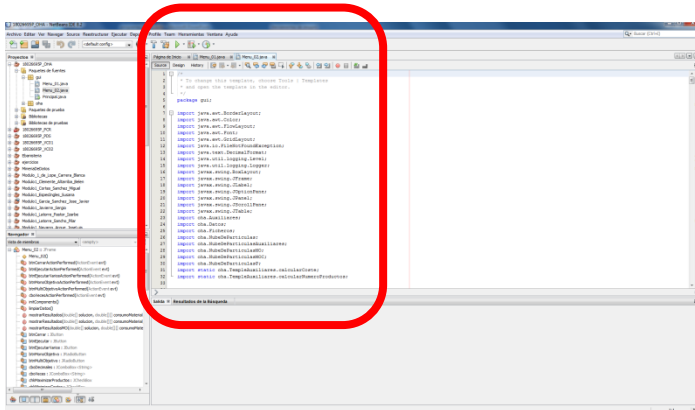
Dadas las crecientes exigencias de los usuarios por disponer de interfaces de usuario amigables surgieron hace unos años distintos programas, conocidos como entornos integrados de desarrollo (IDEs) orientados a facilitar el trabajo al programador que debía desarrollar dicho software.



# Programación

## Los entornos de desarrollo integrado (IDE)

La mayoría de los IDEs tienen una **vista de diseño** en la que el programador puede construir de manera gráfica las distintas ventanas y menús, y añadir los componentes gráficos (cajas de texto, botones, etc.) que aparecen en el programa y una **vista de codificación** en la que el programador accede al código fuente del programa.

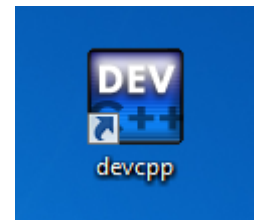
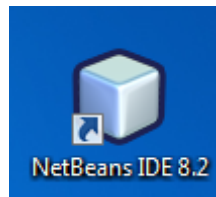
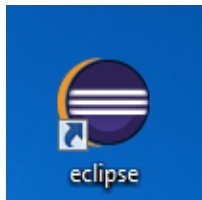


# Programación

## Los entornos de desarrollo integrado (IDE)

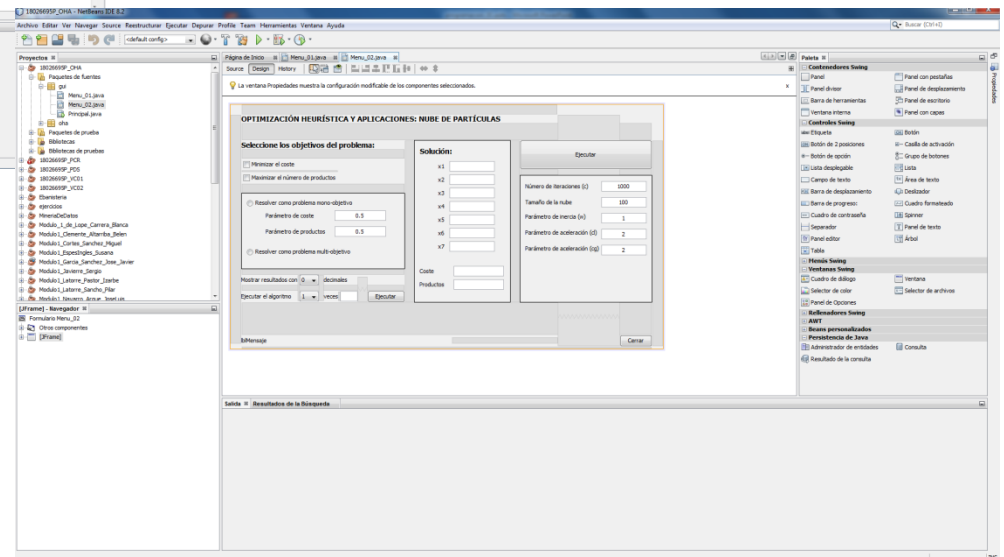
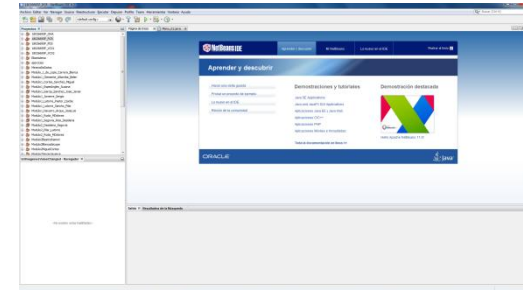
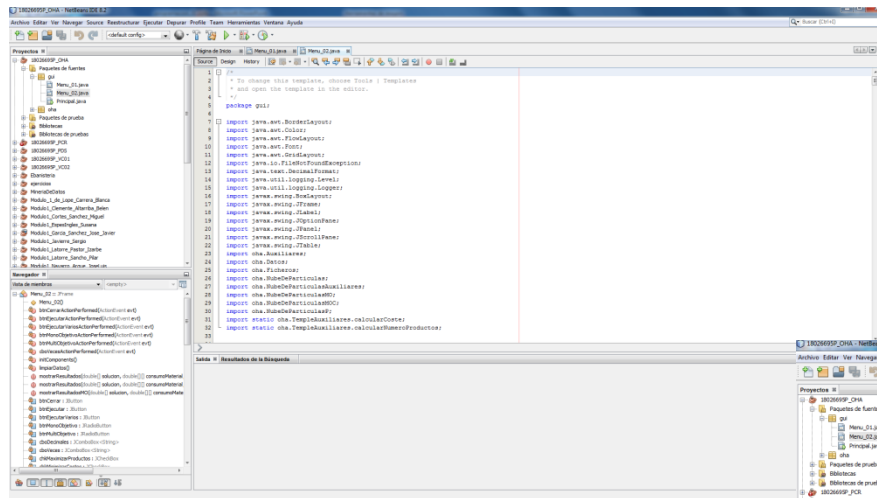
La introducción de la posibilidad de disponer de la vista de diseño en los IDEs fue un gran avance para los programadores y les permitió desarrollar las aplicaciones de forma más rápida, con menos esfuerzo, y ofrecer programas de mayor calidad.

Algunos ejemplos de IDEs son: Netbeans, Eclipse, Dev-C++, BlueJ, JCreator, JBuilder, MS Visual Studio, JDeveloper, etc.



# Programación

## Los entornos de desarrollo integrado (IDE)





Fin