



CEU

*Fundación San Pablo  
Andalucía*

| 25<sup>o</sup> aniversario

# ENTORNOS DE DESARROLLO

1º Diseño Aplicaciones Multiplataformas (DAM)

1º Diseño Aplicaciones Web (DAW)

Profesor: Rafael Madrigal Toscano

# TEMA 1: CONCEPTOS BÁSICOS

1. CONCEPTOS
2. TIPOS DE LENGUAJES DE PROGRAMACIÓN
3. CÓDIGO FUENTE, CÓDIGO OBJETO Y CÓDIGO EJECUTABLE
4. MÁQUINAS VIRTUALES (TIPOS)
5. PROCESO DE OBTENCIÓN DE CÓDIGO EJECUTABLE A PARTIR DEL CÓDIGO FUENTE
6. HERRAMIENTAS CASE
7. HERRAMIENTAS USADAS EN PROGRAMACIÓN

# TEMA 1: Conceptos Básicos

## 1.1 Conceptos

### **Def. Programa informático:**

Se puede decir que se trata de una secuencia de código escrito que tiene por objeto dar instrucciones específicas al ordenador, o computador, para realizar una tarea.

El programa tiene un formato ejecutable que la computadora puede utilizar directamente para ejecutar las instrucciones. El mismo programa en su formato de código fuente legible para humanos, del cual se derivan los programas ejecutables (por ejemplo, compilados), le permite a un programador estudiar y desarrollar sus algoritmos. Una colección de programas de computadora y datos relacionados se conoce como software.

Una definición sencilla de un programa informático, es ese aspecto “funcional” del computador, que se utiliza para ejecutar tareas que el ser humano quiere hacer.

Por ejemplo:

El típico “hola mundo” , ejecutar una base de datos, abrir una Excel o un programa para ver vídeos.

```
//CONSTANTES
const int ruedas = 4;
const int espejos = 2;

//VARIABLES
int coches = 5;

//Declaración de dos variables en la misma línea
int totalRuedas, totalEspejos;

//Operaciones
totalRuedas = ruedas * coches;
totalEspejos = espejos * coches;

//Sacamos el resultado por pantalla
Console.WriteLine(totalRuedas);
Console.WriteLine(totalEspejos);
Console.ReadKey();
```

# TEMA 1: Conceptos Básicos

## Def. Aplicación informática:

Una aplicación (o app) es un **programa informático** creado para llevar a cabo o facilitar una tarea en un dispositivo informático. Cabe sacar a colación que aunque todas las aplicaciones son programas, **NO** todos los programas son aplicaciones.

Existe multitud de software en el mercado, pero sólo se denomina así a aquel que ha sido creado con un fin determinado, es decir para realizar tareas concretas. No se consideraría una aplicación, por ejemplo, un sistema operativo, ni una suite, pues su propósito es general.



# TEMA 1: Conceptos Básicos

## 1.2 Concepto y tipos de lenguaje de programación

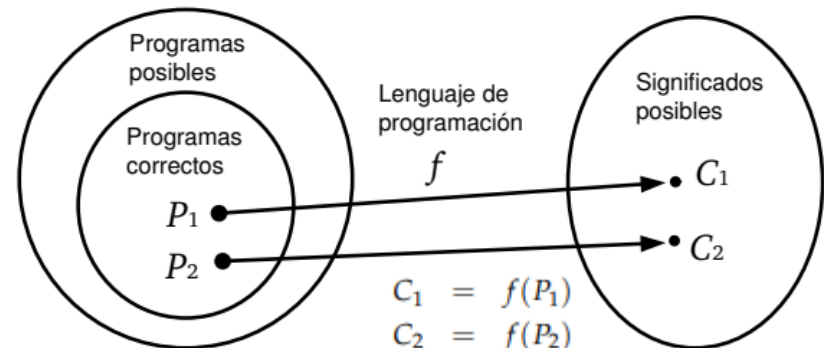
### **Def. Lenguaje de Programación**

Un lenguaje de programación es un lenguaje formal que le proporciona a una persona, en este caso el programador, la capacidad de escribir (o programar) una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico o lógico de un sistema informático, de manera que se puedan obtener diversas clases de datos o ejecutar determinadas tareas. En otras palabras, es un convenio o conjunto de reglas que permiten asociar a cada programa correcto un cálculo que será llevado a cabo por un ordenador.

Algunos lenguajes están definidos por un documento estandarizado en un organismo oficial como ISO.

C++ es el estándar ISO/IEC 14882(1998 )

El lenguaje es como una función que asigna un significado



# TEMA 1: Conceptos Básicos

## Paradigmas de Programación

Es una manera o estilo de programación de software. Existen diferentes formas de diseñar un lenguaje de programación y varios modos de trabajar para obtener los resultados que necesitan los programadores.

Se trata de un conjunto de métodos sistemáticos aplicables en todos los niveles del diseño de programas para resolver problemas computacionales.



<https://youtu.be/bQITKm7-63w?t=28>

Los lenguajes de programación pueden adoptar uno o varios paradigmas según el tipo de órdenes que permiten implementar.

Consisten en una sucesión de instrucciones o conjunto de sentencias

Javascript o Python → Multiparadigma

Pascal, COBOL, FORTRAN, C, C++ → Imperativo

Prolog, Scala, Lisp → Declarativo

C++, Java → Orientado a objetos

Project Reactor, RxJava, RxJs → Reactivo

Describe el problema en lugar de encontrar una solución al mismo

Se construyen modelos de objetos que representan elementos

Escucha lo que emite un evento o cambios en el flujo de datos

# TEMA 1: Conceptos Básicos

## Generaciones de Lenguajes de Programación

### Primera generación (1GL)

Formada por los lenguajes de programación utilizados en los primeros ordenadores: lenguaje máquina y ensamblador.

### Segunda generación (2GL)

Formada por el lenguaje macroensamblador que es ensamblador combinado con instrucciones de control y de manejo de datos más complejas. Estos lenguajes son específicas para una familia de procesadores y el hardware relacionado con él. Aún se siguen utilizando para programar los núcleos (kernel) de los sistemas operativos y los controladores de algunos dispositivos (*device drivers*).



# TEMA 1: Conceptos Básicos

## Generaciones de Lenguajes de Programación

### **Tercera generación (3GL)**

Formada por la mayor parte de los lenguajes de alto nivel actuales. El código es independiente de la máquina y el lenguaje de programación es parecido al lenguaje humano. Por ejemplo: Java, C, C++, PHP, JavaScript, e Visual Basic.

### **Cuarta generación (4GL)**

Formada por lenguajes y entornos diseñados para una tarea o propósito muy específico como acceso a bases de datos, generación de informes, generación de interfaces de usuario, etcétera. Por ejemplo: SQL, Informix 4GL y Progress 4GL.

### **Quinta generación (5GL)**

Formada por lenguajes en las que el programador establece el problema a resolver y las condiciones a cumplir. Se usan en inteligencia artificial, sistemas basados en el conocimiento, sistemas expertos, mecanismos de inferencia o procesamiento del lenguaje natural. Por ejemplo: Prolog, Smalltalk y Lisp.



# TEMA 1: Conceptos Básicos

## 1.3 Código fuente, código objeto y código ejecutable

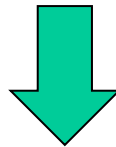
### Definiciones:

- ✓ **Código fuente:** Conjunto de sentencias entendibles por el programador que componen el programa o una parte de ello. Se suele almacenar en un fichero de tipo texto. El código fuente se escribe en un lenguaje de programación concreto.
- ✓ **Código objeto:** Conjunto de instrucciones y datos escritos en un lenguaje que entiende el ordenador directamente: binario o código máquina. Es la traducción de cierto código fuente, es un fragmento del programa final y es específico de la plataforma de ejecución.
- ✓ **Código ejecutable:** Reúne diferentes códigos u objetos generados por los programadores junto con las “librerías de uso general” (propias del entorno o del lenguaje de programación) componiendo el programa final. Éste es el código que ejecutan los usuarios del sistema, y es específico para una plataforma concreta: Windows, Linux, Mac OS, o ciertos sistema hardware.

# TEMA 1: Conceptos Básicos

## 1.4 Máquinas virtuales

- ❑ **Una máquina virtual es un software que crea una capa independiente donde se emula el funcionamiento de un ordenador real con todos los componentes de hardware que necesita para funcionar** (disco duro, memoria RAM, tarjetas de red, tarjeta gráfica, etc.) y que puede ejecutar cualquier sistema operativo o programa, tal y como lo haría un ordenador real. El interior de esa ventana afecta al sistema que la ejecuta.
- ❑ Toda esta emulación se encapsula en una serie de archivos que actúan como contenedor desde el que se ejecuta la máquina virtual, en un trozo acotado de tu ordenador como si de un programa más se tratara y sin que nada de lo que suceda en el interior de esa ventana afecte al ordenador que la ejecuta.



Una máquina virtual es un software que emula un ordenador, es decir, como tener un ordenador dentro de otro ordenador pero funcionando de forma virtual.

# TEMA 1: Conceptos Básicos

## 1.4 Máquinas virtuales

### Definiciones:

- ✓ **Máquinas virtuales de sistema:** (System Virtual Machine).

Permiten ejecutar en la misma máquina física varias máquinas virtuales cada una con un sistema operativo, de esta forma pueden coexistir diferentes S.O. sobre la misma máquina. Ejemplos de este software es VMware Workstation y Virtual Box.



<https://www.youtube.com/watch?v=F-sM9zbk76A>

- ✓ **Máquinas virtuales de proceso:** (Process Virtual Machine)

Llamada también máquina virtual de aplicación, se ejecuta como un proceso normal dentro del S.O. y soporta un único proceso. La máquina se inicia automáticamente cuando se lanza el proceso que se va a ejecutar y se detiene cuando finaliza. El objetivo fundamental es proporcionar un entorno de ejecución independiente de la plataforma hardware y del S.O. Es decir, se ocultan los detalles de la plataforma y permite que un programa se ejecute siempre de la misma forma independientemente de la plataforma mencionada.

# TEMA 1: Conceptos Básicos

## 1.4 Máquinas virtuales sistema

### ANFITRIÓN-HUÉSPED

Ubuntu 20.04 LTS [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

29 de oct. 09:55

Actividades

Q Escribir para buscar

Es decir, el anfitrión (también conocido como host) es que alberga al invitado (también conocido como huésped)

Un anfitrión puede tener varios invitados, no está limitado solo a uno.

Hay un límite al número de invitados activos, dependiendo de la memoria, HD, etc. con que cuente la maquina anfitrión.

SISTEMA ANFITRIÓN

SISTEMA HUÉSPED

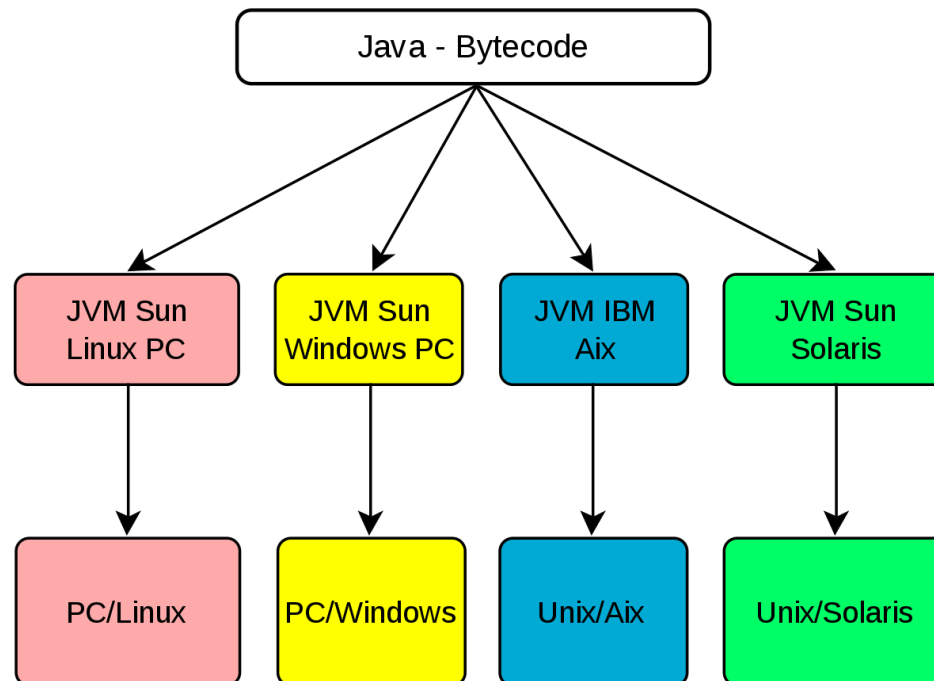
CTRL DERECHA

8:55

# TEMA 1: Conceptos Básicos

## 1.4 Máquinas virtuales de proceso

El ejemplo más conocido es la máquina virtual de java. (en inglés Java Virtual Machine, JVM) es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el bytecode Java), el cual es generado por el compilador del lenguaje Java.



# TEMA 1: Conceptos Básicos

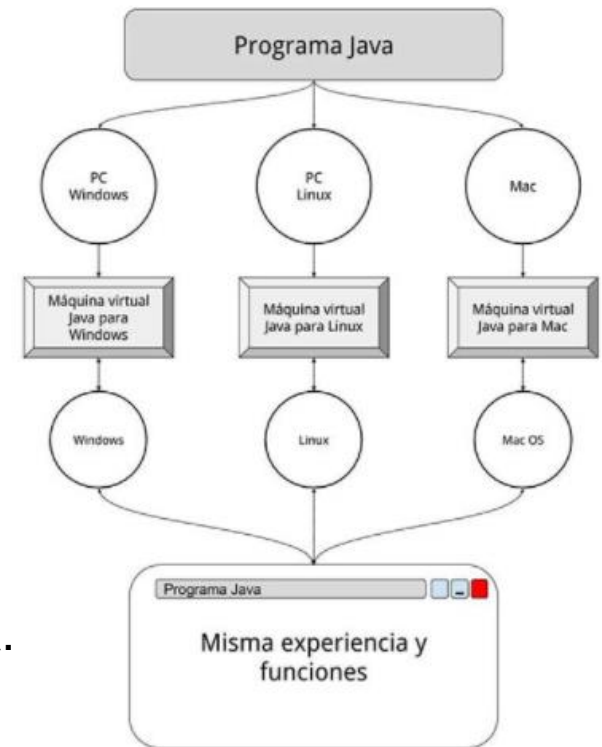
## 1.4 Máquinas virtuales de proceso

- ☐ Una máquina virtual de proceso, a veces llamada “máquina virtual de aplicación”, se ejecuta como un proceso normal dentro de un sistema operativo y soporta un sólo proceso.
- ☐ La máquina se inicia automáticamente cuando se lanza el proceso que se desea ejecutar y se detiene para cuando éste finaliza.
- ☐ Su objetivo es el de proporcionar un entorno de ejecución independiente de la plataforma de hardware y del sistema operativo, que oculte los detalles de la plataforma subyacente y permita que un programa se ejecute siempre de la misma forma sobre cualquier plataforma.
- ☐ Una máquina virtual de proceso es menos ambiciosa que una de sistema. En vez de emular un PC por completo, ejecuta un proceso concreto, como una aplicación, en su entorno de ejecución. Se usa cada vez que ejecutas una aplicación basada en Java o basada en .NET Framework.

# TEMA 1: Conceptos Básicos

## 1.4 Máquinas virtuales de proceso

- ❑ Esto es de utilidad a la hora de desarrollar aplicaciones para varias plataformas, pues en vez de tener que programar específicamente para cada sistema, el entorno de ejecución (es decir, la máquina virtual) es el que se encarga de lidiar con el sistema operativo.
- ❑ Las máquinas virtuales de proceso te permiten disfrutar de aplicaciones que se comportan de forma igual en plataformas tan distintas como Windows, Mac o Linux, pero tú como usuario normalmente no les prestarás mucha atención.
- ❑ Por eso, salvo que seas programador, generalmente cuando se habla de máquinas virtuales nos estamos refiriendo a las de sistema.



# TEMA 1: Conceptos Básicos

## 1.5 Proceso de obtención de código ejecutable a partir del código fuente

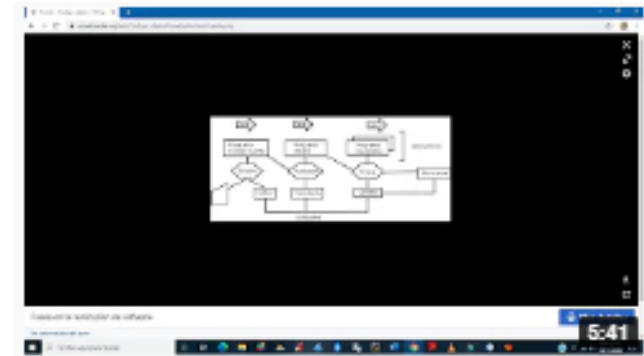


## 1.6 Herramientas CASE

(Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora)

Son diversas aplicaciones informáticas o programas usadas para automatizar actividades de el ciclo de vida de desarrollo de sistemas. Las herramientas CASE son usadas por los Directores de proyectos de software, analistas e Ingenieros para desarrollar sistemas de software

[https://www.youtube.com/watch?v=SAoZ\\_tqweEk](https://www.youtube.com/watch?v=SAoZ_tqweEk)



[https://youtu.be/g\\_BUHY0iotA](https://youtu.be/g_BUHY0iotA)



# TEMA 1: Conceptos Básicos

## 1.6 Herramientas CASE

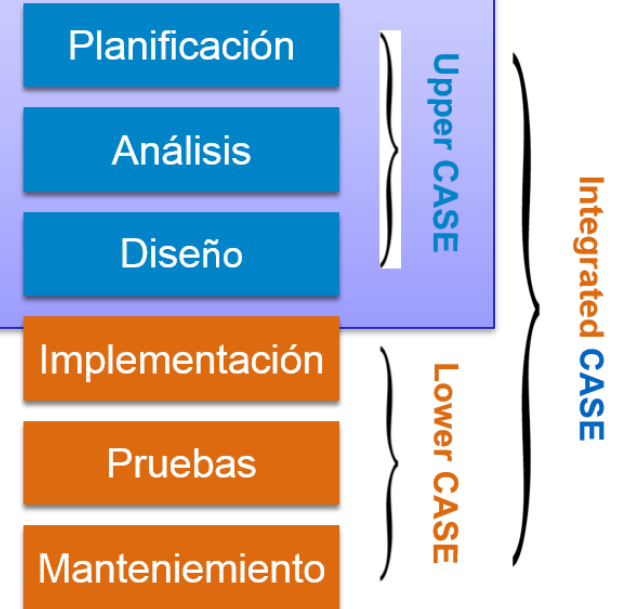
Existen gran cantidad de Herramientas CASE.

Se usan para simplificar varias etapas en el desarrollo del ciclo vital del Software, como por ejemplo:

- Herramientas de análisis,
- Diseño de herramientas,
- Gestión de proyectos de herramientas,
- Proyectos de gestión de herramientas de B.D.
- Gestión de herramientas de Bases de datos,
- Herramientas de Documentación.

### Tipos de Herramientas CASE

- |                              |                         |
|------------------------------|-------------------------|
| • Diagrama                   | • Análisis              |
| • Modelado de Procesos       | • Diseño                |
| • Administración de Procesos | • Gestión de la Config. |
| • Documentación              | • Control de Cambios    |
|                              | etc.                    |



# TEMA 1: Conceptos Básicos

## Ejercicio

Selecciona un ejemplo concreto de herramienta como las que hemos mencionado anteriormente y explícala en 750 palabras como máximo.

Asimismo contesta a las siguientes cuestiones y razona la respuesta.

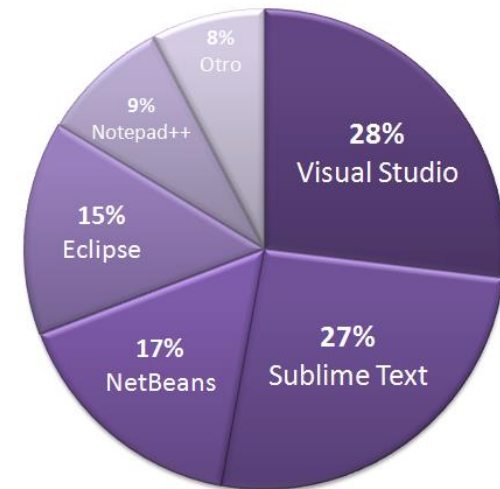
- 1) Diferencias entre compilador e interprete.
- 2) ¿Cuáles son los diferentes estados por los que pasa el código de un programa desde que se escribe hasta que se ejecuta por el ordenador?
- 3) Cuáles de las siguientes afirmaciones sobre la JVM es cierta:
  - a) Un fichero .class contiene el código en lenguaje máquina
  - b) La JVM toma y traduce el bytecode en código binario.
  - c) Los ficheros .class pueden ser ejecutados en Linux y en Windows
  - d) A la hora de instalar el entorno de ejecución de la JVM necesitamos saber en que sistema operativo se va a instalar

# TEMA 1: Conceptos Básicos

## 1.7 Herramientas utilizadas en programación

Para realizar la codificación y pruebas de los programas se suelen utilizar entornos de programación. Éstos nos permiten realizar las siguientes tareas:

- ✓ Crear, editar y modificar el código fuente.
- ✓ Compilar, montar y ejecutar un programa.
- ✓ Examinar el código fuente.
- ✓ Depurar el programa.
- ✓ Realizar test de forma automática.
- ✓ Elaborar documentación.
- ✓ Gestionar cambios (control de versiones)...



A estos entornos de programación se les suele llamar entornos de desarrollo integrado o **IDE** (Integrated Development Environment)



MUCHAS GRACIAS

