

# Entornos de Desarrollo

UD1. Conceptos básicos

Raúl Reyes  
Patricia Vegas  
Adrián Seoane

# Índice

Introducción

Tipos de lenguajes de programación

Características de lenguajes

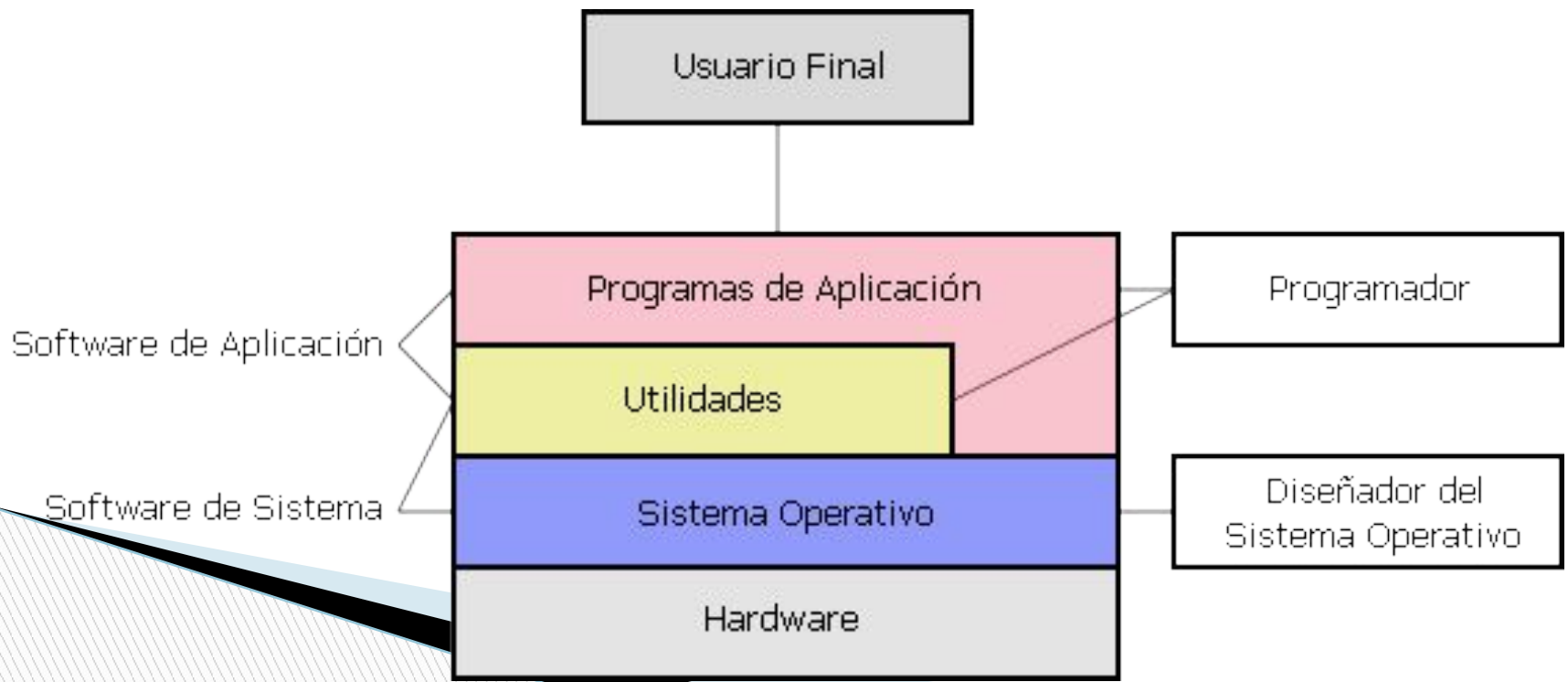
Obtención Código Ejecutable

Fases del desarrollo

Roles que interactúan en el desarrollo

# Introducción

- ? **Software:** Conjunto de programas elaborados por el hombre, que controlan la actuación del **ordenador**, haciendo que este siga en sus acciones una serie de pasos lógicos predeterminados.
- ? **Software:** intermediario entre hardware y hombre

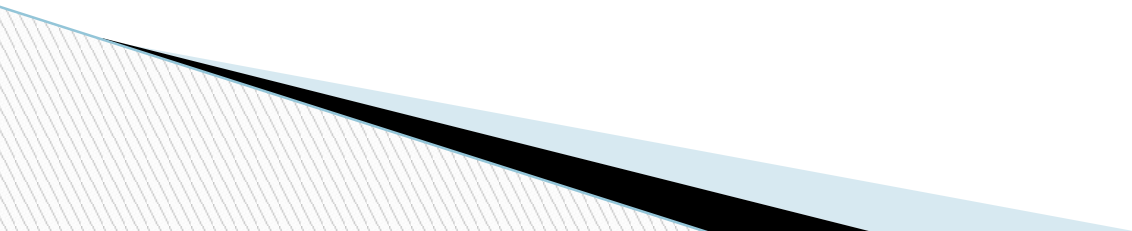


# Introducción

- ? **Ejercicio:** Clasifica el siguiente software en los tipos de software vistos anteriormente:

Software	Tipo
Linux	
Accesorios de Windows	
NetBeans	
Excel	
Clic	
Java	
Símbolo de Sistema	
AutoCad	
Windows XP	

# Introducción

- ? **Algoritmo**: es la forma de resolver un problema, los pasos a seguir para resolver un problema estándar.
  - ? Algoritmos se representan:
    - Lenguaje Natural
    - Pseudocódigo
    - **Diagrama de Flujo**
- 

# Introducción

## Algoritmo *HACER LA COMPRA*

Lenguaje natural	Pseudocódigo	
<p>1) Hago la lista de la compra</p> <p>2) Voy al supermercado</p> <p>3) Reviso la lista de la compra y cojo los artículos</p> <p>4) Vuelvo con la compra</p>	<ol style="list-style-type: none"><li>1. <b>Write</b> “lista de la compra”</li><li>2. <b>Goto</b> “supermercado”</li><li>3. <b>Read</b> “lista de la compra”</li><li>4. <b>Goto</b> “casa”</li></ol>	<pre>graph TD; A[Write "lista de la compra"] --&gt; B[Goto "supermercado"]; B --&gt; C[Read "lista de la compra"]; C --&gt; D[Goto "casa"];</pre> <p>The flowchart illustrates the steps of the shopping algorithm. It starts with a box labeled "Write 'lista de la compra'", followed by an arrow pointing down to a box labeled "Goto 'supermercado'". From there, another arrow points down to a box labeled "Read 'lista de la compra'", and a final arrow points down to a box labeled "Goto 'casa'".</p>

# Introducción



# Introducción

? **Ejercicio:** Escribe la secuencia de pasos (algoritmo) para las siguientes situaciones cotidianas:

Tarea	Algoritmo
Ir al colegio	
Arreglar un ordenador	
Almorzar	



# Introducción

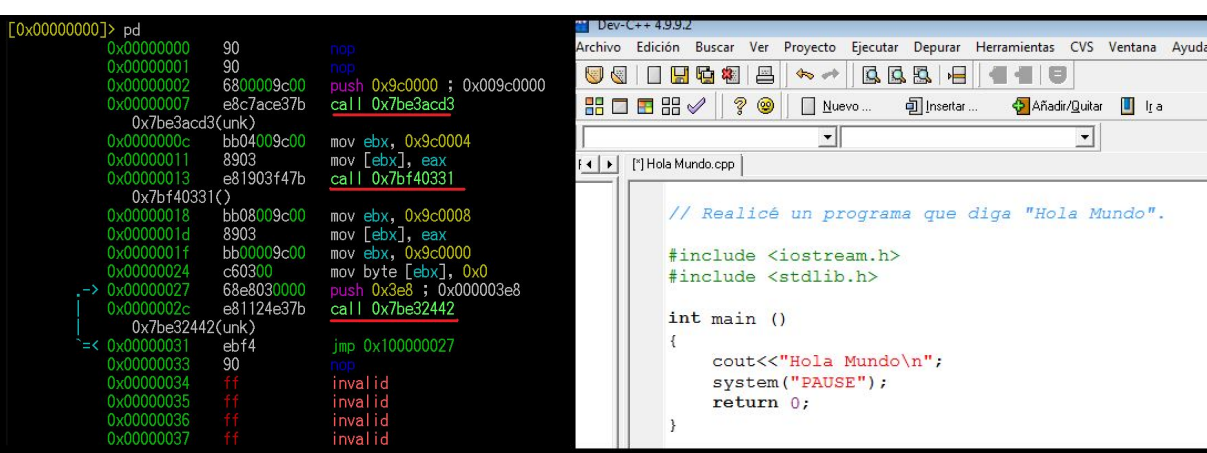
- ? **Lenguaje de programación:** es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar.



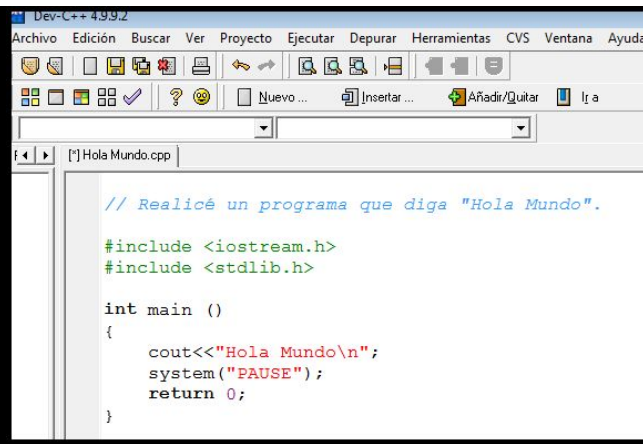
# Tipos de lenguajes de programación

## ? Nivel de abstracción:

- *Bajo Nivel:* La programación se realiza teniendo muy en cuenta las características del procesador. Lenguaje ensamblador
- *Medio Nivel:* Permiten un mayor grado de abstracción pero al mismo tiempo mantienen algunas cualidades de los lenguajes de bajo nivel. C
- *Alto Nivel:* Más parecidos al lenguaje humano. Java, Ruby



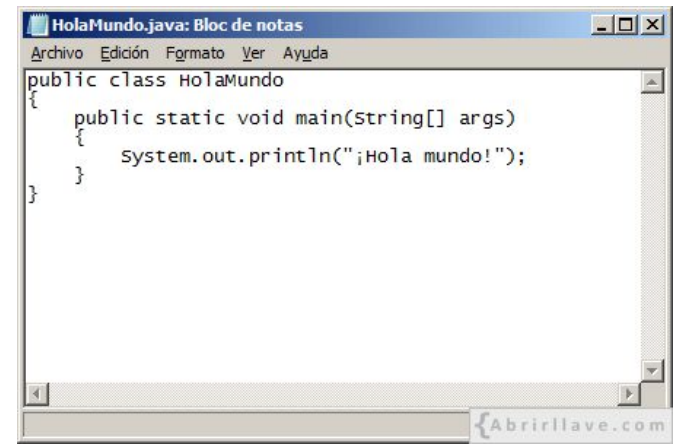
```
[0x00000000]> pd
0x00000000  90          nop
0x00000001  90          nop
0x00000002  6800009c00 push 0x9c0000 ; 0x009c0000
0x00000007  e8c7ace37b call 0x7be3acd3
0x7be3acd3(unk)
0x0000000c  bb04009c00 mov ebx, 0x9c0004
0x00000011  8903        mov [ebx], eax
0x00000013  e81903f47b call 0x7bf40331
0x7bf40331(unk)
0x00000018  bb08009c00 mov ebx, 0x9c0008
0x0000001d  8903        mov [ebx], eax
0x0000001f  bb00009c00 mov ebx, 0x9c0000
0x00000024  c60300     mov byte [ebx], 0x0
-> 0x00000027  68e8030000 push 0x3e8 ; 0x000003e8
0x0000002c  e81124e37b call 0x7be32442
0x7be32442(unk)
<= 0x00000031  eb14        jmp 0x100000027
0x00000033  90          nop
0x00000034  ff          invalid
0x00000035  ff          invalid
0x00000036  ff          invalid
0x00000037  ff          invalid
```



```
// Realicé un programa que diga "Hola Mundo".

#include <iostream.h>
#include <stdlib.h>

int main ()
{
    cout<<"Hola Mundo\n";
    system("PAUSE");
    return 0;
}
```

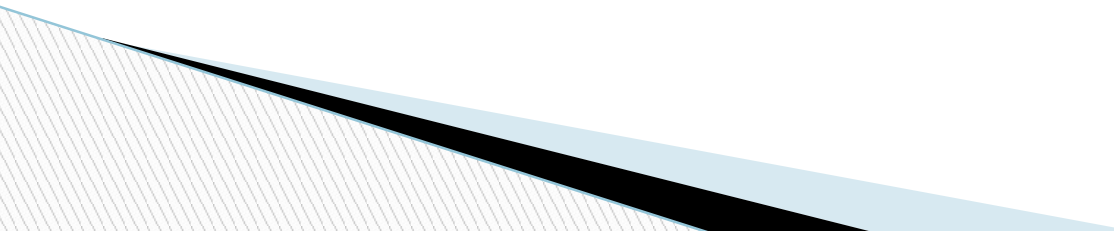


```
HolaMundo.java: Bloc de notas
Archivo Edición Formato Ver Ayuda

public class HolaMundo
{
    public static void main(String[] args)
    {
        System.out.println("¡Hola mundo!");
    }
}
```

# Tipos de lenguajes de programación

## ? Propósito:

- *Propósito general:* Aptos para todo tipo de tareas. C
  - *Propósito específico:* Hechos para un objetivo muy concreto. Csound
  - *Programación de sistemas:* Diseñados para realizar sistemas operativos o drivers. C
  - *Scripts:* Realizar varias tareas de control y auxiliares. JavaScript
- 

# Tipos de lenguajes de programación

## ? **Evolución Histórica:**

- 1GL: Código máquina
- 2GL: Lenguaje ensamblador
- 3GL: Lenguajes modernos (Facilitar programación al humano). C, Java
- 4GL: Diseñados con un propósito concreto. Mathematica
- 5GL: Se usan en inteligencia artificial. Prolog

## ? **Manera de ejecutarse (Los estudiaremos en el tema)**

- Compilados: C
  - Interpretados: Python
  - Mixtos: Java
- 

# Tipos de lenguajes de programación

## ? **Manera de abordar la tarea a realizar:**

- Imperativo: Cómo hay que realizar la tarea. C
- Declarativo: Qué hay que hacer. Prolog

## ? **Paradigma de programación**

- Procedural: Divide el problema en partes más pequeñas. C, Pascal
- Orientada a objetos: Java
- Funcional: Evaluando funciones recursivamente. Lisp, Scala
- Programación lógica: Expresa que computar desde la lógica matemática. Prolog

## ? **Lugar de ejecución**

- Cliente: Javascript
  - Servidor: PHP
- 

# Lenguajes más usados (2021)

## 10 Most Used Programming Languages



1• Python



2• Java



3• JavaScript



4• C++



5• C#



6• R



7• PHP



8• Kotlin



9• Swift



10• Go

# Características de lenguajes más difundidos

- **Python:** Lenguaje de programación de alto nivel. Sintaxis limpia. Multiparadigma

```
def add5(x):  
    return x+5  
  
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]), ast[0])  
    print '    %s [label="%s' % (nodename, label),  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '= %s'];' % ast[1]  
        else:  
            print '"]'  
    else:  
        print '"]';'  
        children = []  
        for n, child in enumerate(ast[1:]):  
            children.append(dotwrite(child))  
        print ',    %s -> {' % nodename  
        for i, name in enumerate(children):  
            print '%s' % name,
```

# Características de lenguajes más difundidos

- **Java:** simple, orientado a objetos, robusto, seguro





# Características de lenguajes más difundidos

- **JavaScript:** Lenguaje más difundido para el desarrollo web, lenguaje orientado a objetos



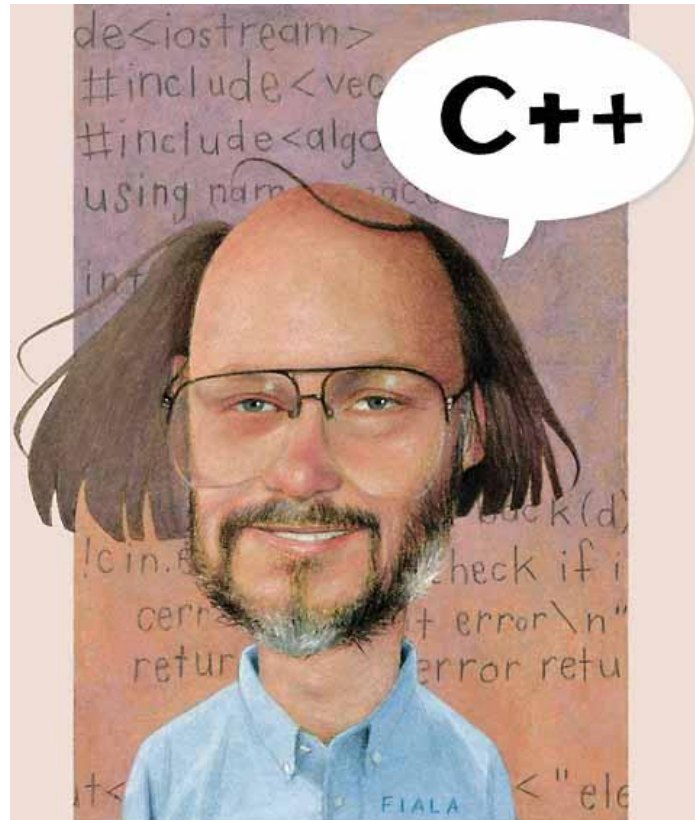
# Características de lenguajes más difundidos

- **C:** Fácil de aprender, permite control ordenador, transportable gracias a ANSI C

```
/** *****  
/** Nombre del archivo: blink.c  
/** Ejemplo de programa que hace parpadear un LED  
/** conectado al PORTB.0  
/** El pic utilizado es el 16F628A de la gama media.  
/** *****  
  
#include <16f628A.h> // Define el PIC  
#fuses XT,NOVDT,NOPROTECT,PUT,BROWNOUT // configuración de fusibles  
  
#use delay(clock=4000000) // Cristal: 4 Mhz  
  
#byte port_b=6 // Dirección en RAM de PORTB  
  
void main() {  
    set_tris_b(0); // Configura Port B como salida  
    port_b=0;  
  
    while(TRUE) {  
        port_b=0; // Bucle infinito  
        delay_ms(200); // Apaga todo el puerto B  
        port_b=1; // Espera 200ms  
        delay_ms(200); // Enciende el bit 0.  
        // Espera otros 200ms  
    }  
}
```

# Características de lenguajes más difundidos

- **C++:** Desarrollo a partir de C++, permite POO (uso en proyectos de mayor envergadura)



# Características de lenguajes más difundidos

- **R:** Lenguaje de programación utilizado para tratamiento de datos y BigData



# Características de lenguajes más difundidos

- **PHP:** Lenguaje de programación para desarrollo de páginas webs



# Características de lenguajes más difundidos

- **Kotlin:** Lenguaje para crear Apps de Android en código nativo



# Características de lenguajes más difundidos

- **Swift:** Lenguaje para crear Apps de IOS en código nativo



Swift

# Características de lenguajes más difundidos

- **GO:** Lenguaje concurrente y compilado, creado por Google y que aúna lo mejor de Python y C



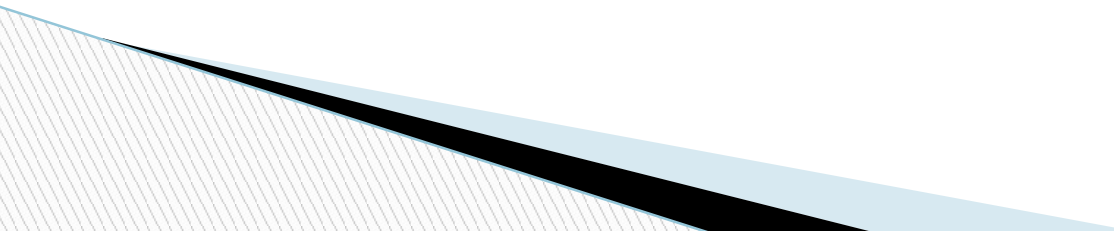


# Características de lenguajes más difundidos

**Ejercicio:** Investiga los lenguajes de programación que se han mencionado y detalla en la siguiente tabla sus características:

	Nivel de abstracción	Propósito	¿Compilado o interpretado?	¿Imperativo o declarativo?	Paradigma	Otros
Python						
Java						
...						

# Características de lenguajes más difundidos

- **Ejercicio:**
  - Investiga programas que estén realizados con el entorno NetBeans (Java)
  - Encuentra ofertas de trabajo relacionadas con dicho lenguaje
  - Importancia de Java en la actualidad
- 

# Lenguajes Ciclo DAM

## Primero

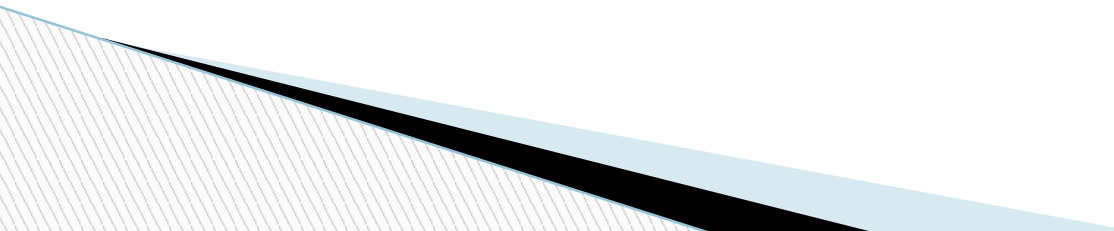
- SQL
- HTML
- CSS
- JAVA
- XML
- JavaScript

## Segundo

- PHP (Laravel Framework)
- JAVA Avanzado
- Ionic (Angular)
- Python (Django Framework)

# Ofertas de Trabajo

## Tarea 1.1.

- ? Busca ofertas de trabajo relacionadas con nuestro sector. Establece distintos niveles: nivel local, provincial, nacional (y si se pudiese, internacional).
  - ? ¿Qué perfiles son los más demandados en cada nivel?
  - ? ¿Cuáles son los requisitos que se piden normalmente?
  - ? Conclusiones obtenidas
- 

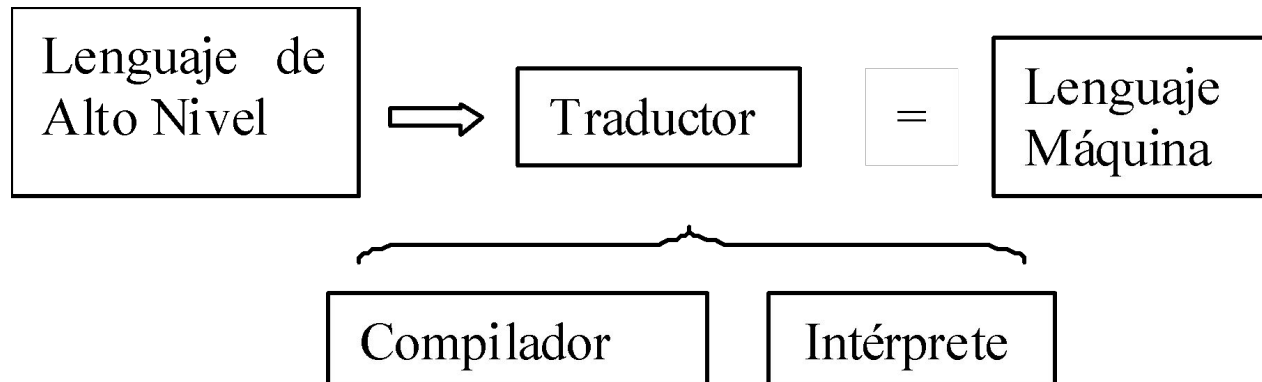
# Obtención del código ejecutable

- **Código fuente:** Programas escritos en un determinado lenguaje de programación.
- **Código objeto:** Código que resulta de la compilación del código fuente.
- **Código ejecutable:** Código obtenido del proceso de enlazar todos los archivos de código objeto con un programa llamado enlazador.



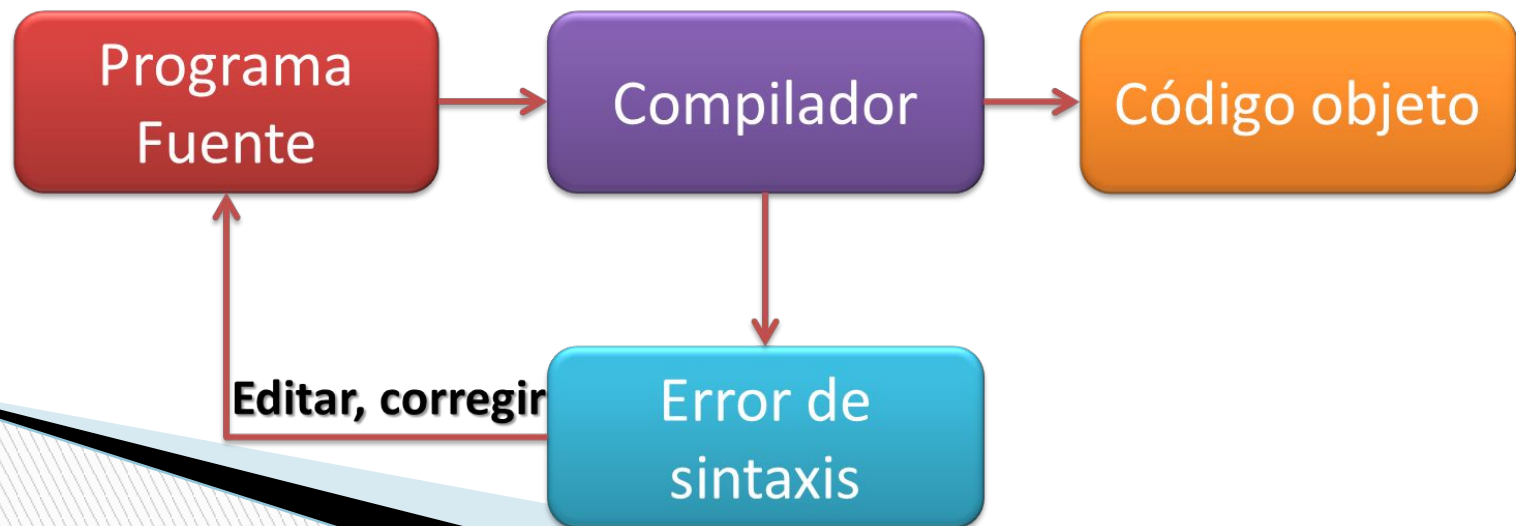
# Obtención del código ejecutable

- **Traductores:** Programas que traducen los programas en código fuente, escritos en lenguajes de alto nivel, a programas escritos en lenguaje máquina



# Obtención del código ejecutable

- **Compiladores:** El programa original (fichero fuente) sólo se traduce una vez, creando un nuevo archivo (fichero objeto) que puede ser entendido por la máquina y ejecutado por el procesador tantas veces como queramos.



# Obtención del código ejecutable

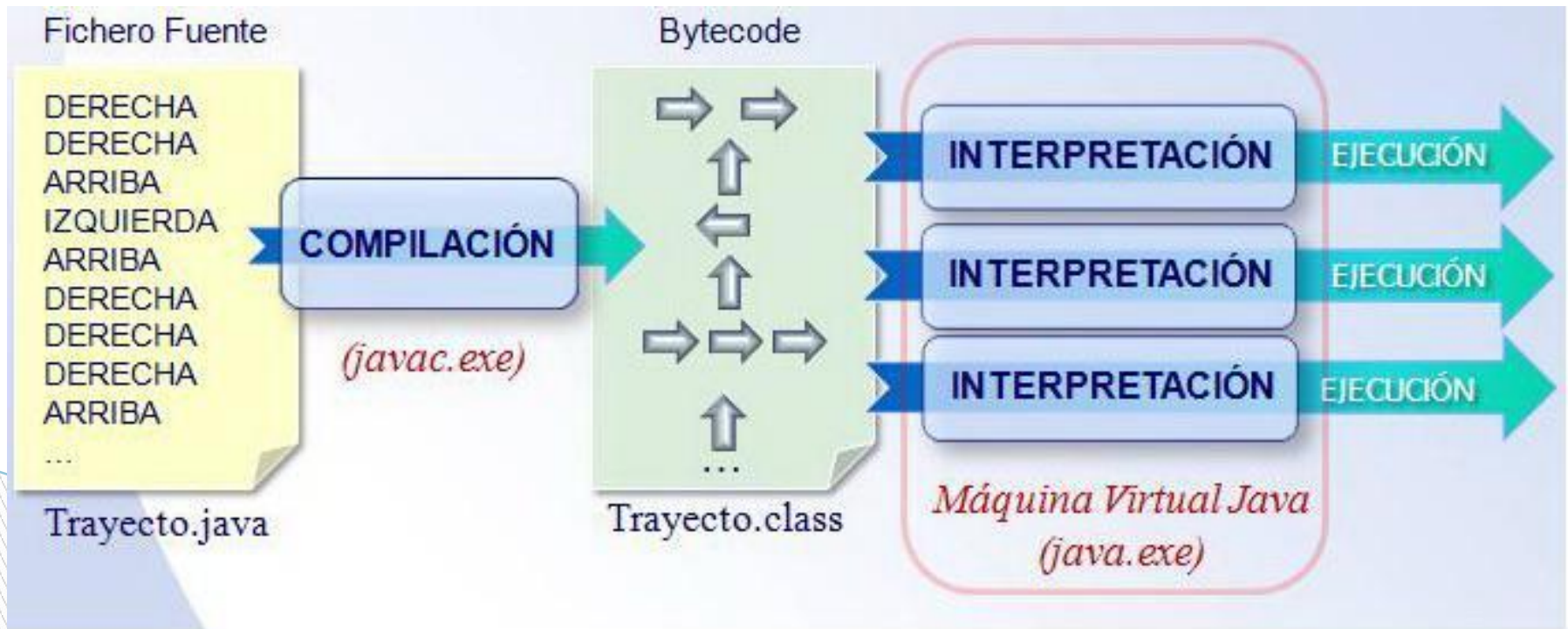
- **Intérpretes:** El programa es traducido cada vez que se va a ejecutar. No producen un código objeto como los compiladores. Lee el código como está escrito y lo ejecuta al instante.





# Obtención del código ejecutable

- **Compilador+Intérprete (Mixto):**  
JAVA



# Obtención del código ejecutable

- Razona las ventajas/desventajas de los compiladores e intérpretes

Compiladores		Intérpretes	
Ventajas		Ventajas	
Inconvenientes		Inconvenientes	

# Fases del desarrollo de una aplicación

- ? El desarrollo de un software pasa por diferentes etapas desde que se produce la necesidad de crear un software hasta que se finaliza y está listo para ser usado por un usuario.



## Ciclo de vida del programa

- ? Hay más de un modelo, vamos a estudiar uno de los modelos más extendidos y completos, el **modelo en cascada**
- ? Las fases o etapas del modelo en cascada son: **Análisis, Diseño, Codificación, Pruebas, Documentación, Explotación, Mantenimiento.**

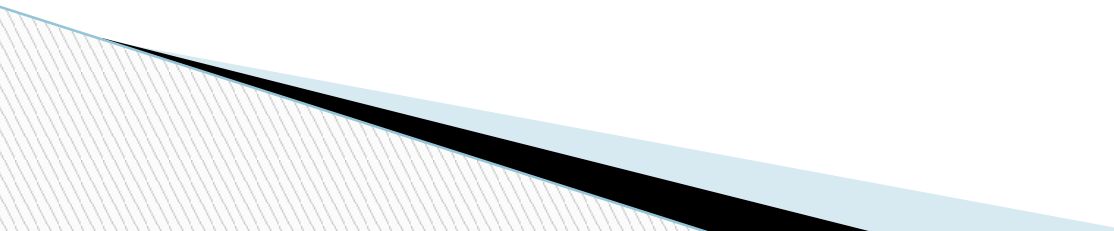
# Fases del desarrollo de una aplicación

## ANÁLISIS

- ? Se establece el producto a desarrollar
- ? Debe existir **gran comunicación**, bilateral, entre el cliente y el analista:
  - El analista debe conocer todas las necesidades que precisa la aplicación.
  - El software debe adaptarse a los recursos existentes: IDEA REALISTA.
- ? Desarrollo de **prototipos** para saber con más precisión sus requerimientos.
- ? **Informe ERS** (Especificación de Requisitos Software)

# Fases del desarrollo de una aplicación

## DISEÑO

- ? Se intenta alcanzar una **solución óptima** de la aplicación, teniendo en cuenta los **recursos físicos** del sistema (tipo de ordenador, periféricos, comunicaciones, etc...) y *los recursos lógicos* (sistema operativo, programas de utilidad, bases de datos, etc...).
  - ? Se define el entorno que requerirá el sistema, aunque también se puede diseñar el sistema en función de los recursos de los que se dispone.
  - ? Se especifica el formato de la información de entrada y salida, las estructuras de datos y la división modular.
  - ? Con todo esto se obtiene el denominado **cuaderno de carga**.
- 

# Fases del desarrollo de una aplicación

## CODIFICACIÓN

- ? Consiste en traducir el diseño a un determinado **lenguaje de programación**, teniendo en cuenta las especificaciones obtenidas en el cuaderno de carga.
- ? No se está exento de necesitar un *reanálisis* o *rediseño* al encontrar un problema al programar el software

# Fases del desarrollo de una aplicación

## Pruebas

- ? Comprobar la calidad y estabilidad del programa:
  - el software no contiene errores
  - el software hace lo que debe hacer.
- ? En general, las pruebas las realiza personal diferente al que codificó la aplicación, con una amplia experiencia en programación, personas capaces de saber en qué condiciones un software puede fallar de antemano sin un análisis previo.

# Fases del desarrollo de una aplicación

## DOCUMENTACIÓN

- Documentación para el usuario
- Documentación técnica
-

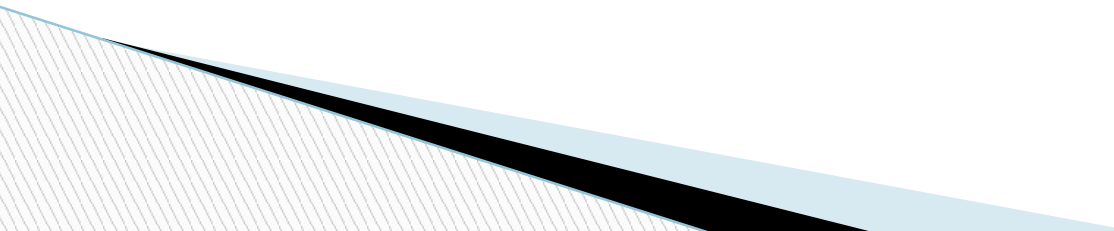


# Fases del desarrollo de una aplicación

## EXPLOTACIÓN

En esta fase se realiza la implantación de la aplicación en el sistema o sistemas físicos donde van a funcionar habitualmente y su puesta en marcha para comprobar el buen funcionamiento.


? Actividades a realizar:

- *Instalación del/los programa/s.*
  - *Pruebas de aceptación al nuevo sistema.*
  - *Conversión de la información del antiguo sistema al nuevo (si hay una aplicación antigua)*
  - *Eliminación del sistema anterior.*
- 

# Fases del desarrollo de una aplicación

## MANTENIMIENTO

Esta es la fase que completa el ciclo de vida de una aplicación.

- **Mantenimiento correctivo:** Consiste en corregir errores no detectados en pruebas anteriores y que aparezcan con el uso normal de la aplicación.
  - **Mantenimiento adaptativo:** Consiste en modificar el programa a causa del cambio del entorno gráfico y lógico en el que estén implantados. Reinicia el ciclo de vida.
  - **Mantenimiento perfectivo:** Consiste en una mejora sustancial de la aplicación al recibir por parte de los usuarios propuestas sobre nuevas posibilidades y modificaciones de las existentes. Reinicia el ciclo de vida.
- 

# Roles que interactúan en el desarrollo

- ? **Analista de sistemas:** Su objetivo consiste en realizar un estudio del sistema para que este cumpla las expectativas del cliente.
- ? **Diseñador de software:** Realiza, en función del análisis de un software, el diseño de la solución que hay que desarrollar.
- ? **Analista programador:** Se suele llamar “desarrollador”, domina una visión más amplia de la programación, aportando una visión general del proyecto más detallada, diseñando una solución más amigable para la codificación y participando activamente en ella.
- ? **Programador:** Se encarga de manera exclusiva de crear el resultado del estudio realizado por analistas y diseñadores. Escribe el código fuente del software.
- ? **Arquitecto de software:** Cohesiona todo el proceso de desarrollo. Conoce e investiga las distintas tecnologías revisando que todo el procedimiento se lleva a cabo de la mejor forma y con los recursos más apropiados.

# Ejercicio Ciclos de vida

- ? Investiga sobre los siguientes ciclos de vida:  
Modelo Incremental, Modelo en Espiral

Modelo Incremental		Modelo en Espiral	
Fases		Fases	
Ventajas		Ventajas	
Inconvenientes		Inconvenientes	
Sistemas		Sistemas	

