

Lógica de programación I

Juan Pablo Restrepo Uribe

Ing. Biomedico

MSc. Automatización y Control Industrial

jprestrepo@correo.iue.edu.co

2023-2

Institución Universitaria de Envigado

Unidad 3: Introducción al lenguaje de programación

Subprogramas - Arreglos

- Definición de programa informático y lenguajes de programación
- Niveles de abstracción
 - lenguaje de máquina
 - lenguaje de bajo nivel
 - lenguaje de alto nivel
- Definición de compiladores y traductores de lenguajes.
- Definición de código fuente y código objeto.
- Paradigmas de programación
- Introducción al lenguaje de programación.
- Tipos de datos, estructuras secuenciales en el lenguaje de programación

Definición de programa informático y lenguajes de programación

Es un conjunto ordenado de instrucciones, escritas en un lenguaje de programación, para realizar una tarea en particular dentro de una computadora.

En palabras sencillas, es una secuencia de órdenes que le indican a una computadora qué hacer.



```
17 string sInput;  
18 int iLength, iN;  
19 double dblTemp;  
20 bool again = true;  
21  
22 while (again) {  
23     iN = -1;  
24     again = false;  
25     getline(cin, sInput);  
26     system("cls");  
27     stringstream(sInput) >> dblTemp;  
28     string str = sInput.length();  
29     iLength = str.length();  
30     if (iLength < 4) {  
31         again = true;  
32         continue;  
33     } else if (sInput[iLength - 3] != '.') {  
34         again = true;  
35         continue;  
36     } while (++iN < iLength) {  
37         if (isdigit(sInput[iN])) {  
38             continue;  
39         } else if (iN == (iLength - 3)) {  
40             continue;  
41         }
```

Definición de programa informático y lenguajes de programación

Finalmente, un lenguaje de programación es la herramienta que permite que una persona (un programador) pueda codificar un cúmulo de instrucciones lógicas (el programa informático), para que una computadora pueda ejecutarla y que esta responda llevando a cabo una serie determinada de acciones



Definición de programa informático y lenguajes de programación



Definición de programa informático y lenguajes de programación

- **Lógico:** Está constituido por un conjunto de procesos lógicos.
- **Intangible:** Es parte del segmento digital (software) de un sistema informático, en contraposición al segmento físico (hardware).
- **Funcional:** Es diseñado para cumplir una tarea o conjunto de tareas.
- **Preciso:** Cumple correctamente con lo programado.
- **Ejecutable:** Se encuentra en un formato para que se pueda accionar dentro una computadora.
- **Secuencial:** El código escrito se ejecuta en un determinado orden.

Definición de programa informático y lenguajes de programación

- Un **programa informático** es una cadena de instrucciones que sigue un patrón lógico para cumplir ciertas tareas definidas
- Un **software** es un paquete informático que puede contener diversos programas informáticos, además de otros elementos, como imágenes, textos, audios, y datos.

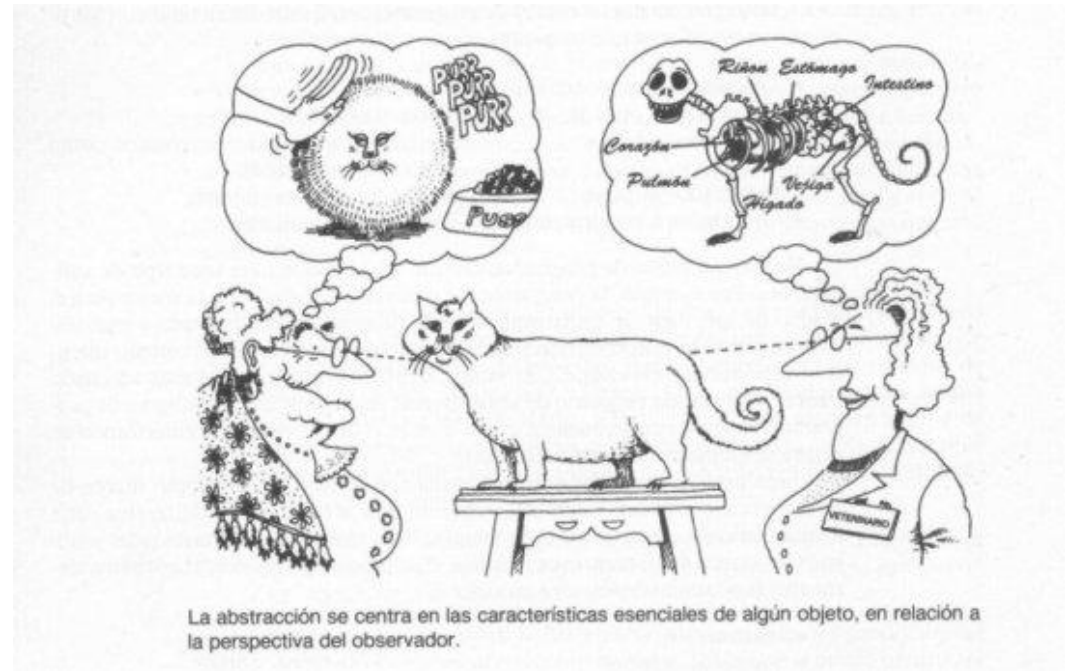


Definición de programa informático y lenguajes de programación

- Realizar cálculos matemáticos.
- Videojuegos.
- Sistemas operativos.
- Procesar texto.
- Reproductores multimedia.
- Visualizar imágenes.
- Crear recordatorios.
- Paquetes ofimáticos
- Navegadores web.
- Agendar citas.
- Aplicaciones de dibujo.
- Antivirus.

Niveles de abstracción

La abstracción es una manera de reducir la complejidad y permitir un diseño e implementación más eficientes en sistemas de software complejos. Oculta la dificultad técnica de los sistemas.



Niveles de abstracción

- Ayuda al usuario a evitar escribir código de bajo nivel.
- Evita duplicar código y aumenta la reusabilidad.
- Se puede cambiar la implementación interna de la clase de forma independiente sin afectar al usuario.
- Ayuda a aumentar la seguridad de la aplicación o programa ya que solo los detalles importantes son proporcionados al usuario.

Niveles de abstracción

El **lenguaje de máquina** o **código máquina** es el sistema de códigos directamente interpretable por un circuito programable, como el microprocesador de una computadora o el microcontrolador de un autómata. Este lenguaje está compuesto por un conjunto de instrucciones que determinan acciones a ser tomadas por la máquina. Un programa consiste en una cadena de estas instrucciones más un conjunto cual se trabaja. Estas instrucciones son normalmente ejecutadas en secuencia, con eventuales cambios de flujo causados por el propio programa o eventos externos. El lenguaje de máquina es específico de la arquitectura de la máquina, aunque el conjunto de instrucciones disponibles pueda ser similar entre arquitecturas distintas.

Niveles de abstracción

```
-u 100 1a
OCFD:0100 BAOB01      MOV     DX,010B
OCFD:0103 B409      MOV     AH,09
OCFD:0105 CD21      INT      21
OCFD:0107 B400      MOV     AH,00
OCFD:0109 CD21      INT      21
-d 10b 13f
OCFD:0100                                48 6F 6C 61 2C
OCFD:0110 20 65 73 74 65 20 65 73-20 75 6E 20 70 72 6F 67
OCFD:0120 72 61 6D 61 20 68 65 63-68 6F 20 65 6E 20 61 73
OCFD:0130 73 65 6D 62 6C 65 72 20-70 61 72 61 20 6C 61 20
OCFD:0140 57 69 6B 69 70 65 64 69-61 24
```

Hola,
este es un prog
rama hecho en as
sembler para la
Wikipedia\$

Lenguaje de máquina del Intel 8088. El código de máquina en hexadecimal se resalta en rojo, el equivalente en lenguaje ensamblador en magenta, y las direcciones de memoria donde se encuentra el código, en azul. Abajo se ve un texto en hexadecimal y ASCII.

Niveles de abstracción

En sentido técnico, el **lenguaje de bajo nivel** incluye el **lenguaje máquina**, pero se refiere más comúnmente a un lenguaje ensamblador que emplea símbolos para crear instrucciones de máquina más fáciles de leer y entender por parte de los programadores. Cada instrucción de un lenguaje ensamblador representa un solo comando para el procesador, lo que permite al desarrollador solamente un bajo nivel de abstracción en cuanto a funciones rutinarias, tales como acceso a memoria y almacenamiento de registro, por lo que tales operaciones requieren suma atención por parte del desarrollador.

Niveles de abstracción

```
SUB AX, AX  
MOV AX, 18D  
SUB AX 18D
```

Código fuente en lenguaje
ensamblador

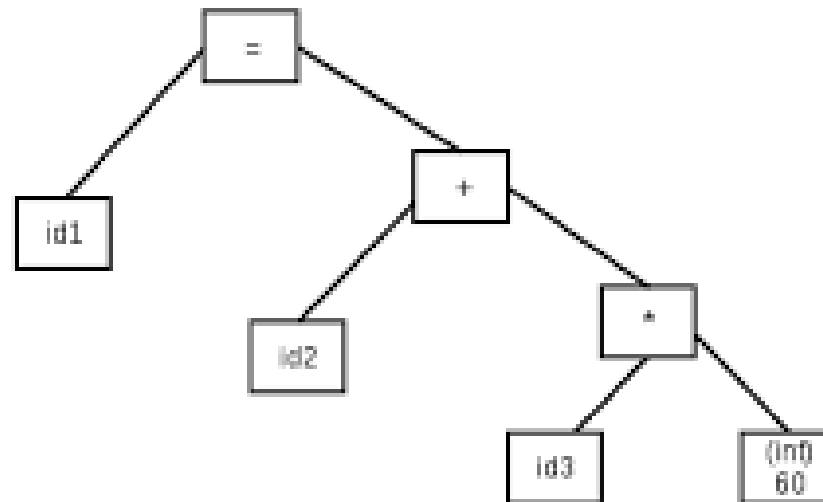
Ensamblador

Traductor

```
0010 0001 1010  
1010 1011 1000  
0110 1101 0001
```

Lenguaje de máquina

Niveles de abstracción



Niveles de abstracción

```
1  section .text
2      global _start
3
4  _start:
5      mov     edx,len
6      mov     ecx,msg
7      mov     ebx,1
8      mov     eax,4
9      int     0x80
10
11     mov     eax,1
12     int     0x80
13
14  section .data
15  msg db 'Hola, mundo!', 0xa
16  len equ $ - msg
```

https://www.tutorialspoint.com/compile_assembly_online.php

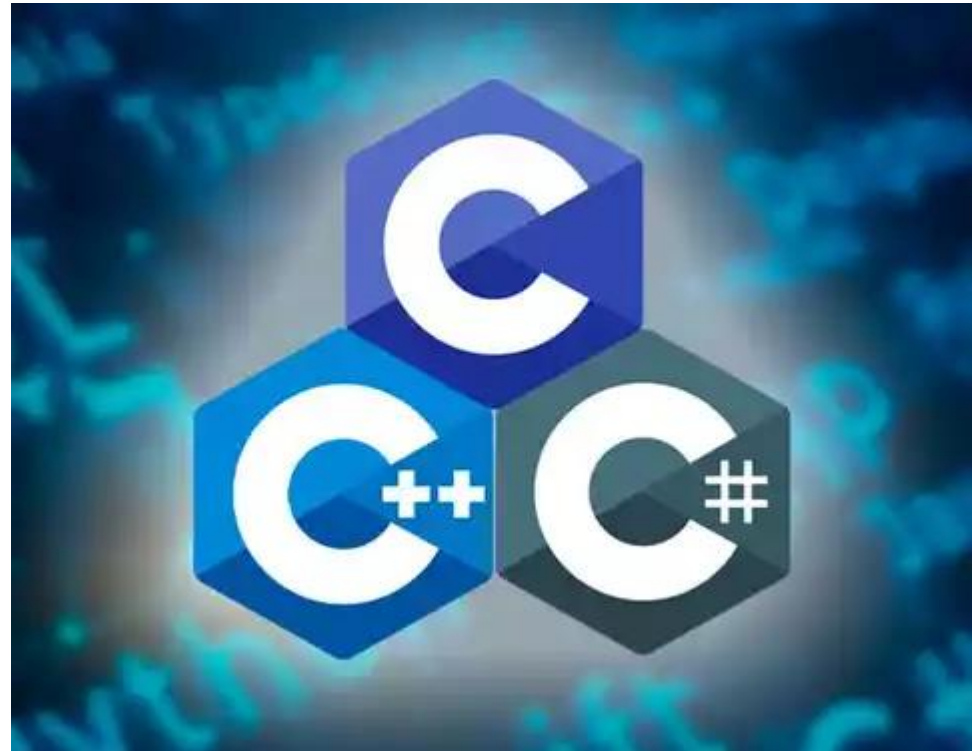
Niveles de abstracción

Un registro es un pequeño espacio de almacenamiento dentro de una unidad de procesamiento central (CPU) que se utiliza para almacenar temporalmente datos que están siendo procesados activamente.

- **Registros de propósito general:** Se utilizan para almacenar datos temporales y resultados de operaciones aritméticas y lógicas.
- **Contador de programa (PC):** Almacena la dirección de la siguiente instrucción que se va a ejecutar.
- **Registro de instrucción (IR):** Almacena la instrucción actual que se está ejecutando.
- **Registro de dirección de memoria (MAR):** Almacena la dirección de memoria de la ubicación de memoria que se va a leer o escribir.
- **Registro de datos (MDR):** Almacena los datos que se van a escribir en la memoria o los datos que se han leído de la memoria.

Niveles de abstracción

- Comprensión de los sistemas computacionales
- Base muy sólida para programar en otros entornos
- Aprendizaje rápido de otros lenguajes de programación
- Uso común en ámbito técnico
- Rápida ejecución.

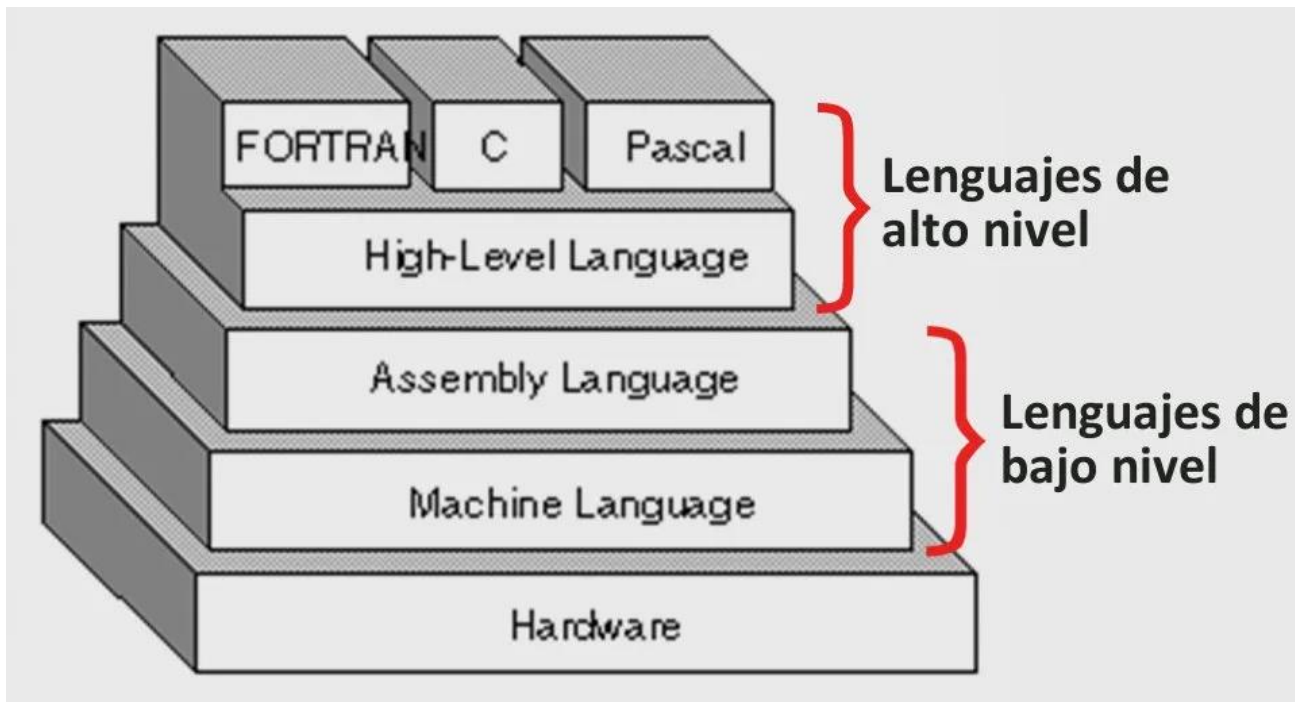


Niveles de abstracción

Un lenguaje de programación de **alto nivel** se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de la capacidad con que las máquinas lo ejecutan. Algunos ejemplos son: Java, PHP, Python, Javascript, C#.



Niveles de abstracción

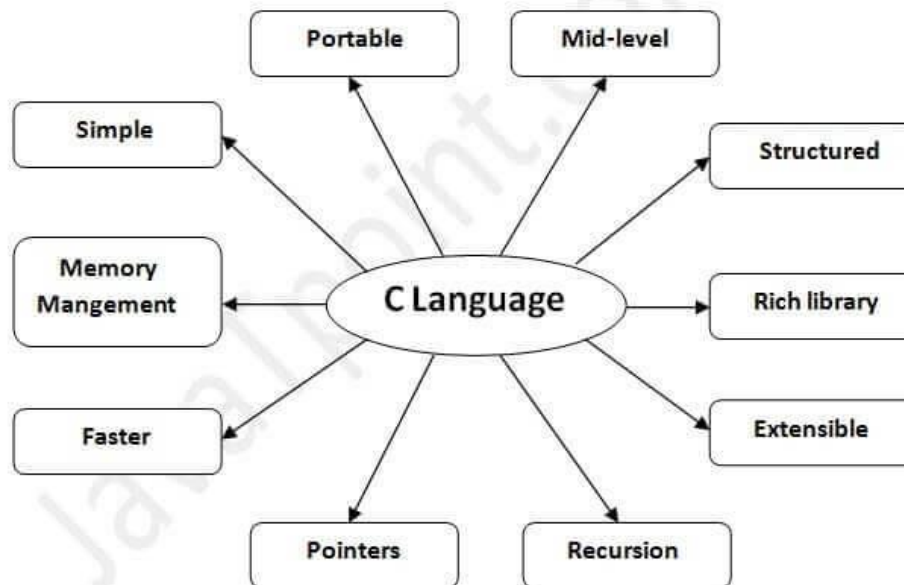


Niveles de abstracción

Lenguajes de medio nivel

Continuando en la escala de clasificación de lenguajes, se encuentran los de nivel medio, cuyas funciones suelen tener mayor complejidad y son implementados en tareas relacionadas al desarrollo de software y manejo de sistemas informáticos.

Niveles de abstracción



Niveles de abstracción

```
/*Programa que permita elegir al usuario si desea convertir una cantidad
de euros a pesetas o realizar una conversión de pesetas a euros*/
#include <stdio.h>

int main (void){

    int opcion;
    float Euros, Pesetas;

    printf("\n 1: Euros a Pesetas");
    printf("\n 2: Pesetas a Euros\n");
    printf("\nEscoja la opcion deseada: ");
    scanf("%d", &opcion);

    if(opcion==1){
        printf("\n\n Introduzca la cantidad en Euros: ");
        scanf("%f", &Euros);
        printf("\n %f Euros son %f Pesetas", Euros, Euros*166.386);
    }else{
        printf("\n\n Introduzca la cantidad en Pesetas: ");
        scanf("%f", &Pesetas);
        printf("\n %f Pesetas son %f Euros", Pesetas, Pesetas/166.386);
    }
    return 0;
}
```

Niveles de abstracción

| Lenguaje | Descripción | Usado en |
|--|--|---|
|  Lenguaje C | <ul style="list-style-type: none"> • Usado para desarrollar sistemas operativos y controlar dispositivos. • Modelo para la generación de procedimientos. • Lenguaje de propósito general. |  Kernel de Linux |
|  Lenguaje C++ | <ul style="list-style-type: none"> • Modelo de programación orientado a objetos y funcional. • Extensión de C. • Lenguaje de más alto nivel (abstrae detalles al desarrollador) |  Navegadores como Firefox y Chrome |
|  Lenguaje Javascript | <ul style="list-style-type: none"> • Es el código que permite la interacción del usuario con interfaces web. • Tecnología clave en el desarrollo de aplicaciones web. • En la actualidad, se usa en otras áreas gracias a ambientes de ejecución con NodeJS |  Aplicaciones Web modernas y Backend de Netflix |
|  Lenguaje Java | <ul style="list-style-type: none"> • Lenguaje multipropósito y de alto nivel. • Modelo orientado a objetos. • Independiente de la plataforma gracias a la máquina virtual de Java. |  Airbnb Backend |
|  Lenguaje Python | <ul style="list-style-type: none"> • Automatiza tareas con scripts en modelo procedural. • Desarrollo backend. P. ej., usando Django. • Lenguaje multipropósito. • Ciencias de datos gracias a un extenso catálogo de herramientas disponibles. |  Airbnb Backend |
|  Lenguaje PHP | <ul style="list-style-type: none"> • Usado principalmente en desarrollo web. • Las páginas HTML se generan en un servidor que interpreta el código PHP. |  Versiones iniciales de Facebook. |

Definición de compiladores y traductores de lenguajes.

A la hora de elegir un lenguaje de programación, se debe prestar especial atención a dos cosas:

- El lenguaje debe contar con todos los componentes básicos necesarios para el proyecto de software que se quiera desarrollar.
- Tiene que permitir programar e implementar este proyecto de la manera más sencilla posible.

La buena legibilidad y simplicidad del código fuente son fundamentales para garantizar lo segundo, porque estas características no solo facilitan el aprendizaje del lenguaje de programación, sino también, obviamente, su posterior utilización en el día a día.

Definición de compiladores y traductores de lenguajes.

Un **intérprete** es un programa informático que procesa el código fuente de un proyecto de software durante su tiempo de ejecución, es decir, mientras el software se está ejecutando, y actúa como una interfaz entre ese proyecto y el procesador.

El proceso de conversión no finaliza hasta que se ha interpretado todo el código. Solo se interrumpe prematuramente si se produce un fallo durante el procesamiento, lo que simplifica mucho la resolución de los errores, ya que la línea de código problemática se detecta inmediatamente después de ocurrir el fallo.

Definición de compiladores y traductores de lenguajes.

Un **compilador** es un programa informático que traduce todo el código fuente de un proyecto de software a código máquina antes de ejecutarlo. Solo entonces el procesador ejecuta el software, obteniendo todas las instrucciones en código máquina antes de comenzar. De esta manera, el procesador cuenta con todos los componentes necesarios para ejecutar el software, procesar las entradas y generar los resultados

https://www.youtube.com/watch?si=Nbp mrb_wLU8sNcKH&v=ojqusTBq4A0&feature=youtu.be

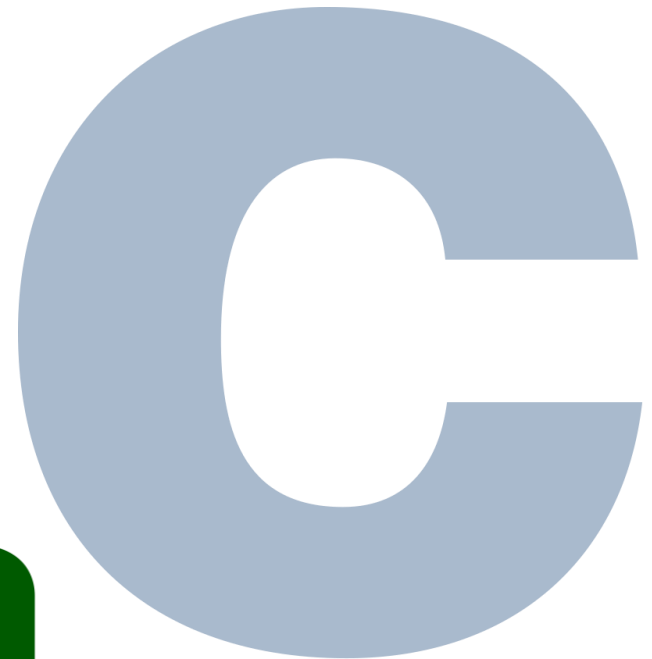
Definición de compiladores y traductores de lenguajes.

| | Intérprete | Compilador |
|--|---|---|
| Momento en que se traduce el código fuente | Durante el tiempo de ejecución del software | Antes de ejecutar el software |
| Procedimiento de traducción | Línea por línea | Siempre todo el código |
| Presentación de errores de código | Después de cada línea | En conjunto, después de toda la compilación |
| Velocidad de traducción | Alta | Baja |
| Eficiencia de traducción | Baja | Alta |
| Coste de desarrollo | Bajo | Alto |
| Lenguajes típicos | PHP, Perl, Python, Ruby, BASIC | C, C++, Pascal |

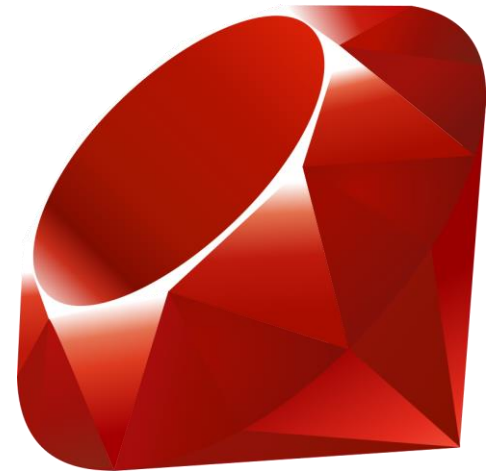
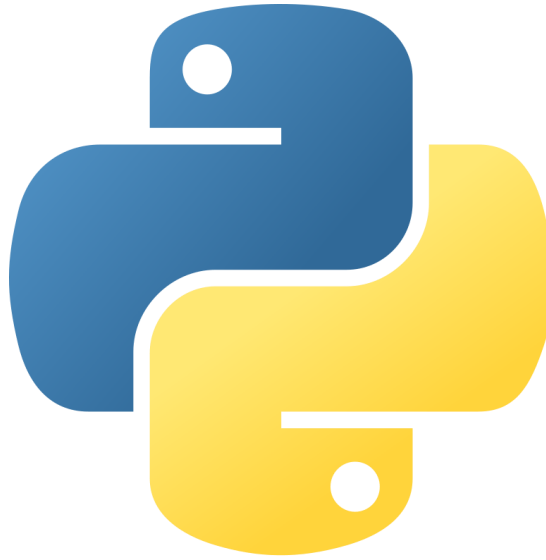
Definición de compiladores y traductores de lenguajes.



In Strong Typing We Trust



Definición de compiladores y traductores de lenguajes.



Editores de código

Los editores de código son las herramientas que todo desarrollador debe tener a mano. Permiten editar código fuente en diversos lenguajes de programación y ofrecen múltiples herramientas para facilitar el trabajo y aumentar la productividad.

Debemos distinguir entre editores de código e IDE. Los editores generalmente son programas ligeros, que ofrecen lo necesario para poder ser productivos y tener una experiencia de desarrollo adecuada, pero sin complicaciones.

Sin embargo, los editores actuales se pueden extender tanto como se quiera, por medio de complementos que los pueden hacer llegar a ser tan avanzados como los IDE.

Editores de código



PyCharm

<https://desarrolloweb.com/coleccion/es/editores-codigo>

Editores de código



ANACONDA®

Definición de código fuente y código objeto.

Se llama **código objeto** al código que resulta de la compilación del **código fuente**. Puede ser en lenguaje máquina o bytecode, y puede distribuirse en varios archivos que corresponden a cada código fuente compilado. Luego un enlazador (linker) se encarga de juntar todos los archivos de código objeto para obtener el programa ejecutable.

Definición de código fuente y código objeto.

El **código fuente** de un programa informático es un conjunto de líneas de texto con los pasos que debe seguir la computadora para ejecuta.

El **código fuente** de un programa está escrito por un programador en algún lenguaje de programación legible por humanos, normalmente en forma de texto plano. Sin embargo, el programa escrito en un lenguaje legible por humanos no es directamente ejecutable por la computadora en su primer estado, sino que debe ser traducido a otro lenguaje o código binario; así será más fácil para la máquina interpretarlo (lenguaje máquina o código objeto que sí pueda ser ejecutado por el hardware de la computadora). Para esta traducción se usan los llamados compiladores, ensambladores, intérpretes y otros sistemas de traducción.

Paradigmas de programación

Un paradigma 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.

Los lenguajes de programación adoptan uno o varios paradigmas en función del tipo de órdenes que permiten implementar como, por ejemplo, Python o JavaScript, que son multiparadigmas.

Paradigmas de programación: Imperativo

Los programas consisten en una sucesión de instrucciones o conjunto de sentencias, como si el programador diera órdenes concretas.

El desarrollador describe en el código paso por paso todo lo que hará su programa.

- **Programación estructurada:** La programación estructurada es un tipo de programación imperativa donde el flujo de control se define mediante bucles anidados, condicionales y subrutinas, en lugar de a través de **GOTO**.
- **Programación procedimental:** Este paradigma de programación consiste en basarse en un número muy bajo de expresiones repetidas, englobarlas todas en un procedimiento o función y llamarlo cada vez que tenga que ejecutarse.
- **Programación modular:** consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más manejable y legible. Se trata de una evolución de la programación estructurada para resolver problemas de programación más complejos.

Paradigmas de programación: Declarativo

Este paradigma no necesita definir algoritmos puesto que describe el problema en lugar de encontrar una solución al mismo. Este paradigma utiliza el principio del razonamiento lógico para responder a las preguntas o cuestiones consultadas.

- Programación Lógica: Prolog
- Programación funcional: Lisp, Scala, Java, Kotlin

Paradigmas de programación

Ventajas

Código más corto y eficiente.

Realizable con métodos no conocidos en el momento de la programación.

Optimización sencilla, ya que la ejecución se gestiona mediante un algoritmo.

Es posible el mantenimiento independiente del desarrollo de la aplicación.

Desventajas

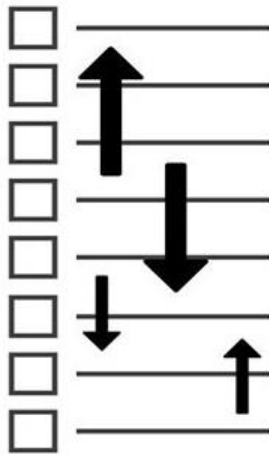
En parte, difícil de comprender para personas ajenas.

Basado en una forma de pensar no habitual en las personas (estado de solución).

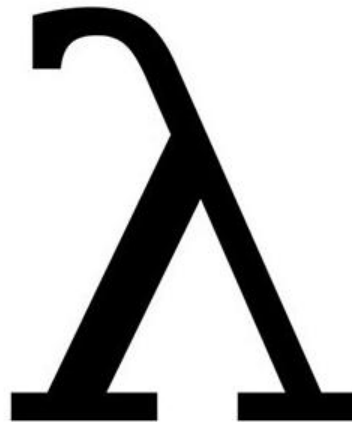
Las características de casos de aplicación individuales se pueden considerar en la programación, pero únicamente de forma compleja.

Paradigmas de programación

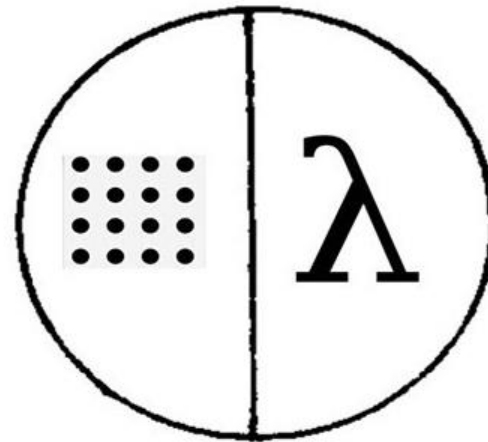
Imperative



Functional

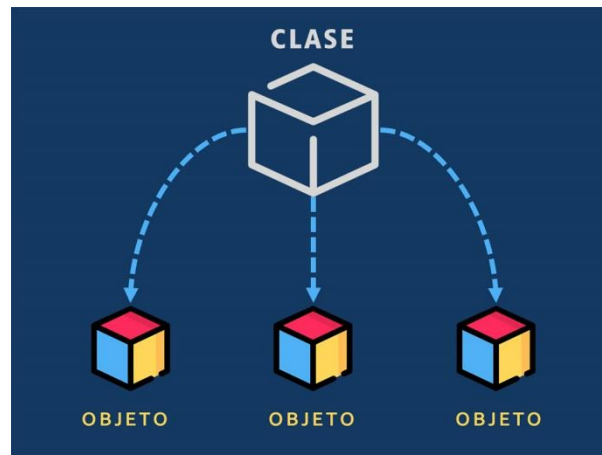


Object-Oriented

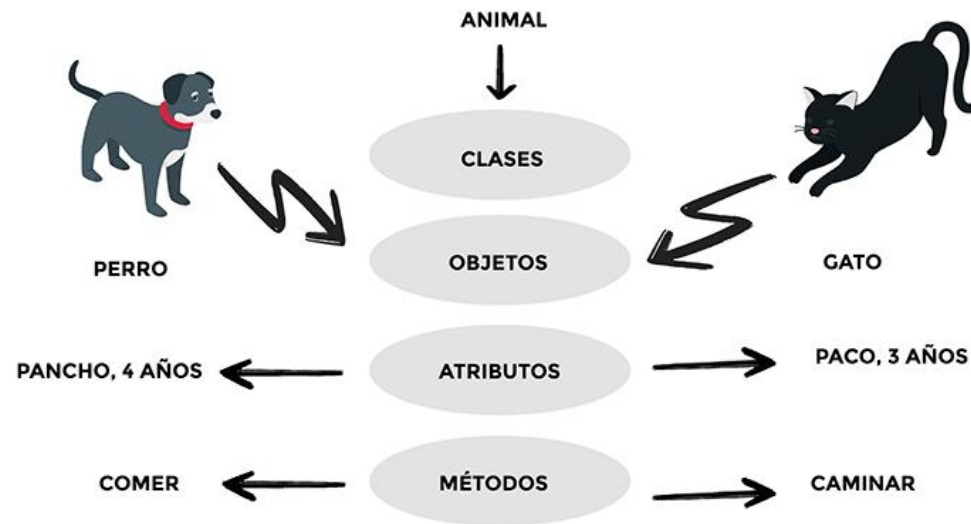


Paradigmas de programación: POO

En este modelo de paradigma se construyen modelos de objetos que representan elementos (objetos) del problema a resolver, que tienen características y funciones. Permite separar los diferentes componentes de un programa, simplificando así su creación, depuración y posteriores mejoras. La programación orientada a objetos disminuye los errores y promueve la reutilización del código. Es una manera especial de programar, que se acerca de alguna manera a cómo expresaríamos las cosas en la vida real.



Paradigmas de programación: POO



Paradigmas de programación: Reactiva

Este paradigma se basa en escuchar lo que emite un evento o cambios en el flujo de datos, en donde los objetos reaccionan a los valores que reciben de dicho cambio. Las librerías más conocidas son Project Reactor, y RxJava. React/Angular usan RxJs para hacer uso de la programación reactiva.

Unidad 3: Introducción al lenguaje de programación

Subprogramas - Arreglos

Estructuras selectivas y repetitivas en el lenguaje de programación

Textos

- Todo String en Python tiene 3 propiedades que exploraremos en detalle en esta sección.
- Los Strings contienen caracteres, los cuales son letras o símbolos individuales.
- Los Strings tienen un tamaño (length), el cual es el número de caracteres que contiene el String.
- Los caracteres en un String aparecen en una secuencia ascendente. Lo cuál significa que cada carácter tiene un número de posición (también llamado índice) dentro del String. Los índices inician desde el número 0.

Textos

mensaje

| | | | | | | | | | | |
|---|---|---|---|---|--|---|---|---|---|---|
| H | o | l | a | , | | G | r | u | p | o |
|---|---|---|---|---|--|---|---|---|---|---|

posición

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|

Textos

- `len()` -> para obtener el tamaño de un texto.
- `mensaje[pos]` -> para extraer un carácter de un texto.
- `mensaje[posInicial:posFinal+1]` -> para extraer un subtexto de un texto.
- `mensaje.find("textoABuscar")` -> para buscar un carácter o subtexto dentro de un texto.
- `mensaje.find("textoABuscar", posInicial)` -> para buscar un carácter o subtexto dentro de un texto basado en una posición inicial.

<https://docs.python.org/3/library/stdtypes.html#string-methods>

Textos

```
1 mensaje = "Hola, Grupo"
2
3 # devuelve la cadena en minúscula
4 mensajeMinus = mensaje.lower()
5 print(mensajeMinus)
6
7 # devuelve la cadena en mayúscula
8 mensajeMayus = mensaje.upper()
9 print(mensajeMayus)
10
11 # cuenta cuantas veces aparece la "o" en mensaje
12 conteo = mensaje.count("o")
13 print(conteo)
14
15 # devuelve la cadena reemplazando las "o" por "u"
16 nuevoMensaje = mensaje.replace("o","u")
17 print(nuevoMensaje)
```


Ejercicios

- Escribe un programa que solicite al usuario ingresar una cadena y luego cuente cuántas vocales (a, e, i, o, u) hay en la cadena. Luego, muestra el resultado.
- Escribe un programa que verifique si una palabra ingresada por el usuario es un palíndromo (se lee igual de izquierda a derecha y de derecha a izquierda - palabra[::-1]).
- Escribe un programa que reemplace todas las letras "a" en una cadena ingresada por el usuario por el carácter "x" y muestre la cadena resultante.
- Escribe un programa que cuente cuántas letras mayúsculas y cuántas letras minúsculas hay en una cadena ingresada por el usuario
(https://www.w3schools.com/python/ref_string_isupper.asp,
https://www.w3schools.com/python/ref_string_islower.asp)

Ejercicios

- Escribe un programa que tome una frase ingresada por el usuario y genere un acrónimo utilizando la primera letra de cada palabra en mayúsculas.
- Escribe un programa que cuente cuántas palabras hay en una oración ingresada por el usuario. Puedes asumir que las palabras están separadas por espacios.