

# INFORME DE LENGUAJES, PARADIGMAS Y ESTÁNDARES DE PROGRAMACIÓN

Pablo Nicolás Soto Irago  
Computer Science & AI

# ÍNDICE

I 2 3 4 5

Introducción

Lenguajes de  
programación

Paradigmas de  
la programación

Estándares de  
la programación

Conclusión

3

4-7

8-9

10-12

13

# INTRODUCCIÓN

El mundo de la programación es un campo tecnológico, complejo y en constante crecimiento. Programar se puede definir como el acto de comunicarse con herramientas o máquinas a través de órdenes y algoritmos, usando un lenguaje entendible por ambos, programador y máquina, conocido como lenguaje de programación. Además, la programación puede tener diferentes usos según su finalidad, estilo o expresión, también llamados paradigmas o estándares de la programación. Todos estos elementos son imprescindibles para que se pueda programar correctamente.

# LENGUAJES DE PROGRAMACIÓN

```
langueta-laptop:~/Backup/Programs/CodeCall/Tutorials/asm/intro$ bimage
=====
          bimage
      Disk Image Creation Tool for Bochs
  $Id: bimage.c,v 1.33 2000/02/05 22:57:42 salwerth Exp $
=====

Want to create a floppy disk image or a hard disk image?
Type hd or fd. [hd] fd

The size of floppy disk image to create, in megabytes.
Type 0.18, 0.38, 0.52, 0.72, 1.2, 1.44, 1.68, 1.72, or 2.88.
0.38
Create a floppy image with
1
4 per track=8
sectors=320
bytes=163840

Add I name the image?
intro.img

[] Done.

163840 bytes to intro.img.

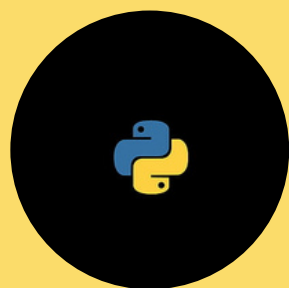
Writing image should appear in your bochsrc:
as image="intro.img", status="inserted"
langueta-laptop:~/Backup/Programs/CodeCall/Tutorials/asm/intro$
```

Los lenguajes de programación han ido cambiando y mejorando en su campo con el paso del tiempo. Esta evolución se suele percibir en como los lenguajes actuales varían según el nivel de abstracción, es decir, los más actuales suelen ser lenguajes más acordes con la forma de comunicación humana. El nivel de abstracción de cada lenguaje nos permite categorizarlos en varios niveles:

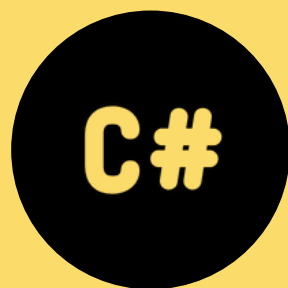
- Alto nivel
- Medio nivel
- Bajo nivel

# Lenguajes de alto nivel

Los lenguajes del nivel alto son los que están más abstraídos, es decir, son los que están más alejados del lenguaje máquina; se acercan más al lenguaje humano, siendo más fáciles de comprender. Algunos ejemplos son:



PYTHON



C#

## PYTHON

Se usa principalmente en el campo de la ciencia de datos y el aprendizaje automático, aunque es un lenguaje muy versátil.

```
def main():  
    palabras =  
    ["Hello", "World",  
    "in", "Python"]  
    for palabra in  
    palabras:  
  
    print(palabra)  
    if __name__ ==  
    "__main__":  
        main()
```

## C #

Se usa principalmente en el desarrollo de software: aplicaciones empresariales o sistemas integrados.

```
class Program{  
    static void  
    Main(){  
        string[]  
        palabras = {  
        "Hello", "World",  
        "in", "C#" };  
        foreach  
        (string palabra  
        in palabras)  
        {  
            Console.WriteLi  
            ne(palabra);  
        }  
    }  
}
```

# Lenguajes de medio nivel

Los lenguajes de programación del anterior nivel, nivel medio, son los lenguajes que surgen con el fin de facilitar la comprensión del lenguaje máquina a través del lenguaje ensamblador. Además, los lenguajes de nivel medio aumentan notablemente las posibilidades para programar, a través de nuevas funciones, frente a las posibilidades que ofrecen los lenguajes de nivel bajo. Algunos ejemplos son:



```
#include  
<stdio.h>  
int main() {  
    printf("Hello,  
World!\n");  
    return 0;}
```

Se usa principalmente para el desarrollo de sistemas operativos y la programación de redes. Es uno de los lenguajes más eficientes para acceder directamente a las funciones del software, y también es muy usado por su distintivo dominio de los punteros.

## BASIC

Se suele usar en desarrollo de software, pero es conocido por ser un lenguaje de enseñanza e introducción a la programación. Esto ocurre por su gramática simple y por ser dinámicamente tipado.

```
10 PRINT "¿Cuál  
es tu nombre?";  
20 INPUT  
nombre$  
30 PRINT "Hola, ";  
nombre$; "!  
Bienvenido a  
BASIC."  
40 END
```

```
100 GOTO 130  
110 PRINT "GUESSING GAME"  
120 PRINT "BY THOMAS E. KURTZ"  
130 PRINT "GUESS THE NUMBER BETWEEN 1 AND 100."  
140 LET X = RND(100)  
150 LET N = 0  
160 PRINT "YOUR GUESS?"  
170 INPUT G  
180 LET N = N+1  
190 IF G = X THEN GOTO 250  
200 IF G < X THEN GOTO 230  
210 PRINT "TOO LARGE, GUESS AGAIN"  
220 GOTO 160  
230 PRINT "TOO SMALL, GUESS AGAIN"  
240 GOTO 160  
250 PRINT "YOU GUESSED IT, IN", N, "TRIES"  
260 PRINT "ANOTHER GAME (YES = 1, NO = 0)"  
270 INPUT A  
280 IF A = 1 THEN GOTO 140  
290 PRINT "THANKS FOR PLAYING! YOU'VE BEEN GREAT!"  
300 END
```



# Lenguajes de bajo nivel

Los lenguajes de bajo nivel son los menos abstraídos y parecidos al lenguaje básico de la máquina. Estos lenguajes son los más exactos respecto a los lenguajes de más alto nivel, Los lenguajes más representativos de este nivel son:

- Lenguaje máquina
- Lenguaje ensamblador

## MÁQUINA

Es poco común y complejo programar con este lenguaje ya que es directamente el lenguaje de la máquina (como su nombre indica). En el caso de programar con este lenguaje, las actividades más común es la optimización de código. También sirve en procesos de depuración avanzada, a nivel de máquina.

01001000 01101111 01101100 01100001



## ENSAMBLADOR

Este lenguaje es una traducción directa del lenguaje máquina usando abreviaciones o mnemónicos (palabra que sustituye a un código de operación). También se usa para la optimización de código, y para el desarrollo de sistemas operativos.



```
.model small
.stack
.data
Cadena1 DB
'Hola Mundo.$'
.code
programa:
    mov ax, @data
    mov ds, ax
    mov dx, offset
Cadena1
    mov ah, 9
    int 21h
    int 20h
end programa
```

# PARADIGMAS DE LA PROGRAMACIÓN

Los paradigmas de programación se refieren a las metodologías empleadas para llevar a cabo tareas concretas y proyectos. En términos sencillos, constituyen enfoques de desarrollo de software que se utilizan con el propósito de abordar problemas en sistemas o lograr los resultados deseados.

En el mundo de la programación, podemos encontrar muchos paradigmas, pero los más conocidos y usados son:

## *PROGRAMACIÓN IMPERATIVA*

También se conoce como paradigma de procedimientos. Como dice su nombre, se usa principalmente para la creación y el modelaje de programas a través de procedimientos. Se caracteriza principalmente por enseñar a la máquina como hacer trabajos secuencialmente. Algunos lenguajes que usan este paradigma son C o Algol.

## *PROGRAMACIÓN DECLARATIVA*

Se caracteriza por ser una forma de programar centrada en consultas, ya que se utiliza para definir que se quiere conseguir o encontrar, a través de reglas o relaciones, en vez de centrarse en como conseguir un objetivo. Algunos lenguajes usados son SQL o Erlang.

## *PROGRAMACIÓN ORIENTADA A OBJETOS (OOP):*

Los programas están hechos a base de la agrupación lógica de objetos. A su vez, los objetos se determinan a partir de los datos que contienen y su conducta. Algunos lenguajes que siguen este paradigma son Python o C++.



# PARADIGMAS DE LA PROGRAMACIÓN

*PROGRAMACIÓN FUNCIONAL: Este paradigma se enfoca en manejar los programas como una valoración de funciones matemáticas, procurando mantener la estabilidad de los valores constantes. Los lenguajes más comunes dentro de este tipo de programación son Haskell o Scheme.*

*PROGRAMACIÓN LÓGICA: Como dice su nombre, se centra en la programación lógica a través de pautas y afirmaciones. Se usa cada vez más en la programación de inteligencias artificiales. Prolog o Mercury.*

*PROGRAMACIÓN ORIENTADA A ASPECTOS (AOP): Es una programación innovadora centrada en el modelaje de las aplicaciones, potenciando la división de los objetos más repercusivos transversalmente en toda la estructura del programa. El lenguaje más usado es AspectJ (extensión de Java).*

*PROGRAMACIÓN CONCURRENTE: Este paradigma se caracteriza por ser una forma polivalente de programar, es decir, se pueden completar varias acciones a la vez. Los lenguajes Ada o Java (multiparadigma) son algunos ejemplos utilizados.*

# ESTÁNDARES DE LA PROGRAMACIÓN

Los estándares de la programación son un conjunto de normas, reglas o directrices que siguen los programadores para escribir un código bien estructurado. Los estándares se han creado con la finalidad de facilitar la comprensión y lectura del código, además de ayudar a los programadores a entender la programación de scripts. Estos estándares suelen variar según el lenguaje y su gramática.

Es importante seguir estándares para que el código tenga consistencia, asegurando que el código mantenga un estilo constante y una estructura coherente. Además, los estándares facilitan su lectura (limpio y legible), y consecuentemente su mantenimiento y revisión. Todo esto conlleva a una mayor eficiencia reflejada en el diseño, la fiabilidad y la seguridad del código. Algunos estándares populares son los siguientes.

# EJEMPLOS

Block Element Modifier (BEM): Es un estándar especializado para CSS. Su función consiste en dividir por categorías, en bloques de código, el programa para facilitar su entendimiento y mejorar el desarrollo web.

Airbnb JavaScript Style Guide: Se puede considerar como un conjunto de normas en JavaScript que regulan la gramática, para conseguir un código limpio y ordenado. Se usa mucho para establecer convenciones para garantizar consistencia en un proyecto.

ISO 9001: Uno de los principales estándares para la gestión de calidad. Esta herramienta se suele usar en empresas con el objetivo de mejorar el negocio, facilitar el control de calidad en todos los procesos empresariales, y aumentar aspectos fundamentales, como la rentabilidad.

IEEE 730 – 2002 Standard: Define el software de alta calidad. Se centra en declarar las condiciones mínimas para crear y elaborar planes de garantía de calidad de software. Sus funciones principales son el “testing”, las mediciones y revisiones, el informe de problemas o la gestión de riesgos.

La utilización de estándares de la programación nos beneficia dado que nos permite realizar trabajos con mayor facilidad, aumenta la legibilidad del código, acelerando su mantenimiento y su compatibilidad entre distintas plataformas.

En el caso de no seguir estándares, nuestro código podría tener fallos que ralentizarían su desarrollo, además de que dentro del mismo código se podrían crear dificultades a la hora de combinar y/o unir varias secciones de código.



# UTILIZACIÓN ESTÁNDARES DE LA PROGRAMACIÓN

# CONCLUSIÓN

La complejidad y diversidad de lo que se conoce como programación, nos impide desarrollar programas con completa libertad. Esto ocurre porque la manera de que las máquinas nos entiendan es a través de los lenguajes de programación. Pero nuestra creación llega mucho más allá de un programa o una base de datos, por ello se establecen, según su necesidad, ciertos estándares y paradigmas. Sin estos tres puntos básicos no se podría programar, y el mundo de la programación se convertiría en algo poco coherente.



# FUENTES UTILIZADAS

## LENGUAJES DE PROGRAMACIÓN:

## PARADIGMAS DE LA PROGRAMACIÓN:

## ESTÁNDARES DE LA PROGRAMACIÓN:

- **APUNTES DE CLASE:**

<https://speckle-porpoise-32d.notion.site/Introducci-n-a-la-programaci-n-906cf521f30f41d9893cd2098b41cd87?pvs=4&authuser=3>

- **EJEMPLO DEL LENGUAJE ENSAMBLADOR:**

<https://arquitectura-17.webnode.com.co/lenguaje-ensamblador/>

- **CHATGPT:** Ejemplos propuestos de lenguajes de programación.

- **PROGRAMACIÓN CONCURRENTE:**

<https://jeisson.ecci.ucr.ac.cr/concurrente/material/#:~:text=El%20paradigma%20de%20programaci%C3%B3n%20concurrente,los%20procesos%20para%20resolver%20problemas.>

[https://www.ctr.unican.es/asignaturas/procodis\\_3\\_ii/doc/procodis\\_1\\_02.pdf](https://www.ctr.unican.es/asignaturas/procodis_3_ii/doc/procodis_1_02.pdf)

- **APUNTES DE CLASE:**

<https://speckle-porpoise-32d.notion.site/Introducci-n-a-la-programaci-n-906cf521f30f41d9893cd2098b41cd87?pvs=4&authuser=3>

<https://cognosonline.com/co/blog/que-son-paradigmas-de-programacion/#:~:text=En%20programaci%C3%B3n%20se%20conocen%20como,llugar%20a%20los%20resultados%20esperado>

- **AOP:**

[https://seminariomirianandres.unirioja.es/2018/01/17/una-breve-introduccion-a-la-programacion-orientada-a-aspectos/#:~:text=La%20programaci%C3%B3n%20orientada%20a%20aspectos%20\(aspect%20orientado%20programming%20o%20AOP,transversales%20a%20todo%20el%20sistema](https://seminariomirianandres.unirioja.es/2018/01/17/una-breve-introduccion-a-la-programacion-orientada-a-aspectos/#:~:text=La%20programaci%C3%B3n%20orientada%20a%20aspectos%20(aspect%20orientado%20programming%20o%20AOP,transversales%20a%20todo%20el%20sistema)

<https://icariatechnology.com/la-calidad-del-software-y-sus-estandares-mas-importantes/>

- **CHATGPT:** Definición y ejemplos

- **BLOCK ELEMENT MODIFIER:**

<https://en.bem.info/methodology/quick-start/>

- **AIRBNB JAVASCRIPT STYLE GUIDE:**

<https://www.makeuseof.com/eslint-with-airbnb-javascript-style-guide/#:~:text=The%20Airbnb%20style%20guide%20is,style%20guides%20for%20JavaScript%20projects.>

- **ISO 9001:**

<https://www.bsigroup.com/es-ES/Gestion-de-Calidad-ISO-9001/>

- **IEEE 730:**

<https://ieeexplore.ieee.org/document/1040117>

<https://www.uv.mx/personal/ermeneses/files/2018/02/7-IEEE-730.pdf>