

LENGUAJE DE PROGRAMACIÓN



**UNIVERSIDAD
DEL PACÍFICO**



Jaminton Asprilla A.

LENGUAJE DE PROGRAMACIÓN

Un lenguaje de programación es un sistema estructurado compuesto por reglas gramaticales, símbolos y sintaxis que permiten a los desarrolladores escribir instrucciones comprensibles para una computadora.

Estas instrucciones, conocidas como código, indican a la máquina cómo ejecutar tareas específicas de manera precisa y eficiente.

```
8 # Use the json module to parse the json object
9 theJSON = json.loads(myJSON)
10
11 # Now we can access the members of the dictionary
12 if "name" in theJSON:
13     print(theJSON["name"])
14
15 # output the number of members in the dictionary
16 count = theJSON["members"]
17 print(str(count) + " members")
```

En esencia, el lenguaje de programación actúa como un intermediario entre el pensamiento humano y el lenguaje binario que las computadoras pueden entender.

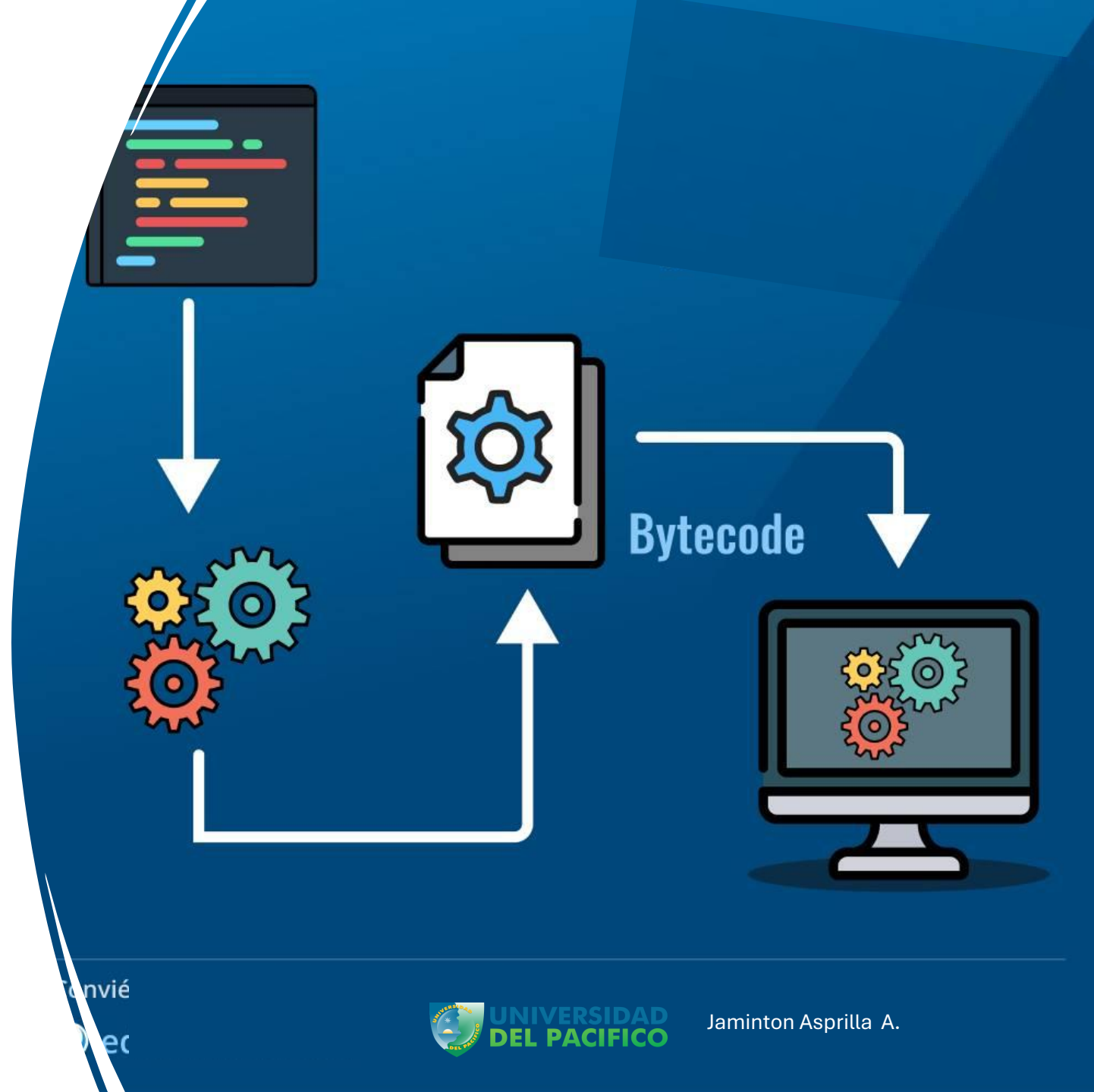
Gracias a estos lenguajes, es posible desarrollar todo tipo de software: desde aplicaciones móviles y páginas web hasta videojuegos, sistemas operativos y programas empresariales, cómo un chatbot o herramientas de social listening para las redes sociales.

En el contexto de la informática moderna, el lenguaje de programación es una herramienta indispensable ya que permite transformar ideas en soluciones digitales funcionales, facilitando la innovación tecnológica en múltiples sectores.

NIVELES DEL LENGUAJE DE PROGRAMACIÓN

• Un lenguaje de programación puede clasificarse según su grado de abstracción en dos grandes categorías:

- Lenguajes de bajo nivel y
- Lenguajes de alto nivel.



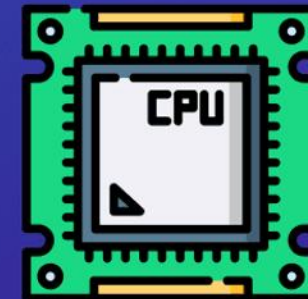


Programador

ALTO NIVEL



BAJO NIVEL



**CPU
(Hardware)**

```
7 # Print the JSON object to the console
8 theJSON = json.loads(response)
9
10
11 # Now we can access the JSON object
12 if "title" in theJSON:
13     print(theJSON["title"])
14
15 # Output the number of items in the JSON object
16 count = theJSON["count"]
17 print(count)
```

Los lenguajes de bajo nivel, como el lenguaje de máquina o el lenguaje ensamblador, están diseñados para comunicarse directamente con el hardware del computador.

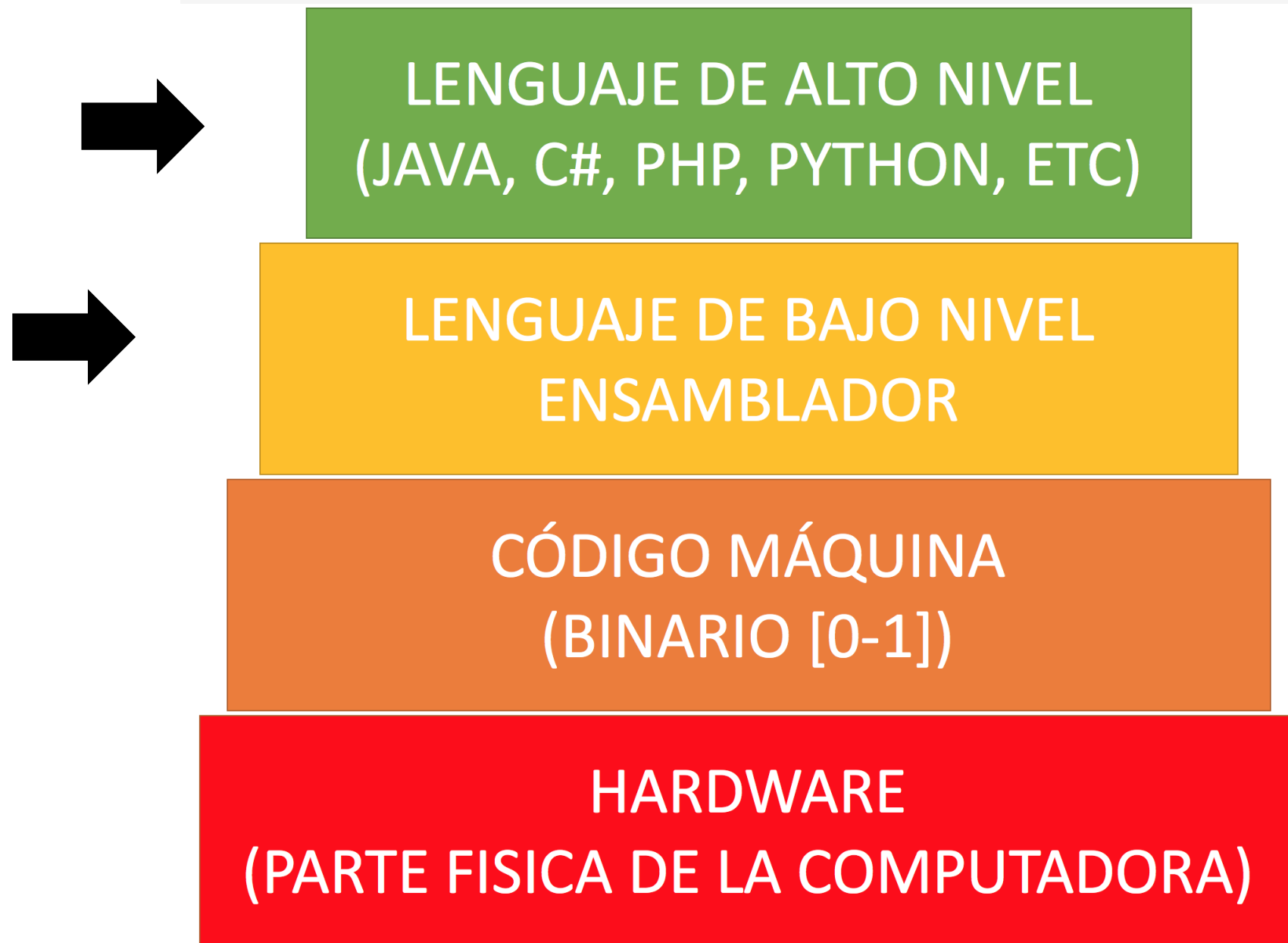
El lenguaje de máquina consiste en secuencias de unos y ceros que la CPU puede interpretar sin necesidad de traducción. Aunque permiten un control muy preciso sobre los recursos del sistema, escribir código en este tipo de lenguaje resulta extremadamente complejo, propenso a errores y difícil de mantener.

En contraste, **los lenguajes de alto nivel**, como Python, Java o JavaScript, están pensados para ser más comprensibles por los humanos. Utilizan una sintaxis más cercana al lenguaje natural y abstraen muchos de los detalles técnicos del hardware.

Esto facilita el aprendizaje, acelera el desarrollo de software y permite que más personas puedan convertirse en programadores. El código escrito en estos lenguajes es convertido en lenguaje de máquina a través de compiladores o intérpretes, que actúan como traductores entre el programador y la computadora.

resumen, el tipo de lenguaje de programación que se elige depende del tipo de proyecto, del nivel de requerimiento y del conocimiento técnico del desarrollador. Mientras que los lenguajes de bajo nivel ofrecen control total, los lenguajes de alto nivel priorizan la productividad y la facilidad de uso, siendo la base del desarrollo de la mayoría de las aplicaciones modernas.





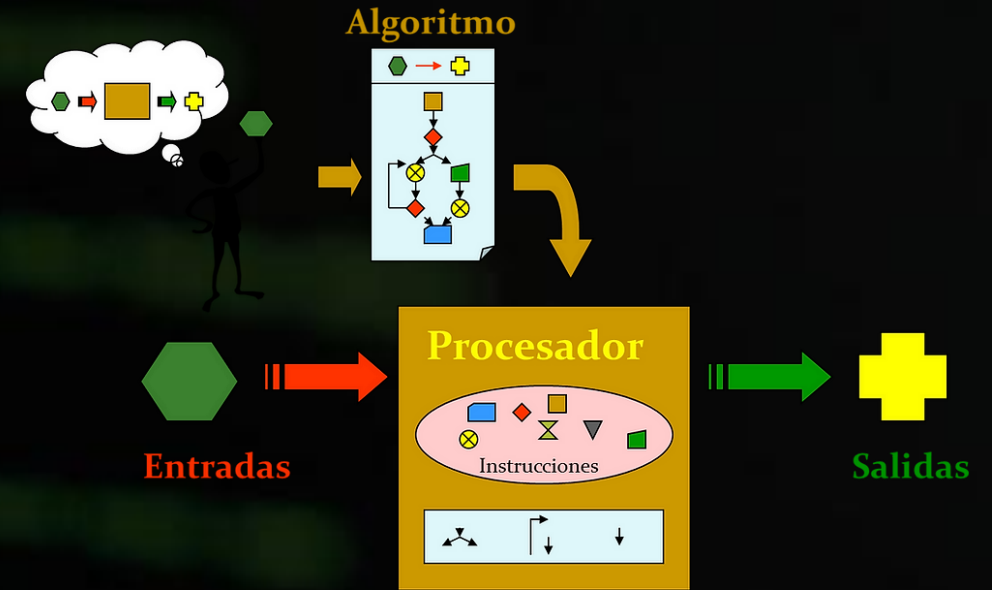
Cómo funcionan los lenguajes de programación

Los lenguajes de programación funcionan, en resumen, a través de instrucciones y algoritmos, además de utilizar compiladores e intérpretes. A continuación, explicamos cada uno de estos elementos con más detalle.

¿Qué son las instrucciones y algoritmos?

Un lenguaje de programación es una herramienta que permite diseñar programas mediante la definición de instrucciones. Estas instrucciones no se escriben de forma aleatoria, sino que se organizan siguiendo una lógica precisa: los algoritmos.

Un algoritmo es una secuencia ordenada de pasos que resuelve un problema específico o ejecuta una tarea determinada. Así, en el contexto de un lenguaje de programación, los algoritmos toman forma a través del código estructurado, que indica cómo la máquina debe procesar datos, tomar decisiones y generar resultados.



Gracias a los lenguajes de programación, es posible representar algoritmos de manera clara, legible y reutilizable, facilitando no solo el desarrollo de software eficiente, sino también el mantenimiento y la escalabilidad de las soluciones tecnológicas.

```
1 # Use the json module to parse the JSON data
2 theJSON = json.loads(json_data)
3
4 # Now we can access the data in the JSON object
5 if "title" in theJSON:
6     print(theJSON["title"])
7
8 # Output the count of the number of items
9 count = theJSON["count"]
10 print(str(count))
11
12 # For each item in the list, print the name and the count
13 for item in theJSON["items"]:
14     print(item["name"], item["count"])
15
16 # Print the total count of items
17 print("Total count: " + str(count))
```

Además, estos lenguajes proporcionan estructuras de control (como bucles, condicionales y funciones) que permiten abordar problemas complejos dividiéndolos en partes más simples y manejables.

En definitiva, el lenguaje de programación no solo sirve para comunicarse con la máquina, sino también para transformar ideas abstractas en soluciones lógicas y funcionales, mediante la correcta implementación de algoritmos.



Principales lenguajes de programación web



WEBGRAFIA

Fabíola Sebastiany

04/07/2025

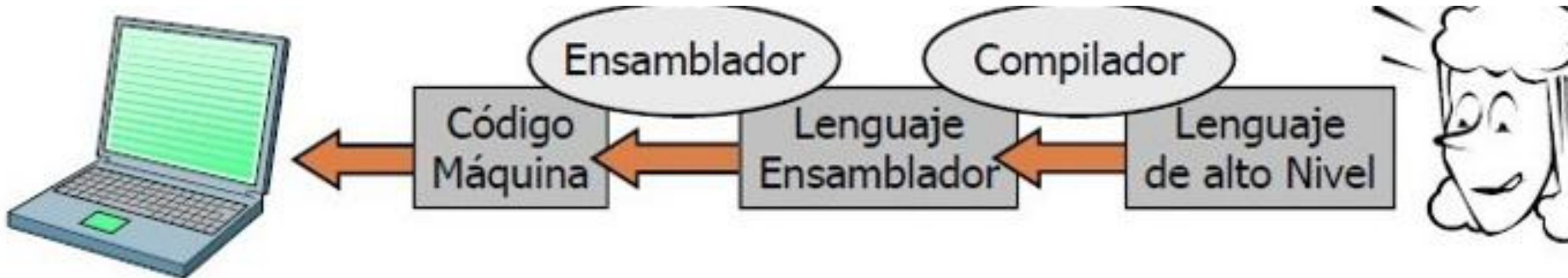
<https://www.salesforce.com/mx/blog/lenguaje-de-programacion/>

Compiladores e Interpretes

Para que las instrucciones escritas en un lenguaje de alto nivel puedan ser comprendidas por una computadora, es necesario traducirlas a lenguaje de máquina, y es ahí donde entran en juego los compiladores y intérpretes.

Los lenguajes compilados, como C o C++, utilizan un compilador que analiza todo el código fuente de una vez y lo convierte en un archivo ejecutable antes de que el programa se ejecute.

En resumen, tanto compiladores como intérpretes son componentes esenciales que permiten que un lenguaje de programación sea funcional. Cada enfoque tiene sus ventajas e desventajas, e a escolha entre um ou outro depende das necessidades do projeto, desempenho esperado e facilidade de desenvolvimento.



Utilidad de los lenguajes de programación

Los lenguajes de programación se utilizan para **transformar ideas en soluciones tecnológicas** que abarcan múltiples campos:

1.Desarrollo de software y aplicaciones: permiten crear programas de escritorio, móviles y web. Lenguajes como **C++**, **Java**, **Python** y **JavaScript** hacen posible desde sistemas operativos hasta plataformas en la nube.

2.Inteligencia artificial y ciencia de datos: lenguajes como **Python** y **R** se usan para crear algoritmos, analizar datos y construir sistemas inteligentes (redes neuronales, recomendadores, visión artificial)



R se usan para crear algoritmos, analizar datos y construir sistemas inteligentes (redes neuronales, recomendadores, visión artificial).

3. Automatización y scripting: herramientas como **Bash, Python y Perl** automatizan tareas repetitivas, mejorando la eficiencia y reduciendo errores en entornos informáticos.

4. Desarrollo móvil: **Swift** (para iOS) y **Kotlin** (para Android) son los lenguajes principales en la creación de apps móviles.

5. Sistemas embebidos: **C y C++** permiten programar dispositivos electrónicos e industriales, garantizando precisión y rendimiento en tiempo real.

En conclusión, los lenguajes de programación son la **base del desarrollo tecnológico moderno**, esenciales para la innovación en todos los sectores.

Lenguajes de programación más populares

1. Python: lenguaje versátil y fácil de aprender, usado en **IA, ciencia de datos, desarrollo web y automatización**. Destaca por su gran comunidad y bibliotecas como *Pandas* y *TensorFlow*.

2. JavaScript: base del **desarrollo web**; permite crear sitios dinámicos y también trabajar en backend (con *Node.js*), móviles e IoT.

3. Java: potente, seguro y **multiplataforma** gracias a la JVM; usado en **aplicaciones empresariales y móviles (Android)**.

4. C++: lenguaje de **alto rendimiento**, ideal para **sistemas operativos, videojuegos y software científico**.

5.C#: desarrollado por Microsoft, se usa en **aplicaciones de escritorio, sistemas empresariales y videojuegos (Unity).**

6.Ruby: conocido por su framework **Ruby on Rails**, facilita el desarrollo rápido de **aplicaciones web.**

7.Swift: lenguaje moderno y seguro para **aplicaciones iOS y macOS**, creado por Apple.

8.Kotlin: lenguaje oficial de **Android**, combina potencia y simplicidad, compatible con Java.

Cada lenguaje responde a necesidades distintas desde desarrollo web y móvil hasta inteligencia artificial y sistemas complejos, siendo herramientas clave en la innovación tecnológica actual.

Tipos de lenguajes de programación

1. Lenguajes de programación procedimental
2. Lenguajes de programación orientada a objetos
3. Lenguajes de programación funcional
4. Lenguajes de programación lógica

Programación imperativa: Se basa en dar instrucciones paso a paso al computador para modificar el estado del programa.

Ejemplo: C, Python, Java.

Programación orientada a objetos (POO): Organiza el código en objetos que combinan datos y funciones, promoviendo la reutilización y modularidad.

Ejemplo: Java, C#, Python.

Programación funcional: Se enfoca en el uso de funciones matemáticas puras, evitando modificar datos o estados.

Ejemplo: Haskell, Lisp, Scala.

Programación lógica: Se basa en reglas y hechos que el sistema usa para deducir conclusiones mediante lógica.

Ejemplo: Prolog.

