



# Los **200** términos más utilizados en programación

**mouredev**<sup>pro</sup>

[mouredev.pro/recursos](https://mouredev.pro/recursos)

# Índice

A	9
Algoritmo	9
Análisis de Complejidad	9
API (Application Programming Interface)	9
API GraphQL	9
API Key	10
API RESTful	10
Argumento	10
Array (Arreglo)	10
Atributo	10
B	11
BaaS (Backend as a Service)	11
Backend	11
Balanceador de carga (Load Balancer)	11
Balanceo de carga	11
Base de datos	11
Batch Processing (Procesamiento por lotes)	12
Biblioteca o Librería (Library)	12
Big Data	12
Big O	12
Binario	12
Bit	12
Booleano	13
Bucles (Loops)	13
Buffer	13
Bug	13
Build (Compilación)	13
C	14
Cache	14
Callback	14
Carga (Load)	14
Clase (Class)	14
CLI (Command-line interface)	14

Closure (Clausura)	15
Código fuente (Source code)	15
Código limpio (Clean Code)	15
Cola (Queue)	15
Commit	15
Compilación (Build)	15
Compilador	16
Compilador Just-in-Time (JIT)	16
Compresión de datos	16
Computación en la nube (Cloud Computing)	16
Concurrency (Concurrencia)	16
Condicional	16
Constructor	17
Contenedor (Container)	17
Contexto	17
Control de versiones (Version Control)	17
CORS (Cross-Origin Resource Sharing)	17
CRUD (Create, Read, Update, Delete)	17
CSS (Cascading Style Sheets)	18
D	19
Data Mining (Minería de datos)	19
Data Science (Ciencia de datos)	19
Dependencia (Dependency)	19
Depuración (Debugging)	19
Desarrollo ágil (Agile)	19
Desbordamiento de pila (Stack Overflow)	20
Deserialización	20
DevOps	20
Diagramas de flujo	20
DOM (Document Object Model)	20
Driver	20
E	21
Eficiencia	21
Elemento	21
Encapsulación	21

Encriptación (Cifrado)	21
Endpoint	21
Ensamblador (Assembler)	22
Entrada (Input)	22
Escalabilidad	22
Esquema (Bases de datos)	22
Estilo de código	22
Evento	22
Excepción (Exception)	23
Expresión regular (Regex)	23
F	24
FIFO (First In, First Out)	24
Firewall	24
Framework	24
Frontend	24
Fullstack	24
Función (Function)	25
G	26
Git	26
GitHub	26
H	27
Hashing (Hash)	27
Heap (Montículo)	27
Herencia (Inheritance)	27
HTML (HyperText Markup Language)	27
HTTP (Hypertext Transfer Protocol)	27
HTTPS (Hypertext Transfer Protocol Secure)	28
Hypervisor	28
I	29
IDE (Entorno de Desarrollo Integrado)	29
Índice (Index)	29
Immutable	29
Instancia (Instance)	29
Instanciación	29
Inyección de dependencias (Dependency Injection)	30

IoT (Internet of Things)	30
Iteración	30
J	31
JavaScript	31
JSON (JavaScript Object Notation)	31
JVM (Java Virtual Machine)	31
JWT (JSON Web Token)	31
K	32
Kernel	32
Key (Llave)	32
Keyword (Palabra clave)	32
L	33
Lambda	33
Latencia	33
Lazy Loading	33
Lenguaje de Alto nivel	33
Lenguaje de Bajo Nivel	33
LIFO (Last In, First Out)	34
Linter	34
LLM	34
Local	34
M	35
Machine Learning (Aprendizaje Automático)	35
Método (Method)	35
Microservicios	35
Middleware	35
Mock (Objeto simulado)	35
Módulo (Module)	36
Monolito	36
N	37
Namespace (Espacio de nombres)	37
Nested (Anidado)	37
Normalización (Normalization)	37
NoSQL	37
Null	37

O	38
OAuth	38
Objeto (Object)	38
Open Source (Código abierto)	38
Operador aritmético	38
Operador de asignación	38
ORM (Mapeo Objeto-Relacional)	39
Overloading (Sobrecarga)	39
Override (Sobrescribir)	39
P	40
PaaS (Platform as a Service)	40
Paralelismo	40
Parámetro (Parameter)	40
Parsing (Análisis sintáctico)	40
Pila (Stack)	40
Pipeline	41
Polimorfismo (Polymorphism)	41
POO (Programación Orientada a Objetos)	41
Programación funcional	41
Programación imperativa	41
Promesa (Promise)	42
Pseudocódigo	42
Pull Request	42
Python	42
Q	43
QA (Quality Assurance)	43
Query (Consulta)	43
Query string	43
R	44
Rama (Branch)	44
Recursión	44
Refactorización	44
Remoto	44
Renderizado	44
Repositorio (Repository)	45

	REST (Representational State Transfer)	45
	Retorno (Return)	45
	Revisión de código	45
	Runtime (Tiempo de Ejecución)	45
S		46
	SaaS (Software as a Service)	46
	Script	46
	Scrum	46
	SDK (Software Development Kit)	46
	Serialización	46
	Servidor Web (Web server)	46
	Shell	47
	Síncrono	47
	Singleton	47
	SQL (Structured Query Language)	47
	SSH (Secure Shell)	47
	Stack Trace	47
	Switch	48
T		49
	Test Unitario (Unit Test)	49
	Testing (Pruebas)	49
	Thread (Hilo)	49
	Timeout	49
	Tipado dinámico (Dynamic Typing)	49
	Tipado estático (Static Typing)	50
	Token	50
	Tupla (Tuple)	50
U		51
	UI (Interfaz de Usuario)	51
	URL (Uniform Resource Locator)	51
	UTF-8	51
	UX (Experiencia de Usuario)	51
V		52
	Variable	52
	Variable Global	52

	Virtualización	52
W		53
	Web scraping	53
	WebSocket	53
	Wireframe	53
	Workflow (Flujo de trabajo)	53
	Wrapper	53
X		54
	XML (Extensible Markup Language)	54
Y		55
	YAML (YAML Ain't Markup Language)	55
Z		56
	Z-index	56
	Zero-Day Vulnerability	56
	Zip	56



# Glosario

## A

### **Algoritmo**

Conjunto de pasos bien definidos que permiten solucionar un problema o realizar una tarea específica. Los algoritmos pueden implementarse en cualquier lenguaje de programación.

### **Análisis de Complejidad**

Evaluación de la eficiencia de un algoritmo en términos de tiempo de ejecución y espacio en memoria en función del tamaño de la entrada. Se suele expresar en notación Big O.

### **API (Application Programming Interface)**

Conjunto de definiciones y protocolos que permiten la interacción entre diferentes sistemas de software. Una API permite que una aplicación acceda a las funcionalidades o datos de otra.

### **API GraphQL**

Un lenguaje de consulta para APIs que permite al cliente solicitar exactamente los datos que necesita. A diferencia de REST, GraphQL puede evitar el envío de datos innecesarios.

## **API Key**

Clave de autenticación utilizada para acceder a una API. Es una forma segura de controlar el acceso a los recursos y gestionar las credenciales.

## **API RESTful**

Estilo de arquitectura que se utiliza para construir APIs, basado en las operaciones HTTP (GET, POST, PUT, DELETE, entre otras). Se usa para interactuar con recursos representados en un servidor.

## **Argumento**

Valor que se pasa a una función cuando es llamada. Este valor es procesado por la función para realizar una tarea específica.

## **Array (Arreglo)**

Estructura de datos que almacena múltiples valores en una secuencia indexada. Los arrays permiten el acceso rápido a cualquier elemento usando su índice (que habitualmente se corresponde con el 0).

## **Atributo**

Propiedad o característica de un objeto en programación orientada a objetos. Un atributo representa el estado de un objeto.

## B

### **BaaS (Backend as a Service)**

Plataforma que proporciona a los desarrolladores todas las herramientas y servicios necesarios para conectar aplicaciones al backend sin necesidad de gestionar la infraestructura.

### **Backend**

Parte del desarrollo de software que se enfoca en la lógica del servidor, la gestión de bases de datos, y la manipulación de datos detrás de una aplicación. No es accesible directamente por el usuario final.

### **Balanceador de carga (Load Balancer)**

Dispositivo o software que distribuye el tráfico de red entre varios servidores para mejorar el rendimiento y evitar sobrecargas en un solo servidor.

### **Balanceo de carga**

Técnica que distribuye el tráfico de red entre múltiples servidores para mejorar el rendimiento, evitar sobrecargas y garantizar la alta disponibilidad.

### **Base de datos**

Sistema estructurado para almacenar, organizar y gestionar grandes cantidades de datos. Pueden ser relacionales (SQL) o no relacionales (NoSQL).

## **Batch Processing (Procesamiento por lotes)**

Método de ejecución en el que las tareas se agrupan y ejecutan en conjunto en lugar de ejecutarse de manera interactiva o en tiempo real.

## **Biblioteca o Librería (Library)**

Colección de funciones, métodos o clases predefinidas que pueden ser reutilizadas en diferentes proyectos. Las bibliotecas permiten evitar la duplicación de código y mejorar la productividad.

## **Big Data**

Conjunto de datos extremadamente grandes y complejos que requieren herramientas y tecnologías especializadas para su análisis, almacenamiento y procesamiento.

## **Big O**

Notación que describe la complejidad temporal o espacial de un algoritmo en el peor de los casos. Mide cómo el rendimiento de un algoritmo escala con el tamaño de la entrada.

## **Binario**

Sistema numérico en base 2, que solo utiliza los dígitos 0 y 1. Es el lenguaje fundamental de los equipos informáticos.

## **Bit**

Unidad mínima de información en computación que puede tener solo dos valores: 0 o 1.

## Booleano

Tipo de dato que solo puede tener dos valores posibles: verdadero (True) o falso (False). Utilizado comúnmente en expresiones condicionales.

## Bucles (Loops)

Estructura de control que permite repetir una secuencia de instrucciones hasta que se cumpla una condición específica. Los tipos más comunes son for, while y do while.

## Buffer

Espacio temporal en memoria que almacena datos mientras son transferidos entre diferentes partes de un programa o hardware. Los buffers se usan comúnmente para mejorar la eficiencia de entrada/salida.

## Bug

Error o fallo en el código que provoca que un programa no funcione correctamente. La depuración (debugging) es el proceso de encontrar y corregir estos errores.

## Build (Compilación)

Proceso de convertir el código fuente en un formato ejecutable por un ordenador. En los lenguajes compilados, como C++ o Java, el código pasa por un proceso de construcción antes de ejecutarse.



## Cache

Almacenamiento temporal de datos que son accedidos con frecuencia para mejorar el rendimiento y reducir la latencia en las solicitudes repetidas.

## Callback

Función que se pasa como argumento a otra función y que es ejecutada después de que la función principal termina su tarea. Común en programación asíncrona.

## Carga (Load)

Se refiere a la cantidad de trabajo que un sistema, servidor o programa está procesando en un momento dado. El manejo de carga es crucial para evitar la saturación de recursos.

## Clase (Class)

Plantilla o definición para crear objetos en programación orientada a objetos. Una clase define los atributos y comportamientos que los objetos creados a partir de ella tendrán.

## CLI (Command-line interface)

Interfaz basada en texto que permite a los usuarios interactuar con el sistema operativo o el software mediante la introducción de comandos.

## Closure (Clausura)

Función que "recuerda" el ámbito en el que fue creada, incluso si se ejecuta fuera de ese ámbito. Común en lenguajes como JavaScript.

## Código fuente (Source code)

Conjunto de instrucciones y declaraciones escritas en un lenguaje de programación, que luego son compiladas o interpretadas para ser ejecutadas por una máquina.

## Código limpio (Clean Code)

Código que es fácil de leer, comprender, y mantener. Se caracteriza por una estructura clara, nombres descriptivos, y la ausencia de redundancias o complejidad innecesaria.

## Cola (Queue)

Estructura de datos que opera según el principio FIFO (First In, First Out), donde el primer elemento añadido es el primero en ser procesado.

## Commit

Acción en un sistema de control de versiones como Git que guarda un estado específico del código en el historial del proyecto.

## Compilación (Build)

Proceso de transformar código fuente en un archivo ejecutable por la máquina. Los compiladores se encargan de esta tarea, generando el código binario necesario (en la mayoría de los casos) para que el sistema operativo ejecute el programa.

## **Compilador**

Programa que convierte el código fuente escrito en un lenguaje de alto nivel a un lenguaje de máquina ejecutable.

## **Compilador Just-in-Time (JIT)**

Compilador que traduce el código en tiempo de ejecución en lugar de hacerlo antes de que comience la ejecución del programa, optimizando su rendimiento.

## **Compresión de datos**

Proceso que consiste en reducir el tamaño de un archivo o conjunto de datos. Puede ser con pérdida (se elimina parte de la información) o sin pérdida (se mantiene toda la información original).

## **Computación en la nube (Cloud Computing)**

Modelo de computación que permite acceder a recursos informáticos (servidores, almacenamiento, bases de datos, etc.) a través de internet, en lugar de gestionarlos localmente.

## **Concurrency (Concurrencia)**

Capacidad de un sistema o programa para ejecutar múltiples tareas de forma simultánea, a menudo mejorando el rendimiento en sistemas multi-núcleo.

## **Condicional**

Estructura de control que permite tomar decisiones en el flujo de un programa según una condición booleana (True o False). Los condicionales comunes incluyen if, else if, else, y switch case.



## **Constructor**

Método especial que se invoca automáticamente cuando se crea una instancia de una clase. En la mayoría de los lenguajes orientados a objetos, se usa para inicializar objetos.

## **Contenedor (Container)**

Unidad de software que empaqueta el código y todas sus dependencias para que una aplicación pueda ejecutarse de manera confiable en cualquier entorno, ya sea en local o en producción.

## **Contexto**

Entorno en el que se ejecuta un bloque de código, que define qué variables y funciones están disponibles en ese momento.

## **Control de versiones (Version Control)**

Sistema que permite gestionar los cambios en el código fuente a lo largo del tiempo, facilitando la colaboración entre desarrolladores y el mantenimiento de un historial de versiones, como Git o Subversion.

## **CORS (Cross-Origin Resource Sharing)**

Mecanismo de seguridad que permite a un servidor especificar quién puede acceder a sus recursos desde un dominio diferente, utilizado para prevenir problemas de seguridad en aplicaciones web.

## **CRUD (Create, Read, Update, Delete)**

Acrónimo que describe las operaciones básicas que se pueden realizar sobre los datos en una base de datos o sistema de almacenamiento. Crear, leer, actualizar y eliminar son las acciones fundamentales de manipulación de datos.

## **CSS (Cascading Style Sheets)**

Lenguaje utilizado para definir la apariencia y el diseño de las páginas web.

Permite separar el contenido HTML de los estilos visuales, como colores, fuentes y diseño de las páginas.

## D

### **Data Mining (Minería de datos)**

Proceso que analiza grandes volúmenes de datos para descubrir patrones ocultos y relaciones útiles para la toma de decisiones.

### **Data Science (Ciencia de datos)**

Campo que utiliza técnicas de estadística, matemáticas y programación para extraer información valiosa a partir de datos.

### **Dependencia (Dependency)**

Componente o módulo externo que una aplicación necesita para funcionar. Por ejemplo, bibliotecas o paquetes que agregan funcionalidades específicas a un proyecto.

### **Depuración (Debugging)**

Proceso de identificar, rastrear y corregir errores o fallos en un programa para asegurarse de que funcione correctamente.

### **Desarrollo ágil (Agile)**

Metodología de desarrollo de software que se enfoca en la entrega rápida y continua de pequeñas mejoras a lo largo del tiempo. Se basa en la colaboración y la adaptación rápida a los cambios.

## **Desbordamiento de pila (Stack Overflow)**

Error que se produce cuando un programa agota la memoria disponible para la pila de ejecución, generalmente debido a una recursión infinita, bucles mal gestionados, objetos de gran tamaño o ausencia de eliminación de recursos.

## **Deserialización**

Proceso de convertir un formato de datos serializado (como JSON o XML) en un objeto nativo del lenguaje de programación que lo pueda manipular.

## **DevOps**

Práctica que combina el desarrollo de software (Dev) y las operaciones de TI (Ops) para mejorar la colaboración entre equipos y automatizar el ciclo de vida de la entrega de software.

## **Diagramas de flujo**

Representación gráfica de un proceso o algoritmo que muestra las decisiones, entradas, salidas y flujo de control mediante símbolos y flechas.

## **DOM (Document Object Model)**

Modelo de programación que representa la estructura de una página web como un árbol de nodos, lo que permite a los lenguajes de programación manipular dinámicamente el contenido y estilo de una página web.

## **Driver**

Programa que permite que el sistema operativo o una aplicación se comuniquen con un hardware específico, como una impresora, tarjeta gráfica o dispositivo USB.

# E

## Eficiencia

Medida que representa cómo un algoritmo o programa utiliza los recursos disponibles, como su tiempo de ejecución y espacio en memoria, para resolver un problema o ejecutar una tarea.

## Elemento

Unidad individual dentro de una estructura de datos como un array o una lista. Por ejemplo, en el array `[1, 2, 3]`, los elementos son 1, 2 y 3.

## Encapsulación

Principio de la programación orientada a objetos que oculta los detalles internos de una clase o módulo y expone solo una interfaz pública, protegiendo el estado interno de cambios accidentales o no deseados.

## Encriptación (Cifrado)

Técnica que transforma los datos en un formato ilegible para garantizar su confidencialidad. Solo quienes tienen la clave adecuada pueden descifrar y leer los datos.

## Endpoint

Punto final en una conexión de red o API. Representa un recurso específico en un servidor al que se accede para interactuar con la aplicación.

## **Ensamblador (Assembler)**

Un programa que traduce código en lenguaje ensamblador (un lenguaje de bajo nivel) a código máquina, que es directamente entendible por el procesador de un ordenador.

## **Entrada (Input)**

Datos o información proporcionados a un programa para ser procesados. La entrada puede venir del usuario, de un archivo, de otro programa o de una base de datos.

## **Escalabilidad**

Capacidad de un sistema, red o proceso para manejar un incremento en la carga de trabajo añadiendo recursos, como aumentar la cantidad de servidores (escalabilidad horizontal) o mejorar los recursos de hardware (escalabilidad vertical).

## **Esquema (Bases de datos)**

Estructura que define cómo se organizan y relacionan los datos dentro de una base de datos. Describe las tablas, columnas, relaciones y restricciones de la base de datos.

## **Estilo de código**

Conjunto de convenciones y reglas sobre cómo debe estructurarse el código en un lenguaje de programación. Facilita la lectura y mantenimiento del código.

## **Evento**

Acción o suceso que ocurre en un programa, como un clic del ratón, el envío de un formulario o la llegada de datos desde un servidor. Los eventos desencadenan respuestas predefinidas.

## **Excepción (Exception)**

Evento anómalo o error que ocurre durante la ejecución de un programa y que interrumpe su flujo normal. Las excepciones pueden ser gestionadas con estructuras como try/catch.

## **Expresión regular (Regex)**

Secuencia de caracteres que define un patrón de búsqueda. Se utiliza para la coincidencia, validación y manipulación de texto en la gran mayoría de lenguajes de alto nivel.

## F

### **FIFO (First In, First Out)**

Estructura de datos en la que el primer elemento añadido es el primero en ser retirado, como en una cola.

### **Firewall**

Sistema de seguridad que controla y filtra el tráfico de red entrante y saliente, basado en reglas de seguridad predeterminadas. Ayuda a proteger las redes de accesos no autorizados.

### **Framework**

Conjunto de herramientas, bibliotecas y convenciones que ofrecen una estructura estándar para desarrollar aplicaciones de manera eficiente.

### **Frontend**

Parte de una aplicación o sitio web que interactúa directamente con el usuario, abarcando la interfaz gráfica y la experiencia de usuario.

### **Fullstack**

Desarrollador que trabaja tanto en el frontend como en el backend de una aplicación, manejando tanto la interfaz de usuario como la lógica del servidor y las bases de datos.



## Función (Function)

Bloque de código que realiza una tarea específica y que puede ser invocado múltiples veces dentro de un programa. Las funciones permiten modularizar y reutilizar el código.

## G

### Garbage Collection (Recolección de basura)

Mecanismo automático que gestiona la memoria en lenguajes de programación de alto nivel como Java o Python, liberando la memoria que ya no está en uso.

## Git

Sistema de control de versiones distribuido que permite a los desarrolladores llevar un historial de cambios en el código y trabajar en colaboración en proyectos de manera eficiente.

## GitHub

Plataforma en línea que permite a los desarrolladores alojar sus repositorios de Git, colaborar con otros programadores y compartir proyectos de código abierto. Entre otras funcionalidades extra.

### GUI (Graphical User Interface)

Interfaz gráfica que permite a los usuarios interactuar con un sistema mediante elementos visuales como ventanas, botones y menús, en lugar de usar solo comandos de texto.

# H

## Hashing (Hash)

Proceso de convertir datos de longitud variable en una representación de longitud fija mediante una función hash. Utilizado para búsquedas rápidas, encriptación y generación de claves únicas.

## Heap (Montículo)

Estructura de datos en la que los elementos se organizan en un árbol binario, útil para implementar colas de prioridad. El elemento con mayor prioridad siempre se encuentra en la raíz.

## Herencia (Inheritance)

Propiedad de la programación orientada a objetos que permite que una clase derive de otra, heredando sus atributos y métodos, y permitiendo la reutilización del código.

## HTML (HyperText Markup Language)

Lenguaje estándar para estructurar contenido en la web. Define elementos como encabezados, párrafos, enlaces, imágenes y otros componentes de una página web.

## HTTP (Hypertext Transfer Protocol)

Protocolo de comunicación utilizado para transferir datos a través de la web. Es el protocolo fundamental para la navegación en sitios web y la interacción con APIs.

## **HTTPS (Hypertext Transfer Protocol Secure)**

Versión segura de HTTP que utiliza encriptación para proteger la información transmitida entre el cliente y el servidor. Asegura la privacidad e integridad de los datos.

## **Hypervisor**

Software que permite ejecutar y gestionar múltiples máquinas virtuales en un único hardware físico, maximizando el uso de recursos.

## **IDE (Entorno de Desarrollo Integrado)**

Software que proporciona un conjunto de herramientas para facilitar el desarrollo de software, como un editor de código, un depurador y un compilador, todo en un solo entorno.

## **Índice (Index)**

Posición numérica asignada a un elemento dentro de una estructura de datos como un array o lista, que permite acceder a él de manera rápida y directa.

## **Inmutable**

Propiedad de un objeto o dato que no puede cambiar después de su creación. Las estructuras inmutables son comunes en programación funcional y ayudan a prevenir errores relacionados con el cambio de estado.

## **Instancia (Instance)**

Objeto creado a partir de una clase. Una instancia representa un ejemplo específico de la clase, con sus propios valores y comportamientos.

## **Instanciación**

Proceso utilizado para crear un objeto a partir de una clase. Cuando se instancia una clase, se crea un nuevo objeto con los atributos y métodos definidos en la clase.

## **Inyección de dependencias (Dependency Injection)**

Patrón de diseño que consiste en proporcionar a un objeto sus dependencias desde fuera, en lugar de crearlas internamente. Facilita el testeo y la reutilización de código.

## **IoT (Internet of Things)**

Red de dispositivos conectados a internet que recopilan y comparten datos, como sensores, electrodomésticos inteligentes, coches autónomos, etc.

## **Iteración**

Proceso de repetir una secuencia de instrucciones o pasos en un bucle, hasta que se cumple una condición de finalización. Los bucles for y while son comunes para realizar iteraciones.

# J

## JavaScript

Lenguaje de programación interpretado que se ejecuta en el navegador para crear experiencias interactivas en la web. También se utiliza en servidores con tecnologías como Node.js.

## JSON (JavaScript Object Notation)

Formato de intercambio de datos ligero y fácil de leer tanto para humanos como para máquinas. Es ampliamente utilizado en APIs y en la transmisión de datos entre cliente y servidor a través de la red.

## JVM (Java Virtual Machine)

Máquina virtual que ejecuta aplicaciones Java (o Kotlin) al interpretar el bytecode generado por el compilador. Ofrece portabilidad y seguridad independiente del hardware.

## JWT (JSON Web Token)

Estándar para crear tokens de acceso seguros que permiten la autenticación entre cliente y servidor. Los tokens JWT se utilizan para validar sesiones en aplicaciones web.

# K

## **Kernel**

Parte central del sistema operativo que gestiona el hardware del equipo informático y proporciona servicios básicos a las aplicaciones.

## **Key (Llave)**

En bases de datos, una columna o combinación de columnas que identifica de manera única un registro en una tabla. Las claves primarias y foráneas son tipos comunes. En estructuras, el elemento que permite acceder a su valor asociado.

## **Keyword (Palabra clave)**

Término reservado en un lenguaje de programación que tiene un significado especial y reservado por dicho lenguaje, como if, else, while o return.



# L

## **Lambda**

Función anónima, generalmente escrita en una sola línea, que se utiliza para realizar tareas simples y rápidas.

## **Latencia**

Tiempo que transcurre entre una solicitud y la respuesta correspondiente. En programación y redes, se refiere al retraso en la transmisión de datos.

## **Lazy Loading**

Técnica de optimización que retrasa la carga de imágenes, archivos u otros recursos hasta que realmente son necesarios, mejorando la velocidad de carga inicial de las aplicaciones.

## **Lenguaje de Alto nivel**

Lenguajes de programación que están más cerca del lenguaje humano y se abstraen de los detalles del hardware, como Python, Java, o JavaScript. Son más fáciles de entender y usar, aunque a veces menos eficientes que los lenguajes de bajo nivel.

## **Lenguaje de Bajo Nivel**

Lenguaje de programación que está más cerca del código máquina y ofrece menos abstracción que los lenguajes de alto nivel, como el lenguaje Ensamblador, o en mejor medida, C.

## **LIFO (Last In, First Out)**

Estructura de datos en la que el último elemento añadido es el primero en salir, como en una pila.

## **Linter**

Herramienta de análisis de código que detecta errores y malas prácticas en el código fuente, mejorando la calidad y consistencia del código.

## **LLM**

Modelo de lenguaje a gran escala entrenado con grandes cantidades de datos para comprender, generar y predecir texto de manera contextual.

## **Local**

Hace referencia a un entorno o recurso que se encuentra en la máquina del usuario (como un servidor, base de datos o archivo) y no en una ubicación remota.

# M

## **Machine Learning (Aprendizaje Automático)**

Rama de la inteligencia artificial que se centra en desarrollar algoritmos que permiten que las máquinas aprendan automáticamente a partir de datos, sin ser explícitamente programadas.

## **Método (Method)**

Función asociada a una clase o un objeto en programación orientada a objetos, que define un comportamiento específico.

## **Microservicios**

Arquitectura en la que una aplicación está dividida en pequeños servicios independientes que pueden desarrollarse, desplegarse y escalarse de manera autónoma.

## **Middleware**

Software que actúa como intermediario entre diferentes sistemas o aplicaciones. Permite que varios componentes se comuniquen de manera más eficiente y segura.

## **Mock (Objeto simulado)**

Objeto utilizado en pruebas de software para simular el comportamiento de componentes reales del sistema. Permite probar una parte del código sin tener que interactuar con el resto del sistema.

## **Módulo (Module)**

Componente reutilizable de software que encapsula un conjunto de funcionalidades relacionadas, como un archivo de Python que contiene funciones y clases.

## **Monolito**

Arquitectura en la que todos los componentes de una aplicación están estrechamente interconectados y se despliegan como una sola unidad. A menudo se contrasta con la arquitectura de microservicios.

# N

## **Namespace (Espacio de nombres)**

Contexto que permite diferenciar múltiples identificadores (nombres de variables, funciones, clases) sin causar conflictos de nombres, al agruparlos bajo un mismo nombre.

## **Nested (Anidado)**

Situación en la que un bloque de código, como un bucle o condicional, está contenido dentro de otro bloque. El código anidado puede aumentar la complejidad de un programa.

## **Normalización (Normalization)**

Proceso que consiste en organizar los datos en una base de datos para reducir la redundancia y mejorar la integridad de los datos, dividiéndolos en múltiples tablas relacionadas.

## **NoSQL**

Tipo de base de datos que no sigue el modelo relacional tradicional. Es útil para manejar grandes volúmenes de datos no estructurados o semi-estructurados.

## **Null**

Valor especial que indica la ausencia de un valor o la inexistencia de un objeto. Es común en muchos lenguajes de programación para representar "sin datos".



## **OAuth**

Protocolo estándar de autorización que permite a los usuarios dar acceso limitado a sus recursos sin compartir sus credenciales. Común en sistemas de autenticación y APIs.

## **Objeto (Object)**

Instancia de una clase que contiene datos (atributos) y comportamientos (métodos) definidos en la clase. Los objetos son fundamentales en la programación orientada a objetos.

## **Open Source (Código abierto)**

Software cuyo código fuente está disponible públicamente para que cualquiera lo use, modifique y distribuya (aunque pueden existir restricciones). Por ejemplo: Linux, Python y Git.

## **Operador aritmético**

Símbolos utilizados en la programación para realizar operaciones matemáticas básicas como suma (+), resta (-), multiplicación (\*), división (/) y módulo (%).

## **Operador de asignación**

Símbolos usados para asignar valores a variables, siendo el más común el operador igual (=), que asigna el valor de la derecha a la variable de la izquierda.

## **ORM (Mapeo Objeto-Relacional)**

Herramienta que permite interactuar con bases de datos relacionales usando objetos. Simplifica la manipulación de bases de datos al eliminar la necesidad de escribir consultas SQL manualmente.

## **Overloading (Sobrecarga)**

Capacidad de definir múltiples versiones de una función o un operador que realizan diferentes tareas en función de sus parámetros.

## **Override (Sobrescribir)**

Proceso de redefinir un método heredado de una clase base en una subclase para cambiar su comportamiento.

# P

## **PaaS (Platform as a Service)**

Modelo de computación en la nube que ofrece una plataforma completa para desarrollar, ejecutar y gestionar aplicaciones sin preocuparse de la infraestructura subyacente.

## **Paralelismo**

Ejecución simultánea de múltiples tareas o procesos en diferentes hilos o procesadores, mejorando el rendimiento general del sistema.

## **Parámetro (Parameter)**

Variable que se pasa a una función o método para que este pueda realizar operaciones basadas en los valores proporcionados.

## **Parsing (Análisis sintáctico)**

Proceso de descomponer una secuencia de texto o código en sus componentes estructurales, como palabras clave y operadores, para analizar su significado o ejecutarlo.

## **Pila (Stack)**

Estructura de datos que sigue el principio LIFO (Last In, First Out), donde el último elemento en entrar es el primero en salir. Se utiliza comúnmente en algoritmos y para la gestión de la memoria.



## **Pipeline**

Conjunto de pasos automatizados que se ejecutan en secuencia para compilar, probar y desplegar una aplicación de software. Común en sistemas de CI/CD (Integración continua y Despliegue continuo).

## **Polimorfismo (Polymorphism)**

Principio de la programación orientada a objetos que permite que un solo método o función se comporte de manera diferente según el objeto que lo invoque.

## **POO (Programación Orientada a Objetos)**

Paradigma de programación que organiza el software en clases y objetos que representan conceptos o entidades del mundo real. Promueve la reutilización del código y la modularización.

## **Programación funcional**

Paradigma de programación que se basa en funciones matemáticas puras, evitando cambios de estado y efectos secundarios, y promoviendo un enfoque declarativo.

## **Programación imperativa**

Paradigma de programación que se centra en describir los pasos necesarios para lograr un resultado, utilizando instrucciones secuenciales, bucles y cambios en el estado del programa.

## **Promesa (Promise)**

En JavaScript, una promesa es un objeto que representa la eventual resolución o rechazo de una operación asíncrona. Permite manejar resultados futuros de manera más ordenada que los callbacks.

## **Pseudocódigo**

Descripción informal de un algoritmo que utiliza una mezcla de lenguaje natural y sintaxis de programación. Se utiliza para planificar el código sin preocuparse por los detalles de la sintaxis.

## **Pull Request**

Solicitud en un sistema de control de versiones, como GitHub, para fusionar los cambios realizados en una rama del proyecto con la rama principal. Es una práctica común para la revisión de código colaborativo.

## **Python**

Lenguaje de programación de alto nivel conocido por su simplicidad, legibilidad y amplia comunidad. Se utiliza en desarrollo web, análisis de datos, inteligencia artificial y más.



## **QA (Quality Assurance)**

Proceso de asegurar que el software cumple con los estándares de calidad establecidos, mediante la planificación, pruebas, revisión y mejora continua a lo largo del ciclo de desarrollo.

## **Query (Consulta)**

Solicitud de información a una base de datos. Por ejemplo, en SQL se usa SELECT para recuperar datos específicos.

## **Query string**

Parte de una URL que contiene parámetros clave-valor para enviar datos a un servidor, como ?id=1&name=mouredev.

# R

## Rama (Branch)

En sistemas de control de versiones, una branch (rama) es una línea paralela de desarrollo que permite trabajar en nuevas características o corregir errores sin afectar la línea principal (llamada habitualmente main).

## Recursión

Técnica de programación en la que una función se llama a sí misma, directa o indirectamente, para resolver problemas que pueden descomponerse en subproblemas más pequeños.

## Refactorización

Proceso de reescribir y mejorar la estructura interna del código sin cambiar su comportamiento externo. Se realiza para mejorar la legibilidad, la eficiencia o la mantenibilidad.

## Remoto

Hace referencia a un recurso, servidor o servicio que se encuentra en una ubicación distinta a la máquina del usuario, generalmente accesible a través de una red como internet.

## Renderizado

Proceso para generar la vista final de una página web o una aplicación. En el contexto web, puede hacerse del lado del cliente (frontend) o del lado del servidor (SSR, Server-side Rendering).

## **Repositorio (Repository)**

Almacén donde se guardan y gestionan los archivos y el código de un proyecto de software, generalmente utilizando un sistema de control de versiones como Git.

## **REST (Representational State Transfer)**

Estilo de arquitectura para diseñar APIs que utilizan los métodos HTTP (GET, POST, PUT, DELETE) para interactuar con recursos. REST es ampliamente utilizado en la comunicación entre aplicaciones web.

## **Retorno (Return)**

Declaración en una función que termina su ejecución y devuelve un valor al código que llamó a esa función.

## **Revisión de código**

Práctica en la que otros desarrolladores revisan el código escrito por un compañero para identificar errores, mejorar la calidad del código y asegurar el cumplimiento de las buenas prácticas.

## **Runtime (Tiempo de Ejecución)**

Fase durante la cual un programa ejecuta sus instrucciones. También puede referirse al entorno que soporta dicha ejecución.

# S

## **SaaS (Software as a Service)**

Modelo de distribución de software en el que una aplicación se ofrece como un servicio a través de internet, sin necesidad de instalarla localmente.

## **Script**

Programa corto o secuencia de comandos que se ejecuta para automatizar tareas específicas. Comúnmente usado en lenguajes como Python o Bash.

## **Scrum**

Metodología ágil que divide el desarrollo de software en ciclos cortos (sprints) para entregar incrementos funcionales del producto.

## **SDK (Software Development Kit)**

Conjunto de herramientas y bibliotecas que facilita el desarrollo de aplicaciones para una plataforma específica.

## **Serialización**

Proceso de convertir un objeto o estructura de datos en un formato que pueda ser fácilmente almacenado o transmitido, como JSON o XML.

## **Servidor Web (Web server)**

Software que almacena, procesa y entrega contenido web a los navegadores mediante el protocolo HTTP.

## Shell

Interfaz que permite a los usuarios interactuar con el sistema operativo mediante comandos escritos en la línea de comandos.

## Síncrono

Modelo de ejecución en el que las tareas se ejecutan de manera secuencial, esperando a que una tarea termine antes de comenzar la siguiente. Común en operaciones de entrada/salida.

## Singleton

Patrón de diseño que asegura que una clase tenga solo una instancia controlada en todo el programa. Es útil para administrar recursos compartidos como conexiones de base de datos o configuraciones globales.

## SQL (Structured Query Language)

Lenguaje de consulta utilizado para gestionar y manipular bases de datos relacionales. Permite realizar operaciones como insertar, actualizar, eliminar y consultar datos.

## SSH (Secure Shell)

Protocolo de red que permite la administración remota segura de servidores y otros dispositivos mediante una conexión encriptada. Se utiliza para acceder de manera segura a sistemas remotos.

## Stack Trace

Lista de las funciones o métodos que están siendo ejecutados cuando ocurre un error en un programa. Los stack traces son útiles para depurar y encontrar la causa de un error.

## Switch

Estructura de control de flujo que permite ejecutar diferentes bloques de código en función del valor de una expresión. Es una alternativa más eficiente que múltiples if-else cuando se evalúa un mismo elemento varias veces.



# T

## Test Unitario (Unit Test)

Tipo de prueba automatizada que verifica que una unidad de código (como una función o método) funcione correctamente de manera aislada, asegurando que produzca los resultados esperados.

## Testing (Pruebas)

Proceso de evaluar software para identificar defectos y asegurar que funciona como se espera. Incluye pruebas unitarias, de integración y funcionales.

## Thread (Hilo)

Unidad más pequeña de procesamiento que puede manejarse de manera independiente dentro de un proceso. Los hilos permiten la ejecución de múltiples tareas concurrentes en un programa.

## Timeout

Periodo de tiempo máximo que se espera para que una operación o proceso finalice antes de que el sistema lo cancele. Es útil para evitar que los sistemas se bloqueen cuando una operación no responde.

## Tipado dinámico (Dynamic Typing)

Característica de algunos lenguajes de programación, como Python o JavaScript, en los que las variables no tienen un tipo fijo y pueden cambiar de tipo durante la ejecución del programa.

## **Tipado estático (Static Typing)**

Sistema de tipos en el que las variables tienen un tipo definido en tiempo de compilación. Por ejemplo, lenguajes como Java o C usan tipado estático.

## **Token**

Unidad básica de un lenguaje de programación, como una palabra clave, un operador o un identificador. En el contexto de seguridad, también puede referirse a un token de autenticación. Puede referirse al uso de tokens para representar información sensible de manera segura.

## **Tupla (Tuple)**

Colección de elementos ordenados que, a diferencia de una lista, es inmutable.

# U

## **UI (Interfaz de Usuario)**

Parte de una aplicación con la que el usuario interactúa directamente, incluyendo elementos gráficos como botones, menús, formularios, y otros componentes visuales.

## **URL (Uniform Resource Locator)**

Dirección única que identifica recursos en la web, como una página web o un archivo, compuesta por el protocolo (ej. http), el dominio y la ruta del recurso.

## **UTF-8**

Formato de codificación de caracteres que puede representar cualquier carácter en Unicode. Es ampliamente utilizado en la web para manejar texto en múltiples idiomas.

## **UX (Experiencia de Usuario)**

La forma en que un usuario percibe e interactúa con un sistema o aplicación, evaluada en términos de facilidad de uso, accesibilidad y satisfacción.

# V

## **Variable**

Contenedor que almacena datos en un programa. Las variables tienen un nombre y un tipo asociado, lo que permite que el programa acceda a los datos almacenados y los manipule.

## **Variable Global**

Variable accesible desde cualquier parte del programa. Su uso excesivo puede dificultar el mantenimiento del código y generar errores inesperados.

## **Virtualización**

Tecnología que permite la creación de versiones virtuales de recursos físicos como servidores, redes o almacenamiento, utilizando software especializado.

# W

## Web scraping

Técnica para extraer datos de sitios web de manera automatizada, utilizando herramientas o scripts que recorren el contenido HTML de una página y extraen la información necesaria.

## WebSocket

Protocolo de comunicación que permite la transmisión de datos bidireccional en tiempo real entre un cliente y un servidor, manteniendo una conexión activa durante toda la sesión.

## Wireframe

Representación visual básica de una página web o aplicación, usada en la fase inicial del diseño para mostrar la disposición de elementos clave.

## Workflow (Flujo de trabajo)

Conjunto de pasos o procesos definidos para llevar a cabo una tarea o serie de tareas de manera estructurada y automatizada en un entorno de desarrollo o producción.

## Wrapper

Código que “envuelve” una función, biblioteca o componente para agregar funcionalidades adicionales o simplificar su uso.



## **XML (Extensible Markup Language)**

Lenguaje de marcado utilizado para estructurar, almacenar y transportar datos de manera legible tanto por humanos como por máquinas. Es ampliamente utilizado en servicios web y configuraciones.

# Y

## YAML (YAML Ain't Markup Language)

Formato de serialización de datos fácil de leer y escribir, utilizado comúnmente para archivos de configuración y almacenamiento de datos. Es más legible para humanos que JSON o XML.

# Z

## **Z-index**

Propiedad en CSS que determina la posición relativa de elementos superpuestos en una página web. Un valor mayor indica que el elemento está más “arriba”.

## **Zero-Day Vulnerability**

Vulnerabilidad de seguridad desconocida para los desarrolladores del software, que puede ser explotada antes de que se publique un parche para corregirla.

## **Zip**

Formato de compresión que permite reducir el tamaño de archivos y carpetas para facilitar su almacenamiento y transmisión.



**mouredev<sup>pro</sup>**

**Estudia programación y desarrollo de software de manera diferente**

mouredev.pro