Prof. Ignacio Traverso y Sofía Mármol

Algoritmos y Estructuras de Datos
Introducción

Contenidos del documento:

- ¿Qué son los algoritmos?
- ¿Qué son las estructuras de datos?
- ¿Qué es un lenguaje de programación y de qué tipos hay?
- ¿Qué es un IDE?

Algoritmos:

En el ámbito de la informática, un algoritmo es una secuencia finita y ordenada de pasos bien definidos que se siguen para resolver un problema específico o realizar una tarea particular.

Un algoritmo tiene las siguientes cualidades:

- Secuencia finita: Un algoritmo debe tener un número limitado de pasos. No puede ser un proceso que nunca termine. Siempre debe haber un punto final. Imagina una receta que nunca termina de decirte qué hacer; ¡sería inútil!

- Ordenada: Los pasos deben seguir un orden específico. Cambiar el orden de los pasos generalmente lleva a un resultado incorrecto o a no resolver el problema. En la receta, no puedes hornear el pastel antes de mezclar los ingredientes.

- Bien definida: Cada paso en el algoritmo debe ser claro, preciso y sin ambigüedad. No debe haber lugar a interpretaciones. Una instrucción como "mezcla un poco" no es bien definida; "mezcla durante 5 minutos a velocidad media" sí lo es. En informática, cada instrucción debe ser entendible por la computadora sin ninguna duda.

- Resolver un problema específico o realizar una tarea particular: Un algoritmo siempre tiene un objetivo claro. Puede ser encontrar la ruta más corta entre dos puntos, ordenar una lista de nombres, calcular el promedio de calificaciones, o cualquier otra tarea que una computadora pueda realizar.

Estructuras de Datos:

Una estructura de datos es una forma específica de organizar y almacenar datos en la memoria de una computadora para que puedan ser accedidos y manipulados de manera eficiente.

Pensá como si fueran diferentes tipos de "contenedores" para tus datos, cada uno con sus propias ventajas y desventajas según cómo necesites usarlos:

- Imaginate que tenemos una lista de tareas, podrías simplemente escribirlas una debajo de la otra en un papel. Es fácil agregar más tareas al final, pero encontrar una tarea específica podría requerir revisar toda la lista.

- Ahora pensá en una pila de platos. Sólo podés poner o sacar platos desde la parte superior. El último plato que pusiste es el primero que sacás. Esto se asemeja a una pila (stack). Es útil para ciertas operaciones donde el orden de llegada importa (cómo deshacer acciones).

- Considera una fila para comprar entradas, La primera persona que llega es la primera en ser atendida. Esto es como una cola (queue). Los elementos entran por un extremo y salen por el otro. Es útil para procesar tareas en el orden en que llegan.

Una estructura de datos define:

I. **Cómo se organizan los datos:** La disposición lógica de los elementos y las relaciones entre ellos.
II. **Cómo se almacenan los datos:** La representación física de los datos en la memoria.
III. **Las operaciones que se pueden realizar sobre los datos:** Las formas en que podemos acceder, insertar, eliminar, buscar y modificar los datos dentro de la estructura.

La elección de la estructura de datos adecuada tiene un **impacto significativo** en:

- **Eficiencia:** La rapidez con la que se pueden realizar las operaciones (buscar, insertar, eliminar).
- **Uso de memoria:** La cantidad de espacio que se necesita para almacenar los datos.
- Facilidad de **implementación** y **mantenimiento** del código.

Lenguaje de Programación:

Un lenguaje de programación es un conjunto de reglas gramaticales y léxicas que permiten a los programadores escribir instrucciones que una computadora puede entender y ejecutar.

Pensá en ello como un idioma especial que usás para comunicarte con una computadora. Así como el español tiene sus propias reglas de gramática y vocabulario para que las personas se entiendan, un lenguaje de programación tiene sus propias reglas (**sintaxis**) y palabras clave (**comandos**) para que puedas darle órdenes precisas a la máquina.

Los lenguajes de programación se pueden clasificar generalmente en dos grandes categorías según su nivel de abstracción del hardware de la computadora: lenguajes de **bajo nivel** y lenguajes de **alto nivel**.

Lenguajes de bajo nivel:

Estos lenguajes están muy cerca del hardware de la computadora. Requieren que el programador tenga un conocimiento profundo de la arquitectura del procesador y la memoria.

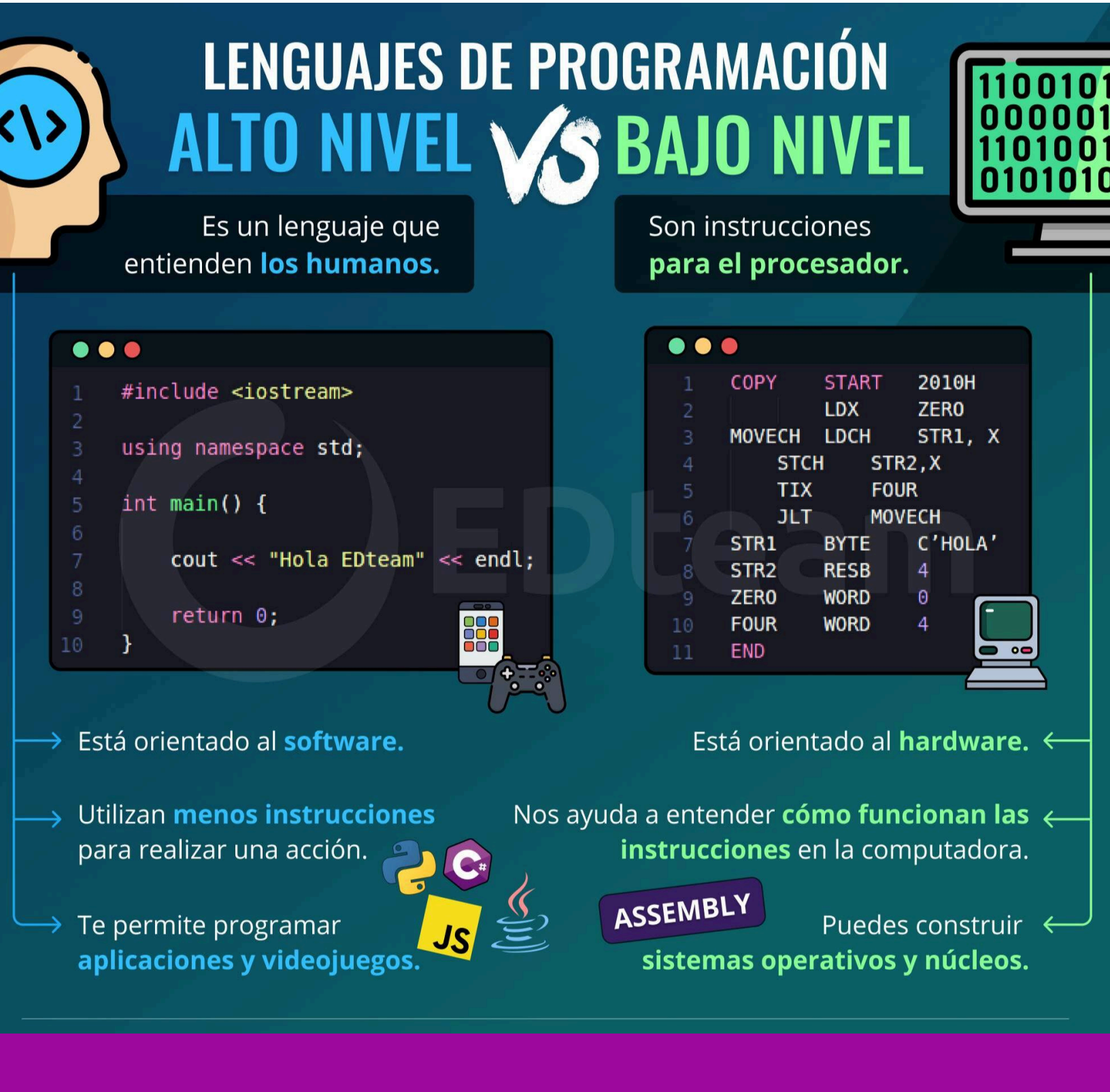
- **Lenguaje Máquina:** Es el lenguaje más básico, consiste en secuencias de 0s y 1s que la computadora entiende directamente. Es muy difícil de escribir y leer para una persona.
- **Lenguaje Ensamblador:** Es una representación simbólica del lenguaje máquina. Utiliza mnemónicos (abreviaturas) para representar las instrucciones del procesador (por ejemplo, **MOV** para mover datos, **ADD** para sumar). Necesita un programa llamado ensamblador para traducirse a código máquina.

Lenguajes de Alto nivel:

Estos lenguajes están más alejados del hardware y se centran en la facilidad de uso y la productividad del programador. Utilizan construcciones lingüísticas más cercanas al lenguaje que usaría cualquier persona.

- **Python:** Conocido por su sintaxis clara y legible, muy usado en ciencia de datos, inteligencia artificial, desarrollo web y más.
- **Java:** Un lenguaje orientado a objetos, popular para aplicaciones empresariales y desarrollo de Android.
- **JavaScript:** Principalmente usado para desarrollo web frontend (interactividad en navegadores).

- **C#:** Desarrollado por Microsoft, ampliamente usado en el desarrollo de aplicaciones de Windows, videojuegos (con Unity) y aplicaciones web con .NET.
- **C++**: Una extensión de C que añade programación orientada a objetos. Ofrece un buen equilibrio entre rendimiento y abstracción, usado en desarrollo de sistemas, videojuegos y software de alto rendimiento.



IDE:

IDE son las siglas de Entorno de Desarrollo Integrado (en inglés, Integrated Development Environment). Imaginá que sos un carpintero. Para construir un mueble, necesitás diferentes herramientas: un serrucho, un martillo, un metro, un taladro, etc. Podrías tener todas las herramientas separadas, pero sería mucho más eficiente tener un taller bien organizado donde todas las herramientas estén a tu alcance, junto con un banco de trabajo, planos y quizás incluso un manual de instrucciones.

Un IDE es como ese taller para un programador. Es una aplicación de software que reúne todas las herramientas esenciales que un desarrollador necesita en un solo lugar para escribir, probar y depurar código de manera más eficiente.