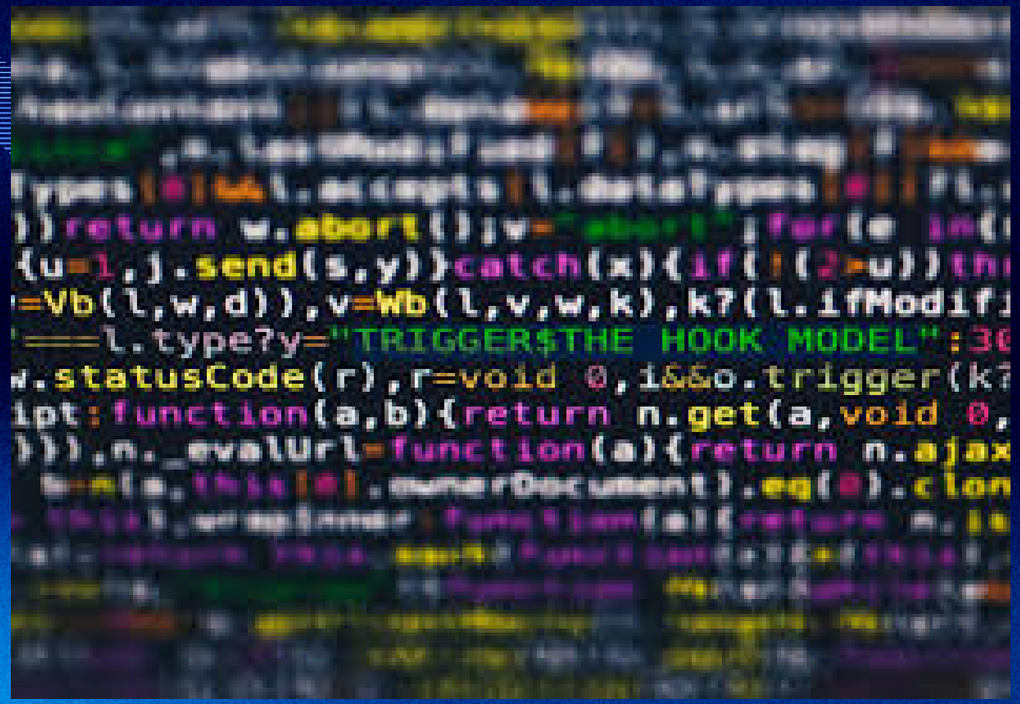
TALLER DE CONCEPTO

INTRODUCCIÓN A LAS COMPUTADORAS Y LOS LENGUAJES DE PROGRAMACIÓN

Las computadoras son máquinas que transforman datos mediante instrucciones programadas, conocidas como código. Estas instrucciones, escritas en diversos lenguajes de programación, permiten a las computadoras realizar una amplia variedad de tareas. La programación es la clave para establecer la comunicación entre los seres humanos y las máquinas, y es una habilidad esencial en el mundo actual (Knuth, 1997).

COMPUTADORAS Y SU FUNCIONAMIENTO BÁSICO

Las computadoras funcionan a través de procesos de entrada, procesamiento, almacenamiento y salida. El código de máquina es la base del funcionamiento de cualquier computadora. Los programadores deben escribir instrucciones que la CPU pueda entender y ejecutar, lo que implica una traducción entre el código fuente y las instrucciones comprensibles para la máquina (Wirth, 2002).



EVOLUCIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

A lo largo de la historia, los lenguajes de programación han evolucionado desde lenguajes de bajo nivel como el lenguaje de máquina y ensamblador, hasta lenguajes más cercanos al lenguaje humano, como Python y Java. Esta evolución ha permitido que la programación sea más accesible y eficiente, reduciendo el tiempo de desarrollo y aumentando la comprensión de los códigos escritos (Turing, 1950).

IMPORTANCIA DE LA PROGRAMACIÓN EN LA ACTUALIDAD

En la actualidad, la programación es esencial para diversas áreas, desde la creación de software hasta la inteligencia artificial. La programación no solo impulsa la tecnología, sino que también tiene un impacto significativo en la economía global y en la forma en que vivimos y trabajamos (McKinsey & Company, 2020).

TIPOS DE LENGUAJES

⚙️ Lenguaje de Máquina

El lenguaje de máquina es el nivel más bajo de programación y está compuesto por cadenas de bits (código binario) que la computadora puede entender directamente. Cada instrucción es ejecutada por la CPU sin necesidad de traducción adicional.

🛠️ Lenguaje Ensamblador

El ensamblador utiliza instrucciones mnemónicas, lo que lo hace más comprensible para los seres humanos que el lenguaje de máquina. Aunque sigue siendo un lenguaje de bajo nivel, permite que los programadores escriban código que se traduce fácilmente a lenguaje de máquina (Ritchie, 1978)..

📜 Lenguajes de Alto Nivel

Los lenguajes de alto nivel están más cerca del lenguaje humano y son más fáciles de entender y escribir. Ejemplos incluyen Java, Python, y C++. Estos lenguajes permiten a los programadores escribir código de manera más eficiente sin preocuparse tanto por los detalles de la arquitectura de la computadora (Stroustrup, 2013).

HERRAMIENTAS NECESARIAS PARA PROGRAMAR

Editor de Texto

Es una herramienta básica que permite escribir código de manera simple. Ejemplos: Notepad++, Sublime Text.

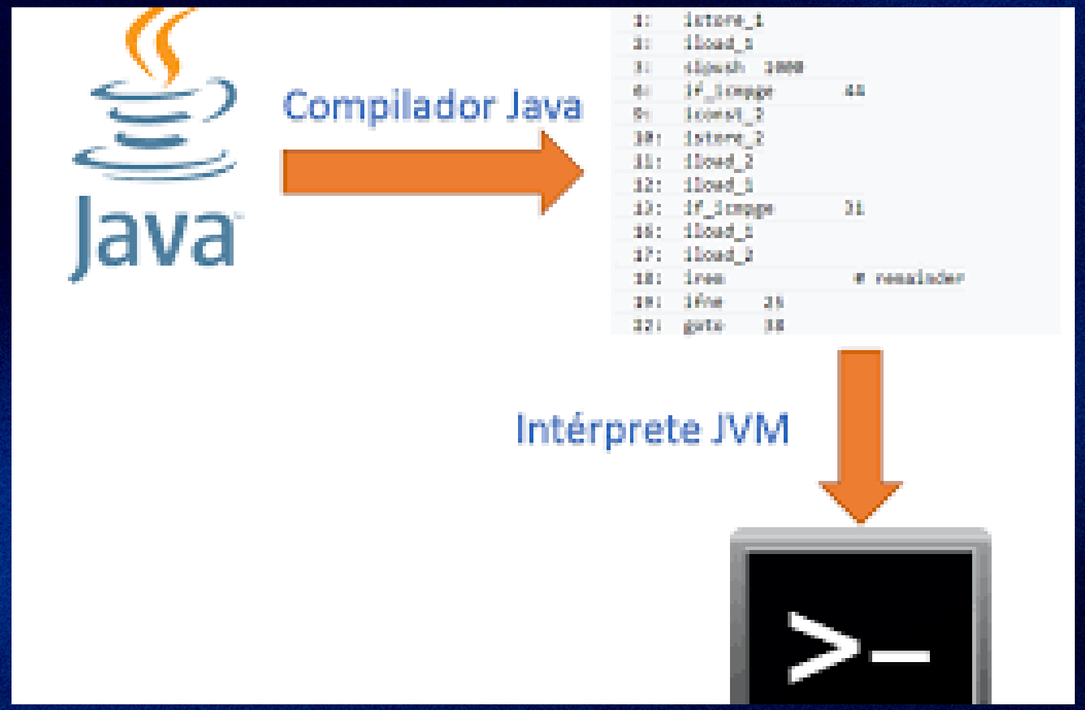
IDE (Entorno de Desarrollo Integrado)

Ofrece características avanzadas, como autocompletado, depuración, y herramientas de análisis de código. Ejemplos: Visual Studio Code, PyCharm.

COMPILADORES E INTÉRPRETES

Compiladores: Traducen el código fuente completo a código máquina antes de la ejecución, generando archivos ejecutables. Ejemplo: GCC para C/C++.

Intérpretes: Traducen y ejecutan el código línea por línea en tiempo real. Ejemplo: Python, JavaScript.



FUNDAMENTOS DE PROGRAMACIÓN

Es el soporte directo de las asignaturas: programación orientada a objetos, estructura de datos, tópicos avanzados de programación y de forma indirecta se relaciona con el desarrollo de sistemas de software, sistemas operativos y programación de sistemas.

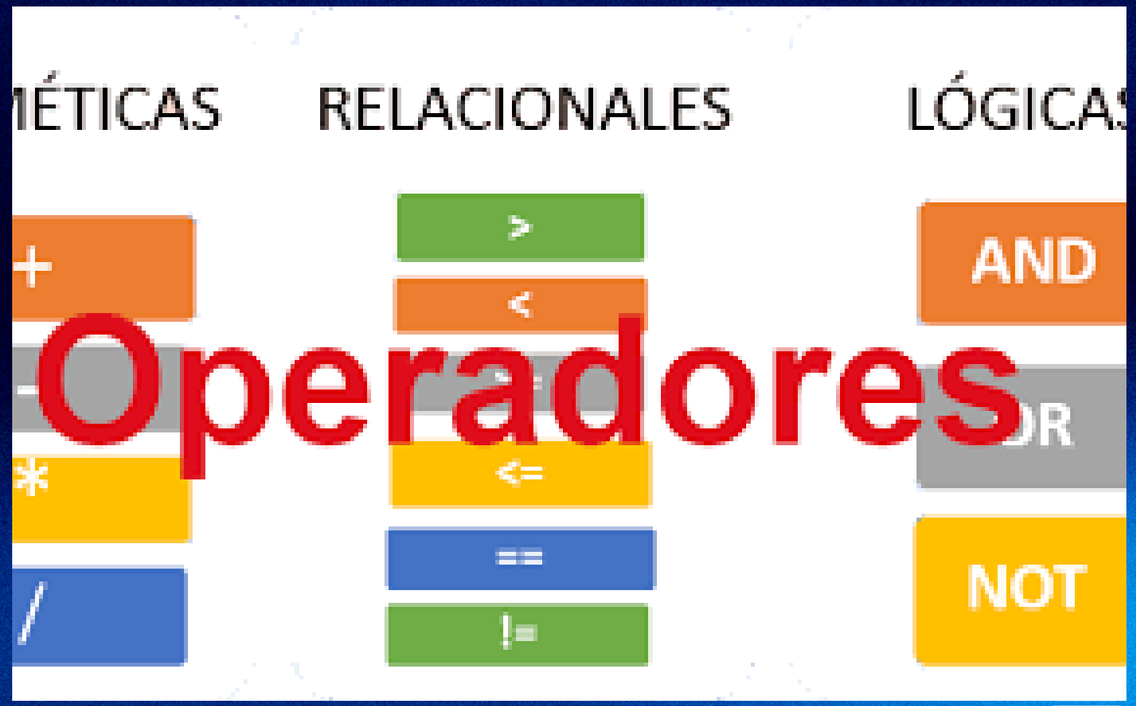
Conceptos Básicos

- Variables: Son espacios de memoria donde se almacena información.
- Constantes: Son valores que no cambian durante la ejecución del programa.
- Tipos de Datos: Definen el tipo de información que puede almacenar una variable, como enteros, cadenas de texto, flotantes, etc.

OPERADORES Y EXPRESIONES

Una expresión es una serie de elementos de valor combinados con operadores, lo que produce un nuevo valor. Los operadores actúan en los elementos de valor mediante cálculos, comparaciones y otras operaciones.

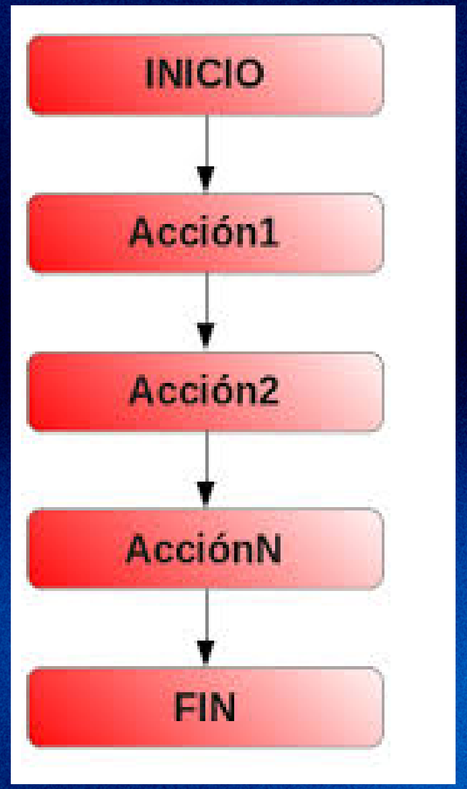
- Operadores Aritméticos: (+, -, *, /) Realizan operaciones matemáticas.
- Operadores Lógicos: (&&, ||, !) Permiten realizar comparaciones booleanas.
- Operadores Relacionales: (==, !=, <, >) Se utilizan para comparar valores.



CONTROL DE FLUJO

El término control de flujo se utiliza para describir el método en el que un dispositivo serie controla la cantidad de datos que se transmiten al mismo.

- Condicionales: Permiten tomar decisiones en el código. Ejemplos: if, switch.
- Bucles: Permiten repetir bloques de código. Ejemplos: for, while.



REFERENCIAS EN NORMAS APA

- American Psychological Association. (2020). Publication manual of the American Psychological Association (7a ed.). https://doi.org/10.1037/0000165-000
- Knuth, D. E. (1997). The art of computer programming (Vol. 1). Addison-Wesley.
- McKinsey & Company. (2020). The future of work in technology. https://www.mckinsey.com/insights
- Ritchie, D. M. (1978). The C programming language. Prentice-Hall.
- Stroustrup, B. (2013). The C++ programming language (4th ed.). Addison-Wesley.
- Turing, A. M. (1950). Computing machinery and intelligence. Mind, 59(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433
- Wirth, N. (2002). Algorithms + data structures = programs. Prentice-Hall.

MUCHAS GRACIAS

www.reallygreatsite.com