

**De computadoras**

**ALGORITMOS Y PROGRAMACIÓN**

**Espacio para el texto**

**Espacio para el texto**



***INDICADOR***

**FUNDAMENTOS DE SOLUCIÓN DE PROBLEMAS POR COMPUTADORA.**

**ESTRUCTURA DE DATOS Y OPERATORES.**

**DISEÑO DE ALGORITMOS.**

**ESTRUCTURA DE CONTROL SECUENCIAL.**

**ESTRUCTURA DE CONTROL CONDICIONAL.**

**ELEMENTOS AUXILIARES DE LOS ALGORITMOS.**

**ESTRUCTURA DE CONTROL ITERATIVAS.**

**ALGORITMIAS CON ESTRUCTURAS COMBINADAS.**

**ARREGLOS UNIDIMENCIONALES.**

**ALGORITMIA DE BÚSQUEDA.**

**ALGORITMIA DE ORDENACIÓN Y DE INSERCIÓN.**

**LA LÓGICA DE PROGRAMCIÓN USANDO MATRICES.**

**ARREGLOS BIDIMENSIONALES.**

**RECORRIDO DE ARREGLOS MULTIDIMENCIONALES.**

**APLICANDO ARREGLOS MULTIDIMENCIONALES.**

**ANÁLISIS PARA EL ALGORITMO DE INTELIGENCIA** ARTIFICIAL**.**

**Fundamentos de la solución de problemas por computadora**

Solución de problemas utilizando una computadora. Incluye la comprensión del problema, el desarrollo de un enfoque de solución, la descomposición del problema en pasos más pequeños y la implementación de una solución utilizando programación.

. Análisis del problema.

. Descomposición del problema en subproblemas.

. Abstracción y generalización.

. Implementación de soluciones utilizando programación.

**Estructura de datos y operadores**

Las estructuras de datos realizan algunas operaciones especiales solo como inserción y estudian las diferentes estructuras de datos, como listas, conjuntos, pilas, colas, etc. Además, se exploran los operadores utilizados para manipular y acceder a los datos almacenados en estas estructuras.

**. Listas (arreglos) y sus operaciones.**

**. Conjuntos y operaciones de conjuntos.**

**. Pilas (stacks) y colas (queues).**

**. Operadores aritméticos y lógicos.**

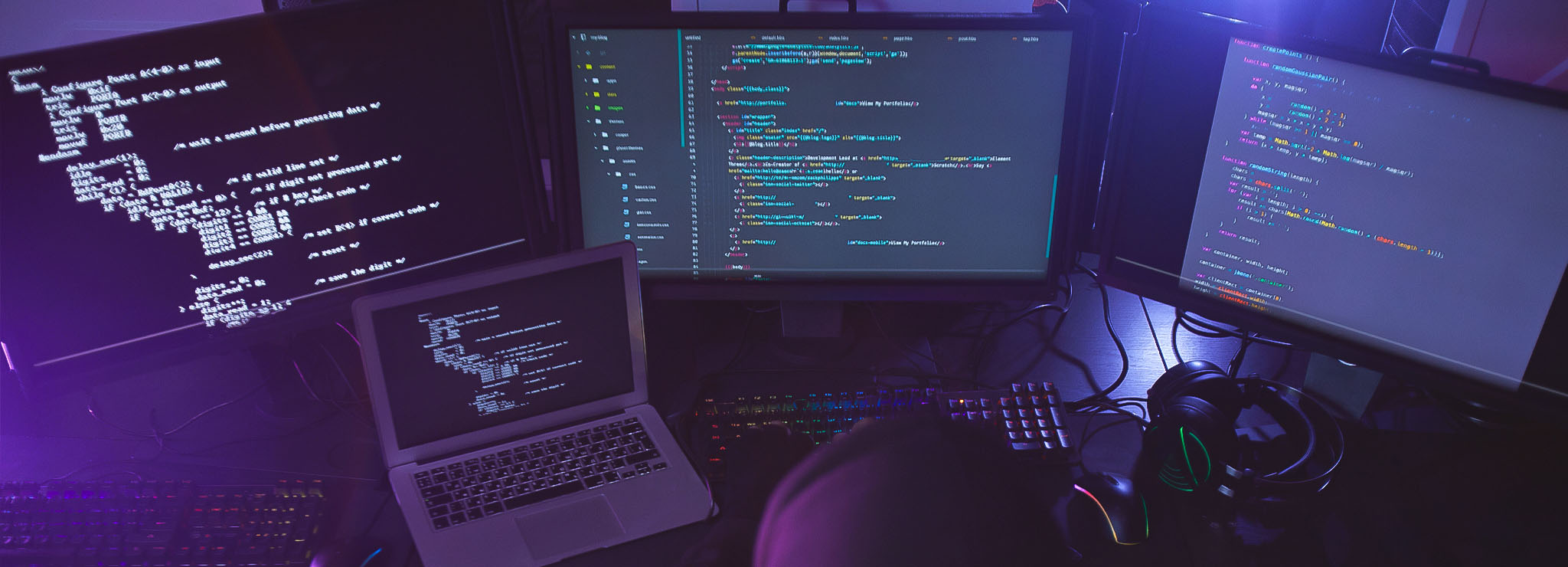


**Diseño de algoritmos**

**. Algoritmos iterativos y recursivos.**

**. Estrategias de diseño de algoritmo.**

**El diseño de algoritmos se enfoca en la creación de procedimientos paso a paso para resolver un problema. Este tema incluye técnicas de diseño, como la subdivisión del problema en subproblemas más pequeños, la recursión y la utilización de estructuras de datos adecuadas.**

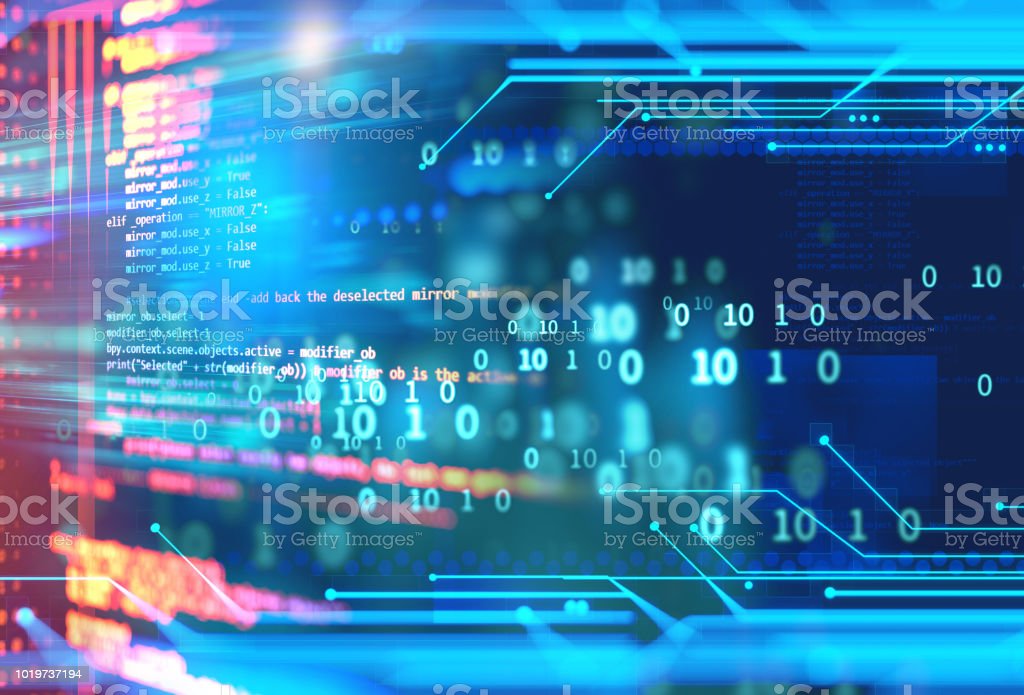
**Estructuras de control secuencial**

**. Secuencia de instrucciones.**

**. Flujo de ejecución del programa.**

**. Instrucciones de entrada y salida.**

**Las estructuras de control secuencial permiten ejecutar instrucciones en un orden específico, de arriba a abajo, sin saltos o repeticiones. Este concepto es fundamental para comprender la ejecución secuencial de un programa.**

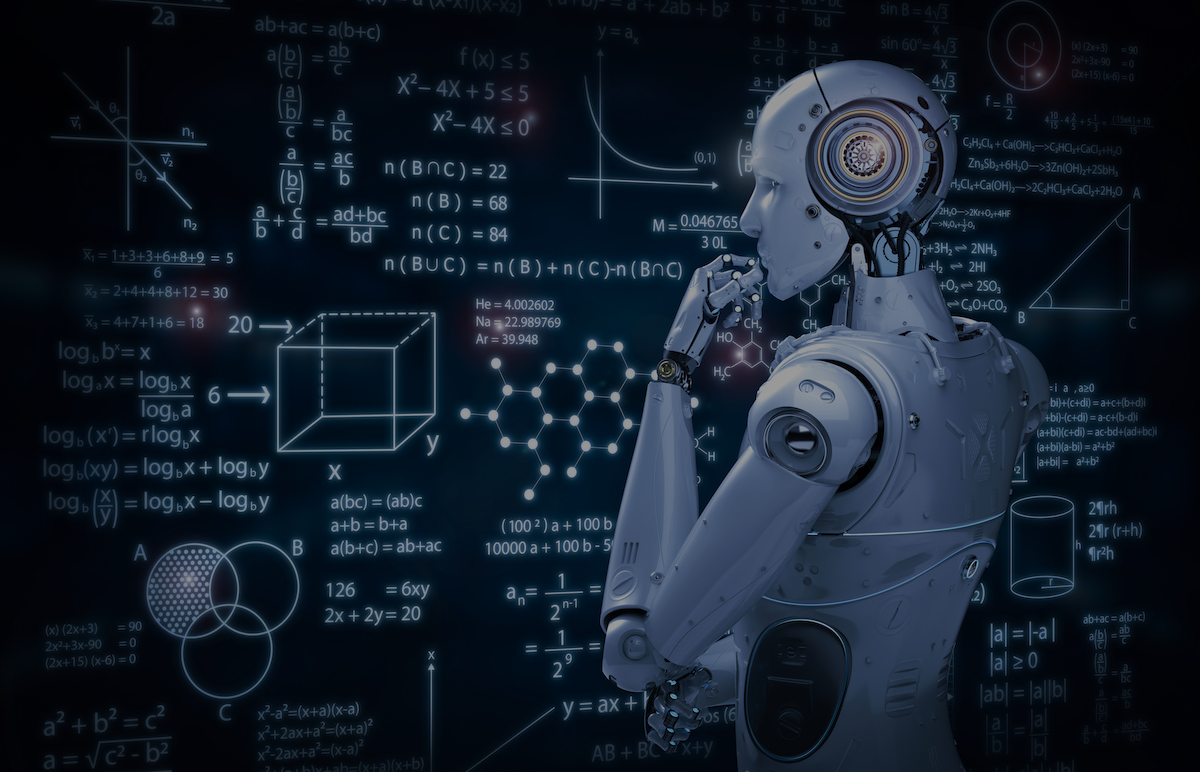
**Estructuras de control condicional**

. Condicionales "si-entonces".

. Condicionales "si-sino".

. Operadores de comparación y lógicos.

Las estructuras de control condicional permiten tomar decisiones en función de ciertas condiciones. Los condicionales, como "si-entonces" y "si-sino", son utilizados para realizar diferentes acciones en base a los valores de las variables o el resultado de una expresión.

**Elementos auxiliares de los algoritmos**

**. Variables y tipos de datos.**

**. Operaciones matemáticas y lógicas.**

**. Expresiones y asignación de valores.**

**Son variables que realizan funciones concretas dentro de un programa, y por su gran utilidad, frecuencia de uso y peculiaridades, conviene hacer un estudio separado, y son de utilidad en la construcción de algoritmos eficientes, como el manejo de variables, operaciones matemáticas básicas, asignación de valores y la comprensión de expresiones lógicas.**

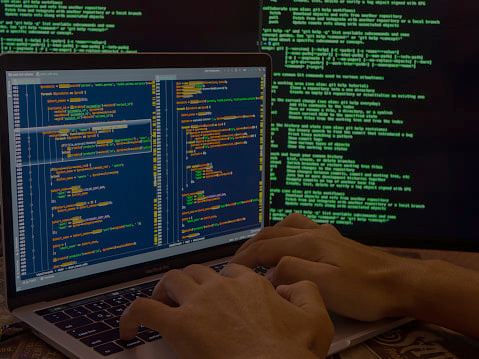
**Estructuras de control iterativas**

**. Bucles "para" (for).**

**. Bucles "mientras" (while).**

**. Bucles "hacer mientras" (do-while).**

**Las estructuras de control iterativas, como los bucles (por ejemplo, bucle "para" y bucle "mientras"), permiten repetir una secuencia de instrucciones múltiples veces. Estas estructuras son útiles para procesar datos en serie o realizar cálculos iterativos.**

**Algoritmias con estructuras combinadas**

**. Combinación de condicionales y bucles.**

**. Uso de variables auxiliares.**

**. Diseño de algoritmos complejos.**

**Se exploran algoritmos más complejos que combinan diferentes estructuras de control, como condicionales y bucles, para resolver problemas más elaborados.**

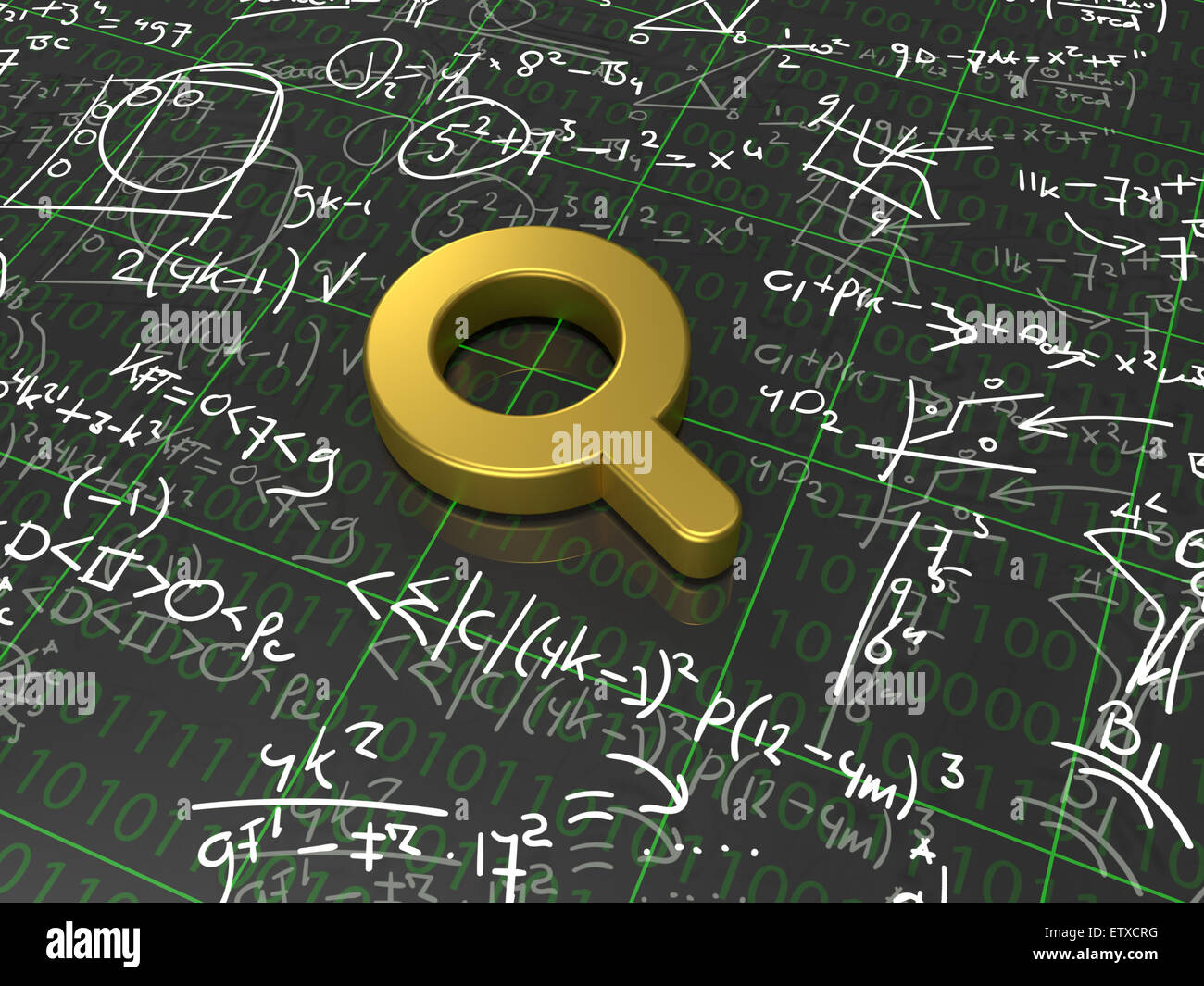
**ARREGLOS UNIDIMENSIONALES**

**Los arreglos unidimensionales, también conocidos como vectores o listas, son estructuras de datos que permiten almacenar y acceder a múltiples valores del mismo tipo. Se analiza cómo declarar, inicializar y manipular estos arreglos.**

**. Declaración e inicialización de arreglos.**

**. Acceso y modificación de elementos.**

**. Búsqueda y modificación en arreglos.**

**Algoritmia de búsqueda**

**. Búsqueda lineal.**

**. Búsqueda binaria.**

**. Búsqueda de patrones.**

La algoritmia de búsqueda se enfoca en la implementación de algoritmos para buscar elementos específicos dentro de una estructura de datos, como un arreglo.

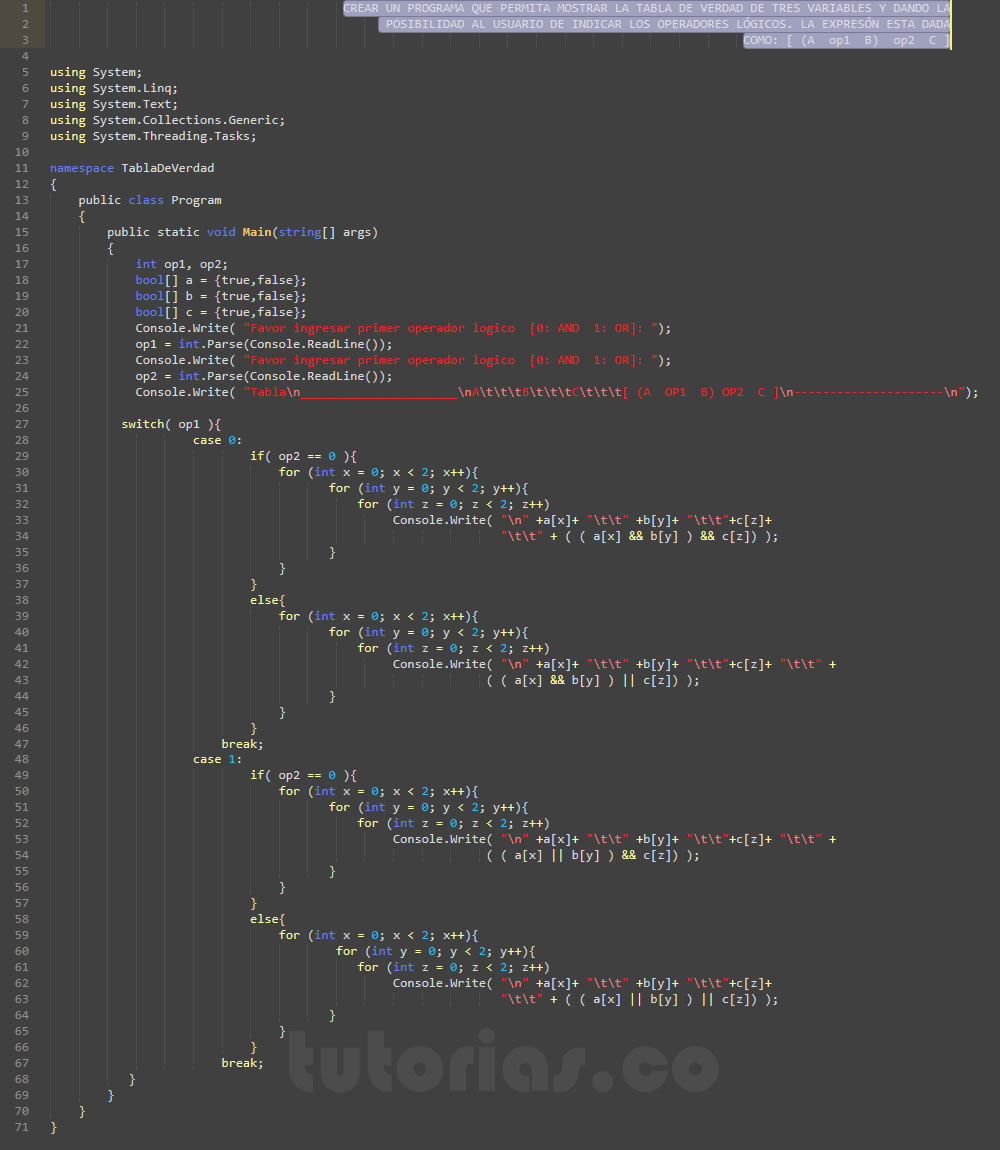
**Algoritmia de ordenación y de inserción**

**Son algoritmos para ordenar elementos en una estructura de datos. Se analizan diferentes métodos de ordenación, como la ordenación por inserción, la ordenación por selección y la ordenación rápida.**

**. Ordenación por inserción.**

**. Ordenación por selección.**

**. Ordenación rápida (quicksort) y ordenación por mezcla (mergesort).**

**La lógica de programación usando matrices**

. **Declaración e inicialización de matrices.**

**. Acceso y modificación de elementos en una matriz.**

**. Operaciones y manipulaciones de matrices.**

**El uso de matrices (arreglos bidimensionales) para almacenar y manipular datos en forma de tablas o grillas. Se explora cómo acceder a elementos específicos de una matriz y realizar operaciones sobre ellas.**

**Arreglos bidimensionales**



**Los arreglos bidimensionales son estructuras de datos que permiten almacenar datos en una tabla o matriz de filas y columnas. Se estudia cómo declarar, inicializar y acceder a los elementos de estos arreglos.**

**. Declaración e inicialización de arreglos bidimensionales.**

**. Acceso y modificación de elementos en una matriz bidimensional.**

**. Recorrido y procesamiento de arreglos bidimensionales.**

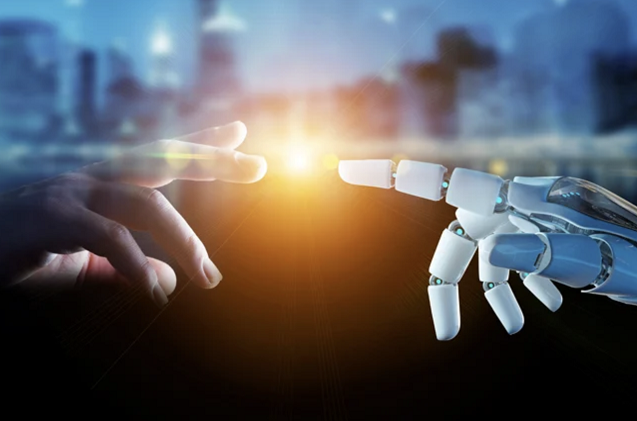
**Recorrido de arreglos multidimensionales**

**Se exploran técnicas para recorrer y procesar todos los elementos de un arreglo multidimensional. Se analizan bucles anidados y otras estrategias para acceder a los elementos de manera ordenada.**

**. Uso de bucles anidados para recorrer y acceder a elementos de arreglos multidimensionales.**

**. Operaciones y algoritmos aplicados a arreglos multidimensionales.**

**Aplicando arreglos multidimensionales**

**Análisis para el algoritmo de inteligencia artificial**

**. Representación y manipulación de datos estructurados en forma de matrices.**

**. Aplicaciones prácticas de arreglos multidimensionales, como imágenes, tablas de datos, etc.**

**cómo utilizar arreglos multidimensionales en la solución de problemas concretos, como la representación de tableros de juegos, imágenes o datos estructurados en forma de matrices.**

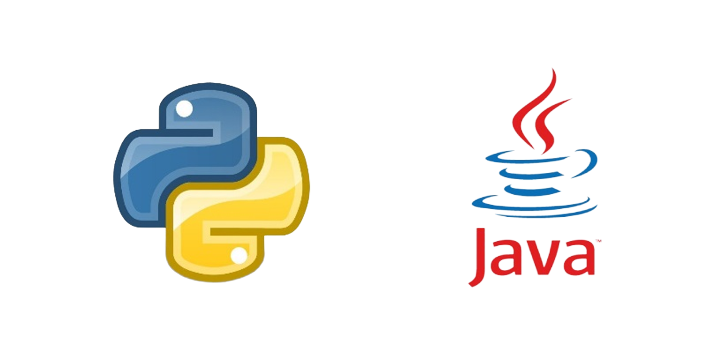
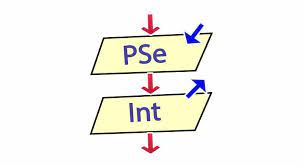
**Se centra en el análisis de algoritmos utilizados en la inteligencia artificial, como los algoritmos de aprendizaje automático y redes neuronales. Se exploran conceptos como la optimización, la evaluación de algoritmos y la eficiencia computacional.**

**. Optimización y evaluación de algoritmos de inteligencia artificial.**

**. Medición de la eficiencia y rendimiento de los algoritmos.**

**. Análisis de complejidad en algoritmos de inteligencia artificial.**

**Integrantes**

****

Collantes Portilla Candy

Montalván Pintado Edilsa

Nanfuñay Carrión Javier

Neyra Quesquén Renzo