**Universidad Nacional de Ingeniería**

**Recinto Universitario Simón Bolívar**

**RUSB**



**ALGORITMIZACION Y ESTRUCTURA DE DATOS**

**Elaborado por:**

* Br. Bejarano Carrión Eduard Antonio.
* Br. Espinoza López María Esther.
* Br. Gutiérrez Espinoza María Natasha.
* Br. Ruiz Sánchez Gabriela Vanessa.

**Docente:**

MSC Jorge Prado D.

**Grupo**:

2M7-CO

**Fecha: martes, 26 de noviembre de 2024**

# **UNIDAD I:**

# **DISEÑOS Y ANALISIS DE ALGORITMOS**

1. Estructura de Datos y Algoritmos
   1. Definición

Un algoritmo es un procedimiento para realizar una tarea específica. Las estructuras de datos son formas de organizar y almacenar datos para que puedan ser utilizados de manera eficiente.

* 1. Características

Características de los Algoritmos:

Las características fundamentales que debe cumplir todo algoritmo son:

1. Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
2. Un algoritmo debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
3. Un algoritmo debe ser finito. Si se sigue un algoritmo se debe terminar en algún momento; o sea, debe tener un numero finito de pasos.
   1. Algoritmo

Ejemplo de Algoritmo:

Un cliente ejecuta un pedido a una fábrica. Esta examina en su banco de datos la ficha del cliente; si el cliente es solvente entonces la empresa acepta el pedido; en caso contrario rechazara el pedido. Redactar el algoritmo correspondiente.

Los pasos del algoritmo son:

* 1. inicio
  2. leer el pedido
  3. examinar la ficha del cliente
  4. si el cliente es solvente aceptar pedido; en caso contrario, rechazar Pedido
  5. fin

1. Necesidad de las estructuras de datos

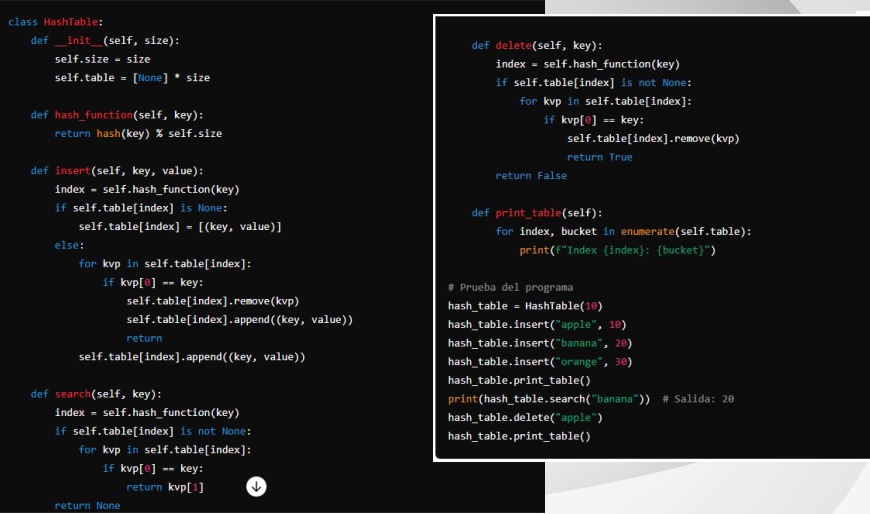
Las estructuras de datos son esenciales para organizar y gestionar eficientemente grandes volúmenes de datos, permitiendo el acceso y modificación rápida de los mismos.

* 1. Tablas Hash
     1. Definición

Una tabla hash es una estructura de datos que asocia llaves o claves con valores.

* + 1. Características

1. Permite el acceso a los elementos almacenados a partir de una clave generada.
2. Transforma la clave con una función hash a un hash, un número que la tabla hash utiliza para localizar el valor deseado.
   * 1. Algoritmo



1. Tipos de datos abstractos (TDA).

Un tipo de dato abstracto es una especificación de un conjunto de datos y las operaciones que se pueden realizar sobre ellos, sin especificar la implementación.

Se consideran en lineales y no lineales.

* 1. Lineales
     1. Pila
        1. Definición

Una pila (stack), es una estructura de datos de tipo lista lineal en la cual la inserción y eliminación de los elementos o datos se realiza solo de un extremo, el cual se denomina cima o tope.

* + - 1. Característica

1. El último elemento insertado en ella es el primero en suprimirse. Por esta razón, una pila se denomina como una estructura LIFO (Last In First Out) en la cual, el último elemento insertado, será el primero en ser eliminado.
   * + 1. Algoritmo



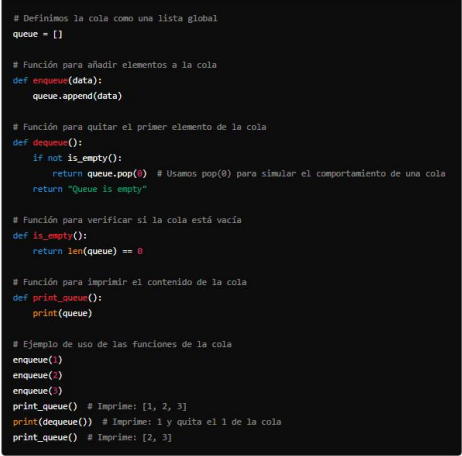
* + 1. Cola
       1. Definición

Una cola es una estructura de datos que almacena elementos en una lista y permite acceder a los datos por uno de los dos extremos de la lista. Un elemento se inserta en la cola (parte final) de la lista y se suprime o elimina por la frente (parte inicial, cabeza) de la lista.

* + - 1. Característica

Una cola (también llamada fila) es un tipo de dato abstracto, caracterizada por ser una secuencia de elementos en la que la operación de inserción push se realiza por un extremo y la operación de extracción pop por el otro.

* + - 1. Algoritmo

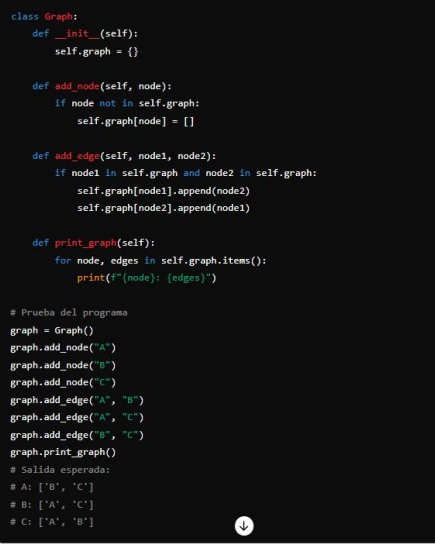


* 1. No lineales
     1. Grafos
        1. Definición

Un grafo en el ámbito de las ciencias de la computación es un tipo abstracto de datos (TAD), que consiste en un conjunto de nodos (también llamados vértices) y un conjunto de arcos (aristas) que establecen relaciones entre los nodos. El concepto de grafo TAD desciende directamente del concepto matemático de grafo.

* + - 1. Característica

1. Se llama bucle o lazo a toda arista de la forma (v, v).
2. Se llaman aristas múltiples a las aristas que aparecen en E.
3. Se dice que dos vértices son adyacentes si están unidos por una arista.
4. Se dice que dos aristas son adyacentes si tienen un vértice en común.
5. Se dice que una arista y un vértice son incidentes si el vértice es extremo de la arista.
6. Se dice que un vértice es aislado si no es adyacente a ningún otro vértice.
7. Se dice que un grafo es simple si no tiene bucles ni aristas múltiples.
   * + 1. Algoritmo



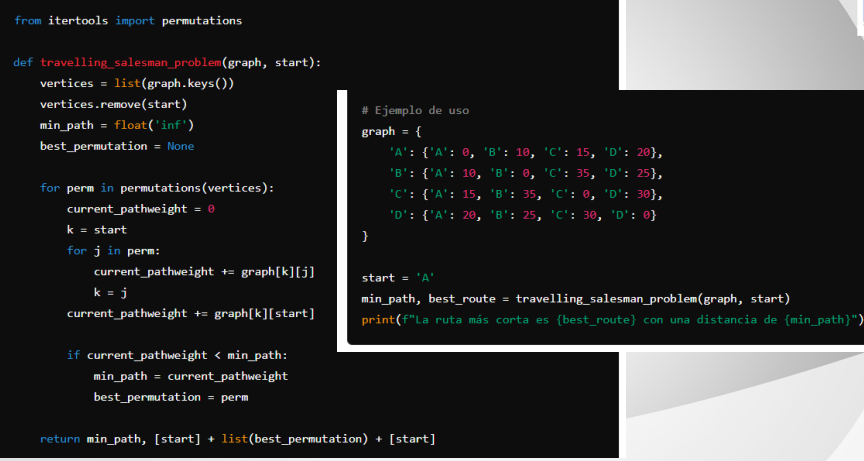
1. Problemas, Algoritmos y Programas

Los problemas son tareas específicas que necesitan ser resueltas. Los algoritmos son soluciones generales a estos problemas y los programas son implementaciones específicas de estos algoritmos en un lenguaje de programación.

* 1. Problema del Viajante

Encontrar la ruta más corta que visita todas las ciudades.

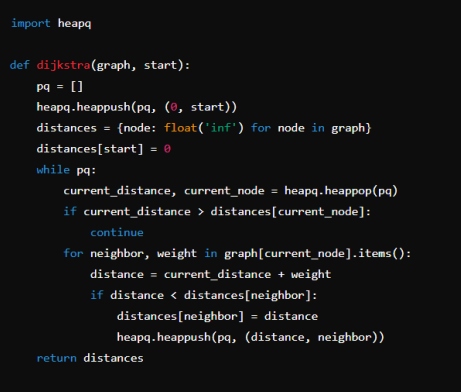
* + 1. Algoritmo

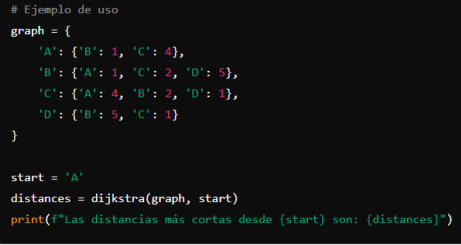


* 1. Algoritmo de Dijkstra

Para encontrar la ruta más corta en un grafo.

* + 1. Algoritmo





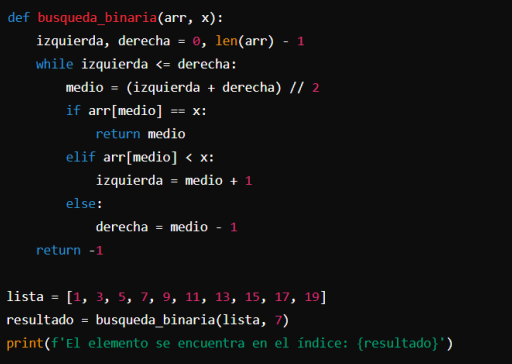
1. Eficiencia de Algoritmos

La eficiencia de un algoritmo se refiere a la cantidad de recursos (tiempo y espacio) que utiliza durante su ejecución.

* 1. Algoritmo de búsqueda binaria
     1. Definición

La búsqueda binaria es un algoritmo eficiente para encontrar un elemento en una lista ordenada de elementos. Funciona al dividir repetidamente a la mitad la porción de la lista que podría contener al elemento, hasta reducir las ubicaciones posibles a solo una.

* + 1. Algoritmo



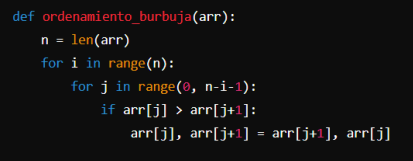
* 1. Algoritmo de ordenamiento por burbuja
     1. Definición

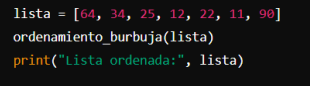
Es un algoritmo de ordenación simple que recorre repetidamente la lista que hay que ordenar, comparando cada par de elementos adyacentes y los intercambia si están en el orden equivocado. Este procedimiento se repite hasta que no se realicen intercambios.

* + 1. Característica

El ordenamiento burbuja hace múltiples pasadas a lo largo de una lista. Compara los ítems adyacentes e intercambia los que no están en orden. Cada pasada a lo largo de la lista ubica el siguiente valor más grande en su lugar apropiado. En esencia, cada ítem “burbujea” hasta el lugar al que pertenece.

* + 1. Algoritmo

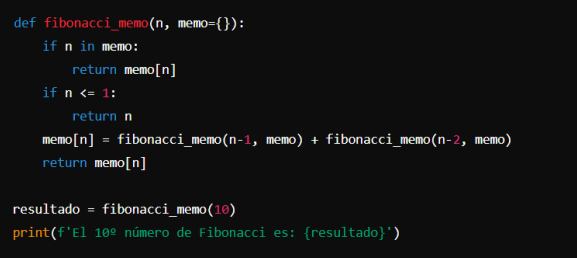




* 1. Algoritmo de Fibonacci usando memorización
     1. Definición

El Algoritmo de Fibonacci es una serie numérica en la que cada número es la suma de los dos anteriores, partiendo de 0 y 1. Es un concepto sencillo y significativo en informática con los casos base F (0) = 0, F (1) = 1, y el caso recursivo F(n) = F(n-1) + F(n-2).

* + 1. Algoritmo



# **UNIDAD II:**

# **ALGORITMOS RECURSIVOS**

1. Recursividad
   1. Definición

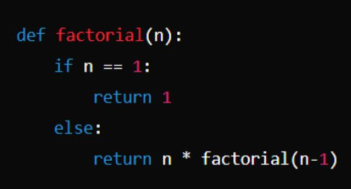
Un procedimiento o función se dice recursivo si durante su ejecución se invoca directa o indirectamente a sí mismo. Esta invocación depende al menos de una condición que actúa como condición de corte que provoca la finalización de la recursión.

Un algoritmo recursivo consta de:

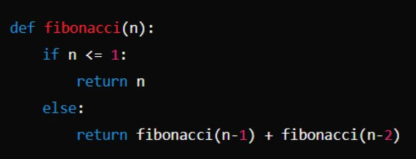
1. Al menos un caso trivial o base, es decir, que no vuelva a invocarse y que se ejecuta cuando se cumple cierta condición, y
2. el caso general que es el que vuelve a invocar al algoritmo con un caso más pequeño del mismo.

Para que una función o procedimiento recursivo funcione se debe cumplir que:

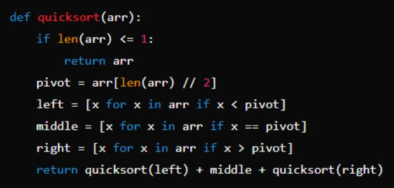
1. Existe una salida no recursiva del procedimiento o función y funciona correctamente en ese caso.
2. Cada llamada al procedimiento o función se refiere a un caso más pequeño del mismo.
3. Funciona correctamente todo el procedimiento o función
   1. Algoritmos
      1. Calcular el factorial de un número.



* + 1. Encontrar el n-ésimo numero de Fibonacci.



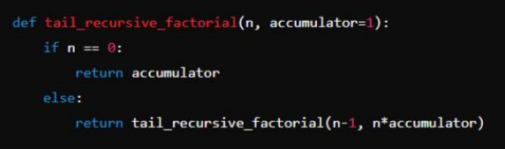
* + 1. Algoritmos de ordenamiento como el quicksort.



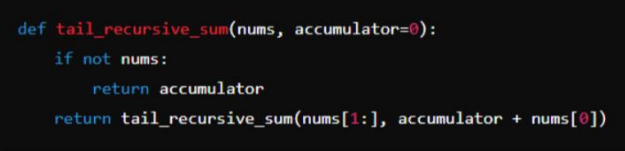
1. Recursión de extremo final
   1. Definición

La recursión de extremo final ocurre cuando la última operación de una función es llamada recursiva. Esto es eficiente porque no necesita mantener un estado intermedio.

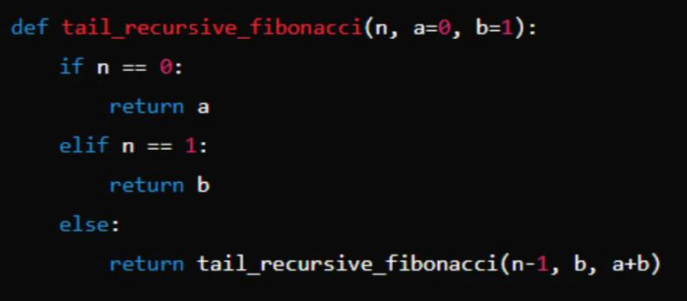
* 1. Algoritmos
     1. Calcular el factorial de un número usando recursión de extremo final.



* + 1. Algoritmo de la suma de los primeros n números naturales.



* + 1. Encontrar el máximo común divisor (MCD) de dos números.



1. Algoritmo de rastreo inverso
   1. Definición

El algoritmo de rastreo inverso (backtracking) es una técnica recursiva para resolver problemas donde se prueba una solución parcial y se retrocede (backtrack) si se determina que la solución no es viable.

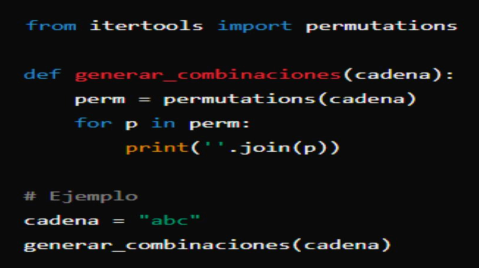
* 1. Algoritmos
     1. Resolver el problema del sudoku.



* + 1. Encontrar todas las soluciones del problema de las n reinas.



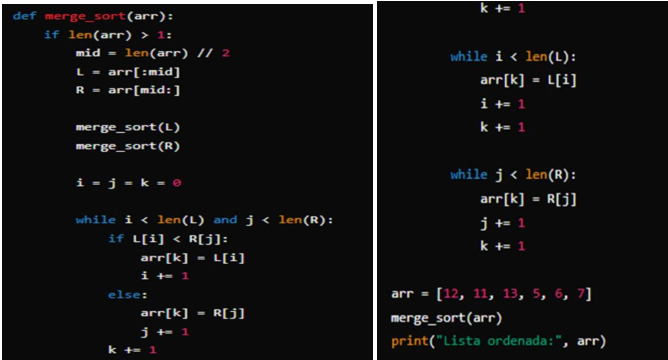
* + 1. Generar combinaciones de cadenas de caracteres.



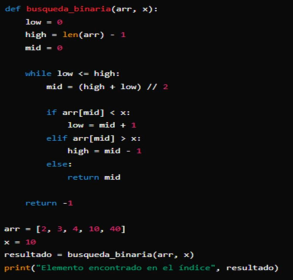
1. Principio Divide y vencerás
   1. Definición.

El principio de divide y vencerás implica dividir un problema en sub-problemas más pequeños, resolverlos de manera recursiva y combinar sus soluciones.

* 1. Algoritmos.
     1. Algoritmo de mergesort para ordenar una lista.



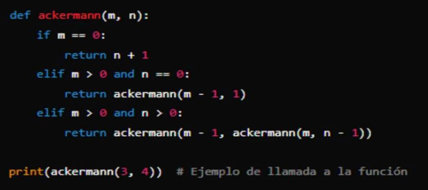
* + 1. Algoritmo de búsqueda binaria.



1. Explicación de los algoritmos: Función de Ackermann, Torre de Hanoi y Las Ocho Reinas.
   1. Función de Ackermann
      1. Definición

La función de Ackermann es un ejemplo de una función recursiva que no es primitiva recursiva, lo que significa que no puede ser expresada sólo con recursión primitiva. Esta función crece muy rápidamente y se define de la siguiente manera: (Algoritmo).

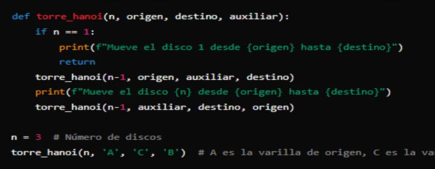
* + 1. Algoritmo



* 1. Torre de Hanoi
     1. Definición

El problema de la Torre de Hanoi es un rompecabezas matemático que consiste en tres varillas y un número de discos de diferentes tamaños que pueden deslizarse sobre cualquier varilla. El rompecabezas comienza con los discos apilados en una varilla en orden ascendente de tamaño. El objetivo es mover toda la pila a otra varilla, respetando las siguientes reglas:

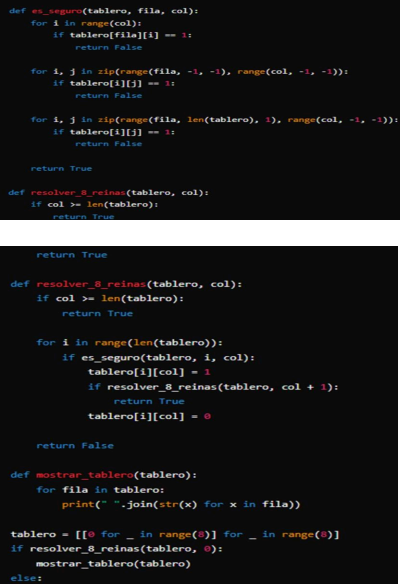
1. Solo se puede mover un disco a la vez.
2. Cada movimiento consiste en tomar el disco superior de una de las pilas y colocarlo en otra pila.
3. Ningún disco puede ser colocado sobre un disco más pequeño.
   * 1. Algoritmo



* 1. Las Ocho Reinas
     1. Definición

El problema de las Ocho Reinas consiste en colocar ocho reinas en un tablero de ajedrez de 8x8 de manera que ninguna de ellas se amenace mutuamente. Es decir, no debe haber dos reinas en la misma fila, columna o diagonal.

* + 1. Algoritmo



# **UNIDAD III:**

# **ESTRUCTURAS DE DATOS FUNDAMENTALES**

1. Listas

Una lista es una colección ordenada de elementos en la que cada elemento tiene una posición determinada. Se pueden añadir, eliminar y acceder a los elementos utilizando su índice.

Características de las listas

1. Ordenadas: Los elementos en una lista tienen un orden específico.
2. Indexadas: Cada elemento tiene un índice asociado que representa su posición en la lista.
3. Mutables: Pueden modificarse después de su creación.
4. Homogéneas o Heterogéneas: Pueden contener elementos del mismo tipo o diferentes tipos.
5. Capacidad Dinámica: Pueden crecer o reducirse según sea necesario.

Ejemplos prácticos:

1. Lista de compras: Una lista que contiene los nombres de los productos que necesitas comprar, como ["Leche", "Pan", "Huevos"].
2. Lista de tareas: Una lista con tareas que debes completar, como ["Hacer la tarea", "Limpiar la casa", "Ir al gimnasio"].
3. Lista de invitados: Una lista de personas invitadas a una fiesta, como ["Ana", "Juan", "María"].

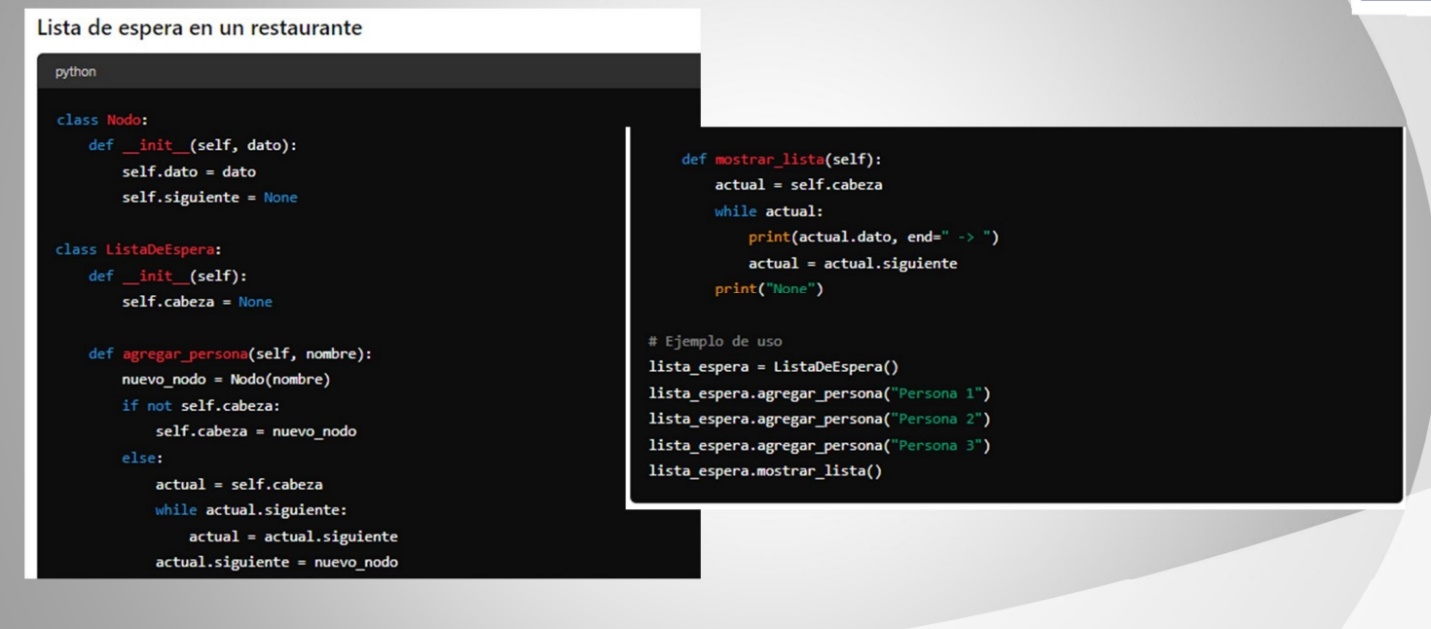


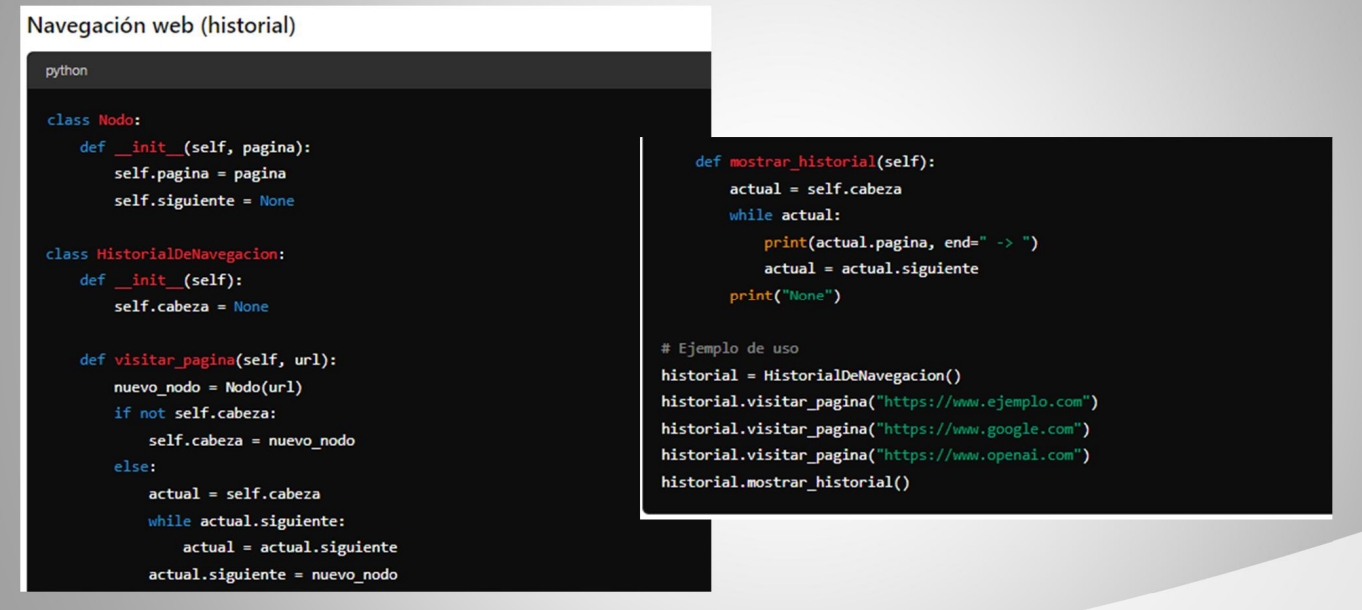
* 1. Listas Enlazadas
     1. Definición

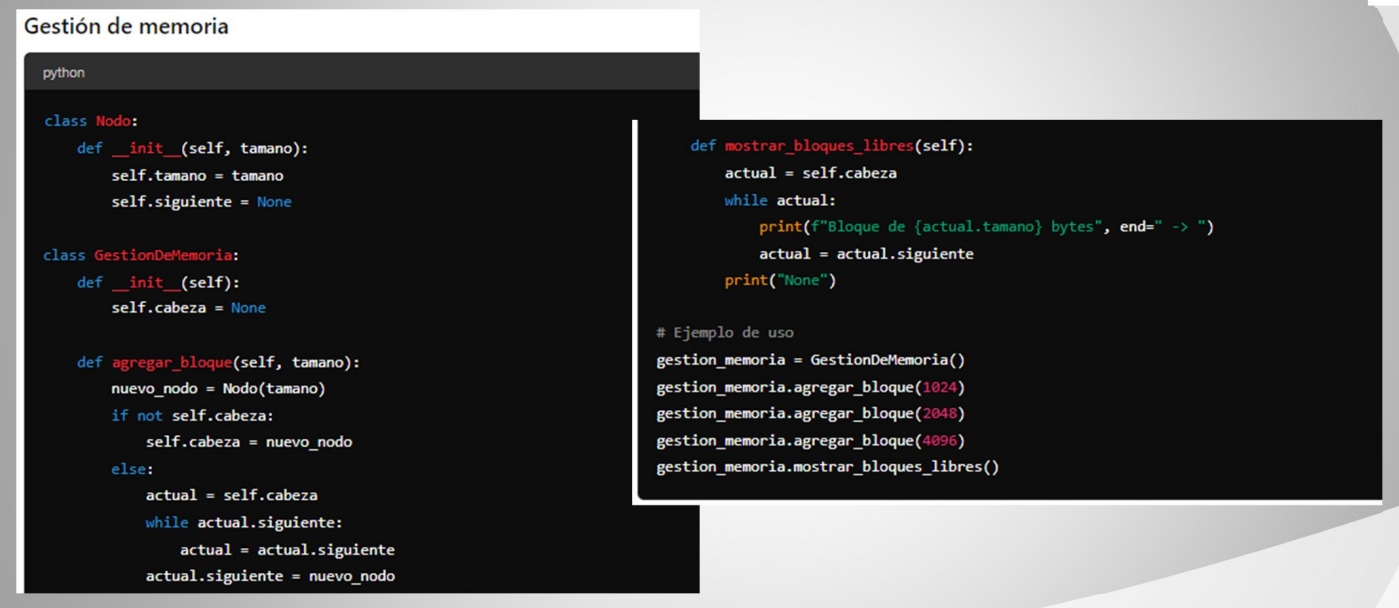
Una lista enlazada es una estructura de datos en la que cada elemento (nodo) contiene un valor y un puntero al siguiente nodo de la lista.

* + 1. Ejemplos Prácticos

1. Lista de espera en un restaurante: Cada nodo representa una persona esperando ser atendida, con un puntero a la siguiente persona en la lista.
2. Navegación web (historial): Cada nodo representa una página web visitada, con un puntero a la siguiente página en el historial.
3. Gestión de memoria: Uso de listas enlazadas para manejar bloques de memoria libre.



****

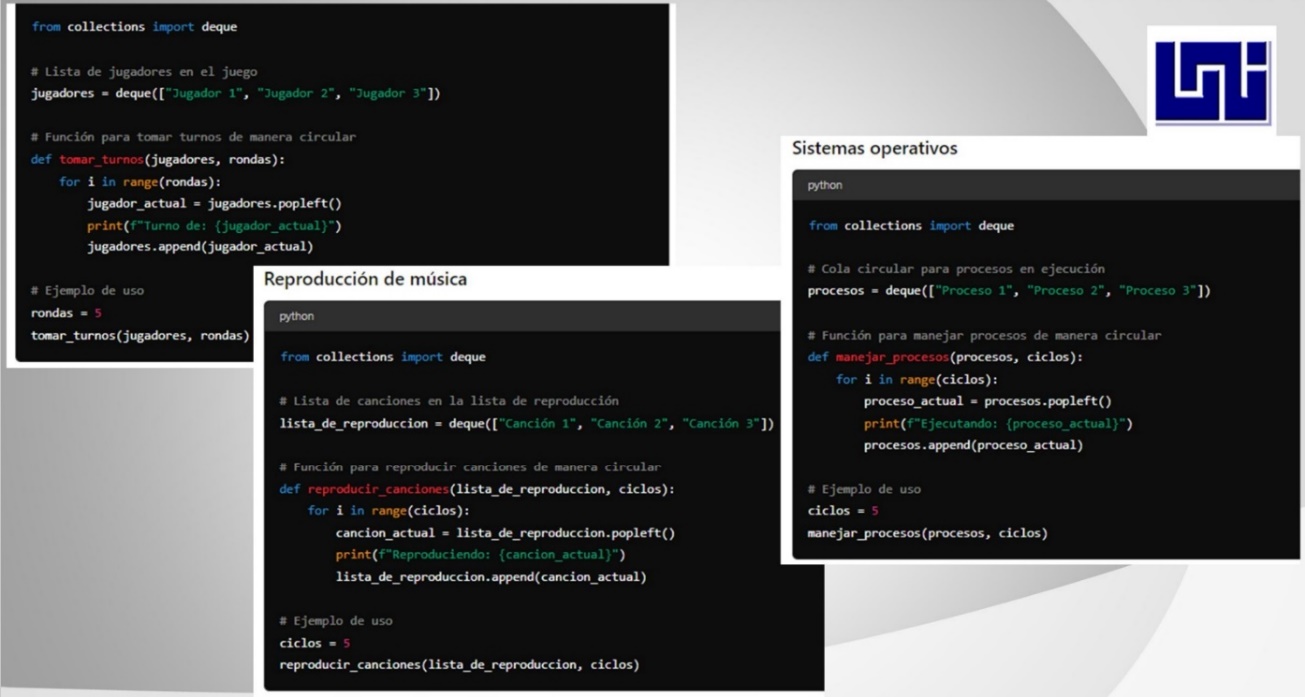
****

* 1. Listas Circulares
     1. Definición

Una lista circular es una lista enlazada en la que el último nodo apunta de nuevo al primer nodo, formando un ciclo.

* + 1. Ejemplos Prácticos

1. Juegos de ronda: Representar a los jugadores de un juego que toman turnos de manera circular.
2. Reproducción de música: Una lista de reproducción que vuelve al primer tema después del último.
3. Sistemas operativos: Manejo de procesos en ejecución usando una cola circular.

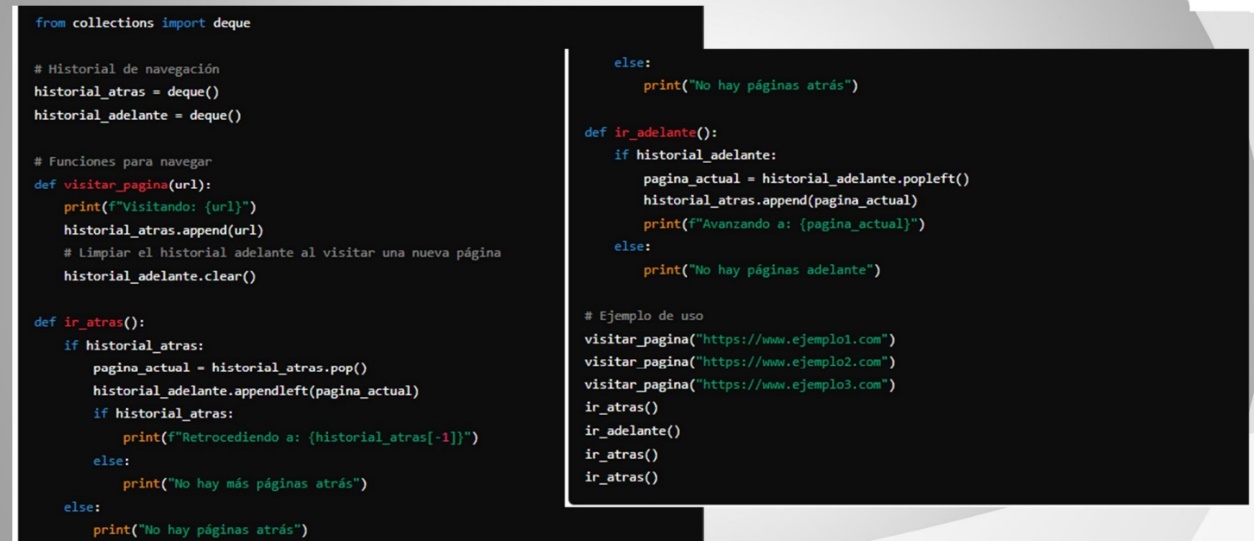


* 1. Listas Doblemente Enlazadas
     1. Definición

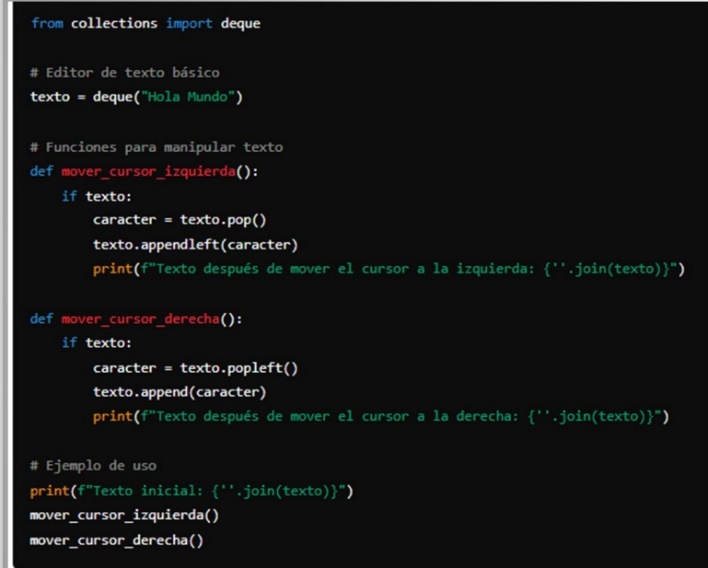
Una lista doblemente enlazada es una lista en la que cada nodo tiene un puntero al nodo siguiente y al nodo anterior.

* + 1. Ejemplos Prácticos

1. Navegadores web: Permiten ir hacia adelante y hacia atrás en el historial de navegación.



1. Editor de texto: Manipulación de caracteres con la capacidad de moverse en ambas direcciones.

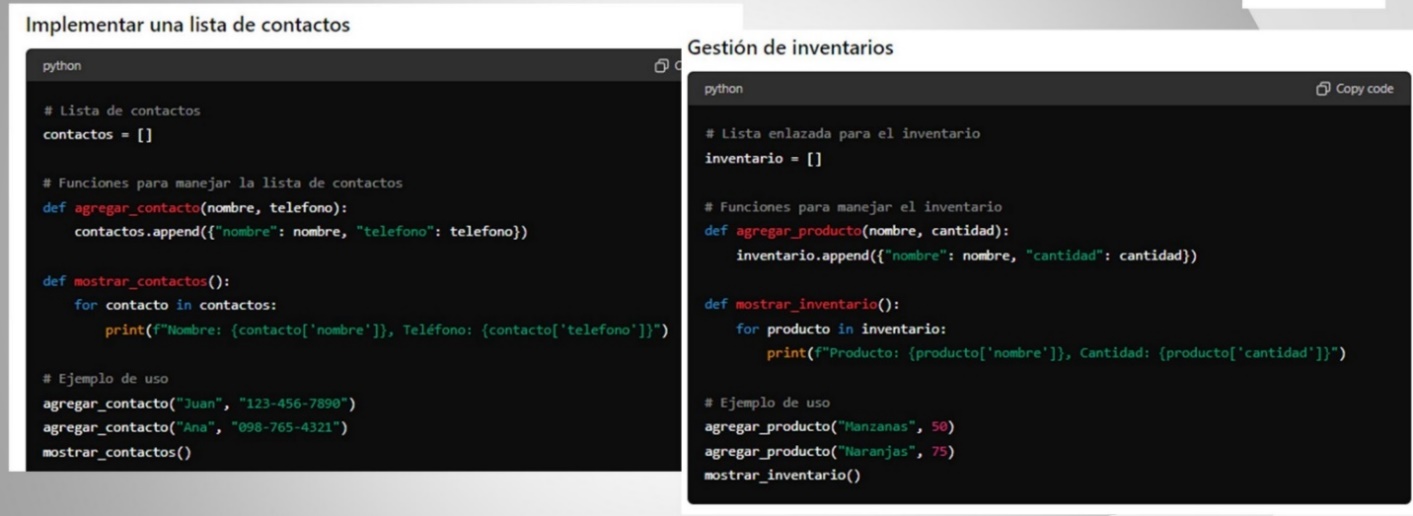


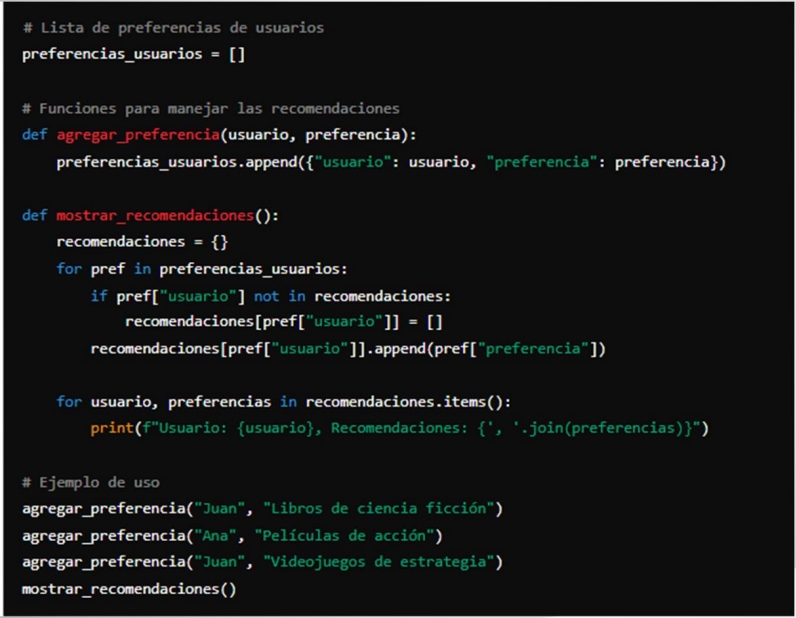
1. Aplicaciones de calendario: Navegación de fechas anterior y siguiente.



* 1. Aplicaciones de Lista
     1. Ejemplos Prácticos

1. Implementar una lista de contactos: Crear una aplicación que maneje una lista de contactos.
2. Gestión de inventarios: Desarrollar un sistema de inventario que utilice listas enlazadas.
3. Sistema de recomendación: Crear una lista de recomendaciones basadas en preferencias de usuarios.





1. Pilas

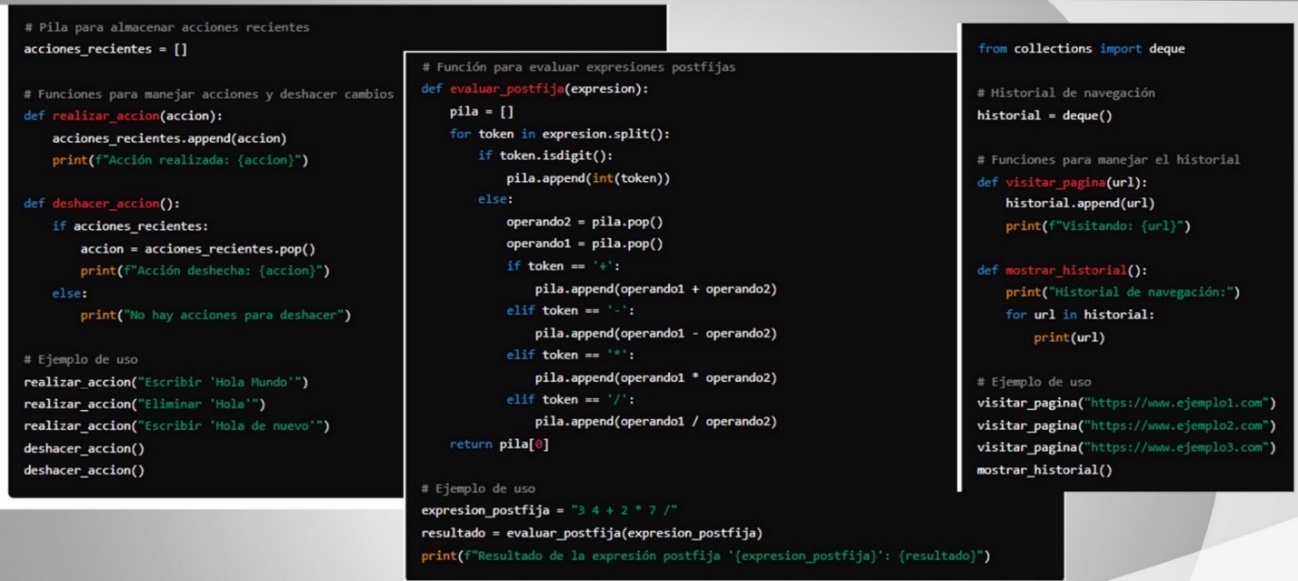
Una pila es una estructura de datos donde el último elemento en ser añadido es el primero en ser eliminado (LIFO).

Características principales de las pilas (stacks)

1. Last In, First Out (LIFO): El último elemento añadido es el primero en ser eliminado.
2. Operaciones Básicas: Las operaciones fundamentales son Push (agregar un elemento en la parte superior) y Pop (eliminar el último elemento añadido).
3. Estructura Lineal
4. Limitación de Capacidad
5. Eficiencia en la Gestión de Datos

Ejemplos prácticos:

1. Deshacer en editores de texto: Almacenar las acciones recientes para permitir deshacer cambios.
2. Evaluación de expresiones matemáticas: Uso de pilas para evaluar expresiones infijas y posfijas.
3. Navegadores web: Manejo del historial de páginas visitadas.

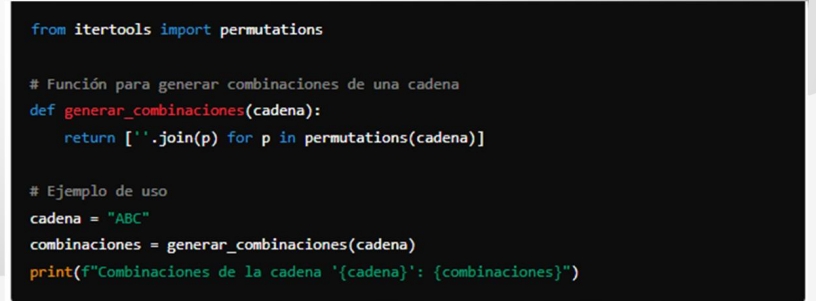


* 1. Pilas Basadas en Arreglos
     1. Definición

El algoritmo de rastreo inverso (back tracking) es una técnica recursiva para resolver problemas donde se prueba una solución parcial y se retrocede (backtrack) si se determina que la solución no es viable.

* + 1. Ejemplo

Generar combinaciones de cadenas de caracteres.

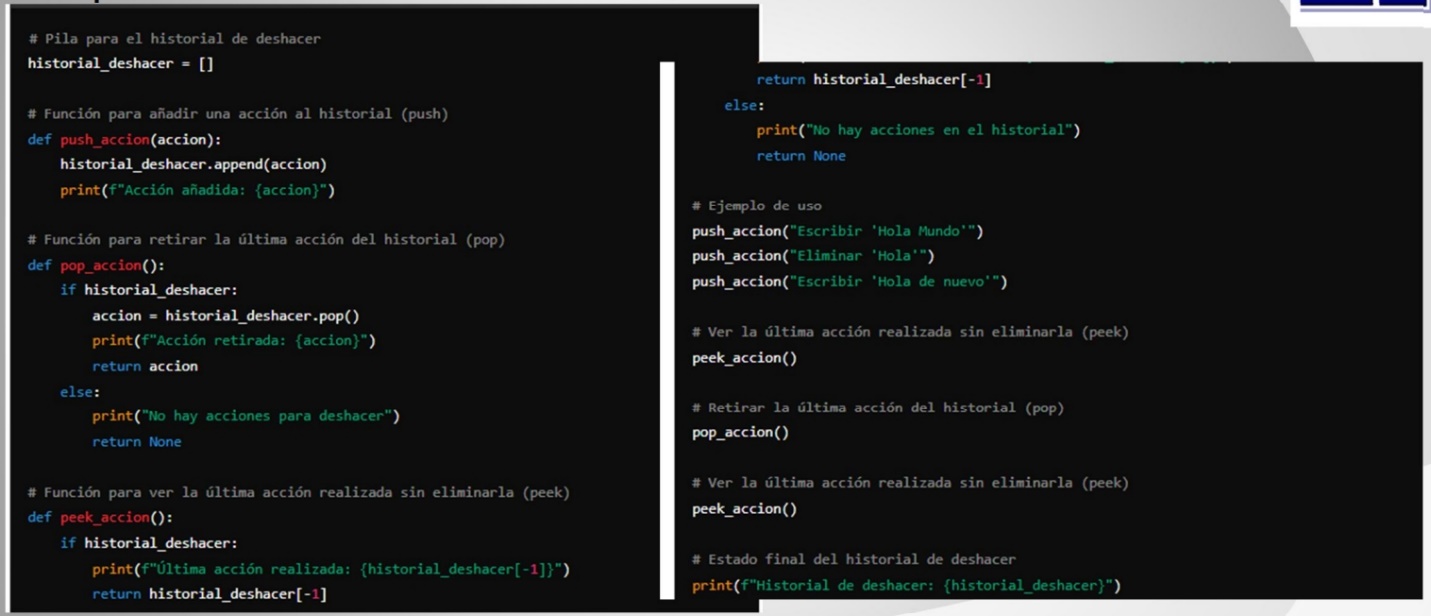


* 1. Operaciones en Pilas
     1. Definición

Las operaciones básicas en pilas son push (añadir un elemento), pop (eliminar el elemento superior), y peek (ver el elemento superior sin eliminarlo).

* + 1. Ejemplo

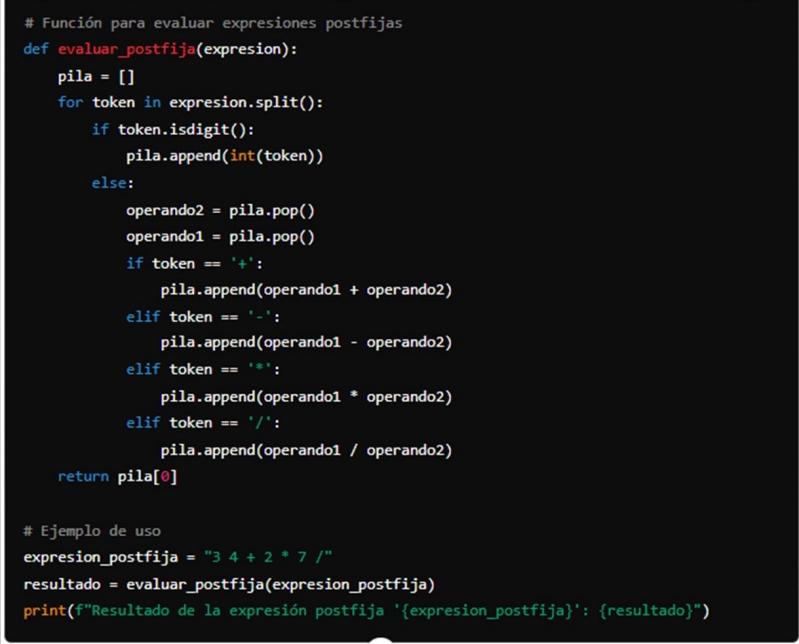
1. Push: Añadir una acción al historial de deshacer.
2. Pop: Retirar la última acción del historial de deshacer.
3. Peek: Ver la última acción realizada sin eliminarla.



* 1. Aplicaciones en Pilas

1. Simulador de deshacer/rehacer: Implementar un simulador que permita deshacer y rehacer acciones.
2. Analizador de expresiones matemáticas: Desarrollar una aplicación que evalúe expresiones matemáticas.
3. Solución de laberintos: Crear una solución de laberintos usando back tracking con pilas.





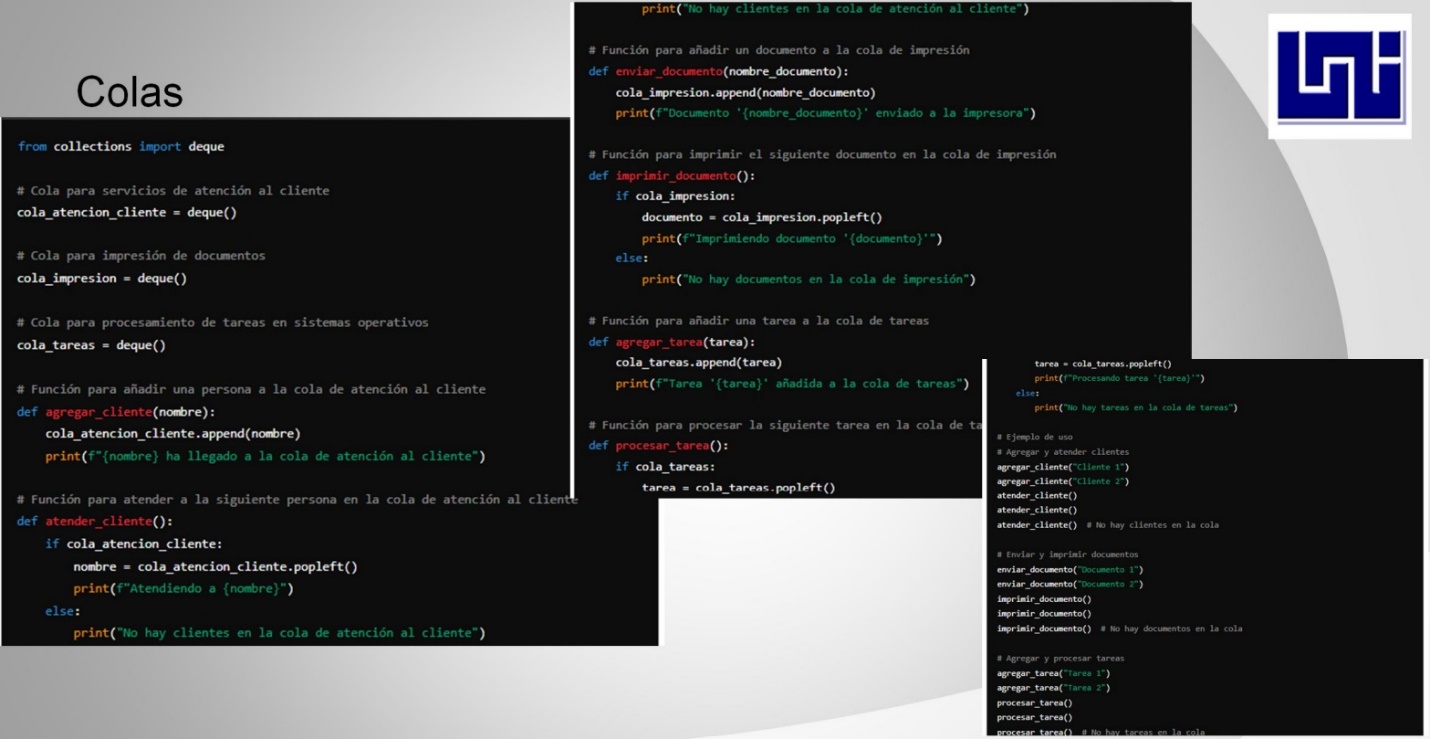
1. Colas
   1. Definición

Una cola es una estructura de datos donde el primer elemento en ser añadido es el primero en ser eliminado (FIFO).

* 1. Características

Características de las colas:

1. *Disciplina Disciplinada:* La disciplina FIFO asegura un orden de procesamiento ordenado y predecible, crucial en situaciones donde la secuencia de llegada es significativa.
2. *Estructura Lineal:*Donde los elementos se almacenan y acceden de manera secuencial. Cada elemento tiene un sucesor y un predecesor, excepto el primero y el último, respectivamente.
3. *Uso en Algoritmos:*Las colas son esenciales en algoritmos como la búsqueda en amplitud (BFS), donde se exploran nodos en un grafo en el orden en que fueron descubiertos.
4. *Aplicaciones Prácticas:*Se utilizan para manejar datos en función del orden temporal de su llegada, como en la impresión de documentos o la gestión de tareas en sistemas operativos.
5. *Colas de Espera:*En sistemas informáticos, se utilizan para gestionar colas de espera, como en el caso de solicitudes a servidores, asegurando un procesamiento ordenado.
6. *Implementación con Listas o Arreglos:* Las colas pueden implementarse utilizando listas enlazadas o arreglos, cada uno con sus propias ventajas y desventajas en términos de complejidad temporal y espacial.
7. *Simplicidad y Poder:* Las colas son fundamentales y poderosas en la programación, proporcionando un enfoque eficiente y ordenado para el procesamiento de datos.
   1. Ejemplos Prácticos
8. Cola en servicios de atención al cliente: Las personas son atendidas en el orden en que llegaron.
9. Impresión de documentos: Los documentos se imprimen en el orden en que se enviaron a la impresora.
10. Procesamiento de tareas en sistemas operativos: Las tareas se procesan en el orden en que se reciben.

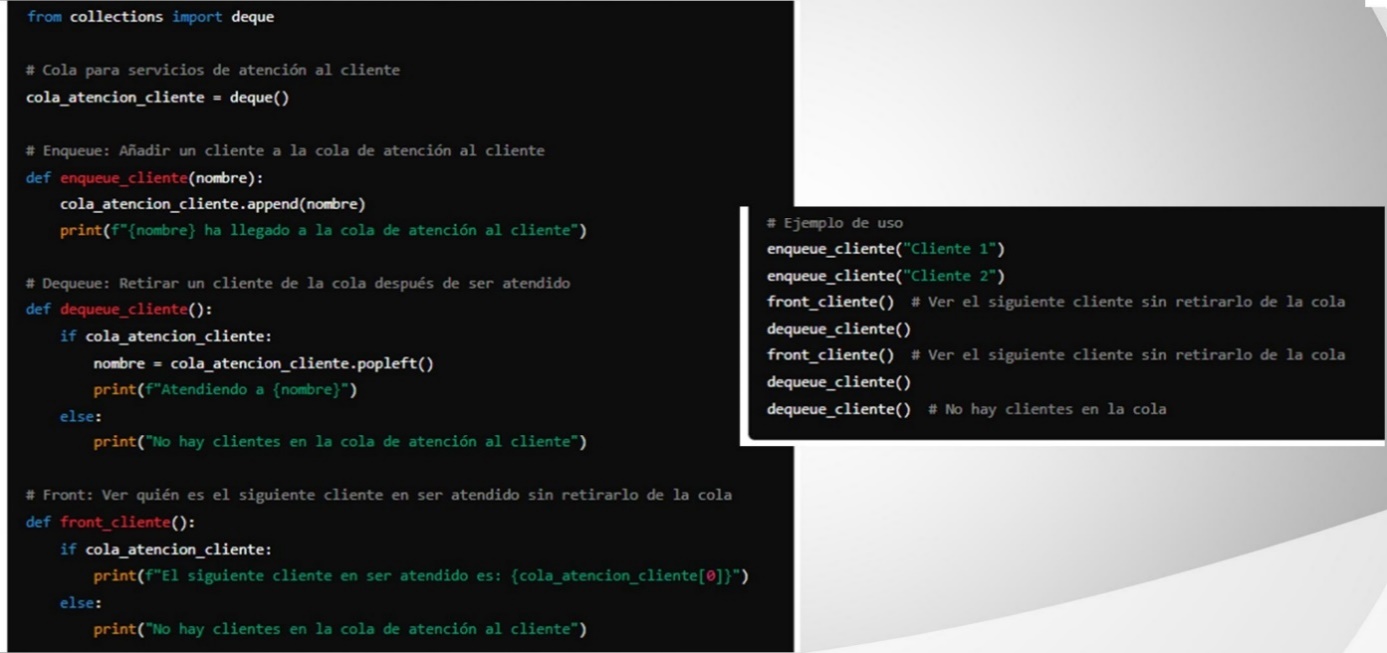


* 1. Operaciones con Cola
     1. Definición

Las operaciones básicas en colas son enqueue (añadir un elemento), dequeue (eliminar el elemento del frente), y front (ver el elemento del frente sin eliminarlo).

* + 1. Ejemplos Prácticos

1. Enqueue: Añadir un cliente a la cola de atención al cliente.
2. Dequeue: Retirar un cliente de la cola después de ser atendido.
3. Front: Ver quién es el siguiente cliente en ser atendido sin retirarlo de la cola.



* 1. Aplicaciones de Cola
     1. Ejemplos Prácticos

1. Simulador de cola de impresora: Crear un simulador que gestione una cola de documentos para impresión.
2. Sistema de atención al cliente: Desarrollar una aplicación que simule una cola de atención al cliente.
3. Planificador de procesos: Implementar un planificador de procesos utilizando colas.

