**Logotipo

Descripción generada automáticamenteInstituto Tecnológico Nacional de México**

**Instituto Tecnológico de Mérida**

**ADA 3**

**Maestro:**

Armando López Valadez

**Grupo:**

3SB

**Alumno:**

David Nicolas Rabiela Blandez

Luis Enrique Carrillo Lopez

Alexis Josimar Bernal Flores

Saul André Cauich Cool

**Fecha de entrega:**

10/05/2025

**Métodos de ordenamiento**

**Código:**

def intercalacion(lista1, lista2):

    resultado = []

    i = j = 0

    while i < len(lista1) and j < len(lista2):a

        if lista1[i] <= lista2[j]:

            resultado.append(lista1[i])

            i += 1

else:

            resultado.append(lista2[j])

            j += 1

    resultado += lista1[i:] + lista2[j:]

    return resultado

a = [1, 3, 5, 7]

b = [2, 4, 6, 8]

print("Intercalación:", intercalacion(a, b))



**Compilación:**

Lo que hace este código es unir 2 listas ya ordenadas en una sola lista ordenada

**¿Por qué su usa?:** es muy útil cuando se tienen 2 listas previas y se necesitan juntar como en los sistemas por partes

**¿Se puede mejorar?:** Si están ordenadas n se necesita mejoras si no están ordenadas hay que ordenarlas primero con otro método como Mergesort

**Mezcla Directa (Mergesort)**

**Código:**

def merge\_sort(lista):

    if len(lista) <= 1:

        return lista

    medio = len(lista) // 2

    izquierda = merge\_sort(lista[:medio])

    derecha = merge\_sort(lista[medio:])

    return intercalacion(izquierda, derecha)

lista = [7, 2, 9, 4, 1]

print("Mezcla Directa:", merge\_sort(lista))

**Pantalla de computadora con letras

Descripción generada automáticamente con confianza mediaCompilación:**

Divide recursivamente la lista en mitades y luego las ordena con intercalación

**¿Por qué se usa?:** Es muy eficiente para listas grandes

**¿Puede mejorar?:** Para datos más pequeños Insertsort es más rápido

**Mezcla Equilibrada:**

Codigo de Mezcla Equilibrada

def mezclaequilibrada(lista):

mid = len(lista) // 2

archivo1 = sorted(lista[:mid])

archivo2 = sorted(lista[mid:])

return intercalacion(archivo1, archivo2)

datos = [20, 0, 5, 200, 125, 25]

print("Mezcla Equilibrada:", mezclaequilibrada(datos))



**Compilación:**

Es similar a la mezcla directa solo que esta pensada para listas grandes en el que se divide el archivo en partes ordenadas se guardan en archivos auxiliares y luego se intercalan para ordenar

**¿Por qué se usa?:** Es bueno para memoria limitada como el procesamiento de grandes datos

**¿Se puede mejorar?:** En memoria es mejor utilizar la mezcla directa, pero en discos es uno de los métodos mas eficientes

**Conclusiones:**

Los 3 métodos se complementan entre si ya que la intercalación necesita tener 2 listas ya ordenadas mientras que tanto la mezcla directa como equilibrada necesitan de la intercalación para funcionar correctamente

La utilización de alguno de estos dependería de lo que se necesita y se tenga ya que la intercalación es para datos ya ordenados la mezcla directa para datos medianos y grandes mientras que la mezcla equilibrada se utilizaría en caso de tener memoria limitada y que se trabajo con archivos o discos