1. **Algoritmos de ordenamiento:**  
    Se implementan dos métodos:
   1. **Burbuja:** compara elementos adyacentes e intercambia sus posiciones hasta ordenar la lista. Tiene una complejidad *O(n²)*, por lo que su rendimiento disminuye con listas grandes.
   2. **Merge Sort:** aplica la técnica *divide y vencerás*, dividiendo la lista en mitades y combinándolas ordenadamente. Su complejidad es *O(n log n)*, siendo más eficiente para grandes volúmenes de datos.
2. **Medición del tiempo:**  
    La función medir\_tiempo() calcula el tiempo promedio que tarda cada algoritmo en ordenar una lista. Cada prueba se repite tres veces usando copias de los mismos datos, midiendo el tiempo con time.perf\_counter() para obtener mayor precisión.
3. **Ejecución experimental:**  
    En la función main() se generan listas aleatorias de distintos tamaños (100, 1000, 5000 y 10000 elementos). Luego, se registra y compara el tiempo de ejecución de ambos algoritmos, mostrando los resultados en consola.
4. **Visualización de resultados:**  
    Los tiempos obtenidos se grafican con **Matplotlib**, mostrando la relación entre el tamaño de la lista y el tiempo de ejecución. La gráfica evidencia que el algoritmo de burbuja crece mucho más rápido en tiempo, confirmando su menor eficiencia frente a Merge Sort.