

# Trabajo Práctico Integrador - Programación 1

**TEMA:** Algoritmos de Búsqueda y Ordenamiento

**CARRERA:** Tecnicatura Universitaria en Programación – UTN

**INTEGRANTES:** Lucas ALVAREZ – Jorge ATENCIO

**FECHA:** 09/06/2025

## INTRODUCCIÓN:

En informática, los **algoritmos** son fundamentales para cada operación. Se puede definir al algoritmo como una secuencia finita o conjunto de instrucciones bien definidas y no ambiguas que están ordenadas en pasos a seguir para alcanzar un objetivo que no es otro que resolver un problema.

Dentro de la programación, dos de las operaciones más fundamentales y recurrentes son la **búsqueda** de información y el **ordenamiento** de datos. Entender y aplicar eficientemente los algoritmos dedicados a estas tareas es crucial para desarrollar un software robusto y optimizado. Este trabajo explorará los principios y las implementaciones de algunos de los algoritmos de búsqueda y ordenamiento más comunes, destacando su funcionamiento, ventajas y desventajas.

## ALGORITMOS DE BÚSQUEDA:

Los algoritmos de búsqueda son métodos utilizados en programación que tienen como objetivo principal encontrar uno o varios elementos específicos dentro de una colección de datos que generalmente se encuentran guardados en estructuras como las listas. Son fundamentales para la eficacia de la búsqueda en bases de datos, motores de búsqueda y otras aplicaciones informáticas.

Se pueden encontrar varios tipos de algoritmo de búsqueda, entre ellos:

- **Búsqueda secuencial o lineal:**

Este es el método de búsqueda más sencillo. Recorre la colección de datos elemento por elemento, desde el principio

hasta el final de la lista y va comparando cada uno de los elementos con el valor que se está buscando. Si encuentra una coincidencia, retorna la posición del elemento. Si llega al final de la colección sin encontrarlo, indica que el elemento no está presente.

Tiene como ventaja que es fácil de entender e implementar y no requiere que la colección de datos esté ordenada. Por otro lado, la desventaja radica

que no es recomendable para grandes bases de datos, toda vez que sino existe el elemento buscado para devolver el resultado deberá recorrer todos los elementos que componen la base.

- **Búsqueda binaria:** a diferencia de la búsqueda lineal, esta es mucho más eficiente y rápida en bases de datos grandes, siempre y cuando la base de datos esté ordenada.

Funciona dividiendo repetidamente la lista por la mitad hasta encontrar el elemento buscado.

Compara el valor buscado con el elemento central de la lista. Si son iguales, lo encuentra. Si el valor buscado es menor, la búsqueda continúa en la mitad inferior de la lista, si el valor es mayor, lo hace en la mitad superior de la misma. Este proceso se repite hasta encontrar el elemento o hasta que la porción de la lista a revisar se termina.

La ventaja de este método es que es mucho más rápido que la búsqueda lineal, pero para ello es requisito que la base de datos esté previamente ordenada caso contrario hay que ordenarla primero.

## ALGORITMOS DE ORDENAMIENTO:

Los algoritmos de ordenamiento se encargan de reorganizar los elementos de una lista en una secuencia específica. Estos algoritmos reubican los elementos de una lista en un orden específico, ya sea ascendente o descendente. Son fundamentales en informática para organizar datos y así optimizar las búsquedas y facilitar los procesos.

Como antes se vio en los tipos de búsqueda son indispensables en la búsqueda de tipo binario.

**Ordenamiento por burbuja:** es uno de los algoritmos de ordenamiento más simples. Funciona revisando repetidamente la lista y comparando pares de elementos adyacentes, intercambiándolos si están en el orden incorrecto. El proceso se repite hasta que no se necesitan más intercambios por estar ya ordenada.

Es fácil de comprender e implementar pero no es eficiente en grandes listas o que se encuentren desordenadas debido a la gran cantidad de comparaciones e intercambios que realiza.

**Ordenamiento por selección:** divide la lista en dos partes, una sub lista ordenada y una sub lista desordenada. Actúa a través de la sub lista desordenada, encontrando el elemento mínimo o máximo, según el orden deseado, y lo coloca al final de la sub lista ordenada o bien al principio de la desordenada que va reduciendo en el proceso. Esto se repite hasta que toda la lista se encuentre ordenada.

Es relativamente simple de implementar y no requiere memoria adicional significativa. Es estable en términos de movimientos de datos, cada elemento se mueve a su posición final solo una vez.

Ahora bien, su rendimiento es pobre en grandes conjuntos de datos ya que debe realizar muchas comparaciones lo que perjudica en el tiempo que emplea.

**Ordenamiento por inserción:** este algoritmo construye la lista final ordenada un elemento por vez. Actúa sobre la lista, tomando cada elemento e insertándolo en su posición correcta dentro de la porción ya ordenada de la lista.

Es eficiente para conjunto de datos pequeños o para listas que ya están casi ordenadas, se implementa fácilmente.

El rendimiento de este método no es bueno en listas grandes.

Caso Práctico:

Se armó un programa en Python con búsqueda lineal.

Conclusiones:

Este trabajo integrador ha sido una inmersión fundamental en el mundo de los algoritmos de búsqueda y ordenamiento. Hemos comprendido que, aunque conceptualmente sencillos, son la base de innumerables aplicaciones informáticas.