

TALLER PRÁCTICO – BÚSQUEDA LINEAL

Nombre: Brahian Alzate Toro

Código: 2460976-2724

Materia: ADA1

Institución: Universidad del Valle

Fecha: Octubre de 2025

Introducción

En este taller de Algoritmos y Estructuras de Datos (ADA1) puse en práctica el algoritmo de **búsqueda lineal**, usando como ejemplo una tienda de electrónica llamada **TechStore**. El objetivo fue entender cómo funciona este tipo de búsqueda y aplicarlo para encontrar información dentro de listas de productos y empleados. Durante el proceso implementé funciones, recorrí listas, usé diccionarios y construí un pequeño sistema con menú interactivo.

¿Qué es la búsqueda lineal?

La **búsqueda lineal o secuencial** es uno de los algoritmos más básicos para encontrar un elemento dentro de una lista. Revisa cada elemento uno por uno hasta que encuentra el que buscamos o llega al final. No necesita que los datos estén ordenados, aunque puede ser más lenta si la lista es muy grande.

Desarrollo de los ejercicios

• Ejercicio 1: Búsqueda simple

Función que busca un número dentro de una lista y devuelve su posición. Sirvió como base para entender el algoritmo.

• Ejercicio 2: Buscar productos

Trabajé con una lista de diccionarios que representaban productos, buscando por nombre, ID o categoría.

• Ejercicio 3: Buscar empleados

Usé el mismo concepto para una lista de empleados, con búsquedas por nombre completo, departamento o estado.

• Ejercicio 4: Búsquedas condicionales

Agregué funciones para buscar productos disponibles, dentro de un rango de precios y contar productos por categoría.

• Ejercicio 5: Sistema completo (TechStore)

Integré todos los ejercicios en un **sistema con menú interactivo**, permitiendo al usuario elegir entre buscar productos o empleados desde la terminal.

Complejidad del algoritmo

Todas las funciones creadas tienen complejidad **O(n)**, lo que significa que el tiempo de búsqueda depende del tamaño de la lista. Mientras más datos haya, más tiempo tarda. Aunque no es el algoritmo más eficiente, funciona muy bien para listas pequeñas o medianas.

Capturas del proceso

1■■ Identificación de entregables y organización del taller

pero yo te dare los demas ejercicios implementados hasta ahora y tu me ayudaras con el menu


Perfecto 🙌🔥

Esa es la mejor forma de hacerlo — tú me compartes **los ejercicios que ya implementaste (1 al 4)**, y yo te ayudo a:

- Integrarlos correctamente dentro del **menú del Ejercicio 5**,
- Organizar las funciones de búsqueda según tu propio código,
- Corregir cualquier error o duplicado,
- Y dejar el sistema totalmente funcional y ordenado.



2■■ Implementación del sistema y estructura del menú

 TALLER_BUSQUEDA_LINEAL.md
Archivo

Necesito que me ayudes a identificar que es lo que debo entregar

Perfecto ✅

Ya revisé el archivo “TALLER_BUSQUEDA_LINEAL.md” y te resumo **exactamente qué debes entregar** según las instrucciones del taller:

ENTREGABLES OBLIGATORIOS

Debes entregar **cinco archivos** bien organizados:

1. `sistema_tienda.py`



→ Contiene el **programa principal** con el menú interactivo.

3■■ Elaboración del README.md basado en el código funcional

en base a el código que hice, ayúdame con el archivo README.md

Perfecto 🍌

Voy a crear tu archivo `README.md` personalizado basándome en el código real que tú implementaste en `funciones_busqueda.py` y el sistema integrado que te pasé.

Este documento explicará claramente qué hiciste en cada ejercicio, cómo ejecutar el sistema, y qué incluye el proyecto.

README.md — Proyecto: Búsqueda Lineal (TechStore)

Proms utilizados

- ChatGPT (asistencia en integración de ejercicios y creación de README.md)
- Visual Studio Code (entorno de desarrollo)
- Python 3.12 (lenguaje de programación)
- Markdown (.md) para documentación
- GitHub (para respaldo y revisión de código)

Conclusiones

La búsqueda lineal es sencilla pero muy útil para comprender la lógica de los algoritmos. Aunque no es la opción más rápida para grandes volúmenes de datos, su aplicación práctica es clara. El proyecto **TechStore** permitió integrar teoría y práctica, fortaleciendo la lógica y el manejo de estructuras en Python.