INFORME TÉCNICO: SISTEMAS DE BÚSQUEDA PARA BIBLIOTECA DIGITAL
Estudiante: Jose Augusto Salgado Murillo
Repositorio GitHub: https://github.com/EtherealWeb5167/SistemaBusquedaBiblioteca

---

1. Introducción y Definición

En el contexto de las ciencias de la computación, un algoritmo de búsqueda se define como un conjunto de instrucciones diseñadas para localizar un elemento específico dentro de una estructura de datos. Estos algoritmos son fundamentales para la recuperación de información, permitiendo transformar grandes volúmenes de datos crudos en información accesible para el usuario (Cormen et al., 2022).

2. Tipos de Búsqueda y Diferencias

Búsqueda Lineal (Secuencial)

Es el método más básico. Consiste en recorrer la estructura de datos elemento por elemento, desde el inicio hasta el final, verificando si el valor actual coincide con el buscado. No requiere que los datos estén ordenados.

Búsqueda Binaria

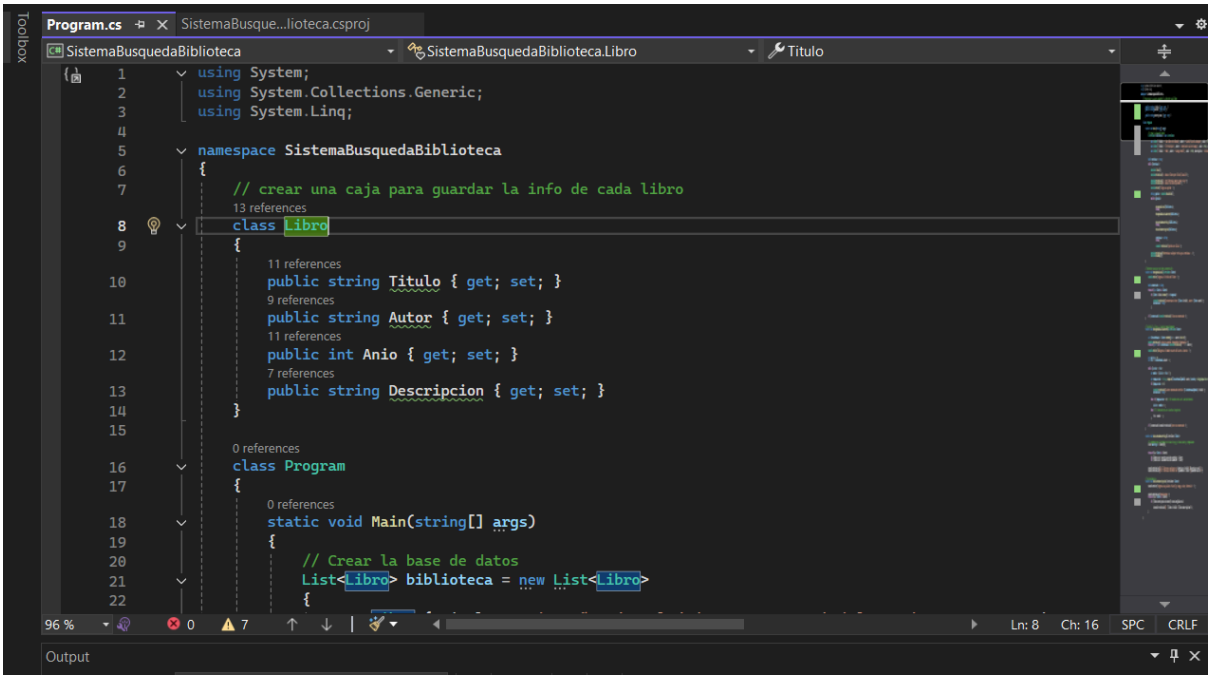
Es un algoritmo eficiente basado en la técnica "divide y vencerás". Requiere obligatoriamente que la lista esté ordenada. El algoritmo compara el valor buscado con el elemento central; si es menor, descarta la mitad superior; si es mayor, descarta la mitad inferior, repitiendo el proceso hasta encontrar el dato.

	Ventajas	Desventajas
Búsqueda Lineal	- Fácil de implementar. - No requiere ordenamiento previo.	- Muy lenta en grandes volúmenes de datos. - Ineficiente.
Búsqueda Binaria	- Extremadamente rápida. - Ideal para bases de datos grandes.	- Requiere datos ordenados. - Más compleja de programar.

---

4. Evidencias del Desarrollo

4.1 Código Fuente

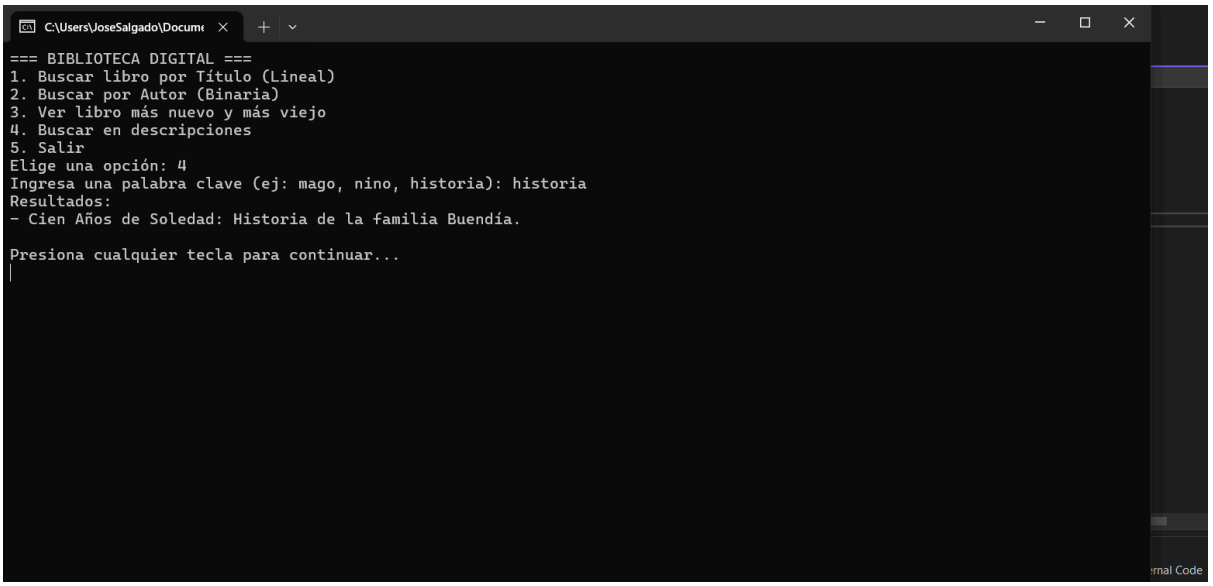


```
using System;
using System.Collections.Generic;
using System.Linq;

namespace SistemaBusquedaBiblioteca
{
    // crear una caja para guardar la info de cada libro
    13 references
    class Libro
    {
        11 references
        public string Titulo { get; set; }
        9 references
        public string Autor { get; set; }
        11 references
        public int Anio { get; set; }
        7 references
        public string Descripcion { get; set; }
    }

    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            // Crear la base de datos
            List<Libro> biblioteca = new List<Libro>
            {
```

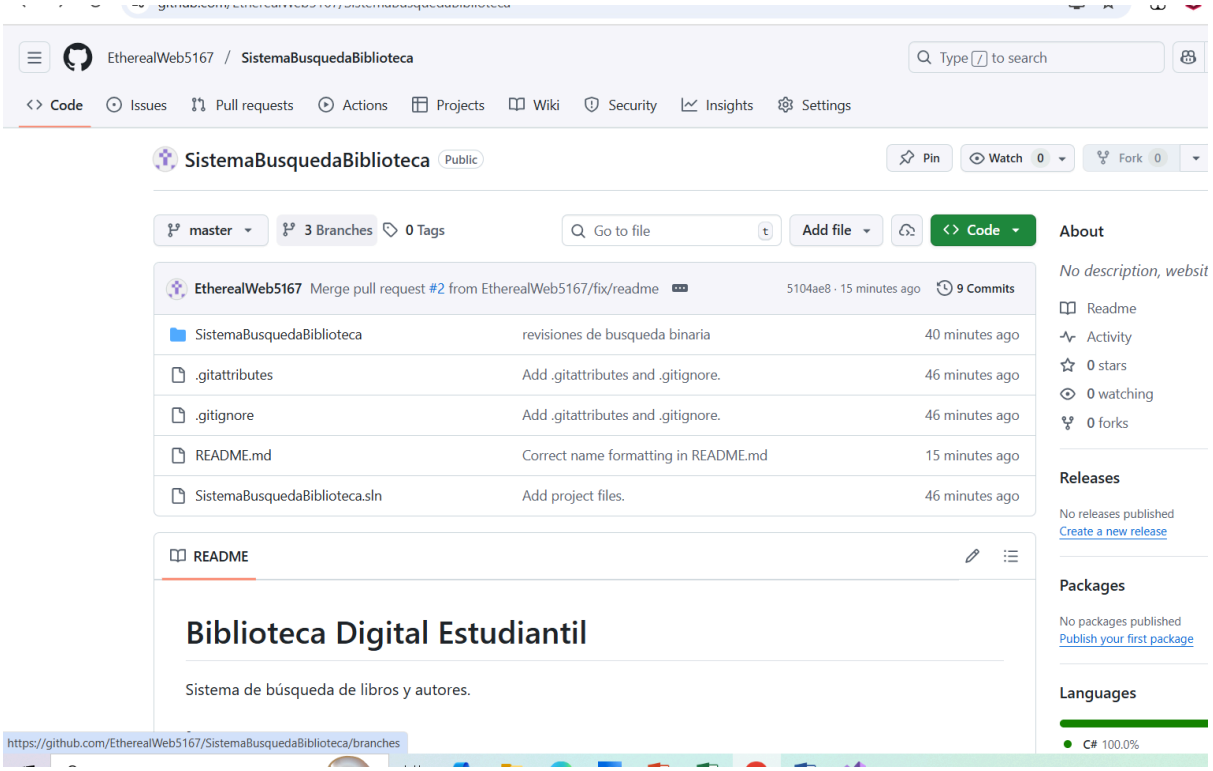
4.2 Ejecución del Programa



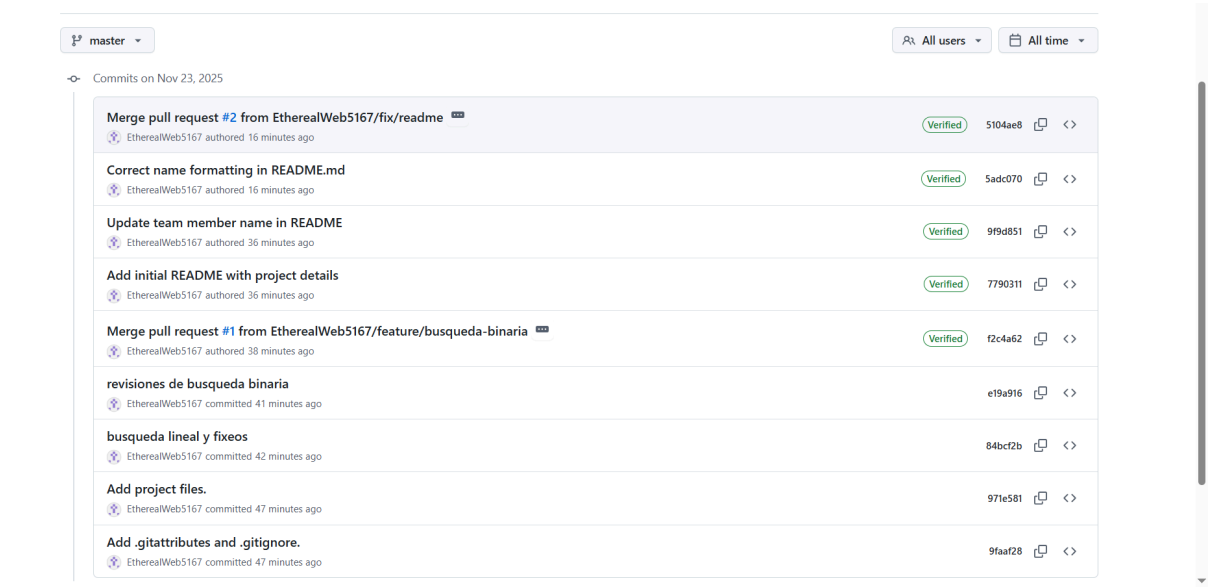
```
=== BIBLIOTECA DIGITAL ===
1. Buscar libro por Título (Lineal)
2. Buscar por Autor (Binaria)
3. Ver libro más nuevo y más viejo
4. Buscar en descripciones
5. Salir
Elige una opción: 4
Ingresa una palabra clave (ej: mago, nino, historia): historia
Resultados:
- Cien Años de Soledad: Historia de la familia Buendía.

Presiona cualquier tecla para continuar...
```

4.3 Repositorio GitHub



4.4 Colaboración (Pull Requests)



5. Reflexión Individual

La capacidad de buscar información es la columna vertebral de la computación moderna. Sin algoritmos de búsqueda eficientes, la interacción con la tecnología sería imposiblemente lenta. Imaginemos buscar un contacto en un teléfono revisando uno por uno 5,000 nombres; sería inútil. En el desarrollo de software, elegir el algoritmo incorrecto puede colapsar un servidor. Por tanto, optimizar la búsqueda ahorra tiempo, reduce el consumo de energía de los procesadores y hace viables los sistemas complejos como la Inteligencia Artificial.

6. Referencias

- Bhargava, A. Y. (2016). Grokking Algorithms: An illustrated guide for programmers. Manning Publications.
- Cormen, T. H., et al. (2022). Introduction to algorithms (4.ª ed.). MIT Press.
- Microsoft. (2023). Documentación de C#: Colecciones y algoritmos. Microsoft Learn.