

Universidad Rafael Landívar

Facultad de Ingeniería

Estructuras de datos II

Sección: 03

Catedrático: Boris José Búcaro Pazzetti

Lab1:

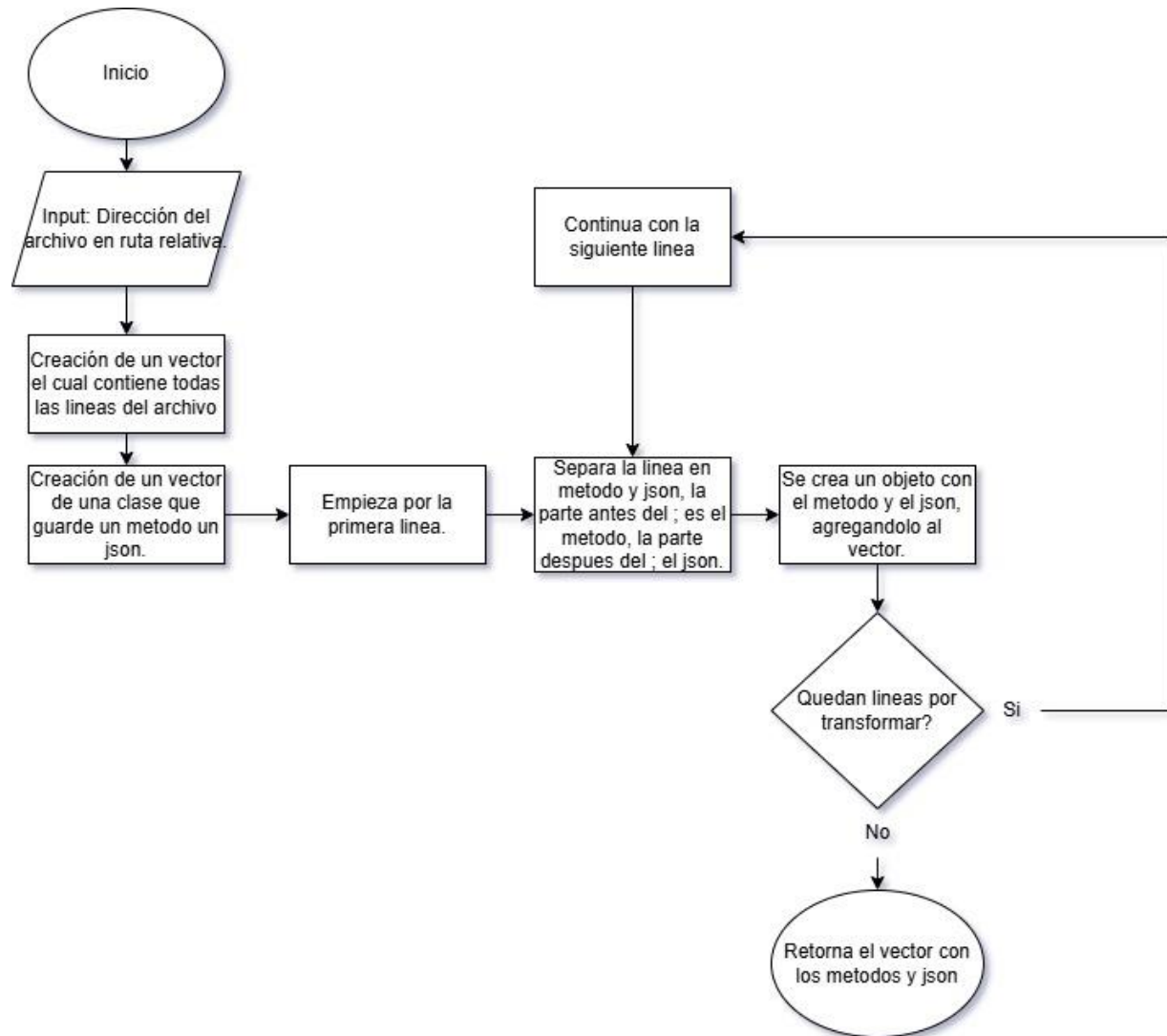
Optimización de Sistema Inventarios de una Librería

Dereck Alexander Cabrera Ng – 1177223

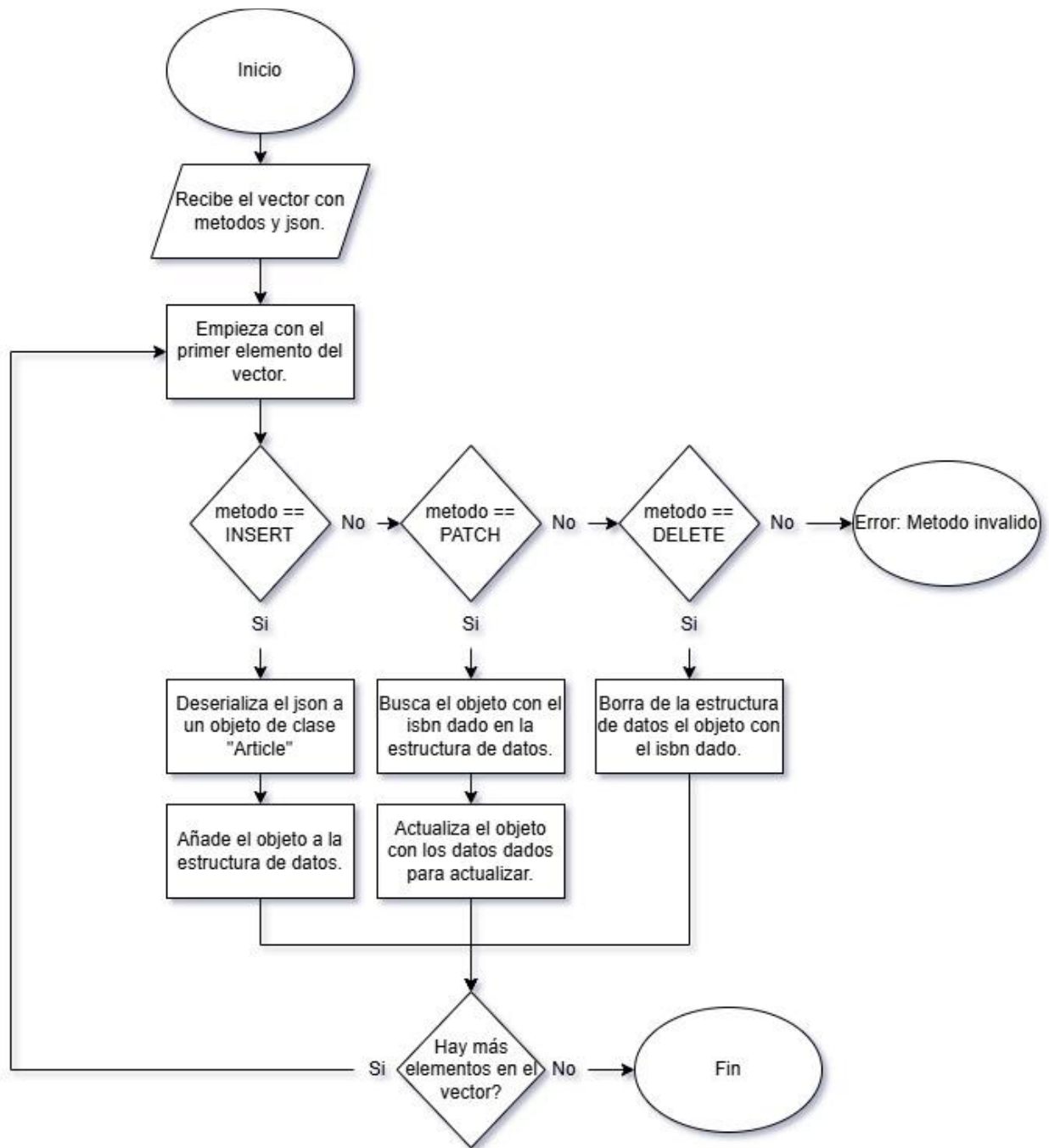
05 de agosto de 2024

Diagramas de flujo del programa

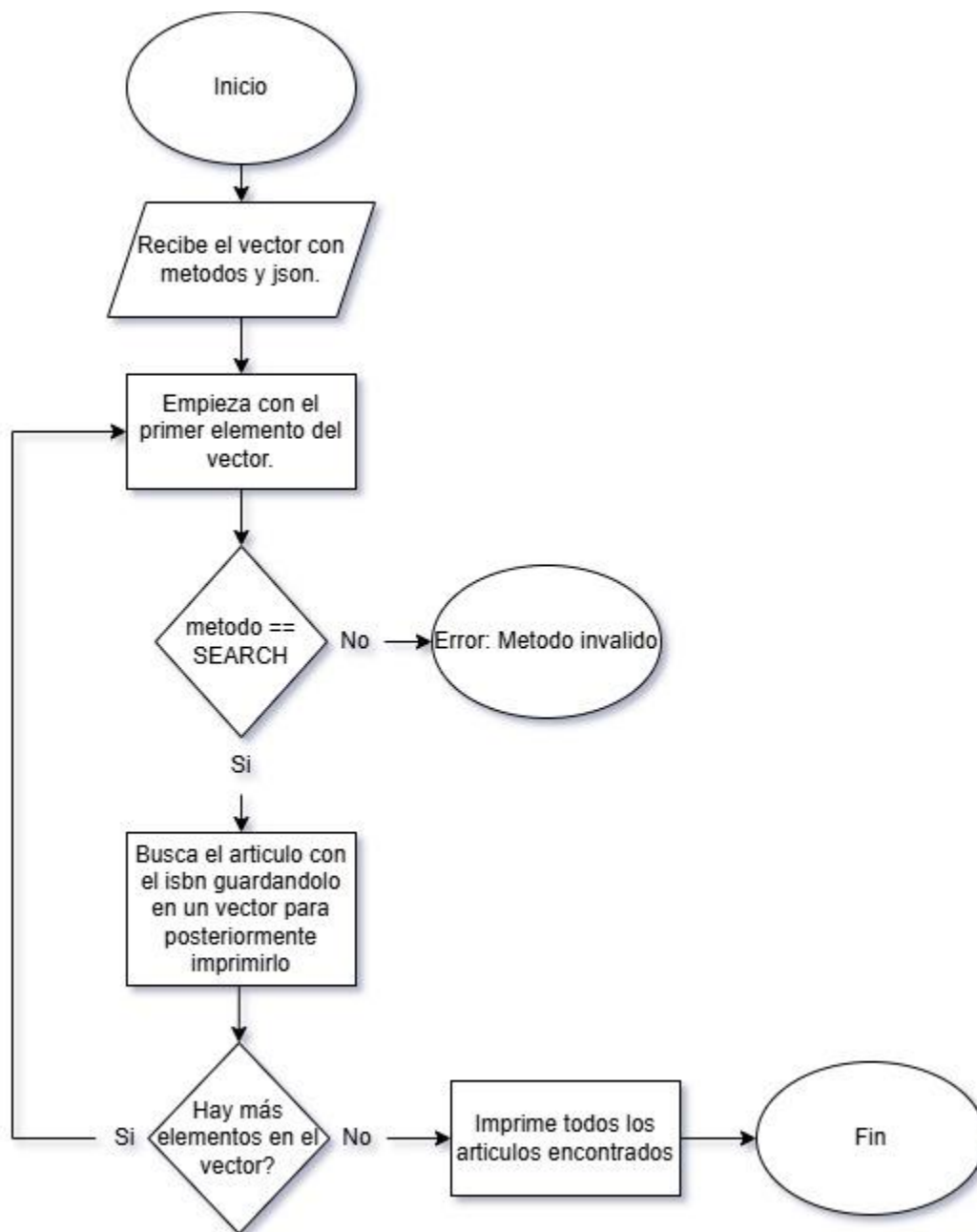
Función lectura de archivos:



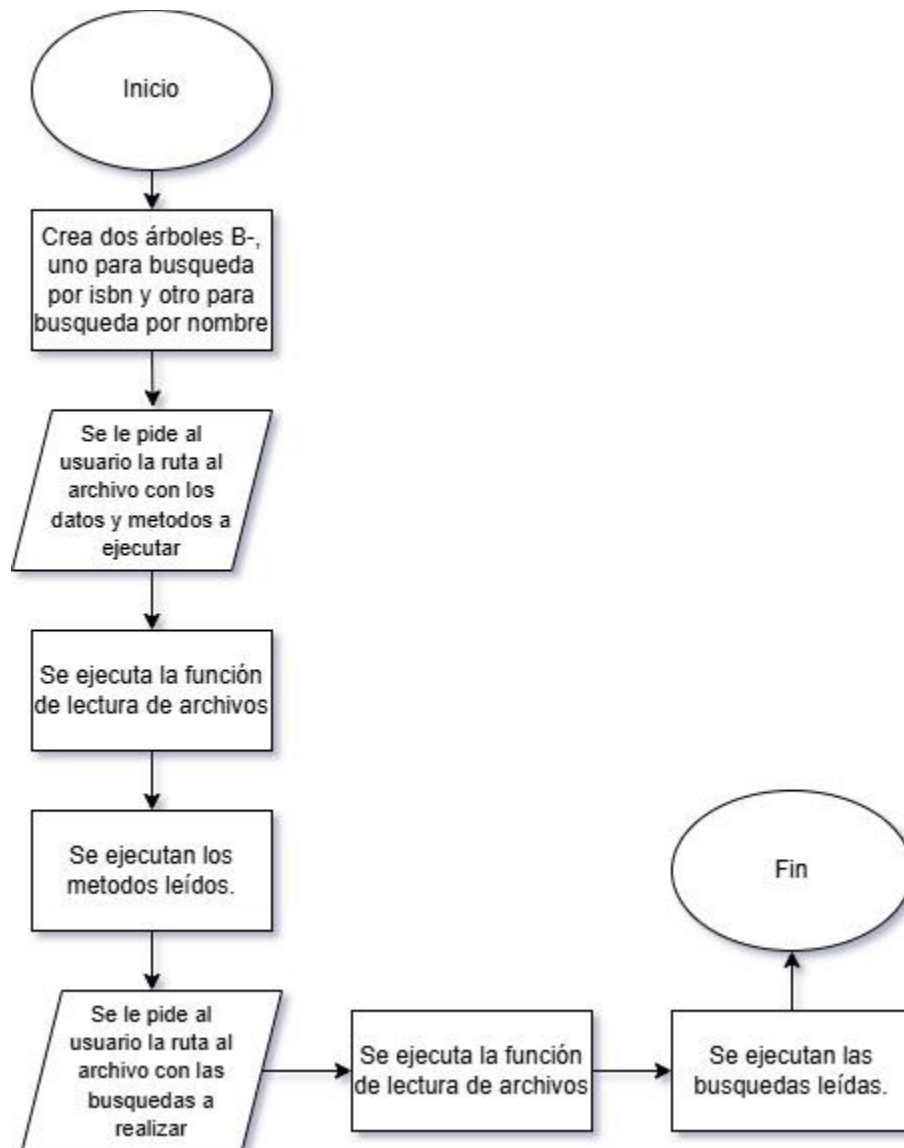
Insertar, Actualizar y Eliminar:



Buscar:



General:



Estructura de datos utilizada

La estructura de datos a utilizar es la de un Árbol B-, se escogió esta estructura debido a su complejidad de $O(\log n)$ para inserción, búsqueda y eliminación. Si la comparamos con Árboles Binarios de Búsqueda, este ultimo tiene el problema de que puede quedar una lista enlazada, haciendo que la complejidad sea $O(n)$, se puede balancear utilizando distintos algoritmos, pero el tiempo que necesita es menor al de un Árbol B-.

También estaba la opción de utilizar un mapa hash, pero no sería una buena idea. Si bien el mapa hash tiene la ventaja de tener una complejidad constante, habría que resolver las colisiones lo que aumentaría la cantidad de memoria utilizada o aumentaría la complejidad. Si la cantidad de datos es pequeña es una buena opción, pero este no es el caso ya que se estará trabajando con una cantidad de datos suficiente para hacer a los mapas hash ineficientes.

Recomendaciones

Para que las búsquedas, inserciones y actualizaciones sean más eficientes, se puede utilizar alguna base de datos relacional o no relacional. Estos sistemas han sido trabajados por profesionales de la industria por un largo periodo de tiempo, haciendo que el crear un sistema más eficiente requiera de muchos recursos y tiempo.

También está la posibilidad de crear una interfaz gráfica sencilla de utilizar para realizar búsquedas, inserciones y actualizaciones de manera sencilla, sin necesidad de modificar directamente un archivo.