

MÉTODO DE LA INGENIERÍA

1. Identificación del problema:

Contexto del problema:

La cultura del libro ha renacido en la ciudad de Cali, es por eso que una Librería muy importante e innovadora se dispone a abrir sus puertas en la sultana del valle. Esta nueva compañía dedicada a la venta de libros de múltiples géneros y diversas lenguas se caracteriza por su estilo muy particular, innovador y eficiente a la hora de atender a sus usuarios.

Definición del problema:

La importante librería desea ofrecer a sus clientes una mejor experiencia dentro de esta, por lo que decide tener un software dividido en secciones, que le permite al usuario:

- a) Tener información detallada de los libros que se encuentran disponibles, esto como, capítulos iniciales, reseñas y críticas de los libros consultados.
- b) Seleccionar y saber el código ISBN de los libros de su interés, para luego añadirlos en el canasto.
- c) Saber cuál será la ruta más “rápida” para recoger los libros deseados.
- d) Hacer compra de libros.

Identificación de necesidades y síntomas:

- La solución a los problemas planteados anteriormente debe ser lo más eficiente posible.

2. Recopilación de información:

¿Qué es una librería?

La librería es un establecimiento comercial destinado a la venta de libros y otros artículos similares.

Aplicaciones para la compra y venta de libros

Parte de estas aplicaciones son de uso online, es decir, no se encuentran directamente en la librería, pero sirven en este laboratorio como referente para la interfaz o del manejo de algunas estructuras.

Vend: esta aplicación tiene muchas cosas buenas a su favor. Es un punto de venta muy asequible y capaz para librerías de cualquier tamaño. Mantiene sincronizados sus inventarios y transacciones en línea / fuera de línea.

ShopKeep: Esta aplicación solo admite hasta 10,000 artículos, por lo que no es la mejor opción para librerías muy grandes. Sin embargo, para las pequeñas librerías que son bastante selectivas con respecto a los títulos que poseen, ShopKeep es una excelente opción por su granular gestión de inventario y ofrecer excelente soporte al cliente.

Bindo: Con el soporte integrado de órdenes de compra de Bindo, se puede facilitar todo el proceso de reposición de existencias, desde pedir los libros a los proveedores hasta llevarlos al piso de ventas. Muestra actualizaciones en tiempo real de las ventas de libros y las existencias, y crea informes personalizados para mostrar el panorama general de la actividad de la librería.

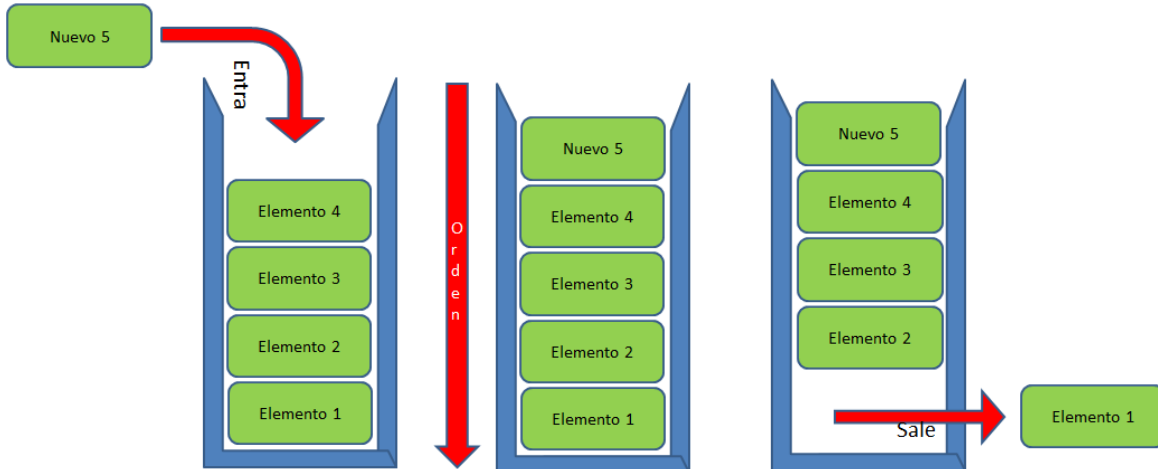
3. **Búsqueda de soluciones creativas:**

Para la solución de los problemas que se han planteado anteriormente se tienen las siguientes alternativas:

Alternativa 1: Colas o Queue

Una cola es una estructura de datos donde el primer elemento en entrar es el primero en salir, también denominadas estructuras FIFO (First In, First Out). Esta estructura de datos se puede definir como una lista enlazada con acceso FIFO a la que sólo se tiene acceso al final de la lista para meter elementos y al principio de esta para sacarlos.

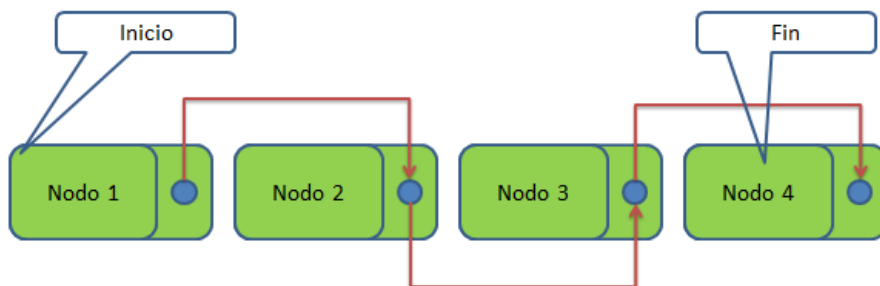
Este tipo de estructura de datos abstracta se implementa en lenguajes orientados a objetos mediante clases, en forma de listas enlazadas.



Alternativa 2: Listas enlazadas o LinkedList

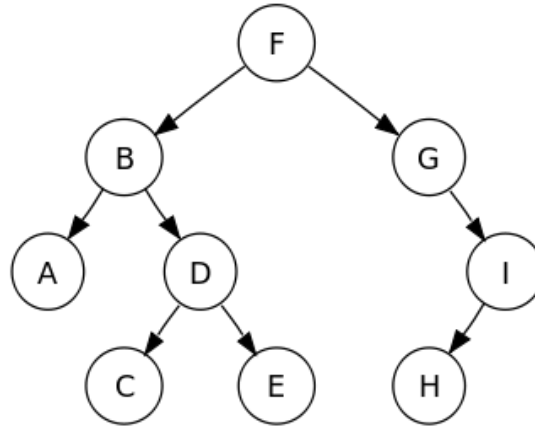
Se define lista enlazada como una colección de elementos que están enlazados entre sí y que cada nodo contiene un valor.

Es la variante más simple que existe pues en esta estructura de datos tenemos un conjunto de nodos que están enlazados solo con el nodo siguiente de tal forma que si queremos recorrer la colección lo haremos del primero hasta el último, pero no se puede regresar.



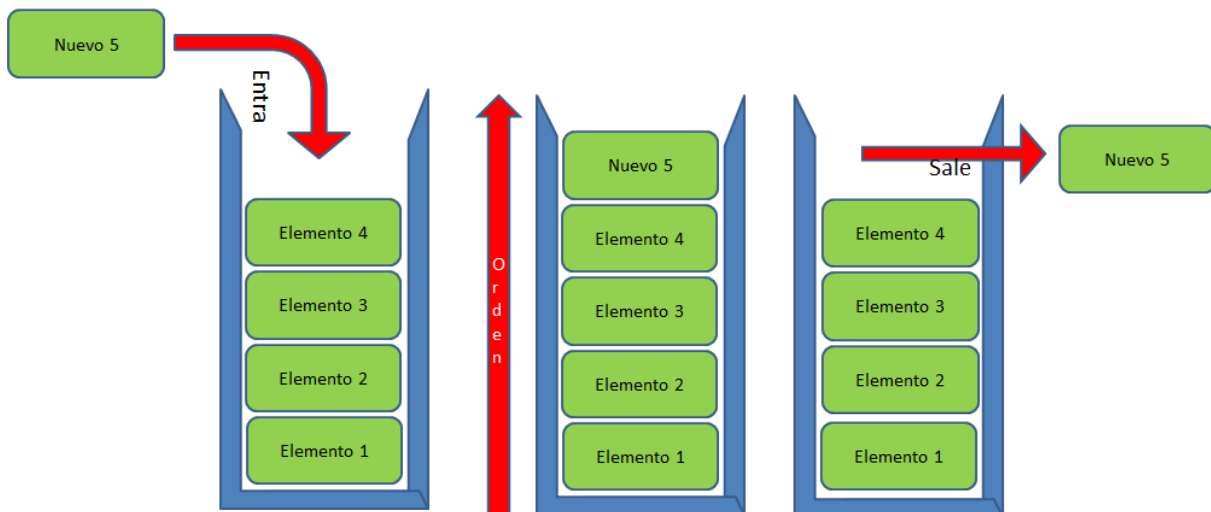
Alternativa 3: Árboles binarios o Binary tree

Un árbol binario es una estructura de datos en la cual cada nodo puede tener un hijo izquierdo y un hijo derecho. No pueden tener más de dos hijos (de ahí el nombre "binario"). Si algún hijo tiene como referencia a null, es decir que no almacena ningún dato, entonces este es llamado un nodo externo. En el caso contrario el hijo es llamado un nodo interno.



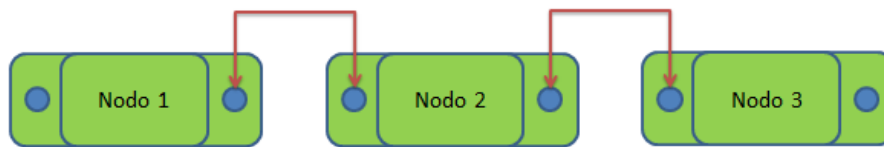
Alternativa 4: Pilas o Stacks

Una pila es una lista ordenada o estructura de datos que permite almacenar y recuperar datos, el modo de acceso a sus elementos es de tipo LIFO (del inglés Last In, First Out, «último en entrar, primero en salir»). Para el manejo de los datos cuenta con dos operaciones básicas: apilar (push), que coloca un objeto en la pila, y su operación inversa, retirar (o desapilar, pop), que retira el último elemento apilado.



Alternativa 5: Listas doblemente enlazadas o Double LinkedList

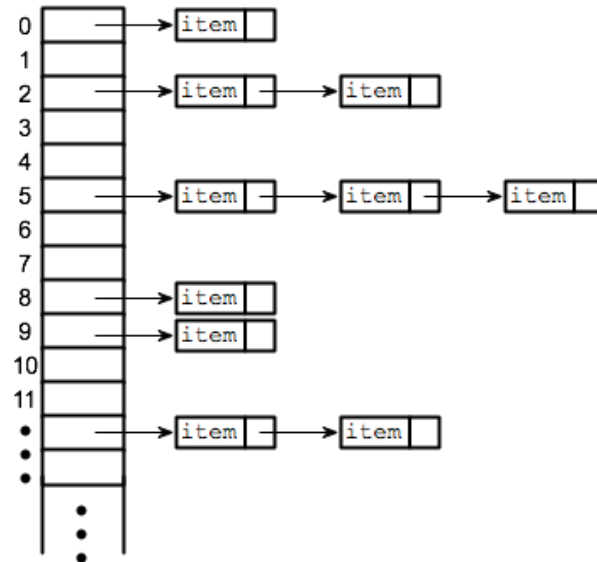
Una lista doblemente enlazada es una lista lineal en la que cada nodo tiene dos enlaces, uno al nodo siguiente, y otro al anterior. Las listas doblemente enlazadas no necesitan un nodo especial para acceder a ellas, pueden recorrerse en ambos sentidos a partir de cualquier nodo, esto es porque a partir de cualquier nodo, siempre es posible alcanzar cualquier nodo de la lista, hasta que se llega a uno de los extremos.



Alternativa 6: Hash tables

Una tabla hash es una estructura de datos que asocia llaves o claves con valores. La operación principal que soporta de manera eficiente es la búsqueda: permite el acceso a los elementos (teléfono y dirección, por ejemplo) almacenados a partir de una clave generada (usando el nombre o número de cuenta, por ejemplo). Funciona transformando la clave con una función hash en un hash, un número que identifica la posición (casilla o cubeta) donde la tabla hash localiza el valor deseado. Las tablas hash se suelen implementar sobre vectores de una dimensión, aunque se pueden hacer implementaciones multi-dimensionales basadas en varias claves. Como en el caso de los arrays, las tablas hash proveen tiempo constante de búsqueda promedio $O(1)$, sin importar el número de elementos en la tabla. Sin embargo, en casos particularmente malos el tiempo de búsqueda puede llegar a $O(n)$, es decir, en función del número de elementos.

Comparada con otras estructuras de arrays asociadas, las tablas hash son más útiles cuando se almacenan grandes cantidades de información. Las tablas hash almacenan la información en posiciones pseudo-aleatorias, así que el acceso ordenado a su contenido es bastante lento.



4. Transición de las ideas a diseños preliminares:

Luego de hacer una profunda investigación de las posibles estructuras de datos que podrían darles solución a los problemas ya propuestos, es momento de analizar cuáles de estas resultarían más eficientes. Al hacer diferentes verificaciones se logró evidenciar que:

Alternativa 2: Listas enlazadas

- Acceso a posiciones intermedias con coste dependiente del tamaño de la lista.
- Necesidad de memoria adicional para almacenar los objetos Node con sus atributos.

Alternativa 5: Listas doblemente enlazadas

- Para la solución de uno de los problemas planteados, es necesario que el acceso y eliminación de elementos agregados en una estructura sea lo suficientemente rápido. e
- Entre una de sus desventajas podemos mencionar que al tener que mantener dos referencias el código se vuelve más complejo.
- Ocupa más espacio en memoria.

5. Evaluación y selección de la mejor solución:

Después de analizar los aspectos necesarios para la solución de los problemas encontrados, se plantean los siguientes criterios:

Criterio A. Se plantea un algoritmo que acceda a los elementos de manera eficiente.

- [4] Constante
- [3] Mayor a constante
- [2] Logarítmica
- [1] Lineal

Criterio B. Los problemas requieren que el programa inserte elementos rápidamente.

- [4] Constante
- [3] Mayor a constante
- [2] Logarítmica
- [1] Lineal

Criterio C. Se necesita de un algoritmo que trabaje eficientemente al momento de eliminar un elemento contenido en una estructura.

- [4] Constante
- [3] Mayor a constante
- [2] Logarítmica
- [1] Lineal

Criterio D. Las estructuras se implementan de manera sencilla.

- [2] Fácil
- [1] La Implementación no es sencilla

	Criterio A	Criterio B	Criterio C	Criterio D	Total
Cola	4	4	4	2	14
Árboles binarios	3	2	2	2	9
Pilas	4	4	4	2	14
Listas doblemente enlazadas	1	1	1	2	5
Hash tables	3	3	3	1	10

Basado en la anterior tabla, se puede inferir que las estructuras de datos más óptimas para la implementación son:

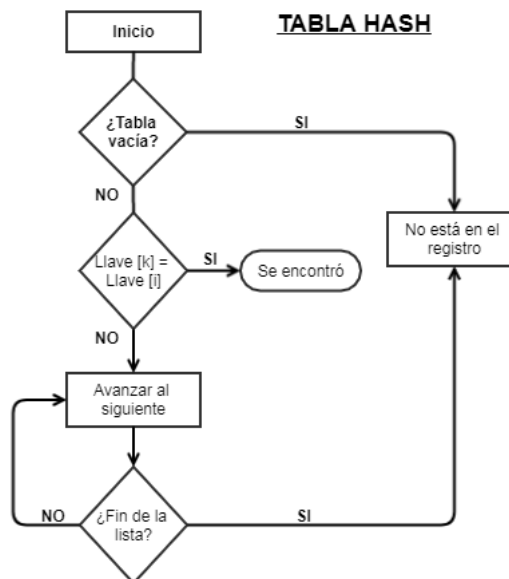
- Tablas hash
- Pilas
- Colas

Los criterios que se plantearon fueron de acuerdo a la solución del problema que se tenía. En primer lugar, las tablas hash ayudaron a la inserción y búsqueda casi inmediata por medio de llaves que venían representadas como los códigos ISBN de cada libro. En segundo lugar, la pila simula la compra de libros, ya que el cliente los va tomando y los agrega en la canasta, y al momento de pagar comienza sacando el último que tomó hasta llegar al primero, haciendo honor a la filosofía LIFO (últimas en entrar, primeras en salir). Por último, la cola permite simular de manera real la fila que se debe hacer para poder comprar los libros deseados, dado que, el primero que está en la fila, es el primero que pasa donde el cajero, mostrando la filosofía FIFO (primeras en entrar, primeras en salir).

6. Documentación:

Diagramas de flujo:

Agregar elemento a la tabla



Agregar, eliminar, visualizar elementos en la pila.



Toda la demás documentación se encuentra en esta carpeta.