

ISWZ1102 – PROGRAMACIÓN 1

PROYECTO PROGRAMACION 01

Autores (Justin Basantes, Luis Vasquez, Ainhoa Salas)

INTRODUCCIÓN:

Existe un negocio de una librería pequeña llamada “Librería Andina” ubicado en , donde tiene una gran cantidad de libros para venderlos, pero estos no tienen una buena organización de almacenado y de inventario, por lo que pierden mucho tiempo cuando buscan un ejemplar en específico o pierden ventas por no saber con exactitud si tienen el libro que buscaban sus clientes. Sin embargo, es muy conocida en su localidad llegando a tener numerosos clientes en su día a día.

Por lo que se ha buscado una solución para resolver su registro de datos manual con inventarios largos y que requieren de mucho tiempo esto hace que la información sobre la cantidad de libros y precios sea fácil de manipular y perder lo que ha llevado a problemas en el manejo del negocio, como pérdidas económicas y clientes insatisfechos por su desorganización.

Esta solución sería la realización de un programa intuitivo para cualquier personal de la librería, por parte de nuestro equipo, que se basaría en el ordenado, almacenado, edición, control y observación de la cantidad, ejemplares y precios que controlan en su librería, para un mejor entendimiento de su mercancía.

OBJETIVOS DEL TRABAJO:

- Realizar un proyecto de código abierto para desarrollar la capacidad de razonamiento lógico para crear algoritmos orientados a la resolución de problemas.
- Analizar a fondo el proceso de diseño y optimización de algoritmos complejos para aplicar el conocimiento que se ha obtenido anteriormente.
- Aplicar las habilidades y conocimientos de programación adquiridos durante el semestre, como el uso de funciones, archivos, arreglos y punteros, al diseñar código de lenguaje C eficiente.

TABLAS:

Formulación del Problema

Nombre del sistema	Librería Andina-Programa de Inventario
Usuarios	Cajeros, jefes de inventarios y encargados de la mercancía.

Objetivos	<ul style="list-style-type: none"> • Ofrecer un sistema digital de registro de inventario para libros de fácil uso para el personal de la librería Andina. • Disminuir la pérdida de ventas y ganancia por el registro anticuado de la librería Andina. • Facilitar y agilizar el proceso de registro, editado y eliminación de inventario de libros para mejorar la eficacia en el lugar.
Contexto del problema (Considerar los contextos económico, social y ambiental, dentro del sector productivo en el que funcionará el sistema de inventarios)	<p>Contexto Ambiental: En el norte de la ciudad de Quito, Ecuador existen una estimación de 30 librerías activas aproximadamente, la cantidad de librerías que se encuentran registradas en Google maps, son un total de 20, entonces le estimamos 10 librerías más que serían las que no tienen registro aquí para nuestra guía; hacemos esto, ya que no existe un registro exacto de la cantidad de librerías que se encuentran en esa parte de la ciudad, por lo que solo podemos estimarlo.</p> <p>Contexto Social y Económico: En Ecuador cada vez existe mayor número de editoriales independientes, las que deben hacer frente a una población que generalmente no tiene un alto consumo de estos productos. (<i>Estudio De Mercado. El Mercado Editorial En Ecuador 2023</i>, n.d.) Que esto se confirma gracias a los estudios del INEC en 2012, donde se ve que la población ecuatoriana tenía un 73,5% de tener hábito lector, pero solo el 50% le dedica de una a dos horas a la semana. Donde Quito sería la cuarta ciudad que más lee, situada detrás de Guayaquil, Ambato y Machala. Además, dice que apenas el 28% lee libros. El resto se distribuye en lectura de periódicos, revistas e Internet. Ningún grupo etario lee por placer o superación personal, las razones principales son profundizar en un tema u obtener información. (Instituto</p>

	<p>Nacional de Estadística y Censos [INEC], 2012).</p> <p>Y basándonos en un reportaje del municipio en el año 2021, más del 70% lo hace por cumplir con una obligación académica o laboral, para obtener información o profundizar en un tema en particular. (<i>Quito: Una Ciudad Devoradora De Libros En Potencia – Quito Como Vamos</i>, n.d.)</p> <p>Concluyendo que la situación económica y social de consumidores en Quito es de un nivel medio estable, ya que si existe una cantidad moderada de consumidores de libros que lo hacen por gusto, además de los que lo hacen también para investigar un tema.</p>
<p>Restricciones (Consideraciones externas que se deben tener para diseñar y desarrollar el proyecto, ejemplo: porcentaje de impuesto, registro sanitario u otros)</p>	<p>Algunas de nuestras restricciones son que se necesitará cumplir con ciertas obligaciones y restricciones impuestas por el Servicio de Rentas Internas (SRI), esto afecta a la creación de nuestro programa dado que el registro y obligaciones tributarias, es algo que no se está tomando en cuenta al momento de hacerlo y es algo que es importante al momento de venta y importación.</p> <p>Aunque los libros impresos están exentos del IVA en Ecuador, pero otros productos o servicios complementarios pueden estar sujetos a este impuesto. Sin embargo, debes verificar las últimas regulaciones ya que pueden cambiar</p> <p>Por lo que se debe inscribir en el RUC, que es un número de identificación tributaria necesario para cualquier actividad económica y emitir comprobantes de venta autorizados por el SRI (facturas, notas de venta) por cada transacción realizada.</p> <p>Además, se tendrás que presentar declaraciones mensuales o semestrales de IVA (Impuesto al Valor Agregado) y de Impuesto a la Renta.</p> <p>Se debe tener en cuenta que se debe cumplir con las normativas de protección al consumidor, ofreciendo políticas claras de devoluciones y atención a reclamaciones.</p> <p>Y también debemos cumplir con las normativas sobre derechos de autor, especialmente con los libros que esté</p>

	<p>protegido por leyes de propiedad intelectual.</p> <p>Cosa que nuestro programa de almacenaje y control no esta previsto a hacer.</p>
<p>Limitaciones (Consideraciones internas que se deben tener para diseñar y desarrollar el proyecto, ejemplo: lenguaje de programación, tipos de almacenamiento u otros)</p>	<p>Las limitaciones de nuestro conocimiento acerca del lenguaje C y recursos que contiene este mismo, pues nos limita en la gestión de memoria y en el tamaño fijo de los arreglos y matices, y dado que el programa está destinado a ser un programa de escritorio de bajo rendimiento, es decir, sin almacenar una gran cantidad de información o no, puede que para un futuro se requiera una gran cantidad de recursos del computador y de datos, por lo que el programa quedara obsoleto en dado caso, por lo que es posible que no sea ideal para almacenar una base de datos a gran escala al realizar el código en lenguaje C, sin embargo para este momento encaja bien con lo que necesita la librería.</p>

PLANTEAMIENTO DE PROPUESTAS DE SOLUCIÓN:

Opción 1: Programación en lenguaje C haciendo uso de conceptos básicos como estructuras de control (Sentencias if-else) y estructuras de repetición (Sentencias while, do-while, for), además del uso de arreglos con un tamaño fijo para el almacenaje de dato.

Opción 2: Programación modular en lenguaje C haciendo uso de módulos, arreglos y matrices de strings, además de aplicar conceptos de estructuras de control, estructuras de repetición, archivos de cabecera y vectores con un tamaño dinámico puesto por el usuario.

ANÁLISIS DE PROPUESTAS DE SOLUCIÓN:

Ventajas Propuesta N1:

- -Tiene facilidad de uso, por lo que es muy sencillo de implementar y entender.
- -Es fácil y rápido de programar, ya que es un Código simple la ejecución y elaboración de este no tardara demasiado.
- -Poca cantidad de variables.

Desventajas Propuesta N1:

- -Alto uso de memoria en el programa, ya que se hace uso de arreglos de tamaño fijo, por lo que, si queremos que el programa funcione bien, tendremos que poner a nuestros arreglos con un tamaño de 1000 o 100.
- -Se volverá obsoleto mucho tiempo después, además de que, sin el uso de conocimiento sofisticado de c, los procesos no estarán optimizados.

Ventajas propuestas N2:

- El código es fácil de leer y entender, con funciones bien definidas y responsabilidades específicas, lo que facilita su mantenimiento.
- Las funciones que leen y validan las entradas del usuario aseguran que los datos ingresados sean correctos y dentro de los límites esperados, reduciendo el riesgo de errores.
- El menú interactivo proporciona una interfaz sencilla para realizar operaciones como ingresar, editar y eliminar libros de manera intuitiva.
- La separación de responsabilidades en funciones distintas mejora la modularidad del código, facilitando su mantenimiento y ampliación.
- Los mensajes detallados guían al usuario a través de cada proceso, mejorando la experiencia de uso y haciendo la interacción con el programa más fácil y agradable.
- Se utilizan arreglos dinámicos, lo que hace que el espacio del programa no se vea afectado brutalmente.

Desventajas propuestas N2:

- Eliminar un elemento de un arreglo en C implica una serie de consideraciones adicionales. Además de simplemente eliminar la entrada del arreglo, se requiere reorganizar los elementos restantes para llenar el espacio vacío dejado por el elemento eliminado.
- Una limitación aquí es que los nombres de los libros no pueden contener espacios, ya que `scanf` no maneja espacios en cadenas de caracteres sin el uso de funciones adicionales o bibliotecas.
- El programa no identifica a los libros por su nombre, sino solo por el número con el que se le asigna.

SELECCIÓN DE LA MEJOR SOLUCIÓN, FUNDAMENTACIÓN TÉCNICA Y TEÓRICA.

La opción seleccionada fue la propuesta número 2 porque comparando las dos propuestas, las ventajas que nos proporciona la opción 2 son más a comparación de las ventajas de la solución 1 y de las desventajas de la solución 2.

Viendo la opción 2 vemos que proporciona una solución clara y estructurada para la gestión eficiente de un inventario de libros. Al utilizar matrices y arreglos de tamaño dinámico para almacenar los nombres, cantidades y precios de los libros, el programa puede manejar un número definido de libros de manera organizada y fácil de acceder, sin que la memoria del programa se vea afectada bruscamente.

La implementación de funciones y archivos de cabecera para leer y validar datos, además de organizar de mejor manera los procesos que suceden en el programa, es una gran

alternativa para su mantenimiento a largo plazo y también para comprender el Código de manera más sencilla.

Además, al permitir al usuario ingresar, agregar, visualizar, editar y eliminar datos de libros, el programa cubre las necesidades esenciales de gestión de inventario.

Aunque la eliminación de un libro en una matriz fija presenta limitaciones, el programa ofrece una solución alternativa viable mediante la marcación de entradas como eliminadas, lo que evita la corrupción de datos y garantiza la integridad del inventario.

Identificación de variables entrada, salida y procesos asociados.

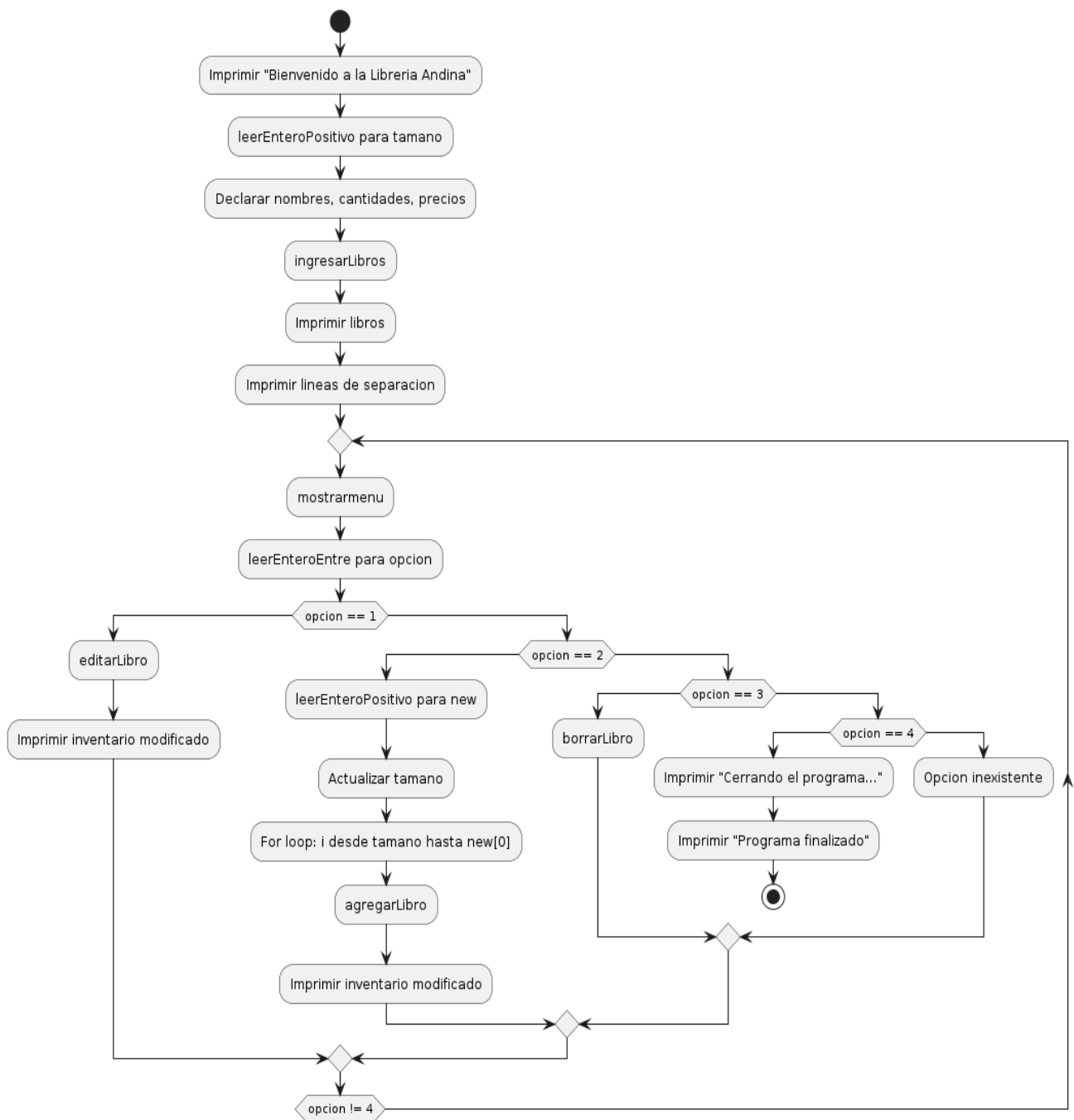
Variables de Entrada	Descripción
tamano	Almacena la cantidad de libros a gestionar, obtenida a través de la función leerEnteroPositivo.
opcion	Almacena la opción seleccionada por el usuario en el menú, obtenida mediante leerEnteroEntre.
new[1] y grade	Utilizadas para almacenar una nueva cantidad de libros cuando se agrega al inventario.
nombres	Arreglo bidimensional de caracteres que almacena los nombres de los libros ingresados.
cantidades	Arreglo de enteros que almacena las cantidades de cada libro.
precios	Arreglo de flotantes que almacena los precios de cada libro.

Variables de Salida	Descripción
nombres	Después de ser editados o modificados, se imprimen para mostrar el inventario actualizado.
cantidades	Actualizadas durante la ejecución y luego mostradas.
precios	Actualizados y luego mostrados.

Procesos	Descripción
Ingresar libros:	Esta función permite al usuario ingresar los detalles de cada libro, incluyendo el nombre, la cantidad y el precio. Utiliza un bucle para solicitar los datos del libro uno por uno hasta que se complete la cantidad de libros especificada por el usuario. Los detalles ingresados se almacenan en las matrices designadas para nombres, cantidades y precios.
Editar libros:	Esta función permite al usuario editar los detalles de un libro específico. Solicita al usuario que seleccione entre todos los datos por su número de índice en la lista, y luego solicita los nuevos valores para el nombre, la cantidad y el precio. Los detalles del libro seleccionado se actualizan en las matrices nombres, cantidades y precios.
Mostrar libros:	Esta función imprime los detalles de todos los libros ingresados, incluyendo el nombre, la cantidad y el precio. Recorre las matrices nombres, cantidades y precios y los imprime en un formato claro y estructurado.
Eliminar libros:	Esta función permite al usuario “eliminar” un libro específico del inventario, ya que lo que se hace es que se llena ese espacio de la matriz con valores nulos. Solicita al usuario que seleccione el libro por su número de índice en la lista y luego procede a eliminar dicho libro del inventario.
Agregar libros:	Es un proceso que permite agregar un nuevo libro al inventario, agregando más tamaño a las matrices para que así no haya problemas con el espacio de la memoria del programa.

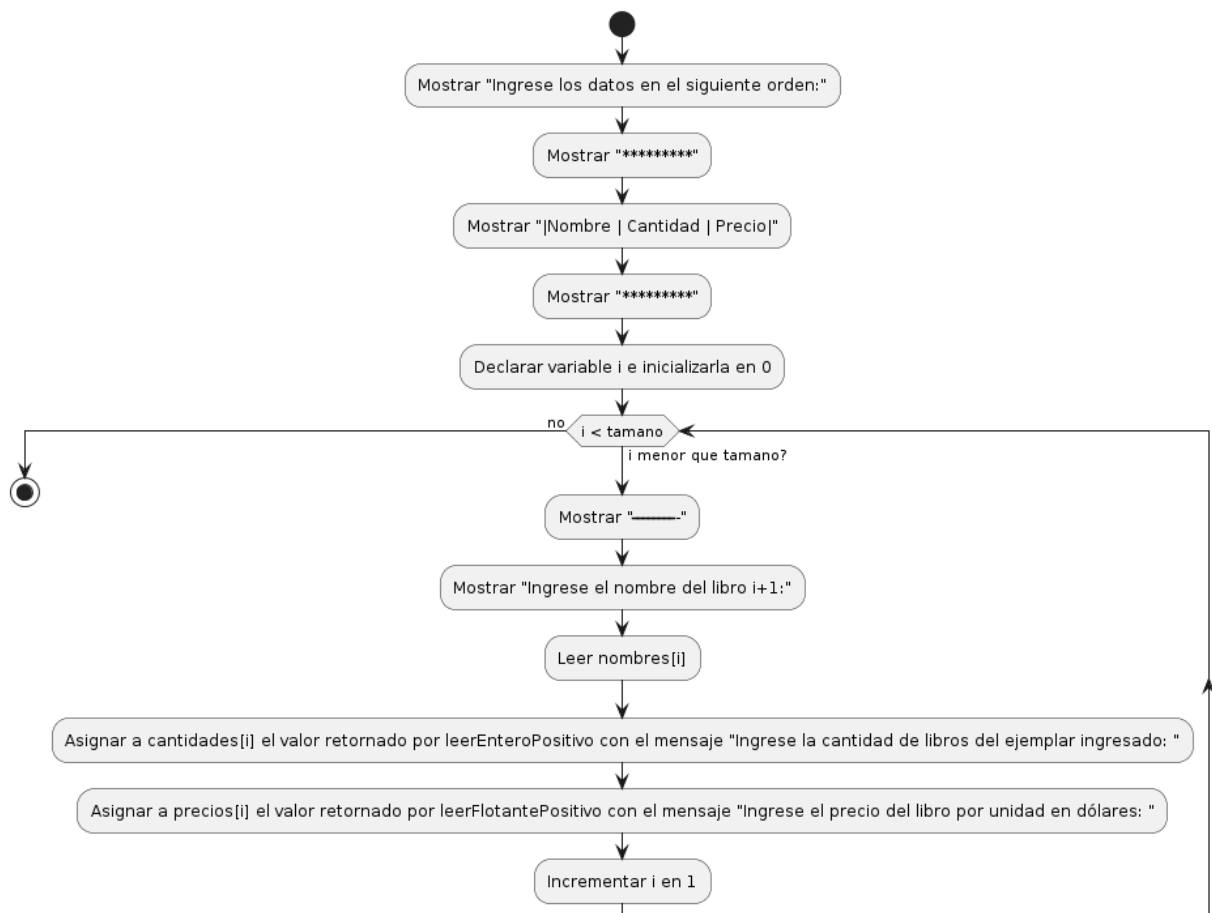
DISEÑO Y DESARROLLO DE LA SOLUCIÓN:

1. Diagrama de flujo del programa.

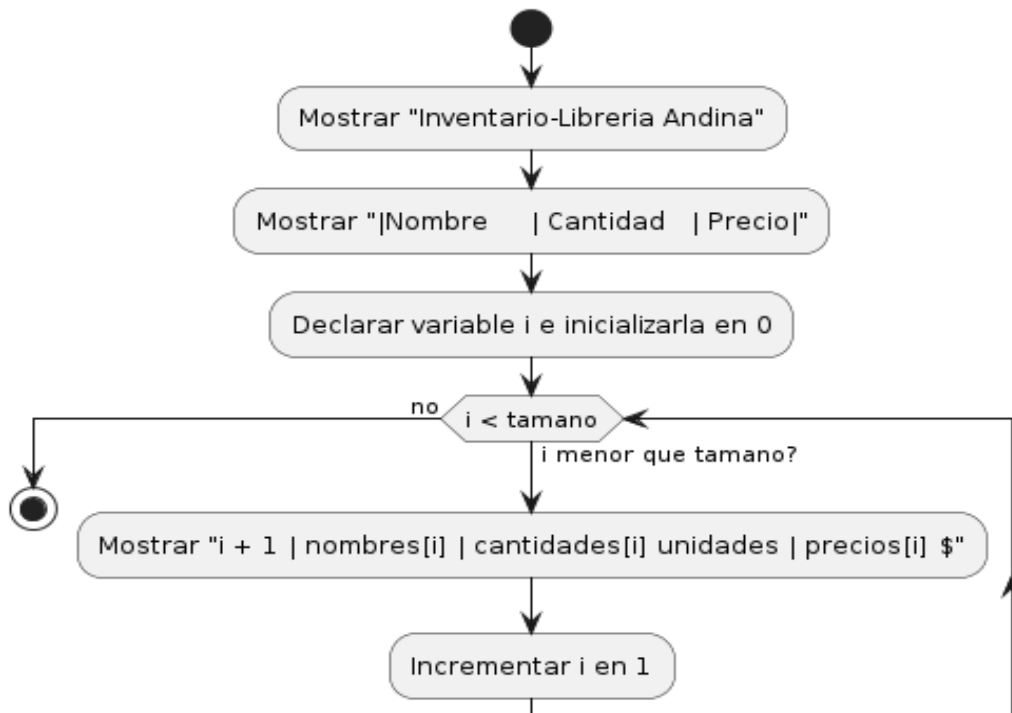


Las librerías usadas son los archivos de cabecera realizados en clase, como el de lecturas.h y operacionesVec.h, y estas son las funciones utilizadas de estas librerías:

IngresarLibro:



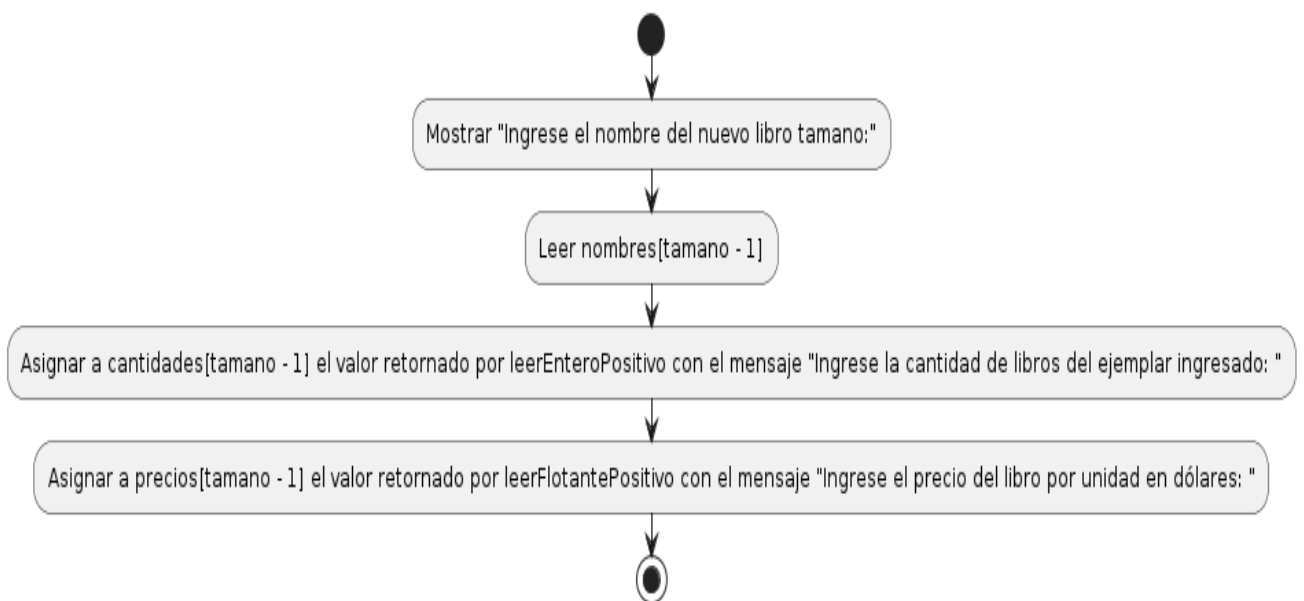
ImprimirLibro:



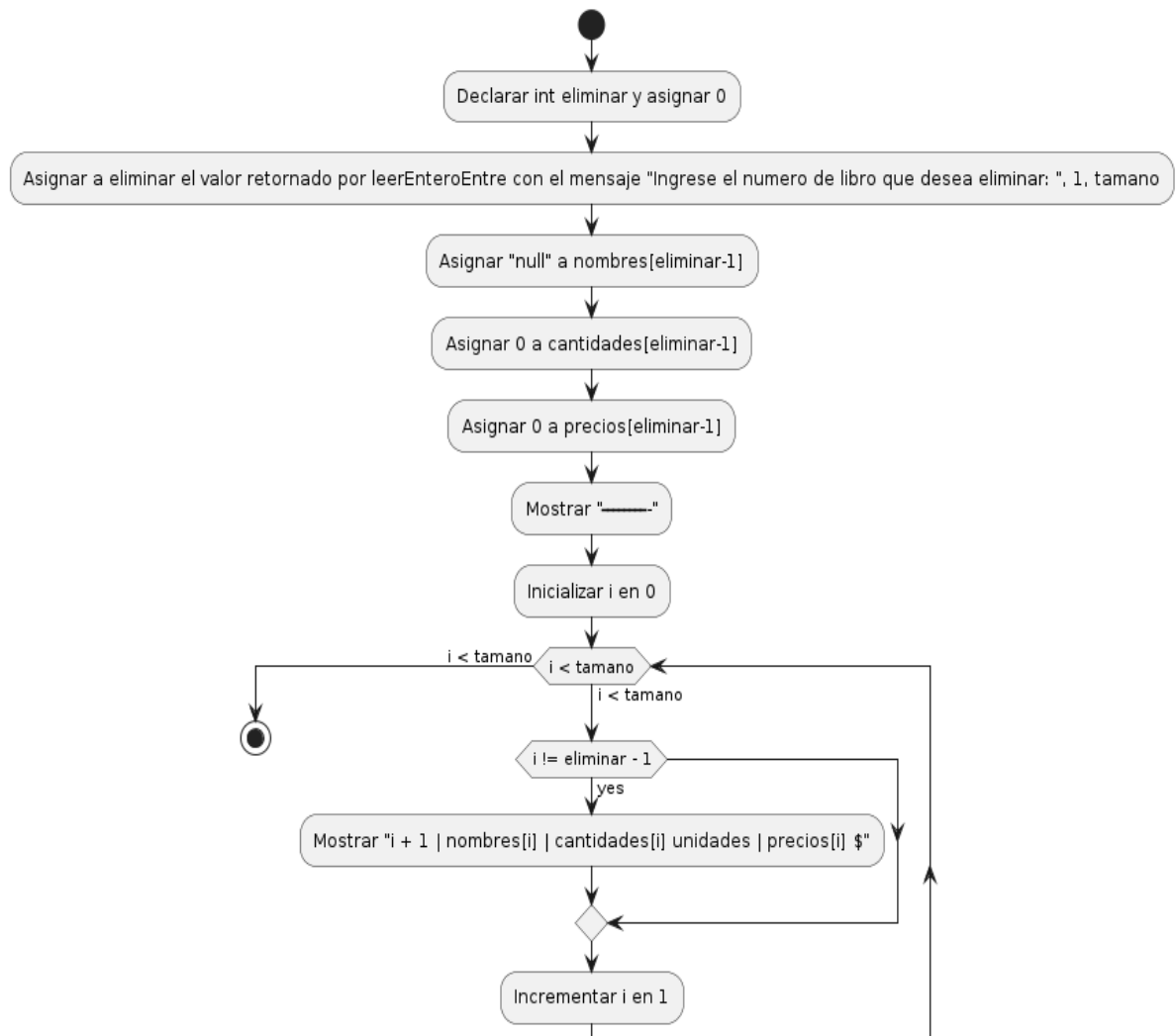
EditarLibro:



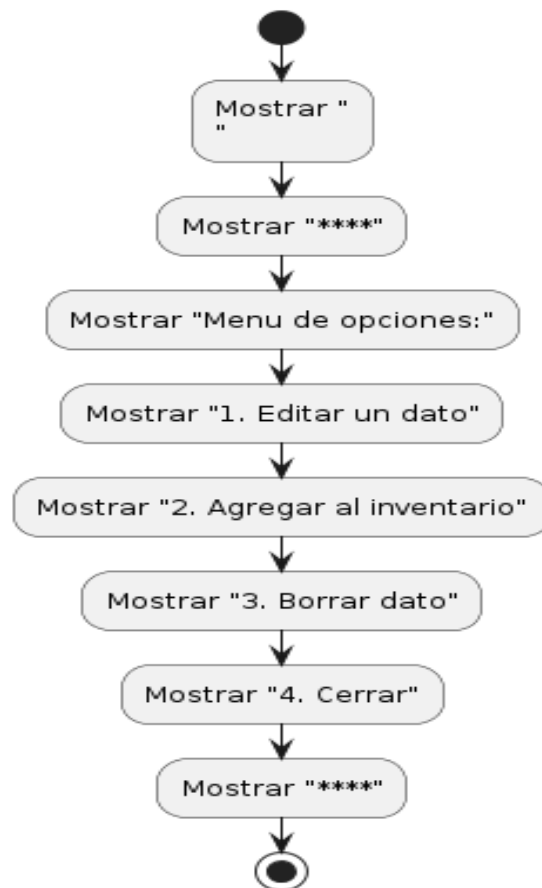
AgregarLibro:



BorrarLibro:

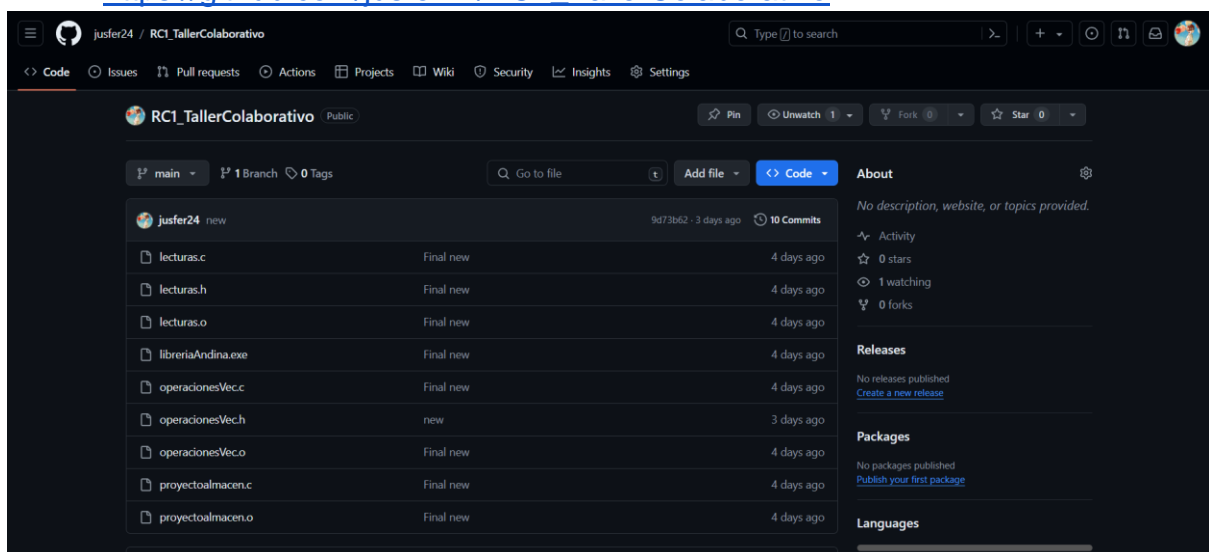


MostrarMenu:



2. Enlace de GitHub al código fuente del programa:

https://github.com/jusfer24/RC1_TallerColaborativo



3. Imágenes de las partes más importantes del código explicadas.
Primero empecemos con las librerías usadas en cada archivo:

```
#include <stdio.h>
#include "lecturas.h"
#include "operacionesVec.h"
```

Como se puede observar se utiliza una librería fuente de C, luego añadimos nuestras propias librerías, que son “lecturas.h” y “operacionesVec.h”, el primero se encarga de la lectura y validación de los datos que necesitan ser ingresados al programa. El segundo es donde están la mayoría de procesos importantes para nuestro controlador de inventario y almacenaje de la librería.

```
int main(int argc, char const *argv[]) {
    int tamano, opcion, new[1], grade;
    printf("--> * Bienvenido a la Libreria Andina* <--\n");
    tamano = leerEnteroPositivo("\nIngrese cantidad de libros a almacenar: ");
    char nombres[tamano][50];
    int cantidades[tamano];
    float precios[tamano];

    ingresarLibros(nombres, cantidades, precios, tamano);
    printf("\n");
    printf("+++++\n");
    printf("\n");
    imprimirLibros(nombres, cantidades, precios, tamano);
    printf("\n");

    do {
        mostrarmenu();
        opcion = leerEnteroEntre("Ingrese su opcion: ", 1, 4);
        switch (opcion) {
            case 1:
                editarLibro(nombres, cantidades, precios, tamano);
                printf("\n Inventario modificado:\n");
                imprimirLibros(nombres, cantidades, precios, tamano);
                break;
        }
    } while (opcion != 0);
}
```

La primera parte del código da la bienvenida al programa, además de ser también la parte donde se declaran las variables, vectores y matrices de caracteres del main, por lo que es la encargada de llamar a las funciones puestas en la librería “operacionesVec.h”.

```
tamano = leerEnteroPositivo("\nIngrese cantidad de libros a almacenar: ");
char nombres[tamano][50];
int cantidades[tamano];
float precios[tamano];
```

```

int main(int argc, char const *argv[]) {
    clean;
    case 2:
        new[0] = leerEnteroPositivo("\nIngrese la nueva cantidad de libros a almacenar: ");
        grade=new[0];
        for (int i = tamano; i < grade; i++){
            printf("Ingrese la informacion de venta del libro %i:\n", i+1);
            agregarLibro(nombres, cantidades, precios, i+1);
            printf("*****\n");
        }
        tamano=new[0];
        printf("\n Inventario modificado:\n");
        imprimirLibros(nombres, cantidades, precios, tamano);
        break;
    case 3:
        borrarLibro(nombres, cantidades, precios, tamano);
        break;
    case 4:
        printf("Cerrando el programa...\n");
        break;
    default:
        printf("Opción inexistente.\n");
    }
} while (opcion != 4);

printf("Programa finalizado\n");
return 0;
}

```

4. Imágenes de la ejecución de cada sección del programa explicadas.

CONCLUSIONES Y RECOMENDACIONES.

Conclusiones:

- Por último, pero no menos importante, el uso de archivos en su programa de facturación de libros es esencial para almacenar y recuperar datos de manera constante.
- En conclusión, es posible compartir los datos entre diferentes ejecuciones del programa al guardar los datos en archivos. En una nueva sesión, se pueden leer los datos almacenados en una ejecución anterior y continuar trabajando con ellos.
- Finalmente, se puede obtener información adicional necesaria para el programa leyendo archivos externos, como el archivo de servicios. Esto permite mantener los datos flexibles y actualizados sin tener que cambiar el código fuente del programa cada vez que se agrega o cambia un servicio.

Recomendaciones:

- Dado que mejora la legibilidad y facilita el mantenimiento del programa, se recomienda que siempre tenga una estructura clara y modular.

- Se debe realizar pruebas al sistema antes de implementarlo así se podrá corregir y detectar algún error antes de que el programa sea presentado ante los clientes.
- Es importante tomar en cuenta la funcionalidad del código ya que probablemente con el tiempo este necesite de arreglos, cambios o mantenimientos para mejorar su calidad.