

Universidad Rafael Landívar  
Facultad de Ingeniería  
Ingeniería en Informática y Sistemas  
Laboratorio de Estructura De Datos I  
Sección: 01  
Ing. Daniel Mejía



## **PROYECTO DE APLICACIÓN #1 “LIBRETA DE CONTACTOS”**

Nahomy Mariángel Chitay Duarte.  
Carné: 1211523.  
Angel Samuel Pérez Cruz.  
Carné: 1135323.

Guatemala, 25 de abril de 2025.

## INTRODUCCIÓN

En el mundo actual, donde la información personal y profesional debe ser gestionada con eficiencia, contar con herramientas digitales que permitan almacenar, consultar y organizar contactos se vuelve indispensable. Este proyecto nace como respuesta a la necesidad de una empresa de desarrollo de software que busca una solución funcional, estructurada y optimizada para la gestión de contactos.

El presente proyecto tiene como objetivo principal el desarrollo de una aplicación de consola en lenguaje Java, que permita almacenar, consultar y manipular información de contactos personales de forma eficiente y estructurada. Para ello, se hace uso de estructuras de datos fundamentales, específicamente Árboles Binarios de Búsqueda (BST) y Árboles AVL, con el fin de optimizar los procesos de búsqueda e indexación.

La aplicación desarrollada, denominada Libreta de Contactos, no solo permite agregar, eliminar y modificar contactos, sino que también brinda la capacidad de crear índices personalizados sobre distintos campos como nombre, apellido y número telefónico, mejorando la velocidad de búsqueda. Además, incorpora funcionalidades para la importación y exportación de datos mediante archivos CSV, lo que garantiza la persistencia y reutilización de la información.

Este documento detalla el análisis, diseño, implementación y pruebas de la aplicación, así como el funcionamiento interno de las estructuras utilizadas, y proporciona una guía clara para su uso. El desarrollo de esta herramienta no solo representa un ejercicio práctico de aplicación de conocimientos adquiridos en el curso de Estructura de Datos I.

## ANÁLISIS

### Entradas

- Desde el usuario (consola):
  - Datos para un nuevo contacto: nombre, apellido, apodo, teléfono, correo, dirección, fecha de nacimiento.
  - ID para eliminar o actualizar contactos.
  - Campo y valor para realizar búsquedas.
  - Campo e índice (BST o AVL) para creación o carga de índices.
  - Archivo .txt para reconstrucción de índices.
- Desde archivos:
  - contacts.csv: archivo de entrada principal con los contactos previos guardados.
  - Archivos .txt con recorridos por niveles de índices guardados (nombre-avl.txt, apellido-bst.txt, etc.).

### Procesos

- Gestión de contactos:
  - Carga inicial desde contacts.csv.
  - Adición de nuevos contactos con generación automática de ID.
  - Eliminación y actualización de contactos.
  - Búsqueda por campo (nombre, apellido, teléfono, etc.).
  - Visualización de todos los contactos.
- Manejo de archivos:
  - Guardado automático de contactos en contacts.csv.
  - Exportación de índices por nivel en archivos .txt.
  - Carga de índices desde archivos .txt.
- Indexación:
  - Creación de árboles BST o AVL por campo seleccionado.
  - Inserción de claves (valor de campo) y asociación con ID.
  - Recorrido por niveles de los árboles para guardar el índice.

### Salidas

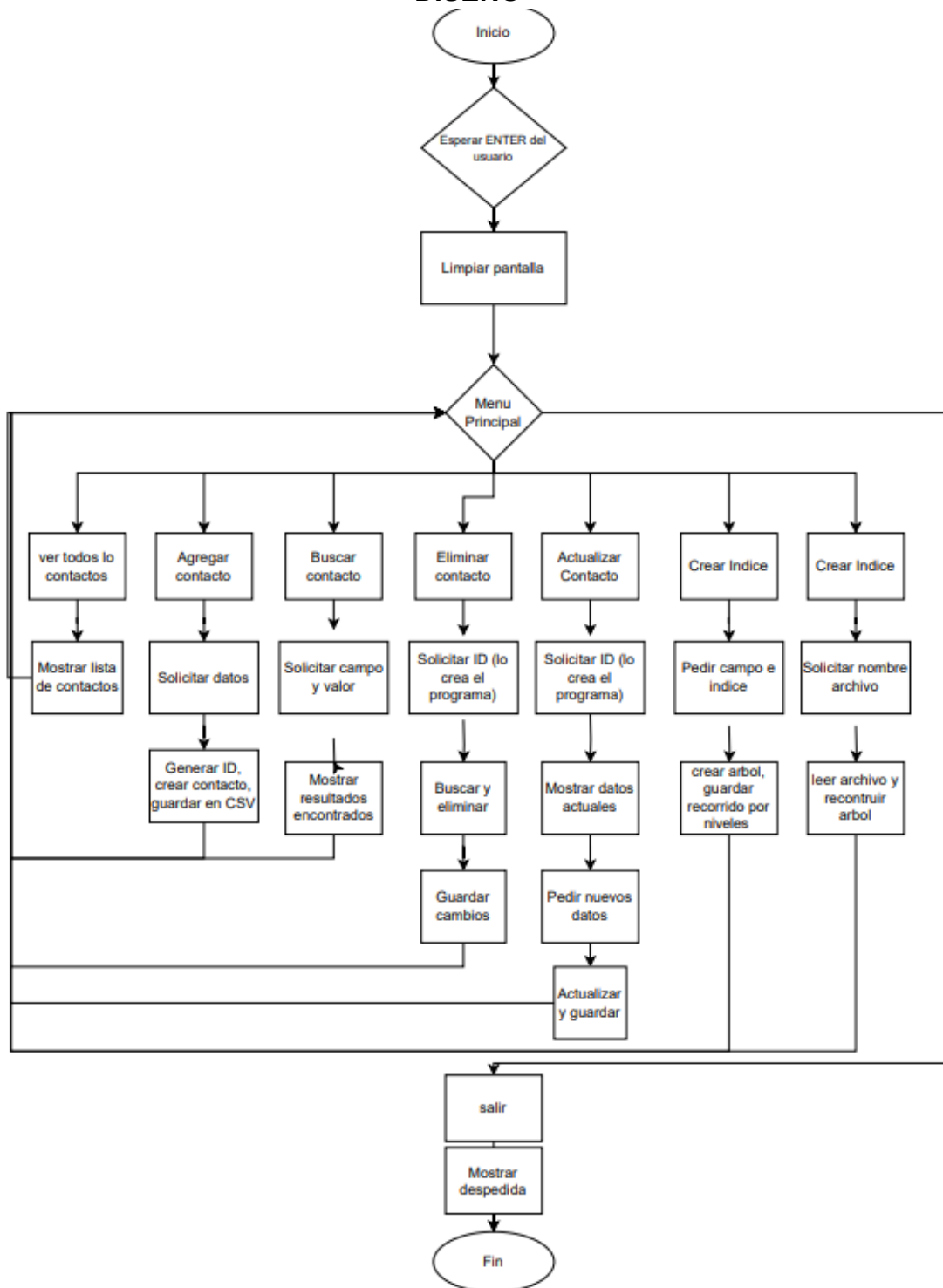
- En consola:
  - Mensajes informativos y de confirmación para cada operación (ej. "Contacto agregado", "Contacto eliminado").
  - Resultados de búsqueda con formato legible.
  - Listado completo de contactos.
  - Listado de IDs al reconstruir un índice desde archivo.
- En archivos:
  - contacts.csv: almacenamiento persistente de todos los contactos.
  - Archivos de índices: campo-tipo.txt (ej. nombre-avl.txt) con IDs en recorrido por niveles.

### Restricciones

- El ID de los contactos debe ser único y asignado automáticamente.
- El archivo contacts.csv debe tener un formato específico y estar ubicado en src/main/resources/.

- No se debe usar librerías de terceros para implementar las estructuras de datos.
- Los índices deben contener únicamente los IDs, no la información completa del contacto.
- Los árboles no deben aceptar claves duplicadas.
- Se debe garantizar que todos los datos sean consistentes entre los archivos y los objetos en memoria.
- La entrada del usuario debe ser controlada para evitar errores por campos vacíos o mal ingresados.

## DISEÑO



## CONCLUSIONES

1. El uso de estructuras de datos como BST y AVL mejora significativamente el rendimiento en la búsqueda e indexación de contactos, permitiendo acceder rápidamente a la información deseada según distintos campos como nombre, apellido o número telefónico.
2. La modularidad y separación en clases facilita la mantenibilidad del código, ya que permite modificar, depurar o extender funcionalidades sin afectar otras partes del sistema.
3. El almacenamiento en archivos CSV garantiza la durabilidad de los datos, permitiendo que los contactos permanezcan guardados incluso después de cerrar la aplicación.
4. La creación y reconstrucción de índices a partir de archivos externos muestra la capacidad del sistema para manejar grandes volúmenes de datos, siendo una solución escalable y eficiente.
5. El desarrollo de este proyecto permitió aplicar conocimientos teóricos de estructuras de datos en un contexto práctico, reforzando la comprensión de algoritmos de inserción, búsqueda, recorrido y balanceo de árboles.

## RECOMENDACIONES

1. Implementar validaciones de entrada más robustas, como verificación de formato de correo electrónico, número telefónico y fecha, para evitar errores y asegurar la calidad de los datos ingresados por el usuario.
2. Incorporar funciones de búsqueda avanzada, como coincidencias parciales, búsquedas por múltiples criterios, o uso de expresiones regulares, para ofrecer una experiencia más flexible al usuario.
3. Automatizar la actualización de índices al modificar contactos, evitando que el usuario deba regenerarlos manualmente después de cada cambio en los datos.
4. Permitir ordenamientos personalizados y filtros dinámicos al visualizar contactos, como ordenar por apellido, mostrar solo contactos con cumpleaños próximos, etc.
5. Incluir pruebas unitarias y de integración para garantizar la estabilidad del sistema, especialmente al agregar nuevas funcionalidades o refactorizar el código existente.

## ANEXOS – MANUAL

### Al ejecutar el programa muestra una “Bienvenida”

```
C:\Users\maria\.jdk\azul-21.0.6\bin\java.exe "-javaagent:C:\Program Files\

;BIENVENIDO A LA LIBRETA DE CONTACTOS!

Administra fácilmente tus contactos
con búsquedas rápidas, actualización
dinámica y respaldo en archivo CSV.

Presiona ENTER para continuar...
```

### Muestra el menú

```
LIBRETA DE CONTACTOS - MENÚ

1. Ver todos los contactos
2. Agregar contacto
3. Buscar contacto por campo
4. Eliminar contacto por ID
5. Actualizar contacto por ID
6. Crear y guardar índice
7. Cargar índice desde archivo
0. Salir

Seleccione una opción: |
```

### Al seleccionar la opción 1 se mostrarán los contactos que se encuentran en el archivo de prueba “contactos.CSV”

```
Seleccione una opción: 1
Contacto #1 - Judit Polgar (Judy)
Teléfono: 98989898 | Email: polgar@judy.com
Dirección: Budapest 1 | Fecha Nacimiento: 19760723

Contacto #2 - Adriana Ruano (Adri)
Teléfono: 89898989 | Email: aruano@mail.com
Dirección: Ciudad de Guatemala 2 | Fecha Nacimiento: 19950626

Contacto #3 - Lionel Messi (Leo Messi)
Teléfono: 50505050 | Email: lio@messi.com
Dirección: Miami 1 | Fecha Nacimiento: 19870624

Contacto #4 - Cristiano Ronaldo (CR7)
Teléfono: 12345678 | Email: cr7@mail.com
Dirección: Arabia 1 | Fecha Nacimiento: 19850205

Contacto #5 - Serena Williams (Serena)
Teléfono: 87654321 | Email: williams@serena.com
Dirección: Michigan 1 | Fecha Nacimiento: 19810923

Contacto #7 - Aitana Bonmati (Aitana)
Teléfono: 99999999 | Email: aitana@fcb.com
Dirección: Barcelona 1 | Fecha Nacimiento: 19980118

Contacto #9 - Michael Phelps (Phelps)
Teléfono: 10101010 | Email: michael@phelps.com
Dirección: Maryland 1 | Fecha Nacimiento: 19850630

Contacto #6 - Kathleen Ledecy (Katie)
Teléfono: 11111111 | Email: katie@mail.com
Dirección: Washington 1 | Fecha Nacimiento: 19970317

Contacto #8 - Magnus Carlsen (Magnus)
Teléfono: 55555555 | Email: magnus@chess.com
Dirección: Noruega 1 | Fecha Nacimiento: 19901130

Contacto #10 - Carlos Ruiz (Pescadito)
Teléfono: 45454545 | Email: fish@soccer.com
Dirección: Ciudad de Guatemala 1 | Fecha Nacimiento: 19790915

Contacto #11 - Luis Lopez (L)
Teléfono: 44590123 | Email: luis@gmail.com
Dirección: Guate | Fecha Nacimiento: 2002-03-21
```



Si se desea agregar un nuevo contacto, elegir la opción 2

```
Seleccione una opción: 2

NUEVO CONTACTO
Nombre: Alejandra
Apellido: Hernandez
Apodo: Ale
Teléfono: 22116709
Email: alejandra@gmail.com
Dirección: Guatemala
Fecha de nacimiento (YYYY-MM-DD): 2002-03-21
Contacto agregado con ID: 12
```

Si se quiere buscar algo en específico, elegir la opción 3

```
Seleccione una opción: 3

BÚSQUEDA DE CONTACTO
Campo a buscar (nombre, apellido, apodo, telefono, email, direccion, fecha): apodo
Valor a buscar: pescadito

RESULTADOS:
Contacto #10 - Carlos Ruiz (Pescadito)
Teléfono: 45454545 | Email: fish@soccer.com
Dirección: Ciudad de Guatemala 1 | Fecha Nacimiento: 19790915
```

Si se quiere eliminar un contacto, elegir la opción 4

```
Seleccione una opción: 4

Ingrese el ID del contacto a eliminar: 12
Contacto eliminado.
```

Si se quiere actualizar información de algún contacto, elegir la opción 5. (llenar solo la información que se desea actualizar.)

```
Seleccione una opción: 5

Ingrese el ID del contacto a actualizar: 8
Deje vacío cualquier campo para conservar el valor actual.
Nuevo Nombre [Magnus]: Magnuss
Nuevo Apellido [Carlsen]:
Nuevo Apodo [Magnus]: Mag
Nuevo Teléfono [55555555]: 12309956
Nuevo Email [magnus@chess.com]:
Nueva Dirección [Noruega 1]:
Nueva Fecha de Nacimiento [19901130]:
Contacto actualizado.
```

Si el usuario quiere crear un índice ordenado (BST o AVL) sobre alguno de los campos del contacto (como nombre, apellido, teléfono, etc). Elegir la opción 6

Seleccione una opción: 6

Campo para indexar (nombre, apellido, etc.): *apellido*

Tipo de índice (BST o AVL): *AVL*

Índice guardado en: src/main/resources/apellido-avl.txt

Si se quiere reconstruir un árbol BST o AVL a partir de un archivo de índice previamente guardado (como los que crea la opción 6). Elegir la opción 7

Seleccione una opción: 7

Nombre del archivo del índice (.txt): *apodo-bst.txt*

Campo del índice (nombre, apellido, etc.): *apodo*

Tipo de índice (BST o AVL): *BST*

Índice reconstruido desde archivo: apodo-bst.txt

IDs en el índice cargado:

1 2 3 4

### Programa finalizado

Seleccione una opción: 0

¡Gracias por usar la Libreta de Contactos!

¡Esperamos verte pronto! :D