

Universidad Rafael Landívar
Facultad de Ingeniería
Ingeniería en Informática y Sistemas
Lab Estructura de Datos Sección 01
Ingeniero René Daniel Mejía



Proyecto de Aplicación No. 1 2025 Libreta de Contactos

Nombres: González Cantoral, Andry Paolo - 2000924
Palencia Aldana, Kevin Rodrigo - 1045324

Guatemala, 25 de abril del 2025

ÍNDICE

RESUMEN	3
ANÁLISIS Y DISEÑO	4
Main.java	4
Contacto.java	4
GestorCSV.java	4
ArbolBST.java y NodoBST.java	4
ArbolAVL.java y NodoAVL.java	4
Diseño Lógico del Sistema Entradas	5
Procesos	6
Salidas.....	6
DIAGRAMA DE FLUJO.....	7
MANUAL DE USUARIO.....	10
Ejecución del programa	10
Menú principal	10
Funciones disponibles	11
Agregar contacto	11
Eliminar contacto	12
Actualizar contacto	12
Buscar contacto	12
Mostrar todos los contactos	12
Exportar contactos.....	12
Importar contactos	12
Crear o reconstruir índice.....	13
Estructura del archivo CSV	13
Notas importantes.....	13
Finalizar programa.....	13

RESUMEN

Este proyecto está basado en una aplicación de consola en Java para la gestión de contactos. Permite almacenar, buscar y organizar la información de personas de forma eficiente. Cada contacto incluye datos como nombre, teléfono, correo, dirección, entre otros, y se guarda en un archivo CSV.

Para mejorar el rendimiento en la búsqueda, se implementaron estructuras de datos como Árboles Binarios de Búsqueda (BST) y Árboles AVL, con la opción de crear índices por diferentes campos. Estos índices se guardan en archivos separados y permiten búsquedas rápidas.

La aplicación permite agregar, eliminar, modificar, importar y exportar contactos, así como reconstruir los índices. Todo el sistema está dividido en clases y sigue principios de programación orientada a objetos.

ANÁLISIS Y DISEÑO

Estructura general del sistema: El proyecto está dividido en varias clases, cada una con una función específica:

Main.java

El método main actúa como la interfaz de usuario por consola. Aquí se crean instancias de las clases principales, se crea el menú interactivo con todas las opciones disponibles y se crean los métodos bases que es agregar contacto, imprimir dependiendo el tipo de árbol, buscar contacto dependiendo del tipo de árbol, guardar y cargar los contactos y convertir el árbol en archivo .txt.

Contacto.java

La clase contacto contiene todos los atributos que lleva un contacto a la hora de ser guardado (nombre, apellido, apodo etc..). Tiene métodos get y set para acceder a los datos y un método toString() para mostrar la información del contacto.

GestorCSV.java

En esta clase se crea el método para guardas los contactos en un archivo .CSV el cuál está redirigido a una carpeta local, también posee el método para cargar archivos de tipo CSV por medio un FileReader.

ArbolBST.java y NodoBST.java

La clase Implementa un Árbol Binario de Búsqueda, ordenado por nombre. Este árbol permite insertar, buscar y recorrer los contactos de forma ordenada (inorden). También implementa un recorrido por niveles para generar archivos .txt de índices.

ArbolAVL.java y NodoAVL.java

Implementan un Árbol AVL (autobalanceado), también ordenado por nombre. Este árbol permite inserción balanceada, búsqueda y recorrido inorden. Incluye rotaciones simples y dobles. Se usa para crear índices eficientes. También implementa recorrido por niveles.

Diseño Lógico del Sistema

Entradas

Selección de opción del menú (1 al 12)

1. Agregar Contacto:

- Nombre
- Apellido
- Apodo
- Teléfono
- Correo electrónico
- Dirección
- Fecha de Nacimiento

2. Mostrar Contactos (Árbol BST)

3. Mostrar Contactos (Árbol AVL)

4. Buscar Contactos (Árbol BST)

5. Buscar Contactos (Árbol AVL)

6. Guardar Contactos en archivo CSV

7. Cargar Contactos desde un archivo CSV

8. Generar archivo de recorrido por niveles (Nombre - BST)

9. Generar archivo de recorrido por niveles (Apellido - AVL)

10. Generar archivo de recorrido por niveles (Apodo - AVL)

11. Reconstruir Árbol desde archivo de ID

12. Salir del Programa...

Procesos

Operaciones internas que realiza el sistema:

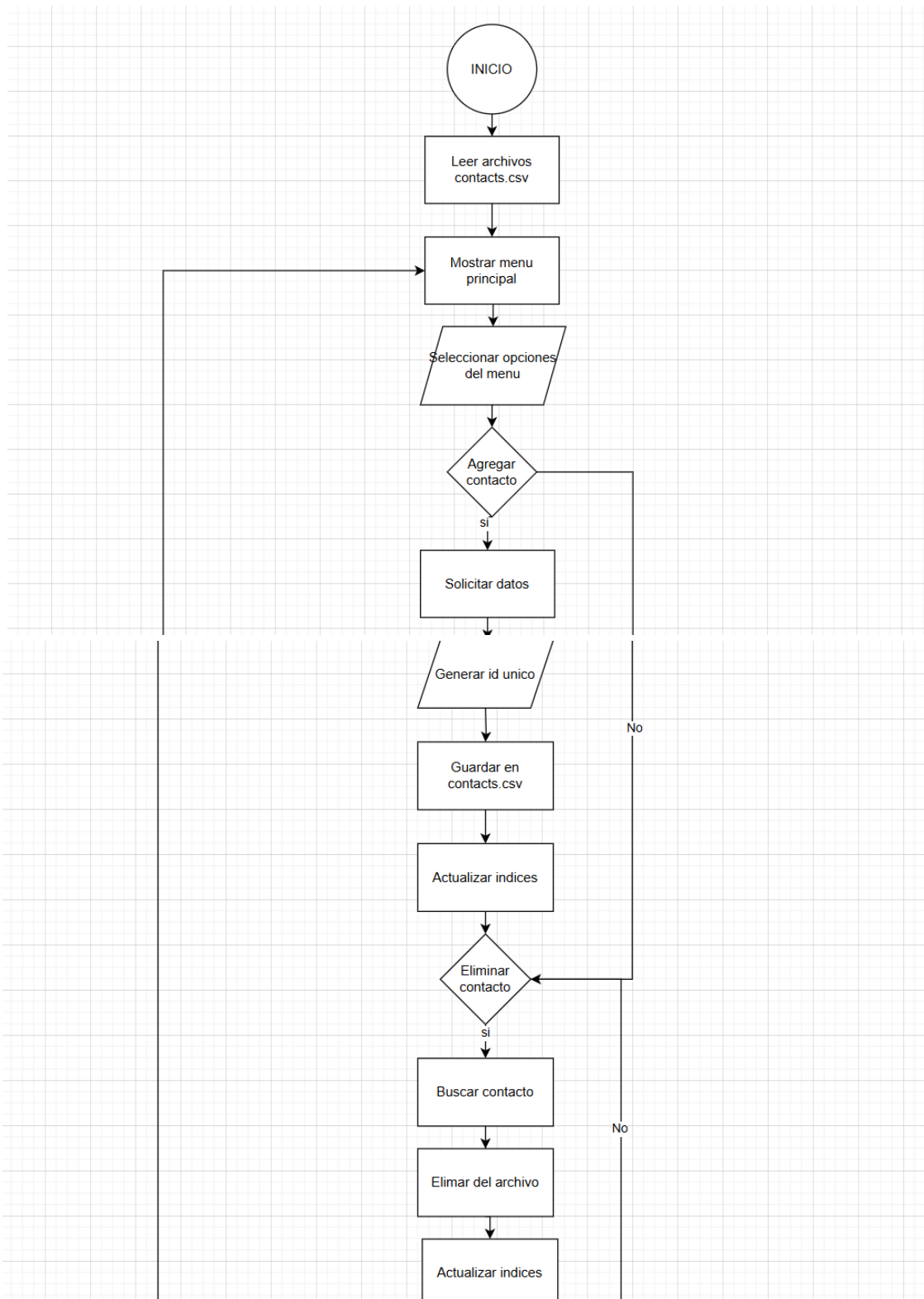
1. Leer archivo contacts.csv
2. Generar un ID único
3. Agregar contacto al archivo y estructuras
4. Eliminar contacto del archivo e índices
5. Actualizar contacto en el archivo e índices
6. Buscar en índice (si existe) o crear uno nuevo
7. Mostrar todos los contactos
8. Exportar datos a un nuevo CSV
9. Importar contactos desde un CSV externo
10. Crear índice BST o AVL
11. Reconstruir índices desde archivos de recorrido
12. Actualizar archivos e índices luego de cualquier cambio

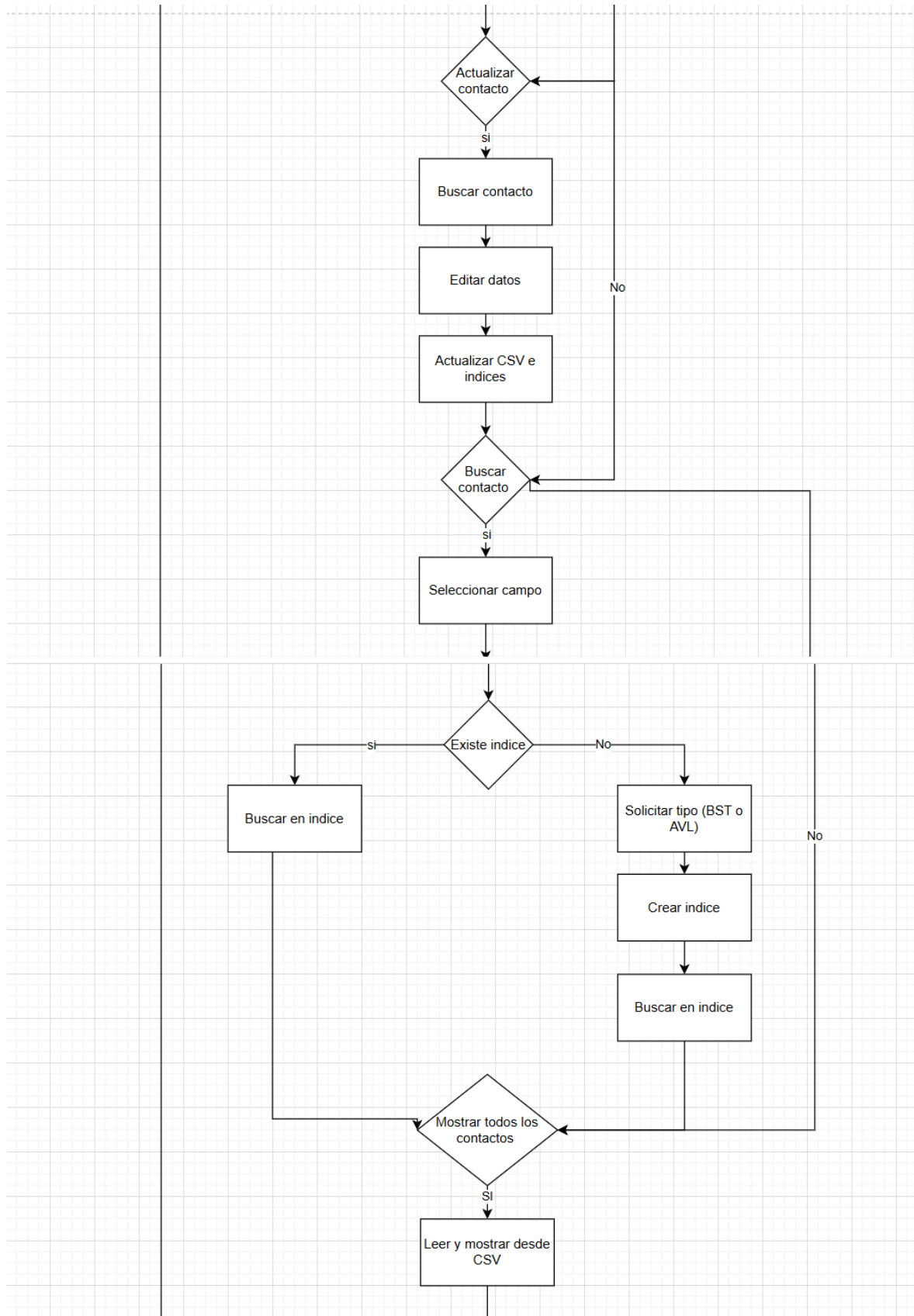
Salidas

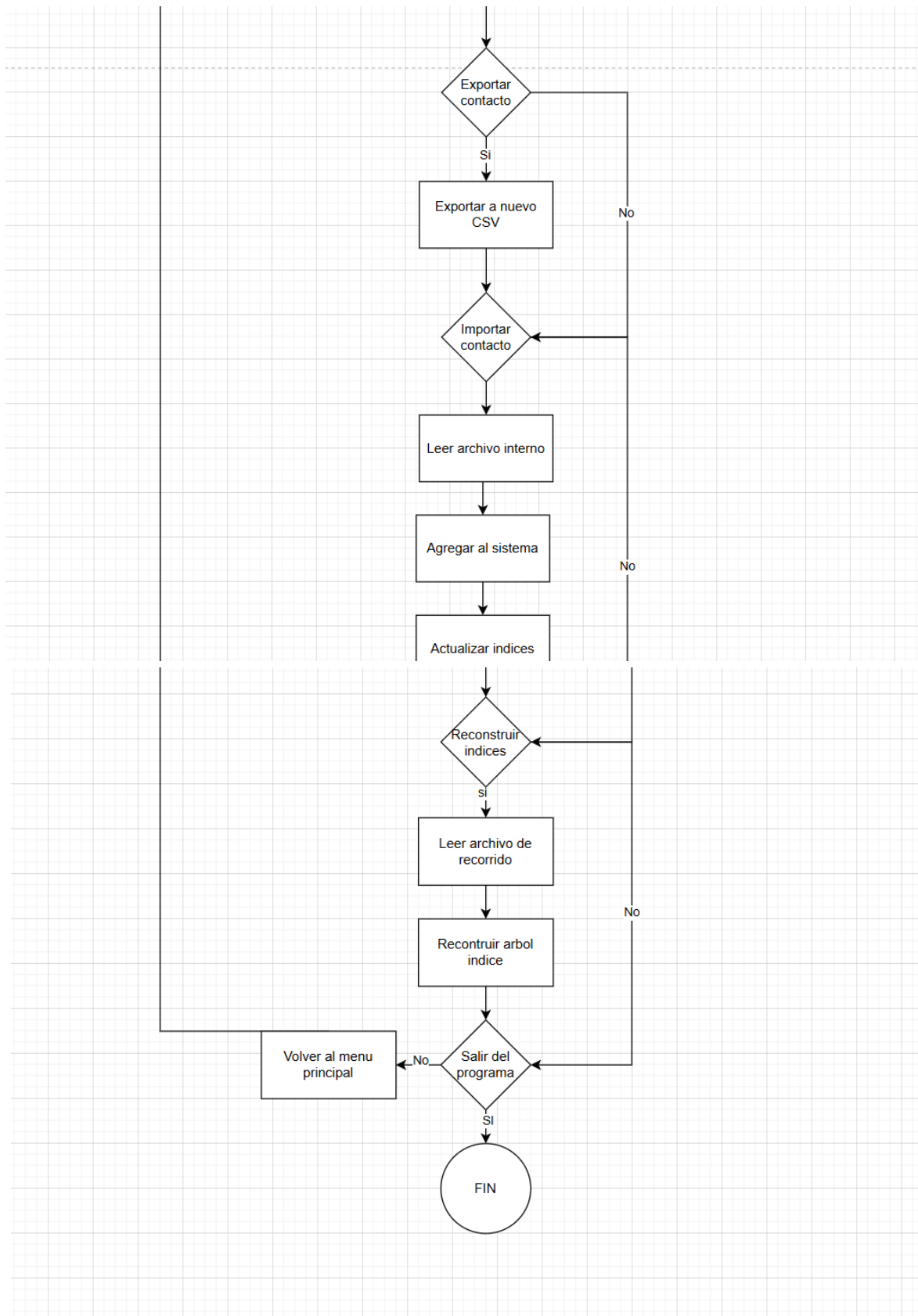
Resultados que el sistema muestra o guarda:

1. Contacto agregado exitosamente (mensaje en consola)
2. Contacto eliminado/actualizado
3. Resultados de la búsqueda (impresos en consola)
4. Listado de todos los contactos
5. Archivo contacts.csv actualizado
6. Archivo exportado (nuevo CSV)
7. Archivos de índice generados (ej. apellido-avl.txt)
8. Mensajes de error o confirmación (ej. "No se encontró el contacto", "Índice creado con éxito")

DIAGRAMA DE FLUJO







MANUAL DE USUARIO

Ejecución del programa

1. Abrir la terminal en su programa de preferencia.
2. Navegar hasta la carpeta donde se encuentra el archivo Main.java.
3. Compile el proyecto

Menú principal

Cuando se ejecuta, el programa muestra un menú con varias opciones. El usuario debe ingresar el número de la acción que desea realizar:

1. Agregar contacto
2. Mostrar Contactos (Árbol BST)
3. Mostrar Contactos (Árbol AVL)
4. Buscar Contactos (Árbol BST)
5. Buscar Contactos (Árbol AVL)
6. Guardar Contactos en archivo CSV
7. Cargar Contactos desde un archivo CSV
8. Generar archivo de recorrido por niveles (Nombre - BST)
9. Generar archivo de recorrido por niveles (Apellido - AVL)
10. Generar archivo de recorrido por niveles (Apodo - AVL)
11. Reconstruir Árbol desde archivo de ID
12. Salir del Programa

Funciones disponibles

Agregar contacto

- Se solicita: nombre, apellido, apodo, teléfono, email, dirección, fecha de nacimiento.
- El sistema genera un ID único.
- Se guarda en el archivo contacts.csv.
- Se actualizan los índices si ya existen.

Eliminar contacto

- Se solicita el ID o campo de búsqueda.
- El contacto es eliminado del archivo y los índices son actualizados.

Actualizar contacto

- Se busca el contacto.
- Se editan los campos deseados.
- Se guarda el cambio y se actualizan los índices.

Buscar contacto

- Se elige el campo de búsqueda (ej. nombre).
- Si no existe un índice, el sistema pregunta si desea crearlo (BST o AVL).
- Se realiza la búsqueda y se muestra el resultado.

Mostrar todos los contactos

- Lee todos los registros desde contacts.csv y los muestra en consola.

Exportar contactos

- Crea una copia del archivo actual en un nuevo CSV.

Importar contactos

- Se importa un archivo externo CSV con el mismo formato.
- Los nuevos contactos se agregan al sistema y los índices se actualizan.

Crear o reconstruir índice

- Se solicita el campo (ej. apellido) y el tipo de árbol (BST o AVL).
- Se genera un archivo con el recorrido por niveles del índice (ej. apellido-avl.txt).

Estructura del archivo CSV

id, nombre, apellido, apodo, telefono, email, direccion, fecha nacimiento
1, Juan, Pérez, Juanchi, 5551234, juan@email.com, Colonia Reforma, 1990-05-02

Notas importantes

- Si no hay índice, el sistema puede buscar de forma secuencial, pero será más lento.
- No edite el archivo CSV manualmente a menos que sepa lo que hace.
- Los índices se deben crear por cada campo que quiera buscar de forma eficiente.

Finalizar programa

Selecciona la opción 12 - Salir del menú para cerrar el sistema de forma segura.