

Barajas Mirada Zihary Leticia: Es la encargada de realizar los puntos 3 y 4 del trabajo. En primer lugar, se encargará de implementar la opción de eliminar los datos de un vendedor del archivo. En segundo lugar, debe asegurarse de que los usuarios puedan modificar cualquier información de un vendedor, con la excepción de su número de empleado. (Métodos implementados: deleteVendor() y modifyVendor() ).

Lopez Felix Julio Cesar: La tarea asignada a esta persona es desarrollar una función que permita a los usuarios realizar consultas por zona. Cuando los usuarios utilicen esta función, deberán mostrar todos los detalles de los vendedores que están asignados a una zona específica. (Método implementado: queryByZone() ).

José José Alex Gabi: Se va a encargar de realizar los dos primeros puntos del proyecto. En primer lugar, debe agregar un campo adicional al documento que contiene la información de los vendedores. Este nuevo campo se centrará en registrar las ventas mensuales realizadas por cada vendedor. Por último, debe implementar la opción de agregar un nuevo vendedor al archivo. (Método implementado: addVendor() ).

main (método principal): Este método es el punto de entrada de la aplicación. Realiza las siguientes acciones:

- Inicializa una lista llamada vendors que se utilizará para almacenar los datos de los vendedores.
- Utiliza un bucle do-while para mostrar un menú de opciones al usuario y recibir la selección del usuario.
- Las opciones incluyen agregar, borrar, modificar un vendedor, consultar vendedores por zona y salir del programa.
- Según la opción seleccionada por el usuario, llama a los métodos correspondientes para realizar la operación deseada.
- Al inicio del programa, carga los datos de los vendedores desde el archivo vendors-data.dat llamando a loadVendorsFromFile.
- Antes de salir del programa, guarda los cambios en el archivo llamando a saveVendorsToFile.

loadVendorsFromFile(): Este método se utiliza para cargar los datos de los vendedores desde el archivo vendors-data.dat. Detalle de su funcionamiento:

- Utiliza un objeto RandomAccessFile para abrir el archivo en modo lectura ("r").
- Lee el archivo en un bucle mientras haya datos por leer.

- Lee cada campo de un registro de vendedor, que incluye código, nombre, fecha de nacimiento, zona y ventas.
- Convierte la fecha de nacimiento de su formato de cadena (dd/MM/yyyy) a un objeto Date utilizando SimpleDateFormat.
- Crea un objeto Vendor con los datos leídos y lo agrega a la lista vendors.
- Maneja excepciones relacionadas con la lectura de datos incorrectos o fechas mal formateadas.
- Finalmente, muestra los vendedores cargados en la consola con fines de depuración.

Alex - addVendor(Scanner scanner): Permite al usuario agregar un nuevo vendedor. Detalles del proceso:

- Solicita al usuario que ingrese el código, nombre, fecha de nacimiento, zona y ventas del nuevo vendedor utilizando un objeto Scanner.
- Crea un nuevo objeto Vendor con los datos ingresados.
- Agrega el nuevo vendedor a la lista vendors.
- Llama a saveVendorsToFile para guardar la lista actualizada en el archivo.
- Muestra mensajes de éxito y confirma que los cambios se han guardado en el archivo.

Zihary - deleteVendor(Scanner scanner): Permite al usuario eliminar un vendedor existente.

- Solicita al usuario que ingrese el código del vendedor que desea eliminar.
- Busca en la lista vendors el vendedor con el código proporcionado y lo elimina si se encuentra.
- Si se elimina un vendedor, llama a saveVendorsToFile para guardar la lista actualizada en el archivo.
- Maneja excepciones relacionadas con entradas incorrectas del usuario.

Zihary - modifyVendor(Scanner scanner): Permite al usuario modificar los detalles de un vendedor existente.

- Solicita al usuario que ingrese el código del vendedor que desea modificar.
- Muestra un menú de opciones para seleccionar qué campo del vendedor se desea modificar (nombre, fecha, zona o ventas).
- Realiza la modificación solicitada y actualiza el vendedor en la lista vendors.

- Llama a `saveVendorsToFile` para guardar la lista actualizada en el archivo.
- Maneja excepciones y asegura que se realice la modificación correctamente.

César - `queryByZone(String zonaConsulta)`: Permite al usuario consultar vendedores por zona.

- Recibe como entrada la zona que se desea consultar.
- Utiliza `stream` y `filter` para buscar los vendedores que pertenecen a la zona proporcionada.
- Muestra los detalles de los vendedores encontrados en la consola, incluyendo su código, nombre, fecha de nacimiento, zona y ventas.

Alex - `saveVendorsToFile(List<Vendor> vendors)`: Guarda la lista de vendedores en el archivo `vendors-data.dat`.

- Utiliza un objeto `RandomAccessFile` para abrir el archivo en modo escritura ("rw").
- Ubica el puntero al final del archivo para agregar nuevos registros.
- Escribe los datos de cada vendedor en el archivo, incluyendo su ID, nombre, fecha de nacimiento y zona.

`readString(RandomAccessFile randomAccessFile, int length)`: Lee una cadena de caracteres de longitud fija desde el archivo binario. Se utiliza para leer campos de texto de longitud fija, como el nombre y la zona de los vendedores, del archivo.

`writeString(RandomAccessFile randomAccessFile, String value, int length)`: Escribe una cadena de caracteres en el archivo binario, asegurándose de que tenga la longitud especificada y llenando con espacios en blanco si es necesario. Se utiliza para escribir campos de texto de longitud fija en el archivo.