

Universidad Rafael Landívar
Facultad de Ingeniería
Ingeniería en Informática y Sistemas
Laboratorio Programación Avanzada, Sección 06
Ing. Daniel Mejía

PROYECTO 01

Erwin Andrés Lima Núñez
Carnet: 1158120

Guatemala, 18 de octubre de 2021

Introducción

El objetivo de este proyecto fue el de elaborar un programa que reciba un archivo del tipo csv y que genere una playlist de canciones. También se pueden añadir canciones a una fila de reproducción y esta se puede sincronizar con la playlist. Todo esto se hizo aplicando conceptos de estructuras de datos dinámicas, como pilas y colas. Estas son estructuras que se les puede añadir o quitar datos en tiempo de ejecución. Este programa se hizo en un form de Windows Forms.

Análisis

Crear las listas

Entradas	Salidas	Procesos	Validaciones
Archivo .csv separado por comas.	Lista con nodos simples guardando la información de las canciones.	<ol style="list-style-type: none">1. Inicio2. Se presiona el botón "Abrir"3. Se crea un filtro para archivos .csv4. Se limpia el textBox que guardará el nombre del archivo.5. Si se pudo abrir el explorador de archivos se guarda la dirección de memoria del archivo.6. Se crea un arreglo que guarde las líneas del archivo.7. Si la longitud del arreglo es mayor a 0 se crea un ciclo for. Dentro de este ciclo se separarán los elementos por comas y se guardarán en un arreglo.8. Se crea otro ciclo for en el que se verificará que la posición no este vacía. De no estarlo separará el nombre de la canción y el artista en base a "-" y guardará los elementos en un arreglo.9. Si el campo de nombre no esta vacío, detectará si el campo artista está vacío. Si artista esta vacío, se añadirá a la playlist el nombre y "desconocido".10. Si no esta vacío, se añadirá a la playlist el nombre y el artista.11. Fin	<p>Toda canción debe de tener nombre.</p> <p>Las canciones deben estar separadas por comas.</p> <p>El nombre y el artista deben estar separadas por "-".</p>

Guardar las listas en un archivo .csv

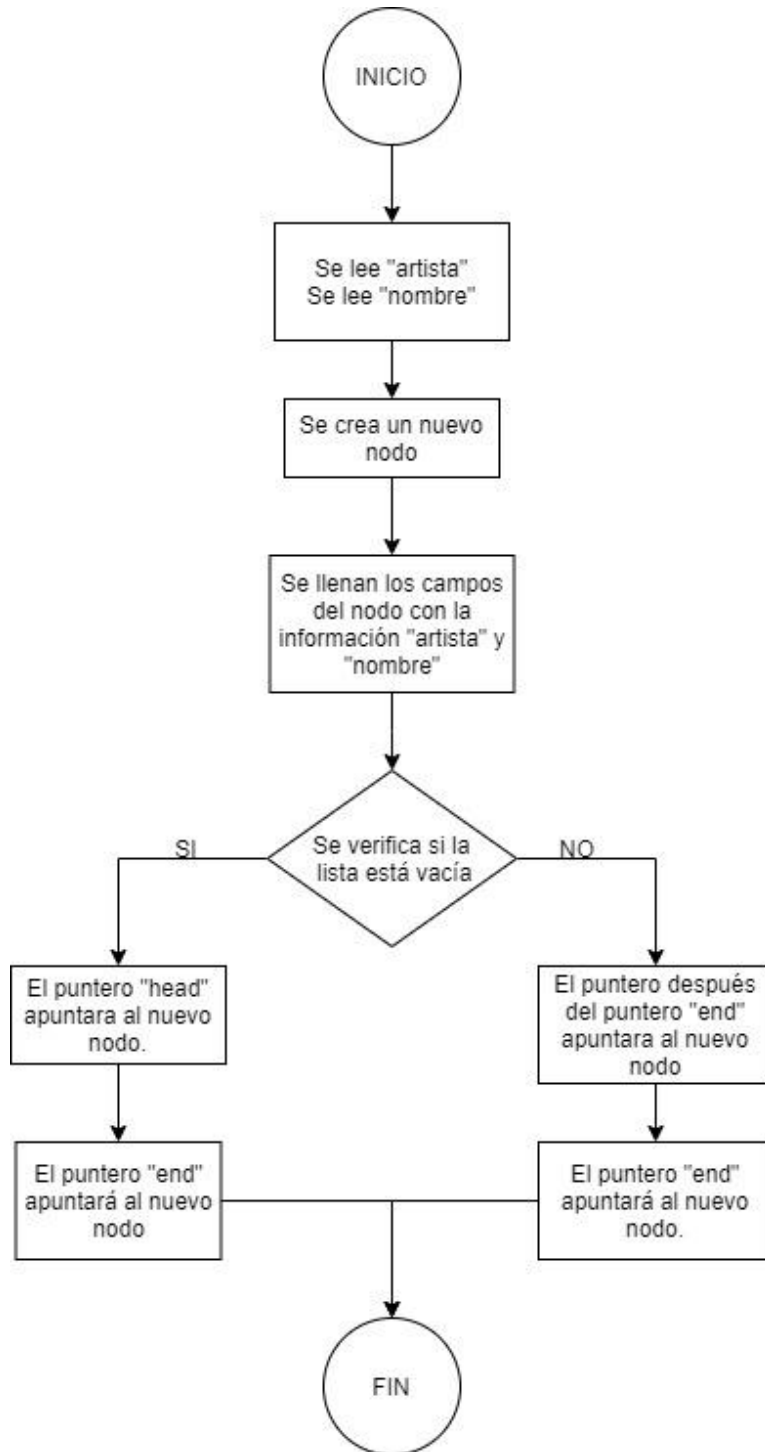
Entradas	Salidas	Procesos	Validaciones
<p>Lista simple</p> <p>Archivo csv con los datos originales.</p>	<p>Archivo .csv con las canciones separadas por comas y el nombre y el artista separados por "-".</p>	<ol style="list-style-type: none"> 1. Inicio 2. Se guardan la cantidad de columnas del archivo original. 3. Si la lista no esta vacía se crea un filtro para guardar archivos .csv. 4. Si se pudo abrir el explorador de archivos, se guarda el nombre de archivo. 5. Se crea una variable entera i para recorrer la lista. 6. Se crea una variable string para guardar los datos en una sola línea en el formato correcto. 7. Se crea una variable j para recorrer las columnas en el archivo. 8. Siempre que i sea menor que la cantidad de elementos de la lista se va a comparar si j es menor a la cantidad de columnas menos 1. Si es menor, la variable string guardará el nombre de canción y al artista en la posición i separados por un "-" con una coma al final. Si no, la variable string guardará el nombre de canción y al artista en la posición i separados por un "-" y agregará los caracteres para terminar una línea y crear una nueva. 9. Se aumentan los valores de i y j. 10. Cuando termine ciclo, se escribe un nuevo archivo, enviando como parámetro el nombre con el que se guardará al archivo y la variable string donde se guardaron los datos. 11. Fin 	<p>La lista no debe estar vacía.</p>

Ordenar las listas

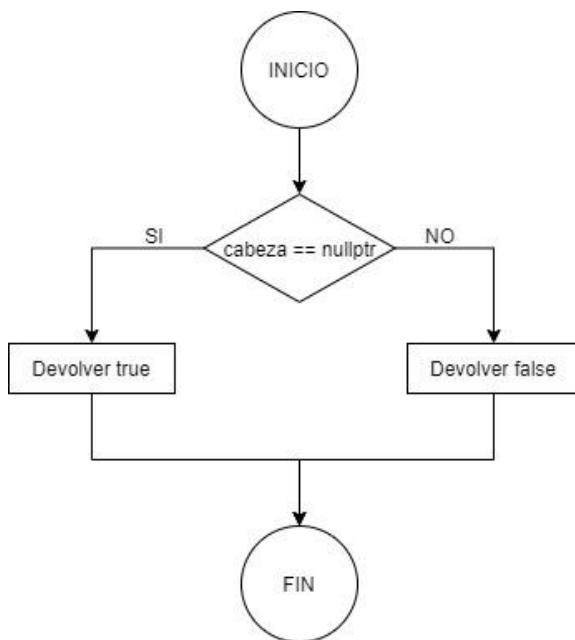
Entradas	Salidas	Procesos	Validaciones
Lista simple	Lista ordenada	<ol style="list-style-type: none">1. Inicio2. Se crea una variable int swapped que servirá para ver si se hizo algún cambio de posición.3. Se crean dos apuntadores de nodos, izquierda y derecha. El apuntador derecha apuntará al final de la lista.4. Se crea un ciclo do while. Se indica que swapped = 0 y el apuntador de nodos apuntará a la cabeza de la lista. Si al final del ciclo swapped = 0 significa que ya no se realizaron cambios por lo que la lista esta ordenada.5. Se crea un ciclo while. Si el nodo siguiente de izquierda es derecha se rompe el ciclo.6. Se verifica que opción de ordenamiento se escogió.7. Se verifican las condiciones necesarias. Si se cumplen las condiciones, se intercambia el valor de los nodos.8. Se indica que swapped = 19. Se indica que izquierda apuntará a al nodo siguiente de izquierda.10. Al final del ciclo do while, derecha apuntará a izquierda.11. Fin	<p>La lista no debe estar vacía.</p> <p>Se debe seleccionar una opción de ordenamiento.</p>

Diseño

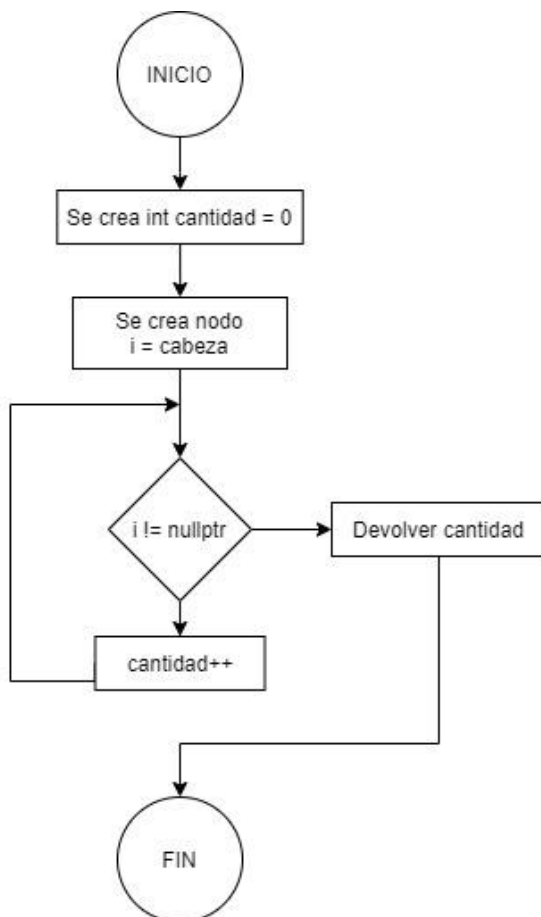
Añadir un nuevo nodo



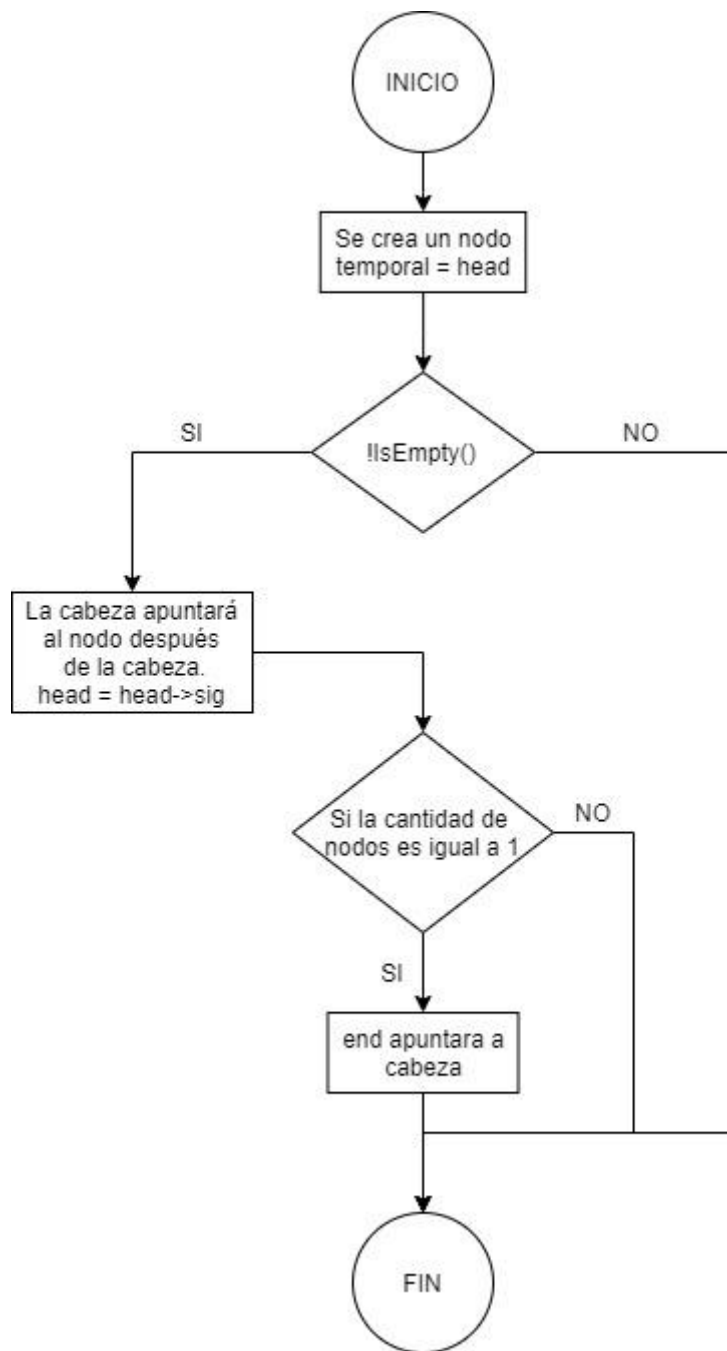
Verificar si esta vacía la lista



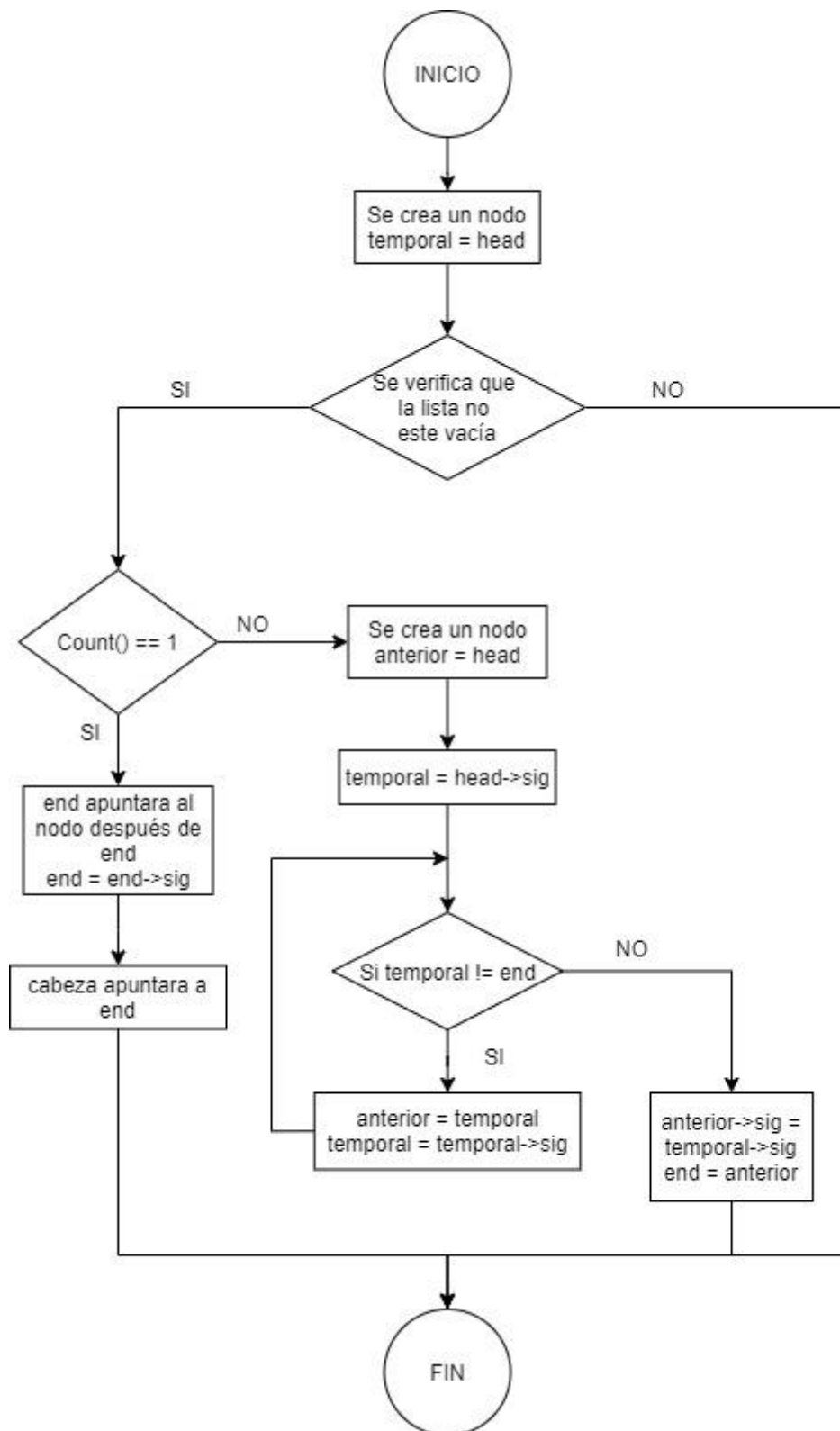
Contar la cantidad de elementos de la lista



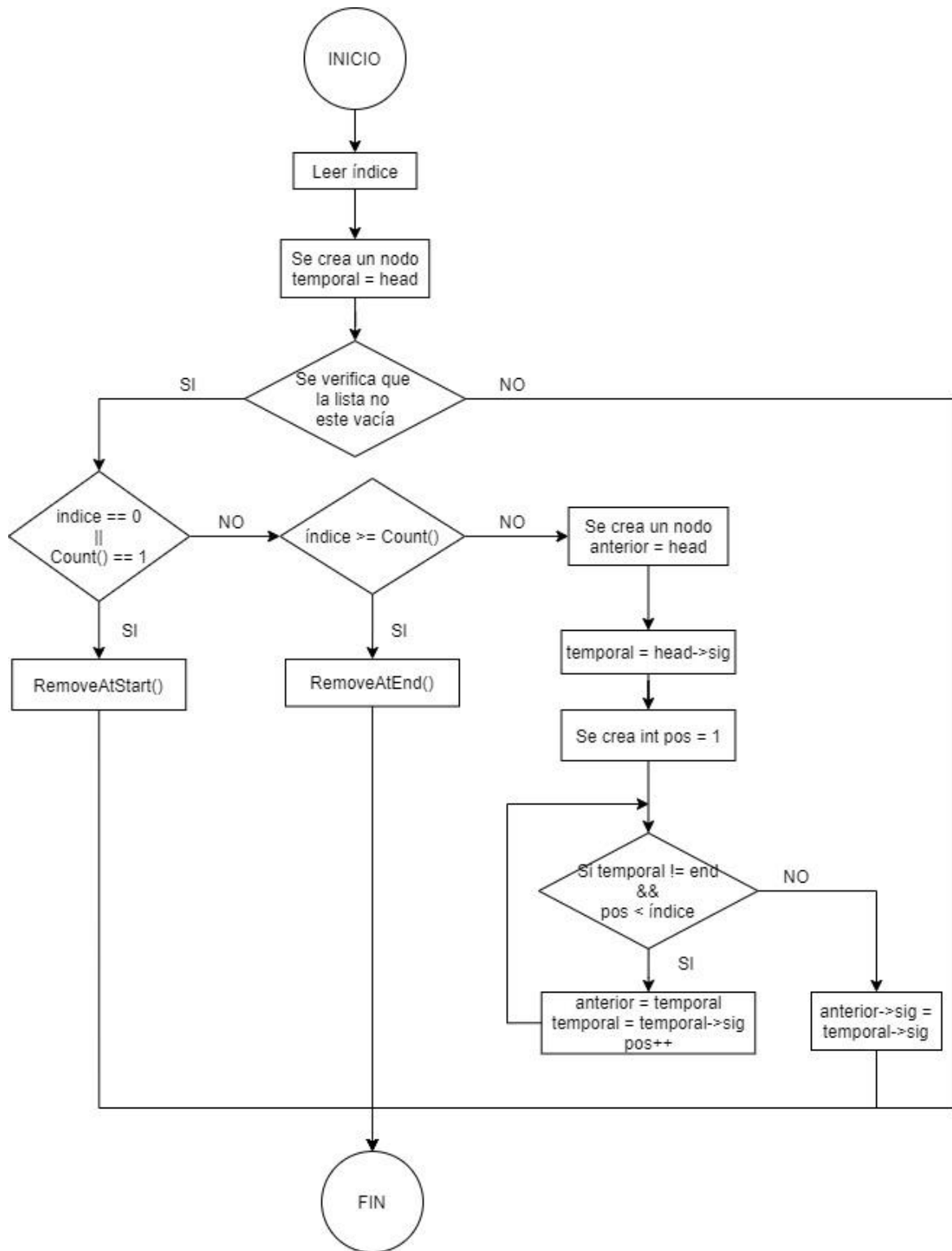
Remove un elemento al inicio de la lista



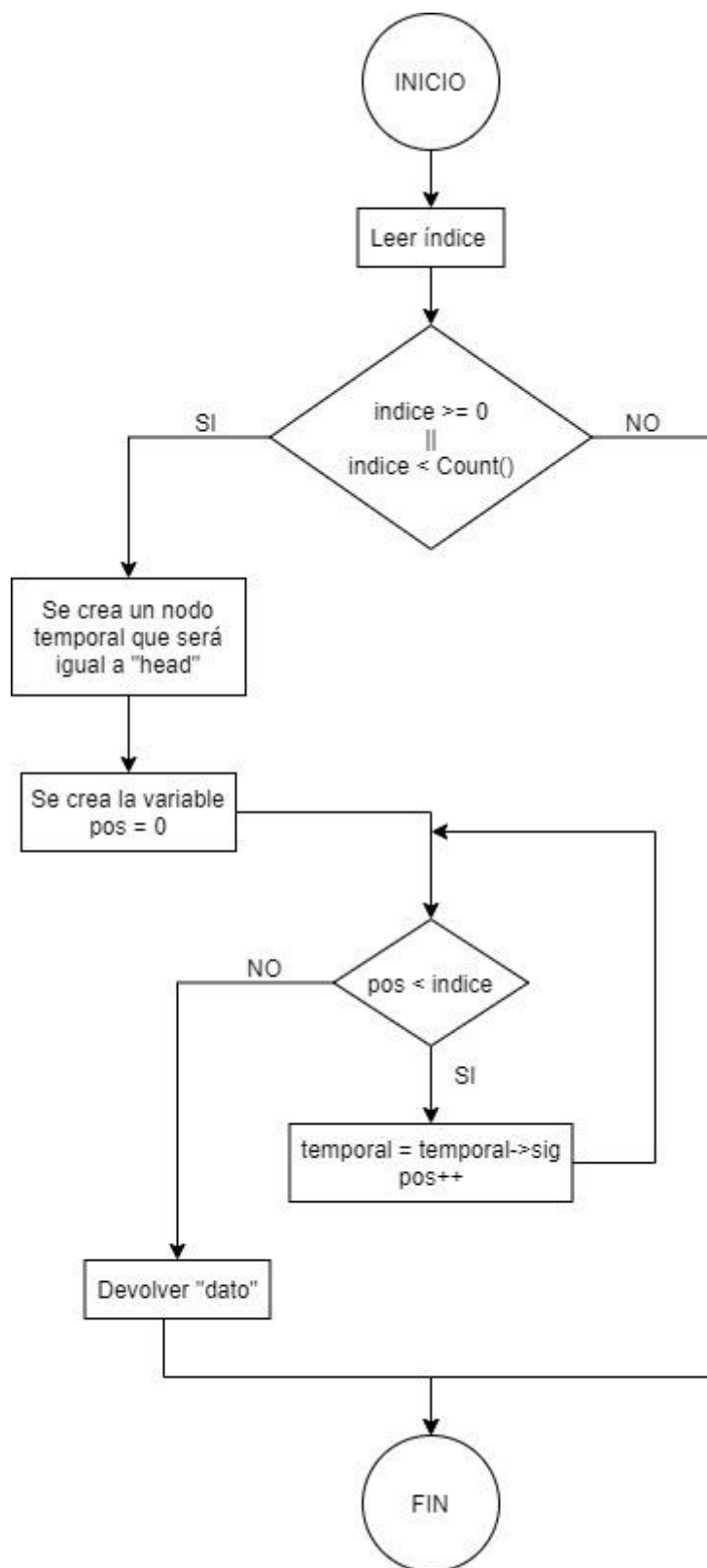
Remove un elemento al final de la lista



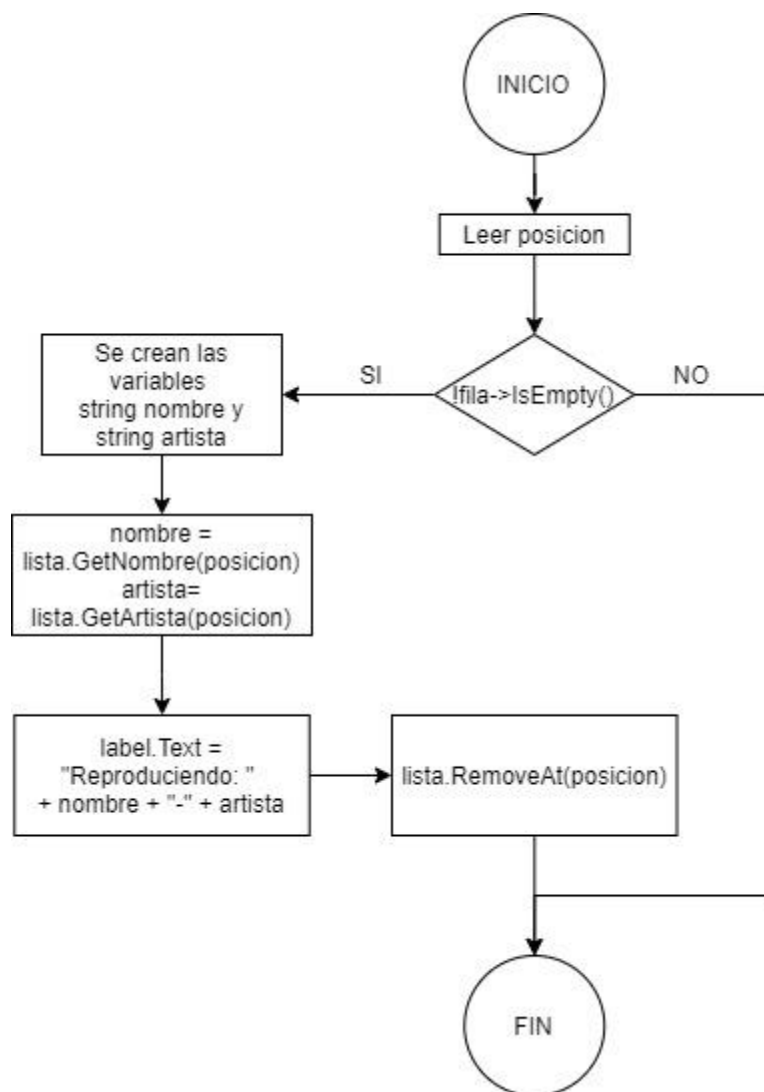
Remove en una posición específica



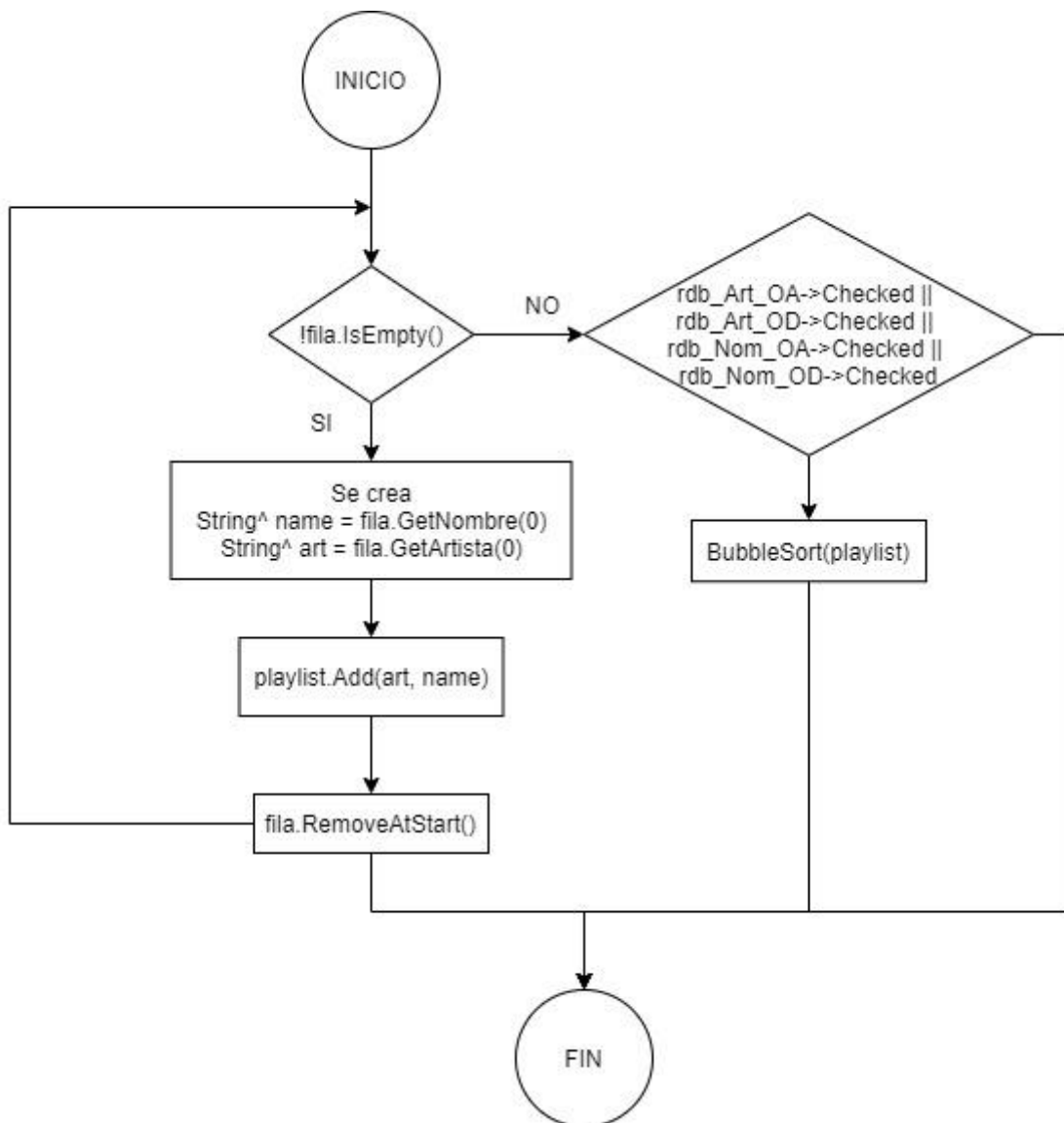
Obtener un dato en la posición indicada



Reproducir una canción



Sincronizar las listas



Conclusiones

- Las estructuras de datos dinámicas pueden modificar su cantidad de elementos durante su tiempo de ejecución.
- Dependiendo de que aplicación se esta creando, hay estructuras de datos que son más efectivas que otras.
- Los reproductores de música utilizan estructuras dinámicas, pues la cantidad de elementos que se pueden añadir depende completamente de cuanta memoria se disponga.

Recomendaciones

- Al llamar un método de búsqueda, ordenamiento o de extracción es necesario verificar que la lista no este vacía y que la posición indicada este en un rango aceptable. De no hacerlo, se puede obtener un error del tipo "NullReferenceException".
- Utilizar el elemento "ListBox" para imprimir las listas. Este elemento resulta más útil que un "DataGrid" o un "RichTextBox" pues permite manipular cada elemento individualmente.
- Si se va a trabajar con archivos, es esencial añadir un filtro para trabajar con el formato correcto.
- Al realizar búsquedas por internet, buscar los términos o palabras en ingles para obtener más resultados.

Referencias

- **System namespace:** Se utilizó crear tipos de datos `String^` para los nodos.
- **System.IO namespace:** Se utilizó para leer y exportar los archivos del tipo CSV.