

Estrategia de resolución:

Diagrama de estructuras de datos:

A través de esta imagen, se puede ver como se organizaron los datos en cada una de las listas.

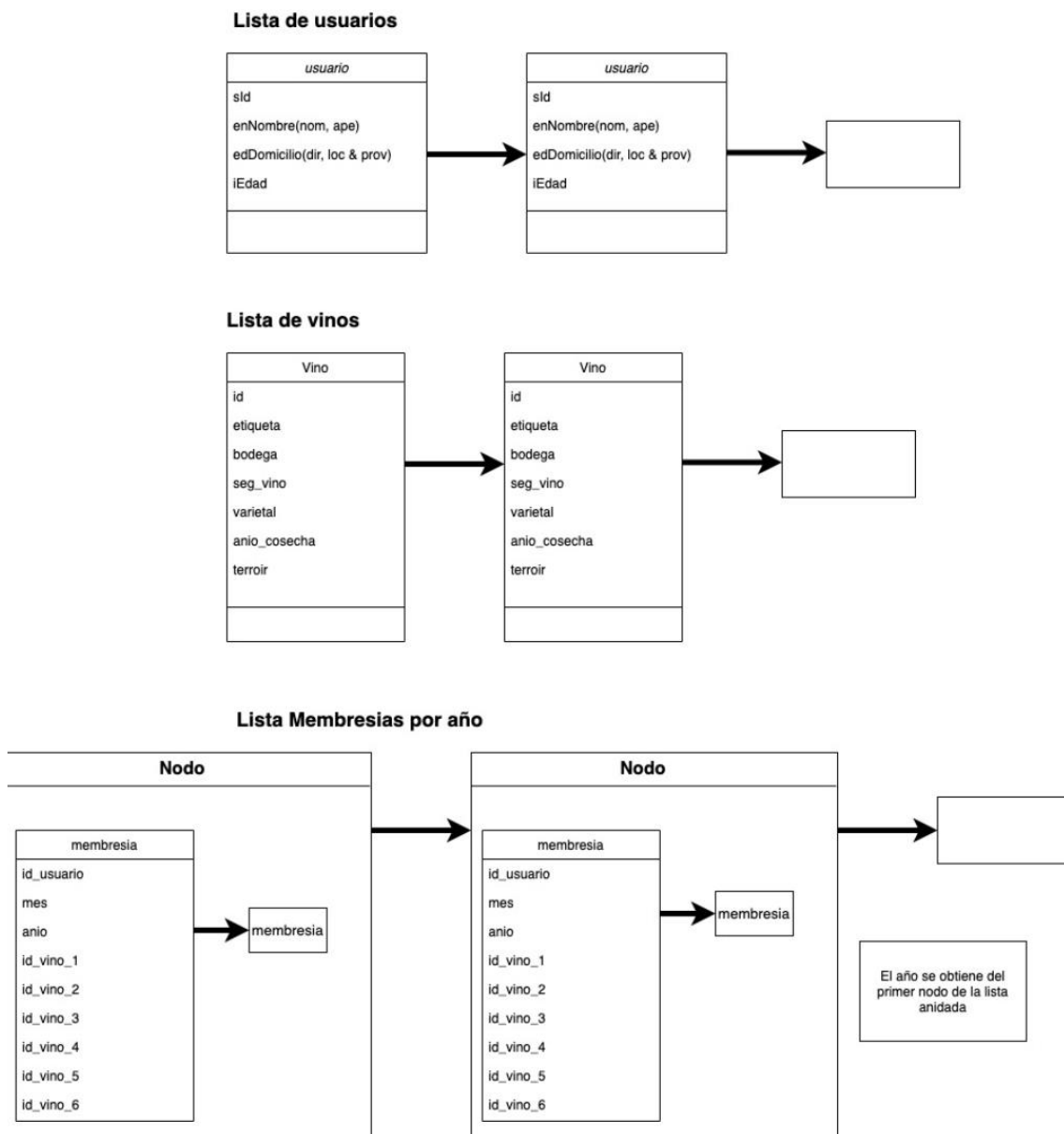
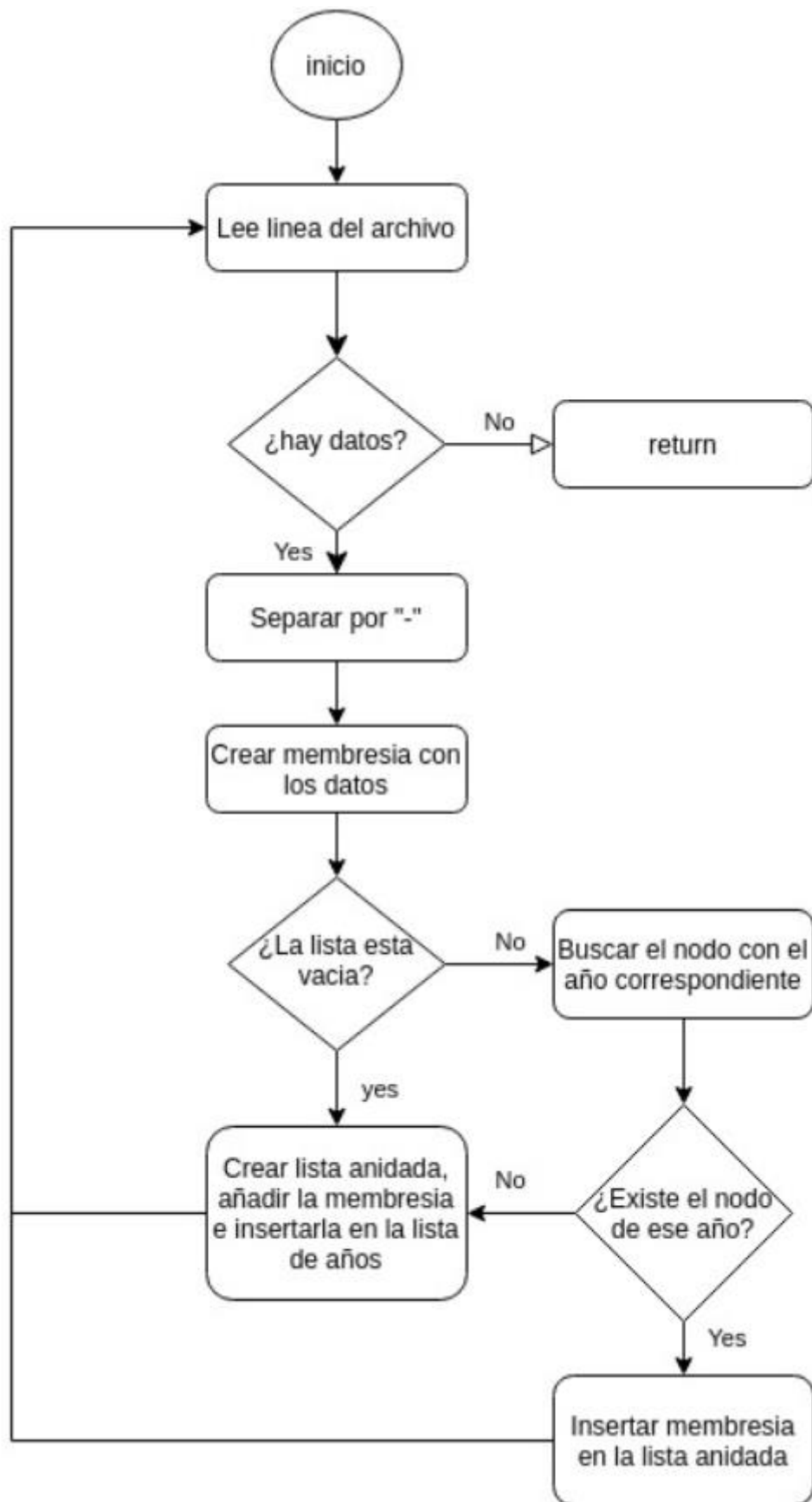


Diagrama de flujo de carga de membresías:

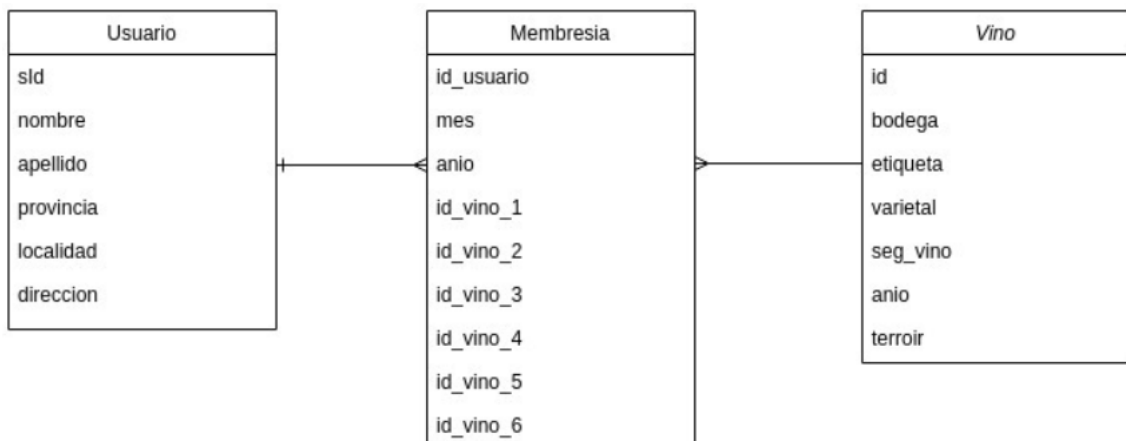
Muestra el algoritmo utilizado para cargar la lista que contiene las membresías.



Estructura a utilizar:

Luego de analizar la consigna en cuanto al trabajo a desarrollar se estableció que la mejor forma de llevar a cabo este desarrollo de una forma óptima, es utilizando principalmente el concepto de la estructura “Lista”, ya que esta es una estructura que nos permite acceder a un dato sin importar el orden o posición en el que este, a comparación de las estructuras “Pilas” o “Colas” que permiten acceder al último o primer dato. En principio se utilizara una lista conforme a su TDA.

Diagrama de relaciones:



División de tareas/cronograma:

Orden y Tareas principales:

- Definir TDA y primitivas
- Implementar listas
- Definir estructuras y cargar archivos de prueba
- Desarrollar las validaciones principales

Las tareas se dividirán equitativamente entre los integrantes utilizando una herramienta como GitHub para control de versiones, seguimiento de errores, trabajar en equipo con distintas ramas y para que los profesores puedan acceder más fácilmente al código fuente.

Minutas

Algunos cambios, tareas, observaciones, discusiones y funciones que fueron surgiendo y agregándose al proyecto con el tiempo para ver como resolvíamos ciertos problemas:

- Desarrollo de diagrama, definir TDAs.
- Creación de TDA lista.
- Nodo separado de lista ya que pensamos que no íbamos a necesitar utilizar pilas o colas. Cada nodo contiene como dato un puntero a otra lista que almacena por ejemplo las membresías.
- Definir membresía como struct.
- Lectura y carga de .txt a listas omitiendo datos repetidos.
- Ordenar e imprimir listas cargadas.
- Duda resuelta sobre métodos de ordenamiento como función de lista.
- Preparar los rankings para el sector de marketing según corresponda.
- Creíamos que el ranking de vinos sería igual al de bodegas pero una bodega puede tener más de un vino, entonces modificamos el enfoque.
- Ordenar ranking de vinos y bodegas por cantidad de mayor a menor y mostrar la cantidad total.
- Mostrar las elecciones de varietales de usuarios con sus datos (Id, nombre).
- Búsqueda de bugs y errores que se podrían dar en la carga por los usuarios (datos incompletos y caracteres inválidos, mal formato de data entry como saltos de línea, entre otros).
- Duda resuelta sobre que constituye “imprimir ranking de ultimo año”, últimos 12 meses desde la fecha actual o en el año en curso.
- Destrucciones de listas para mejor manejo de memoria.

- Uso de un menú para obtener un programa con mayor accesibilidad, amigable y comprensible.
- Verificación y validación de requisitos en el enunciado del trabajo práctico.
- Documentación y confección, tanto de manuales como de código (comentarios).