Materia: Algoritmos

Y Estructura

De Datos  
  
  
Trabajo Práctico de Integrador de la materia

Segundo Cuatrimestre 2021

**Índice**

* [Integrantes](#_Integrantes:)
* [Introducción](#_Introducción)
* [Compilación](#_Compilación)
* [Casos de error](#_Casos_en_de)
* [Manual técnico](#_Manual_técnico)
  + [TDA](#_TDA:)
  + [Librerías/Funciones](#_Librerías/Funciones:)
* [Manual de usuario](#_Manual_de_usuario)
* [Menú](#_Menú)
* [Ranking de vinos](#_Ranking_de_vinos)
* [Ranking de bodegas](#_Ranking_de_bodegas.)
* [Ranking de varietales](#_Ranking_de_varietales)
* [Salir](#_Salir)

# **Integrantes**

|  |  |
| --- | --- |
| Fernandez, Nahuel | 35992399 |
| GramejoRodriguez, Santiago | 40878531 |
| De Laforé, Pablo | 37750103 |
| Nordinelli, Franco | 37995141 |
| Rosales N. Alejandro | 41005121 |
| Quintana, Aaron | 41397796 |

# Introducción

El programa levanta, ordena y muestra de forma cómoda y sencilla, por medio de diferentes rankings, la información guardada en los archivos elecciones, catálogos y usuarios. Los ranking de vinos y bodegas se corresponden siempre con las elecciones hechas por los usuarios durante el último año cargado en el archivo elecciones.txt, en tanto el ranking de varietales muestra la predilección de diferentes vinos según el grado etario de los clientes.

# Compilación

Un IDE para la compilación y ejecución del programa escrito en lenguaje C++. Se recomienda el uso de CODEBLOCKS para poder abrirlo desde el ejecutable.

Por otra parte, los archivos eleciones.txt, catalogos.txt y usuarios.txt se deben mantener actualizados para que la información de entrada en el programa sea precisa. Además, se debe usar como delimitador un guión medio (-) entre las diferentes columnas de dichos textos.

# Casos en de error

* En caso de que cualquier dato en los .txt que se importan tengan datos con acentos o ñ, se mostrarán mal los resultados en algunas ocasiones (Depende de la configuración de la PC donde se esté ejecutando).
* En caso de que cualquier dato en los .txt tenga espacios entre el delimitador (Guion medio) y el dato puede que se muestren los espacios en los resultados, dejándolo menos amigable a la vista.

# Manual técnico

## TDA:

Los TDA solo contienen los datos correspondientes, sus gets y sets, constructor y destructor. Se informan a continuación los TDA del proyecto y sus datos:

### Cliente:

* IdUsuario
* Nombre
* Dirección
* edad

### ÍnfoMensual

* idUsuario
* mesSeleccion
* anioSeleccion
* idVino1
* idVino2
* idVino3
* idVino4
* idVino5
* idVino6

### Vino

* dVino
* etiqueta
* bodega
* segmento\_del\_vino
* varietal
* anioCosecha
* terroir

## 

## Librerías/Funciones/Procedimientos:

|  |  |
| --- | --- |
| **menu** | |
|  | |
| **Funciones** | |
| **Función** | **Descripción** |
| void IniciarMenu(); | Hace el llamado a todas las funciones elementales del programa y despliega el menú en la pantalla. |

|  |  |
| --- | --- |
| **FuncionesCargarArchivo** | |
| **Encargado:Nahuel Fernández** | |
| **Funciones** | |
| **Función** | **Descripción** |
| voidleerYpasarAusuarios(Lista clien); | Lee el archivo de usuarios.txt, transformando cada dato en una estructura de Cliente, y agrandándolos a la una lista que se envía por parámetro |
| voidleerYpasarAseleccion(Lista selec); | Lee el archivo de seleccionados.txt, transformando cada dato en una estructura de InfoMensual, y agrandándolos a la una lista que se envía por parámetro |
| voidleerYpasaraVinos(Lista vin); | Lee el archivo de seleccionados.txt, transformando cada dato en una estructura de Vino, y agrandándolos a la una lista que se envía por parámetro |
| Cliente pasarCharAStructConStringUsuario  (string texto); | Recibe por parámetro una cadena de texto , la cual la separa por delimitador “ -“ , para poder obtener cada atributo de la entidad Cliente y devolver la misma |
| Vino pasarCharAStructConStringVino  (string texto) | Recibe por parámetro una cadena de texto , la cual la separa por delimitador “ -“ , para poder obtener cada atributo de la entidad Vino y devolver la misma |
| infoMesualpasarCharAstructSeleccion(string texto) | Recibe por parámetro una cadena de texto , la cual la separa por delimitador “ -“ , para poder obtener cada atributo de la entidad InfoMensual  y devolver la misma |

|  |  |
| --- | --- |
| Lista | |
| **Encargado:Alejandro Rosales** | |
| **Nodo (ESTRUCTURA)** | |
| Dato | Tipo |
| Nro | int |
| dato | Puntero |
| siguiente | Nodo |
| **Lista (ESTRUCTURA)** | |
| tamanio | int |
| inicio | Nodo |
| **Funciones** | |
| **Función** | **Descripción** |
| Lista crearLista(); | Lista queda creado y con tamaño 0 y inicio null |
| voidagregarNodo(Lista lista,void\* dato); | Agrega nodo a la lista, calcula el nro del nuevo nodo en la lista y asigna el nuevo nodo como el siguiente al último nodo(Si la lista no está vacía) |
| voideliminarNodo(Lista lista,intnro); | Busca el nodo en la lista y lo elimina, recalcula el nro de los nodos restantes. Si no lo encuentra escribe error en pantalla. |
| voideliminarLista(Lista lista); | Elimina todos los nodos de la lista y la lista en si de memoria. |
| BoollistaVacia(Lista lista); | Verifica si la lista esta vacio |
| Nodo nodoSeleccionado(Lista list,intnro) | Devuelve un nodo según su nro, recorre toda la lista en busca de ese nro. |
| Nodo ultimoNodo(Lista lista); | Devuelve el ultimo nodo de la lista(El nodo cuyo siguiente es NULL) |
| Nodo anteriorNodo(Lista lista,Nodo nodo); | Devuelve el nodo cuyo siguiente es el nodo enviado por parámetro |
| voidmostrarLista(Lista lista); | Muestra todos los nodos, muestra: nro,direccionDato.  Utilizado para pruebas. |
| intultimoAnio(); | Devuelve el mayor anio en una lista de infoMensual. Recorre la lista recolectando el mayor año. |
| Ranking Vinos | |
| **Encargado:** GramejoRodriguez, Santiago /Nahuel Fernández (ordenamiento) | |
| **Estructuras:** | |
| **RankingVinosStruct** |  |
| **Dato** | **Tipo** |
| *cantidadSelecciones* | *Int* |
| *vino* | *Vino(estructura)* |
| **Funciones:** | |
| **Función** | **Descripción** |
| voidrankingPorVinos  (Lista seleccionados,Lista vinos); | Recibe por parámetro la lista de InfoMensual y la lista de Vinos, con dicha información, crea una lista ordenada, de mayor cantidad de vinos a menor, y la muestra por consola |
| RankingVinosbuscarVino  (Lista lista,intidVino); | Recibe por parámetro una lista de ranking, y un idVino, con el cual buscara si se encuentra en dicha lista, de ser así lo retorna, de lo contrario retorna Null |
| voidagregarVinos(Lista ranking, Vino vino) | Recibe por parámetro una lista de ranking, de no encontrar una estructura de ranking, la crea y la agrega, por el contrario si la encuentra suma una unidad al atributo de “cantidadSelecionados” |
| voidordenarListaAux(Lista lista, Lista ordenada) | Recibe por parámetros una lista cargada con rankigVinosStruct y una lista vacía, buscara el número mayor de “cantidadSeleciones” y lo guardara en la lista vacía, paso siguiente eliminara el nodo y el dato de la lista de rankigVinosStruct, asi sucesivamente hasta obtener una lista nueva ordenada y eliminar por completo la otra lista |
| voidmostrarListaRankigVino(Lista lista) | Recibe una lista de rankingVinosStruct y la imprime por consola |

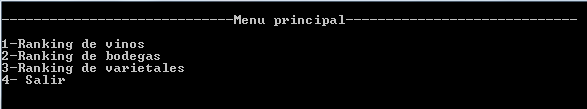
|  |  |
| --- | --- |
| Ranking Bodegas | |
| **Encargado:** Franco Nordinelli | |
| **Funciones:** | |
| **Función** | **Descripción** |
| void lstBodegas(Lista catalogo, Lista info) | Recibe por parámetro la lista de InfoMensual y la lista de Selección. Crea una lista ordenada con la cantidad de vinos escogidos por bodega |
| Bodega buscarBodega(Lista lista,char nombre[50]) | Recibe por parámetro la lista de bodegas y el nombre de una de ellas. Retornará null si el vino no está en lista, o el nombre de la bodega si está. |
| void agregarBodega(Lista bodega, Vino vino) | Recibe por parámetro una lista de bodegas y la struct Vino. La función hace un llamado a buscarBodega. Si esta función devuelve null, crea un nuevo nodo a la lista de bodegas; si se encuentra, suma una unidad al atributo cantidad de la struct Bodega. |
| void ordenarListaAuxBodega(Lista lista, Lista ordenada); | Recibe por parámetros una lista de bodegas (lista)y una lista vacía(ordenada). Busca la bodega con mayor cantidad de selecciones y lo guardara en la lista vacía. Se elimina el nodo de la primer lista. El proceso se repite hasta obtener una lista nueva y ordenada. |
| void mostrarListaBodega(Lista ordenada); | Hace llamada a ordenarListaAuxBodega… y muestra la lista ordenada por pantalla. |

|  |  |
| --- | --- |
| Ranking Varietales | |
| **Encargado:** Pablo de Laforé | |
| **Estructuras:** | |
| **RankingVarietalesStruct** |  |
| **Dato** | **Tipo** |
| *posicion* | *Int* |
| *cantidadSelecciones* | *Int* |
| **Funciones:** | |
| **Función** | **Descripción** |
| int traerEdadCliente(Lista clientes,int idBuscado) | Recibe como parámetros una lista de clientes y el id de un cliente. Devuelve la edad del cliente con ese id de la lista de usuarios. |
| Vino traerVino(Lista vinos,int idBuscado) | Recibe por parámetro la lista de de vinos y el id del vino buscado. Retorna la estructura con el id de vino buscado. |
| void agregarVarietal(Lista lista,char varietalSeleccionado[]) | Recibe por parámetro la lista del ranking desordenado y el nombre del varietal seleccionado para insertar o incrementar su contador de seleccionados según corresponda. |
| void rankingDeVarietales(Lista seleccionados, Lista clientes, Lista vinos); | Recibe por parámetro la lista de vinos seleccionados por los clientes, la lista de los clientes y la lista de los vinos, se encarga de llamar a las funciónes encargadas de: discriminar los grupos etarios para los distintos ranking (rankingPorRangoEtario), ordenar los ranking (ordernarListaAuxVarietales) y mostrarlos por pantalla(mostrarRanking). |
| Lista rankingPorRangoEtario(Lista seleccionados, Lista clientes,Lista vinos, int valorEtarioInf, int valorEtarioSup); | Recibe por parámetro la lista de vinos seleccionados por los clientes, la lista de los clientes, la lista de los vinos, el valor del rango etario inferior y superior y se encarga de discriminar por grupo etario las selecciones de los clientes, extraer la información de los vinos, y llamar a la funcion encargada del armado del ranking (agregarVarietal). |
| void mostrarRanking(Lista ranking); | Recibe por parámetro una lista de ranking ordenada para mostrarla por pantalla |
| Void ordernarListaAuxVarietales(Lista lista,  Lista ordenada); | Recibe por parámetro una lista con un ranking desordenado y una lista vacia para insertar los valores ordenados de una a la otra. |

# Manual de usuario

## Menú

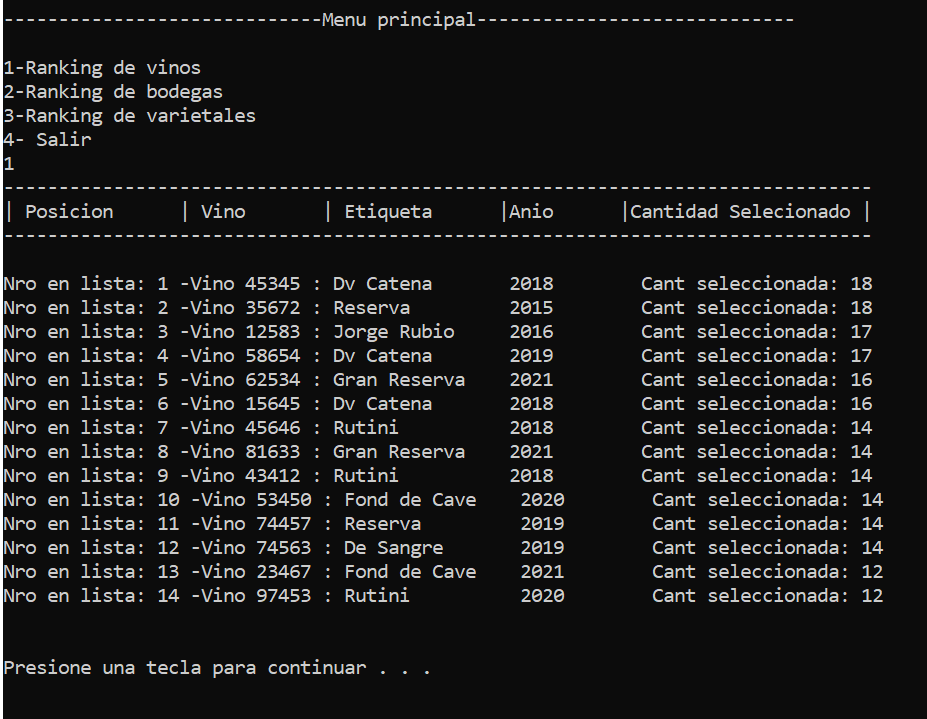
En el menú principal del programa se encuentran las opciones para visualizar los diferentes tipos de ranking. Para acceder a ellos, sólo se debe presionar mediante teclado la opción correspondiente.



Una vez seleccionado, se mostrará por pantalla el ranking junto con la información correspondiente.

## Ranking de vinos

La cantidad corresponde a los pedidos realizados el último año



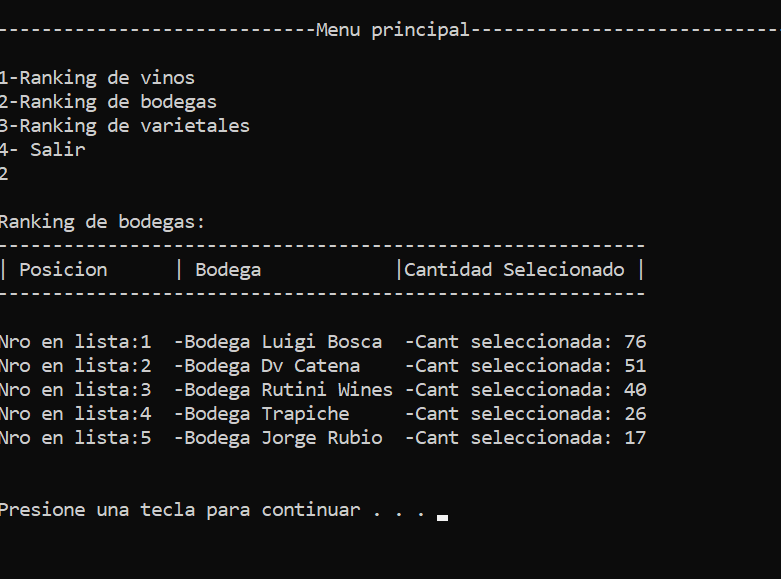
ID Vino

Opción Elegida

Anio de cosecha

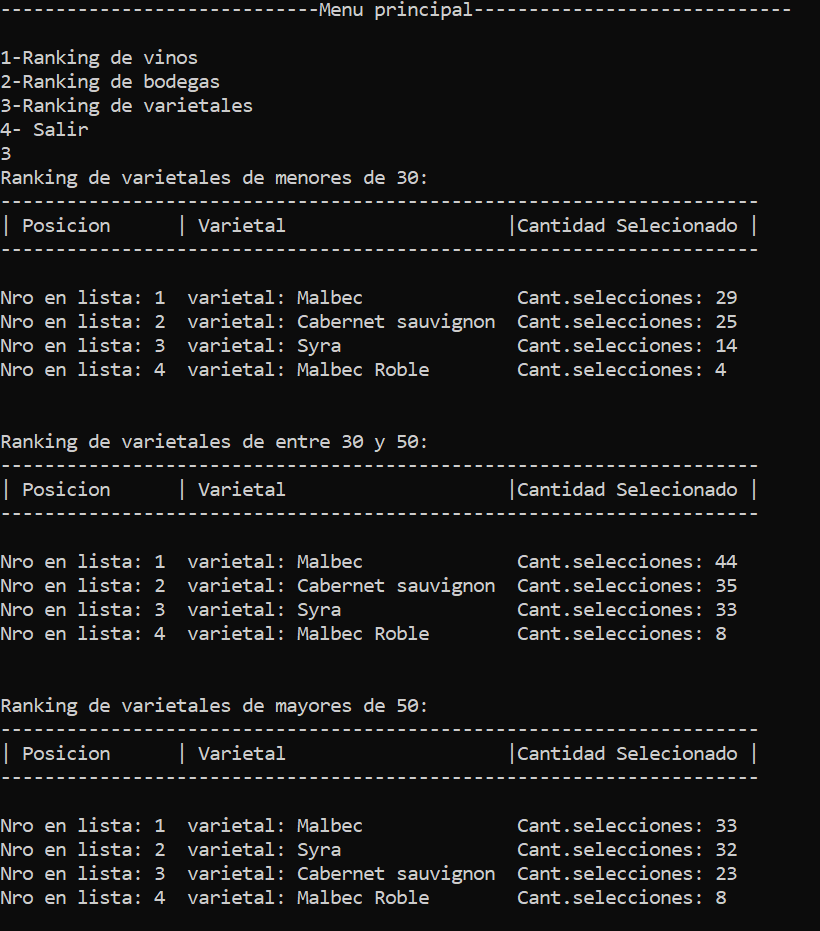
## Ranking de bodegas.

La cantidad corresponde a los pedidos realizados el último año



Opción Elegida

## Ranking de varietales

El ranking de varietales se corresponde a los gustos de los clientes de diferentes edades por distintos vino 

Opcion Elegida

## Salir

Para salir del programa sólo debe presionar la opción 4 en el menú.