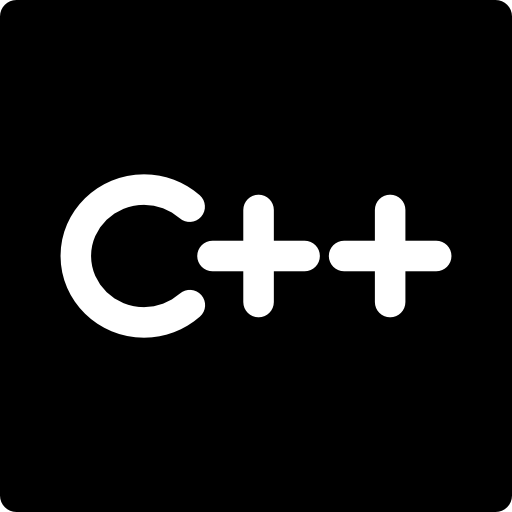
****

****

**TP algoritmo y estructura de datos**

**NOMBRE Y APELLIDO DE LOS INTEGRANTES:**

# Benitez franco

# Índice de Contenidos

Este es un trabajo practico donde se nos pidio resolver cierto problema en grupo, El trabajo se hizo en C++ donde utilizamos varias cosas que se nos brindo durante la cursada de “algoritmo y estructura de datos”.

***Definición general del proyecto de software***

El cliente nos vino con el siguiente problema : “Una empresa de franquicias tiene locales distribuidos por todo el país, más allá de los sistemas contables y administrativos que los maneja cada dueño de las franquicias, la marca ha decidido encarar un concurso interno entre sus franquiciados para levantar las ventas de sus productos. La franquicia ha decidido encarar un ranking de sus sucursales, eligiendo a la que más factura a nivel: nacional, y dentro de cada provincia. Para ello nos han pedido a nosotros, que somos el Depto de sistemas, el desarrollo de un programa que arme los listados de cada concurso y elabore el ranking mensual.

La información se recibirá en un archivo que contendrá el código de sucursal, la provincia, la cantidad de artículos vendidos, el monto mensual de la facturación y los centímetros cuadrados del local. Se pide elaborar:

Un listado con el orden de las sucursales que más facturaron a nivel nacional y en cada provincia, (incluir los totales por provincia).

Un ranking con las sucursales que mayor cantidad de artículos vendieron a nivel nacional y por provincia.

Un ranking de rendimiento donde se busca las sucursales que mejor rendimiento por metro cuadrado tienen, ordenando por el cociente de facturación/metros cuadrados del local.

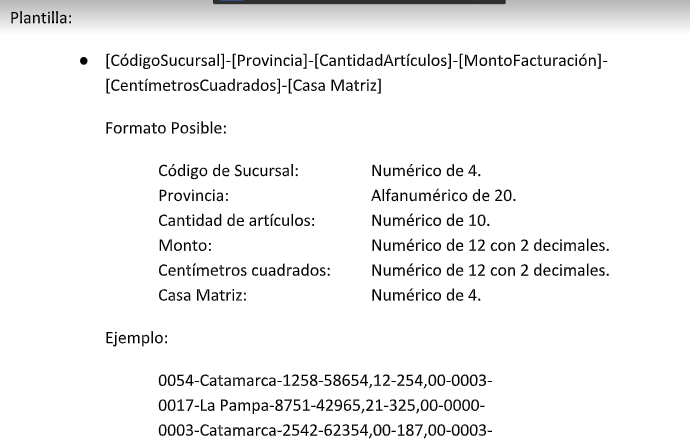
A su vez, las franquicias, pueden ser sucursales que responden a un mismo dueño u organización, en tal caso el campo Matriz contendrá el número de la sucursal “madre” a la que responden, en caso de ser “0000”, la sucursal será Matriz o no tendrá sucursales asociadas.

Si el local fuera una sucursal en los rankings anteriores participa como un local individual, pero en el ranking de rendimiento (Facturación/sobre Mts2) solo participan las matrices con sus sucursales sumadas.”

En este trabajo queremos que funcione lo mas eficientemente posible con ningún error que haga que el programa tenga errores y se caiga, que sea sencillo para cualquier persona utilizarlo que con solo leer el manual de usuario sea una experta en utilizarlo, manejarlo a la perfección y se capacite rápido con nuestro programa. Esto esta echo para empresas que quieran manejar sucursales, su facturación y información para esa empresa que le pueda ayudar a manejar cuales son sus mejores y peores sucursales a niveles de metros cuadrados, provincias en el ámbito de la facturación.

***Especificación de requerimientos del proyecto***:

Un requerimiento principal para que funcione correctamente este programa es que para que funcione bien cuando ingresan un .txt con los datos ordenado de la forma que se nos pidió funcione bien (se refiere al .txt con toda la información de las sucursales ), si eso no entra como se nos mostro a nosotros no funcionara, ese es el requerimiento principal del programa, a continuación un ejemplo:



Como se ve en cada planilla entra un código de sucursal, Provincia, cantidad de artículos, monto, centímetros cuadrados y casa Matriz. Es así como tiene que entrar, separados por un “-” uno debajo de otro.

Requisitos funcionales de este sistema son: dar a conocer el ranking de la sucursales con facturación mas alta, dar a conocer el ranking de la sucursales con facturación mas alta por provincia, dar a conocer el ranking de la sucursales con facturación mas alta y dar a conocer el ranking de los artículos vendidos, y la lista de las sucursales así por como se ingresaron al sistema por el .txt.

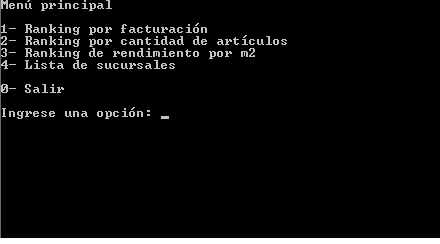
Este trabajo fue hecho por 5 integrantes, este proyecto fue echo totalmente de 0 dando muchísima chance de darse una actualización y que tenga un alcance mucho más lejos así como meterse más en temas como en que artículos son y no de un numero de artículos en general

***Procedimientos de instalación y prueba:***

Este programa no se va a instalas solamente con compilar va a estar funcionando perfectamente. Este programa va a necesitar un archivo .txt con datos del tipo String y enteros que van a manejarse como estructuras y esa estructuras estarán en una lista que se va a usar para dar diferentes resultados que quiera el cliente sobre las sucursales.

#### Arquitectura del sistema

El sistema utiliza TDA para tener todo de forma ordenada y poder acceder a todos los módulos de forma más cómoda y ordenada mente sin necesidad de que sea un código espagueti y nos perdamos en cada momento. Utiliza un tda para Sucursales y otro TDA para ListaSucursales. En este software en el main automáticamente cuando compilamos automáticamente lee el .txt y se genera una lista de sucursales de forma ordenada y todo va a sus respectivas variables, luego se abre un menú con las siguientes opciones a elegir:



El funciones principales son: rankingFacturacion(), rankingFacturacionPorProvincia(), rankingArticulos(); rankingArticulosPorProvincia(), rankingRendimiento(), mostrar\_lista(). Describimos una por una:

**rankingFacturacion():**

**Descripción general y propósito**: Esta procedimiento creo una lista con los rankings de facturación ordenados de todas las sucursales y las muestra.

**Responsabilidad y restricciones:** No puede excluir a ninguna sucursal y si alguna sucursal no tiene facturación el programa cae.

**Dependencias**: Esto depende de otras funciones tales como “crearLista()” ,”reOrdenar()” ,”mostrar\_Lista()”,”eliminarLista()”.

**rankingFacturacionPorProvincia():**

**Descripción general y propósito**: Esta procedimiento creo una lista con los rankings de facturación ordenados Por provincia de todas las sucursales y las muestra.

**Responsabilidad y restricciones:** No puede excluir a ninguna sucursal, o provincia y si alguna sucursal no tiene facturación el programa cae.

**Dependencias**: Esto depende de un cursor para movernos en las listas y otras funciones tales como “crear”(crea una sucursal que usamos para guardar datos),“crearLista()”(la utilizamos para crear una lista para guardar los datos), ”copiarLista()”(esta función la utilizamos para devolver una lista igual al puntero de lista que entra por parámetro), ”primero”(devuelve el primer puntero de la lista que lo usamos de referencia para el cursor) ,”obenerDato()”(obtengo la sucursal en ese cursor que entra por parametro) , “adicionarFinal()” (adiciona al final de la lista), “eliminarNodo()”(eliminar el nodo),”sigiente()”(accedemos al siguiente puesto de la lista),”reOrdenar()”(reordena por facturación), ”mostrar\_Lista()” ,”eliminarLista()”.

**rankingArticulos():**

**Descripción general y propósito**: Esta procedimiento crea una lista con los rankings de artículos ordenados de todas las sucursales y las muestra.

**Responsabilidad y restricciones:** No puede excluir a ninguna sucursal, o provincia y si alguna sucursal no tiene artículos el programa cae.

**Dependencias**:Esto depende de funciones tales como “crearLista()”(la utilizamos para crear una lista para guardar los datos), ”copiarLista()”(esta función la utilizamos para devolver una lista igual al puntero de lista que entra por parámetro),”reOrdenar()”(reordena por articulos), ”mostrar\_Lista()” ,”eliminarLista()”.

**rankingArticulosPorProvincia():**

**Descripción general y propósito**: Esta procedimiento creo una lista con los ranking de artículos ordenados (de mayor a menor) Por provincia de todas las sucursales y las muestra.

**Responsabilidad y restricciones:** No puede excluir a ninguna sucursal, o provincia y si alguna sucursal no tiene artículos el programa cae.

**Dependencias**: Esto depende de un cursor para movernos en las listas y otras funciones tales como “crear”(crea una sucursal que usamos para guardar datos), “crearLista()” (la utilizamos para crear una lista para guardar los datos), ”copiarLista()” (esta función la utilizamos para devolver una lista igual al puntero de lista que entra por parámetro), ”primero”(devuelve el primer puntero de la lista que lo usamos de referencia para el cursor) ,”obenerDato()”(obtengo la sucursal en ese cursor que entra por parametro) , “adicionarFinal()” (adiciona al final de la lista), “eliminarNodo()”(eliminar el nodo), ”sigiente()” (accedemos al siguiente puesto de la lista),”reOrdenar()”(reordena por facturacion), ”mostrar\_Lista()” ,”eliminarLista()”.

**rankingRendimiento():**

**Descripción general y propósito**: Este procedimiento muestra una lista de sucursales ordenadas y mostrando su rendimiento que está dado por sus metros cuadrados y su facturación.

**Responsabilidad y restricciones:** No puede excluir a ninguna sucursal, o provincia y si alguna sucursal no tiene productos o metros cuadros el programa cae.

**Dependencias**: Esto depende de un cursor para movernos en las listas y otras funciones tales como “crear”(crea una sucursal que usamos para guardar datos), “crearLista()”(la utilizamos para crear una lista para guardar los datos), ”copiarLista()” (esta función la utilizamos para devolver una lista igual al puntero de lista que entra por parámetro), ”adicionarFinal()”(adiciona al final de la lista),”primero”(devuelve el primer puntero de la lista que lo usamos de referencia para el cursor) ,”obenerDato()” (obtengo la sucursal en ese cursor que entra por parámetro) , “adicionarFinal()” (adiciona al final de la lista), “eliminarNodo”(eliminar el nodo),”colocarDato()”(coloca el dato donde nosotros queramos ),”sigiente”(accedemos al siguiente puesto de la lista), ”reOrdenar” (reordena por facturación), ”mostrar\_Lista” , ”eliminarLista”, ”destruir” (destruye una sucursal que le pongamos el puntero por parámetro).

**mostrar\_Lista ():**

**Descripción general y propósito**:mustra una lista que le mandemos por parámetro el puntero

**Responsabilidad y restricciones:,**No puede excluir a ninguna sucursal y muestra toda la lista de principio a fin.

**Dependencias**:Esto depende de otras funciones tales como “crear()”(crea una sucursal),”primero()”(devuelve el primero de la lista),”obtenerDato()”(obtengo un dato de la lista) ,”eliminarLista()”,”sigiente()”(accedemos al siguiente puesto de la lista).