Universidad Tecnológica Nacional - FRBA

Trabajo Practico Final

Algoritmos y Estructura de datos

K1028

Santiago Illi, Legajo: 156.186-8

Martin González, Legajo: 156.242-3

Matías Verón Peralta, Legajo: 156.361-0

Maximiliano Avalle, Legajo: 155.819-5

Sol Ottaviani, Legajo: 155.830-4

# Esquema de solución y subproblemas:

Se planteó en primer lugar pasar el archivo de cobranzas a una lista para asi poder tener los datos ordenados por número de cliente. Luego se utilizó otra lista para poder guardar los datos actualizados de forma eficiente y de fácil manejo. Se eligió utilizar un vector de 5 elementos compuesto por punteros a listas para poder mostrar los pagos excedentes e insuficientes por punto de cobro.

**Inicio**

**Abrir los archivos Cobranzas y Maestro**

**Pasar el archivo Cobranzas a una lista**

**Aparear la Lista de cobranzas con el archivo maestro, guardando los datos en una nueva lista (Lista Actualizada)**

**Guardar Datos de lista actualizada en un ArchivoMaestroActualizado**

**Guardar los datos de los pagos excedidos e insuficientes por punto de cobro en un arreglo de punteros a listas**

**Mostrar el Array**

**Mostar los clientes con mas de 5 cuotas impagas**

**Cerrar Archivos Maestro, Cobranzas y Maestro Actualizado**

**Fin**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** *Se asume que no se pagan cuotas que ya se encuentran saldadas. No se toma en cuenta el caso para los puntos de cobro en el cual la cuota es pagada con el monto justo, es decir que no se listara con ninguna leyenda ese tipo de caso. Se asume que las cuotas no tienen valores negativos, así como los pagos.*

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## Diseño del Algoritmo en seudocódigo (Procedimientos generales):

**Inicio**

**Abrir Archivos (ArchivoCobranzas, ArchivoMaestro, ArchivoMaestroActualizado)**

**ArchivoALista (ArchivoCobranzas, ListaCobranzas)** //*OrdenadoPorNroCliente*

**ApareoYPuntoDeCobro (ArchivoMaestro, ListaCobranzas, PuntoDeCobro)** //*Guarda los //datos actualizados en la lista, y los pagos excententes e insufientes en el array PuntoDeCobro*

**ListaAArchivo (ListaActualizada, ArchivoMaestroActualizado)** //*Guarda los datos //actualizados en el Archivo*

**MostrarPagosExc/Insuf (PuntoDeCobro)** //*Muestra Pagos excedidos o insuficientes por PuntoDeCobro*

**MostrarClientesImpagas (ListaActualizada)** //*Muestra clientes con más de 5 impagas*

**CerrarArchivos (ArchivoMaestro, ArchivoCobranzas, ArchivoMaestroActualizado)**

**Fin**

## Division de tareas:

* Generadores de archivos [Cobranzas & Maestro]: *Maximiliano & Sol*
* Lectura de archivos [Cobranzas, Maestro & Actualizado]: *Maximiliano & Sol*
* Main principal: *Santiago, Matias & Martin*
* Pruebas hechas al software: *Todo el grupo*
* Documentación: *Matias, Santiago & Martin*

## Manual del usuario:

Para que el programa funcione primero se deben generar dos archivos (un ArchivoMaestro.dat y otro ArchivoCobranzas.dat) cada uno con los generadores para cada archivo particularmente. Para hacerlo abrir cada respectivo generador e ingresar los datos en el orden que el programa los requiere, para así tener los dos archivos en el formato necesario para que puedan ser utilizados por el programa que actualizara el archivo maestro. Esta es la única ocasión en la cual el usuario deberá ingresar datos.

Luego de tener los dos archivos generados, colocarlos en el directorio en el cual se encuentra el programa de la Actualización del Maestro a partir de las novedades y correr el programa. En pantalla aparecerán los clientes que realizaron un pago excedente o insuficiente con una leyenda indicando cual es el caso y que cliente lo realizo. Debajo se listaran los clientes que tiene una cantidad de cuotas impagas mayor a 5 ordenado de mayor a menor por la cantidad de cuotas impagas.

Lecciones aprendidas:

La mayor lección aprendida es que cambia drásticamente en desarrollar un programa o pensarlo en lápiz y papel donde se pueden omitir varios errores de sintaxis mientras que la idea general y conceptos quedan más que claros, mientras que en una IDE o editores de texto a la hora de compilar son rígidos y más aún en C++ que es un lenguaje fuertemente tipado, a la hora de pasar parámetros o de manejarse con listas se nos presentaron los mayores inconvenientes que tal vez en otros lenguajes o en el seudocódigo no se hacen tan presentes.

A lo largo de este proceso fuimos buscando posibles soluciones en la web, y nos dimos cuenta que es la herramienta más útil que hay a la hora de programar. La mejor lección que nos llevamos de hacer este trabajo práctico es que siempre podemos encontrar una respuesta a nuestro problema revisando foros, manuales del lenguaje y apuntes de la materia.