



TRABAJO PRACTICO I – LISTADO ESPECIFICO – LISTADO GENERICO
Version 2.0 (Agosto 2024)

1. Desarrolle un simple programa para el manejo de una lista de clientes, en la cual el usuario pueda guardar y recuperar fácilmente los siguientes datos:

- Apellido
- Nombre
- Código de Cliente (número entero de 4 dígitos, mayor que cero)

Ejemplo de datos que pueden cargarse en cada registro del archivo de clientes:

#Registro	Apellido	Nombre	Código de Cliente
1	Pérez	Juan	4125
2	Suárez	Enrique José	1385
3	Gómez	Juana	128

Teniendo en cuenta las siguientes premisas:

- Utilice la estructura de datos que considere apropiada para guardar y recuperar estos datos
- No puede utilizar ningún tipo de base de datos, este es un programa tradicional basado en archivos
- El programa es un simple ABM (altas, bajas, modificaciones) sobre los datos de los clientes
- No es necesario realizar una interfaz gráfica de usuario, puede ser un programa de consola
- Utilice el lenguaje de programación y sistema operativo que desee
- Las altas, bajas y modificaciones (ABM) **se hacen por posición física de registro, no maneja clave** (el código de cliente, apellido y nombre pueden repetirse)

2. Desarrolle un nuevo programa para el manejo de esta lista de clientes, teniendo en cuenta que el campo Código de Cliente es la clave primaria del archivo. Utilice un algoritmo de hashing apropiado para su implementación. Se supone que no hay más de 700 clientes. Ahora el ABM se hace por código de cliente, que no admite repeticiones.

3. Desarrolle un programa para el manejo de cualquier tipo de archivo, en donde el usuario pueda guardar y recuperar fácilmente los datos que desee del archivo. El usuario del programa definirá la estructura del archivo. El programa tendrá dos opciones:

1. Definición de la estructura del archivo:

- a. El programa podrá solicitarle al usuario la cantidad de ítems (o campos) que tendrá la lista de clientes
- b. Para cada ítem (o campo) el programa solicitará el nombre del campo y la cantidad de caracteres como máximo que tendrá dicho



- ítem. Para facilitar el desarrollo, se asumirá que todos los ítems definidos por el usuario serán de tipo carácter.
- c. De alguna forma, el usuario le indicará al programa cuando ha terminado de definir la estructura de la lista de clientes
 - d. Una vez definida la estructura del archivo, el programa utilizará la misma para realizar el ABM
 - e. Una vez definida la estructura del archivo, el usuario ya no tendrá posibilidades de cambiar la estructura.
2. Utilización del archivo, se permitirá hacer Altas, Bajas y Modificaciones sobre los registros del archivo, teniendo en cuenta la estructura previamente definida.

Puede optar por hacer cualquiera de las siguientes 3 variantes de este programa:

- a) Definir estructura sin indicar clave primaria y luego hacer ABM utilizando #Registro (puede tener una opción para agregar registros vacíos al final del archivo o bien, luego de definir la estructura del archivo, el usuario indica la cantidad de registros a crear vacíos en el archivo). Se debe validar que el número de registro ingresado para hacer un cambio o dar de baja, sea un registro válido. Las bajas, obviamente, son lógicas.
- b) Definir estructura indicando campo que será clave primaria (se usará un único campo como clave primaria). Se indicará la cantidad de registros del archivo (N) y se implementará el archivo utilizando un algoritmo de hashing. Luego hacer ABM usando el campo clave primaria, no permitir duplicados de clave primaria.
- c) Definir estructura indicando campo que será clave primaria (se usará un único campo como clave primaria). Se implementará el archivo utilizando un algoritmo de indexación (se recomienda utilizar índice denso de un único nivel). Luego hacer ABM usando el campo clave primaria, no permitir duplicados de clave primaria. Se admite usar índice en memoria principal y luego guardarlo en memoria secundaria.

El programa puede asumir el tamaño que desee en cuanto al bloque físico del disco, también puede asumir ciertas limitaciones en la implementación: por ejemplo un límite en cuanto a la cantidad de registros o claves que soporta el programa, la implementación puede usar un único archivo o requerir de varios archivos, puede hacer un programa bajo una arquitectura cliente-servidor o bien file-server, puede hacer pantallas interactivas o bien puede hacer una interfaz tipo shell o línea de comandos, etc.

En todos los casos, debe documentar: la organización de archivo a utilizar y la forma de uso del programa.