

# INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

---

Departamento de Electrónica, Sistemas e Informática

INGENIERÍA EN SISTEMAS COMPUTACIONALES



## PROGRAMACIÓN CON MEMORIA DINÁMICA TAREA 2. MEMORIA DINÁMICA Y ARCHIVOS

Autor: Méndez Méndez, Sergio Alejandro

Presentación: 5 pts.  
Funcionalidad: 55 pts.  
Pruebas: 20 pts.

12 de junio de 2018. Tlaquepaque, Jalisco,

- Falta describir las pruebas (escenario, y resultados de la experimentación).
- Tienes algunos detalles en el tema de funcionalidad, verifica lo que se debe imprimir cuando consultas una cuenta.

## Objetivo de la actividad

El objetivo de la tarea es que el alumno aplique los conocimientos y habilidades adquiridos en el tema de manejo de memoria dinámica y archivos utilizando el lenguaje ANSI C.

## Descripción del problema

Ahora tienes los conocimientos para enfrentarte a un nuevo proyecto llamado **MyDB**. En este proyecto vas a recrear una parte de un sistema de transacciones bancarias. Para esto vas a requerir del uso de:

- Estructuras
- Funciones y paso de parámetros
- Apuntadores
- Memoria Dinámica
- Archivos binarios

El sistema **MyDB** al ser ejecutado deberá mostrar al usuario una interfaz con el siguiente menú principal:

<< Sistema MyDB >>

1. Clientes
2. Cuentas
3. Transacciones
4. Salir

El sistema **MyDB** debe realizar automáticamente, las siguientes operaciones:

- A) Si el sistema **MyDB** se ejecutó por primera vez, este deberá crear tres archivos binarios: **clientes.dat**, **cuentas.dat** y **transacciones.dat**. Para esto el sistema debe solicitar al usuario indicar la **ruta de acceso** (por ejemplo, c:\\carpeta\\) en donde se desea crear los archivos (esta información deberá ser almacenada en un archivo de texto llamado **mydb.sys**).

### Clientes

La opción **Clientes** debe mostrar un submenú con las siguientes opciones:

- |                           |   |
|---------------------------|---|
| - <b>Nuevo</b> cliente    | Registra los datos de un nuevo cliente del banco  |
| - <b>Buscar</b> cliente   | Permite consultar la información de un usuario a partir de su id_cliente.   |
| - <b>Eliminar</b> cliente | Si existe, elimina un usuario deseado del sistema. Esto implica que deben Borrarse las cuentas registradas a nombre del usuario |

(utilice id\_usuario para buscar).

- **Imprimir** clientes      Imprime la información de todos los clientes registrados en el sistema.

La información que el sistema requiere almacenar sobre cada cliente es la siguiente:

- Id\_usuario (es un número entero que se genera de manera consecutiva, clave única)
- Nombre
- Apellido materno
- Apellido paterno
- Fecha de nacimiento (tipo de dato estructurado: dd/mm/aaaa)

Para gestionar la información de los clientes, defina un tipo de dato estructurado llamado **Usuario**, utilice instancias de Usuario para capturar la información desde el teclado y posteriormente guardarlo en el archivo usuario.dat.

Un ejemplo del contenido que se estará almacenando en el archivo **usuario.dat** es el siguiente:

id_usuario	nombre	apellido_paterno	apellido_materno	fecha_nacimiento
1	Ricardo	Perez	Perez	{3,10, 2010}
2	Luis	Rodriguez	Mejía	{2,7, 2005}
3	Gabriela	Martínez	Aguilar	{7,11,2015}

**Importante:** considere que no pueden existir datos **id\_usuario** repetidos y que es un valor autonúmerico. Adicionalmente, recuerde que al inicio el archivo no tendrá datos.

## Cuentas

La opción **Cuentas** debe mostrar un submenú con las siguientes opciones:

- **Nueva** cuenta      Registra una cuenta nueva a nombre de un usuario, utilice **id\_cliente** para relacionar el usuario y la cuenta. Antes de crear la nueva cuenta se debe verificar que el usuario exista en el sistema. Adicionalmente, se debe indicar el saldo con el que se abre la cuenta. Por ejemplo; \$1000.
- **Buscar** cuenta      Permite consultar en pantalla la información de una cuenta en el sistema a partir de su **id\_cuenta**. En pantalla debe mostrarse: **id\_cuenta, nombre de cliente, saldo de la cuenta**.
- **Eliminar** cuenta      Si existe, elimina la cuenta deseada en el sistema.
- **Imprimir** cuentas      Imprime la información de todas las cuentas registradas en el sistema. En pantalla debe mostrarse un listado con la siguiente información de las cuentas: **id\_cuenta, nombre de cliente, saldo de la cuenta**.

La información que el sistema requiere almacenar sobre cada cuenta es la siguiente:

- **id\_cuenta** (es un número entero que se genera de manera consecutiva, clave única)
- **id\_usuario** (indica a quien pertenece la cuenta)
- **Saldo**
- **Fecha de apertura** (tipo de dato estructurado: dd/mm/aaaa)

Para gestionar la información de las cuentas, defina un tipo de dato estructurado llamado **Cuenta**, utilice instancias de **Cuenta** para capturar la información desde el teclado y posteriormente guardarlo en el archivo **cuenta.dat**.

Un ejemplo del contenido que se estará almacenando en el archivo **cuenta.dat** es el siguiente:

id_cuenta	Id_usuario	Saldo	fecha_apertura
1	1	Perez	{12,6, 2018}
2	2	Rodriguez	{2,7, 2018}
3	1	Martínez	{7,3,2018}

**Importante:** considere que no pueden existir valores de **id\_cuenta** repetidos y que es un valor autonómico. Adicionalmente, observe que un usuario puede tener más de una cuenta.

## Transacciones

La opción **Transacciones** debe mostrar un submenú con las siguientes opciones:

- **Depósito** Permite incrementar el saldo de la cuenta, para esto el sistema requiere: **id\_cuenta, monto a depositar** (valide que la cuenta exista).
- **Retiro** Permite a un cliente disponer del dinero que tiene una cuenta bancaria. Para esto el sistema requiere: **id\_cuenta, monto a retirar** (valide que la cuenta existe y que tiene fondos suficientes).
- **Transferencia** Permite a un cliente transferir dinero de una cuenta origen a una cuenta destino. Para esto el sistema requiere: **id\_cuenta origen, id\_cuenta destino, monto a transferir** (valide que existan ambas cuentas y que la cuenta origen tiene fondos suficientes).

La información que el sistema requiere almacenar sobre cada transacción es la siguiente:

- **id\_transacción** (es un número entero que se genera de manera consecutiva, no se puede repetir)
- **Tipo de operación** (depósito, retiro, transferencia)
- **Cuenta origen**
- **Cuenta destino** (se utiliza para las operaciones de transferencia, en otro caso, NULL)

- Fecha de la transacción
- Monto de la transacción

Para gestionar la información de las transferencias, defina un tipo de dato estructurado llamado **Transferencia**, utilice instancias de Transferencia para capturar la información desde el teclado y posteriormente guardarlo en el archivo transferencia.dat.

Un ejemplo del contenido que se estará almacenando en el archivo **transferencia.dat** es el siguiente:

id_transaccion	tipo_transaccion	Id_cuenta_origen	Id_cuenta_destino	fecha_transaccion	monto_transaccion
1	Retiro	1	Null	{12,6, 2018}	\$100
2	Deposito	2	Null	{12,6, 2018}	\$5000
3	Transferencia	2	1	{12,6,2018}	\$1500

**Importante:** considere que no pueden existir datos **id\_transaccion** repetidos y que es un valor autonúmerico. Adicionalmente, recuerde que al inicio el archivo no tendrá datos y que los saldos de las cuentas deberán afectarse por las transacciones realizadas.

## SOLUCIÓN DEL ALUMNO, PRUEBAS Y CONCLUSIONES

### Código fuente

<<Copie y pegue su código fuente aquí.>>

### Ejecución

#### Registro clientes

```
<<Sistema MyDB>>
1) Clientes
2) Cuentas
3) Transacciones
4) Salir
Opcion: 1

1) Nuevo Cliente
2) Buscar Cliente
3) Eliminar Cliente
4) Imprimir Cliente
Opcion: 1
Nombre: Sergio
Apellido Paterno: Mendez
Apellido Materno: Mendez
Fecha de nacimiento (dd/mm/aaaa): 08/06/1999
```

#### Buscar clientes

```
<<Sistema MyDB>>
1) Clientes
2) Cuentas
3) Transacciones
4) Salir
Opcion: 1

1) Nuevo Cliente
2) Buscar Cliente
3) Eliminar Cliente
4) Imprimir Cliente
Opcion: 2
Id de usuario: 1

ID Usuario  Nombre          Apellido paterno  Apellido materno  Fecha de nacimiento
1           Sergio          Mendez           Mendez           08/06/1999
```

## Eliminar cliente / Imprimir clientes

```
<<Sistema MyDB>>
1) Clientes
2) Cuentas
3) Transacciones
4) Salir
Opcion: 1

1) Nuevo Cliente
2) Bucar Cliente
3) Eliminar Cliente
4) Imprimir Cliente
Opcion: 3
Id de usuario: 1
```

```
<<Sistema MyDB>>
1) Clientes
2) Cuentas
3) Transacciones
4) Salir
Opcion: 1

1) Nuevo Cliente
2) Bucar Cliente
3) Eliminar Cliente
4) Imprimir Cliente
Opcion: 4
```

ID Usuario	Nombre	Apellido paterno	Apellido materno	Fecha de nacimiento
2	Juan	Ramirez	Zepeda	09/11/1998
3	Marco	Suarez	Salazar	19/02/0000

## Registro cuentas

```
<<Sistema MyDB>>
1) Clientes
2) Cuentas
3) Transacciones
4) Salir
Opcion: 2

1) Nueva cuenta
2) Buscar cuenta
3) Eliminar cuenta
4) Imprimir cuenta
Opcion: 1
Id del cliente: 2
Saldo de apertura: 9000
Fecha de apertura(dd/mm/aaaa): 22/06/2018
```

## Buscar cuenta

```
<<Sistema MyDB>>
1) Clientes
2) Cuentas
3) Transacciones
4) Salir
Opcion: 2

1) Nueva cuenta
2) Buscar cuenta
3) Eliminar cuenta
4) Imprimir cuenta
Opcion: 2
Id de cuenta: 1
|
```

ID Cuenta	ID Usuario	Saldo	Fecha de apertura
1	2	9000	22/06/2018

## Borrar cuenta/ Imprimir cuentas

```

<<Sistema MyDB>>
1) Clientes
2) Cuentas
3) Transacciones
4) Salir
Opcion: 2

1) Nueva cuenta
2) Buscar cuenta
3) Eliminar cuenta
4) Imprimir cuenta
Opcion: 3
Id de cuenta: 1

<<Sistema MyDB>>
1) Clientes
2) Cuentas
3) Transacciones
4) Salir
Opcion: 2

1) Nueva cuenta
2) Buscar cuenta
3) Eliminar cuenta
4) Imprimir cuenta
Opcion: 4

```

ID Cuenta	ID Usuario	Saldo	Fecha de apertura
2	3	6000	03/02/2019
3	2	1000	05/08/0000

## Deposito

ID Cuenta	ID Usuario	Saldo	Fecha de apertura
2	3	6000	03/02/2019
3	2	1000	05/08/0000

```

<<Sistema MyDB>>
1) Clientes
2) Cuentas
3) Transacciones
4) Salir
Opcion: 3
Tipo de transaccion:
1) Retiro
2) Deposito
3) Transferencia
2
Id cuenta destino: 3
Monto: 3000
Fecha(dd/mm/aaaa): 08/09/2019

<<Sistema MyDB>>
1) Clientes
2) Cuentas
3) Transacciones
4) Salir
Opcion: 2

1) Nueva cuenta
2) Buscar cuenta
3) Eliminar cuenta
4) Imprimir cuenta
Opcion: 4

```

ID Cuenta	ID Usuario	Saldo	Fecha de apertura
2	3	6000	03/02/2019
3	2	4000	05/08/0000



## Retiro

ID Cuenta	ID Usuario	Saldo	Fecha de apertura
2	3	6000	03/02/2019
3	2	4000	05/08/0000

<<Sistema MyDB>>  
1) Clientes  
2) Cuentas  
3) Transacciones  
4) Salir  
Opcion: 3  
Tipo de transaccion:  
1) Retiro  
2) Deposito  
3) Transferencia  
1  
ID de cuenta origen: 2  
Monto: 8000  
Saldo insuficiente

<<Sistema MyDB>>  
1) Clientes  
2) Cuentas  
3) Transacciones  
4) Salir  
Opcion: 2

1) Nueva cuenta  
2) Buscar cuenta  
3) Eliminar cuenta  
4) Imprimir cuenta  
Opcion: 4

ID Cuenta	ID Usuario	Saldo	Fecha de apertura
2	3	6000	03/02/2019
3	2	4000	05/08/0000

ID Cuenta	ID Usuario	Saldo	Fecha de apertura
2	3	6000	03/02/2019
3	2	4000	05/08/0000

<<Sistema MyDB>>  
1) Clientes  
2) Cuentas  
3) Transacciones  
4) Salir  
Opcion: 3  
Tipo de transaccion:  
1) Retiro  
2) Deposito  
3) Transferencia  
1  
ID de cuenta origen: 2  
Monto: 1000  
Fecha(dd/mm/aaaa): 08/09/2020

<<Sistema MyDB>>  
1) Clientes  
2) Cuentas  
3) Transacciones  
4) Salir  
Opcion: 2

1) Nueva cuenta  
2) Buscar cuenta  
3) Eliminar cuenta  
4) Imprimir cuenta  
Opcion: 4

ID Cuenta	ID Usuario	Saldo	Fecha de apertura
2	3	5000	03/02/2019
3	2	4000	05/08/0000

## Transferencia

ID Cuenta	ID Usuario	Saldo	Fecha de apertura
2	3	5000	03/02/2019
3	2	4000	05/08/0000

<<Sistema MyDB>>  
1) Clientes  
2) Cuentas  
3) Transacciones  
4) Salir  
Opcion: 3  
Tipo de transaccion:  
1) Retiro  
2) Deposito  
3) Transferencia  
3  
Id cuenta origen: 2  
Id cuenta destino: 3  
Monto: 2000  
Fecha(dd/mm/aaaa): 08/06/2021

<<Sistema MyDB>>  
1) Clientes  
2) Cuentas  
3) Transacciones  
4) Salir  
Opcion: 2  
  
1) Nueva cuenta  
2) Buscar cuenta  
3) Eliminar cuenta  
4) Imprimir cuenta  
Opcion: 4

ID Cuenta	ID Usuario	Saldo	Fecha de apertura
2	3	3000	03/02/2019
3	2	3000	05/08/0000

## Conclusiones:

Esta practica fue bastante complicada, debido a que no tenia casi ningun conocimiento sobre el tema, tuve que consultar mucha documentación y aprender del manejo, principalmente de archivos. Incluso pedí ayuda de compañeros con el fin de poder verificar soluciones o poder obtener ideas de cómo implementar soluciones.

En general no hubo ningun elemento del problema que no haya podido superar, sin embargo, siento que la forma en que lo desarrolle no fue del todo optima, siento que el programa es poco legible y hace falta hacer bastantes comentarios con el fin de que pueda ser entendido por alguien más.

Lo que más me costó trabajo fue comprender como funcionaban los archivos y el como manejar los datos que estan dentro de ellos, para esto requeri de mucha documentación y de bastante prueba y error, con el fin de revisar que todo funcionara de forma correcta; la verdad es que no fue una tarea fácil, en total quiza tarde 10 horas en realizarla toda contando códigos fallidos, por lo que fue uno de los trabajos más demandantes hasta ahora.