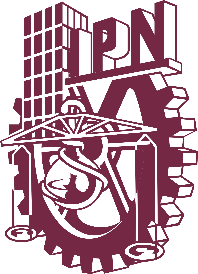
**FUNDAMENTOS DE PROGRAMACIÓN**

**PROYECTO FINAL**

**MORA CAMPOS RICARDO URIEL**

**GUTIÉRREZ JIMÉNEZ CINTHIA NAYELI**

**GONZÁLEZ OLOARTE DIEGO ENRIQUE**

**CASTAÑEDA ESPINOSA DANIEL IVÁN**

**ICV2**

**Planteamiento del problema:**

**Objetivo:**

Aplicar los conocimientos adquiridos para el desarrollo de programas en

Lenguaje C, con la finalidad de utilizar buenas prácticas en el diseño,

implementación, pruebas y depuración de programas.

**Instrucciones:**

Desarrolla en pareja el ejercicio solicitado en Lenguaje C.

Recuerda: Deberás identificar las entradas, salidas y procesos, así como elaborar tu algoritmo previamente de manera escrita (Diagrama de Flujo), para

posteriormente desarrollarlo en Lenguaje C y finalmente realizar capturas de

pantalla del funcionamiento del programa.

**Entradas:** Cada que se selecciones alguna opción de los menús, hay que recibir la opción. Al hacer registro de alguna cuenta o cliente, hay pedir datos para su autorización. Así como solicitar el número de cliente para su obtención de una cuenta. Al hacer depósitos o pedir préstamos también hay que pedir los datos al cliente y los montos requerido. Tampoco podemos olvidar el inicio de sesión de un cliente, donde se pide el número de cuenta y su contraseña.

**Procesos:** En el siguiente programa se simula una base de datos de un banco, declaramos primero las variables globales que se están manipulando durante el llamado de las otras funciones

Menús Principales

Se tienen dos funciones para despegar los menús principales, a su vez esta nos darán paso a operarlos.

Al iniciar secion se desplegaran las opciones de entrar como cliente o administrador

Administrador

El administrador tendrá opción de registrar un cliente o una cuenta nueva, al igual que ver los que ya están registrados.

Registrar Cliente

Al momento de registrar al cliente solicitara los datos como

* Nombre
* Apellido Paterno
* Apellido Materno
* Edad
* Calle
* Número interior
* Número exterior
* Colonia
* Alcaldía
* Ciudad
* Contraseña

Una vez ingresados correctamente, seremos registrados como clientes otorgándonos un ID

Registrar Cuenta

Al registrar la cuenta nos solicitará el ID del cliente y después nos dará el número de cuenta que corresponde

Registrar deposito

Con el número de cuenta asignado podemos realizar un depósito para lo cual solo tenemos que iniciar sesión e ingresar un monto que no exceda el saldo inicial

Lo mismo sucede con el préstamo

Como cliente tienes la opción de consultar tu saldo actual el cual se actualizará conforme a los depósitos y préstamos que se realicen en las cuentas

**Salidas:** Cada que se pida una letra, la va a pedir la terminal, mostrando los intentos restantes. Si se encuentra o no la letra en la Salidas:

• Ver clientes registrados.

Una vez que el usuario registra un cliente y esta información se registra en un archivo, se pueden mostrar los diferentes clientes que se hayan registrado con la siguiente información:

1. El número de cliente (ID).

2. El nombre completo del cliente.

3. La contraseña del cliente.

Toda esta información se mostrará en pantalla en la opción Ver clientes, en la parte de Administración.

En caso de que no haya ningún cliente registrado se mostrará el mensaje “No hay ningún cliente registrado”, para que se proceda a registrar alguno.

• Ver cuentas registradas.

Una vez se ingrese a alguna cuenta de algún cliente a través del uso de su ID y su contraseña, se puede conocer la cantidad de dinero que tiene mostrando:

1. Las diferentes cuantas que pueda tener el cliente.

2. La deuda que deba.

3. El dinero que tenga depositado.

4. Los diferentes depósitos que el cliente deseé hacer en el futuro.

En caso de que el cliente no tenga ninguna cuenta registrada se mostrara el mensaje “¿Desea crear una cuenta? (s/n)”, y se procederá a realizar la acción que el usuario deseé.

**Implementación de solución:**

﻿#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <ctype.h>

#include <string.h>

/\*FUNCIONES DE PREPROCESADOR\*/

/\*variables globales\*/

int numClienteSesion;

char contrasenaSesion [15];

char registro;

struct Domicilio

{

char calle [15];

int numeroInt;

int numeroExt;

char colonia [15];

char alcaldia [15];

char ciudad [15];

};

struct Cliente

{

int numCliente1;

char nombre [15];

char apePat [15];

char apeMat [15];

int edad;

struct Domicilio domicilioCliente;

char contrasena [15];

} miCliente;

bool logueado = false;

struct Cuenta

{

int numCuenta;

int numCliente2;

double saldoDeudor;

} miCuenta;

char opcion;

/\*funciones\*/

/\*funciones que muestran menús\*/

void menuCliente();

void menuAdmin();

/\*funciones de administración\*/

void registrarCliente();

void registrarCuenta();

bool existeCliente(int numCliente); //esta función comprueba que un cliente exista

void verClientes();

void verCuentas();

/\*funciones de cliente\*/

void iniciarSesion();

void registrarDeposito();

void registrarPrestamo();

void verSaldoCuentas(int cliente);

/\*función que valida que el usuario ingrese una opción adecuada\*/

void validarOpc(int limInf, int limSup);

/\*función para salir\*/

void salir();

int main()

{

system("cls");

printf("MENU PRINCIPAL\n");

printf("\n1. Administracion");

printf("\n2. Cliente");

printf("\n3. Salir");

validarOpc(1, 3);

switch(opcion)

{

case '1':

menuAdmin();

break;

case '2':

iniciarSesion();

menuCliente();

break;

case '3':

salir();

break;

default:

printf("Error");

break;

}

getch();

return 0;

}

//Esta función lee y valida que opción sea un número y se encuentre entre dos números

void validarOpc(int limInf, int limSup)

{

/\*

limInf = límite inferior

limSup = límite Superior

\*/

printf("\nIngrese una opcion: ");

scanf(" %c", &opcion);

while(!isdigit(opcion) || opcion-'0'<limInf || opcion-'0'>limSup)

{

printf("Ingrese una opcion correcta (un numero del %d al %d): ", limInf, limSup);

scanf(" %c", &opcion);

}

}

void menuAdmin()

{

system("cls");

printf("ADMINISTRACION\n");

printf("\n1. Registrar un cliente");

printf("\n2. Registrar una cuenta (Credito)");

printf("\n3. Ver clientes");

printf("\n4. Ver cuentas");

printf("\n5. Regresar");

validarOpc(1, 5);

switch(opcion)

{

case '1':

registrarCliente();

break;

case '2':

registrarCuenta();

break;

case '3':

verClientes();

break;

case '4':

verCuentas();

break;

case '5':

main();

break;

default:

printf("Error");

break;

}

}

void iniciarSesion(){

system("cls");

FILE \*fp=fopen("Clientes.txt", "r+");

if (fp == NULL) {

fclose(fp);

printf("No hay clientes registrados. ¿Registrar uno con un asesor? (s/n).\n");

scanf("%s", &registro);

if (registro == 's') registrarCliente();

else main();

}

printf("Inicie sesion.\n\nNo. de cliente: ");

scanf("%d", &numClienteSesion);

while (!feof(fp)) {

fscanf(fp, "%d %s %s %s %s\n", &miCliente.numCliente1, &miCliente.nombre, &miCliente.apePat, &miCliente.apeMat, &miCliente.contrasena);

if (numClienteSesion==miCliente.numCliente1) {

break;

}

}

do {

printf("Contrasenia: ");

scanf("%s", &contrasenaSesion);

} while(strcmp(contrasenaSesion,miCliente.contrasena));

}

void menuCliente()

{

system("cls");

printf("CLIENTE\n");

printf("\n1. Registrar un deposito");

printf("\n2. Registrar un prestamo");

printf("\n3. Ver saldo de cuentas");

printf("\n4. Regresar");

FILE \*fp;

fp = fopen("Clientes.txt", "r+t");

/\*----------------------Se lee el archivo de Clientes---------------------\*/

if(fp == NULL)

{

printf("No hay clientes registrados.");

}

else

{

fclose(fp);

fp = fopen("Clientes.txt", "a+t");

while(!feof(fp))

{

fscanf(fp, "%d %s %s %s %s\n", &miCliente.numCliente1, &miCliente.nombre, &miCliente.apePat, &miCliente.apeMat, &miCliente.contrasena);

if(numClienteSesion == miCliente.numCliente1 && strcmp(contrasenaSesion, miCliente.contrasena)==0)

{

logueado = true;

break;

}

}

fclose(fp);

miCliente.numCliente1++;

}

if(logueado)

{

system("cls");

printf("CLIENTE\n");

printf("\n1. Registrar un deposito");

printf("\n2. Registrar un prestamo");

printf("\n3. Ver saldo de cuentas");

printf("\n4. Regresar");

validarOpc(1, 4);

switch(opcion)

{

case '1':

registrarDeposito();

break;

case '2':

registrarPrestamo();

break;

case '3':

verSaldoCuentas(numClienteSesion);

break;

case '4':

main();

break;

default:

printf("Error");

break;

}

}

else printf("Datos incorrectos");

}

void registrarCliente()

{

system("cls");

struct Cliente miCliente;

FILE \*fp;

FILE \*fp2;

fp = fopen("Clientes.txt", "r+t");

fp2 = fopen("Domicilio.txt", "r+t");

/\*----------------------Se lee el archivo de Clientes---------------------\*/

if(fp == NULL)

{

fp = fopen("Clientes.txt", "w+t");

fprintf(fp, "%d ", 1);

miCliente.numCliente1 = 1;

}

else

{

fclose(fp);

fp = fopen("Clientes.txt", "a+t");

while(!feof(fp))

{

fscanf(fp, "%d %s %s %s %s\n", &miCliente.numCliente1, &miCliente.nombre, &miCliente.apePat, &miCliente.apeMat, &miCliente.contrasena);

}

fclose(fp);

miCliente.numCliente1++;

}

/\*----------------------Se lee el archivo de Domicilio---------------------\*/

if(fp2 == NULL)

{

fp2 = fopen("Domicilio.txt", "w+t");

}

else

{

fclose(fp2);

fp2 = fopen("Domicilio.txt", "a+t");

fclose(fp2);

}

printf("DATOS PERSONALES: \n");

printf("Nombre: ");

scanf(" %[^\n]", miCliente.nombre);

printf("Apellido paterno: ");

scanf(" %[^\n]", miCliente.apePat);

printf("Apellido materno: ");

scanf(" %[^\n]", miCliente.apeMat);

printf("Edad: ");

scanf(" %[^\n]", &miCliente.edad);

printf("\nDOMICILIO: \n");

printf("Calle: ");

scanf(" %[^\n]", miCliente.domicilioCliente.calle);

printf("Numero Interior: ");

scanf(" %d", &miCliente.domicilioCliente.numeroInt);

printf("Numero Exterior: ");

scanf(" %d", &miCliente.domicilioCliente.numeroExt);

printf("Colonia: ");

scanf(" %[^\n]", miCliente.domicilioCliente.colonia);

printf("Alcaldia: ");

scanf(" %[^\n]", miCliente.domicilioCliente.alcaldia);

printf("Ciudad: ");

scanf(" %[^\n]", miCliente.domicilioCliente.ciudad);

printf("Contrasenia: ");

scanf(" %s", miCliente.contrasena);

fp = fopen("Clientes.txt", "a+t");

fp2 = fopen("Domicilio.txt", "a+t");

fprintf(fp, "%d ", miCliente.numCliente1);

fprintf(fp, "%s ", miCliente.nombre);

fprintf(fp, "%s ", miCliente.apePat);

fprintf(fp, "%s ", miCliente.apeMat);

fprintf(fp2, "%d ", miCliente.numCliente1);

fprintf(fp2, "[%s] ", miCliente.domicilioCliente.calle);

fprintf(fp2, "%d ", miCliente.domicilioCliente.numeroInt);

fprintf(fp2, "%d ", miCliente.domicilioCliente.numeroExt);

fprintf(fp2, "[%s] ", miCliente.domicilioCliente.colonia);

fprintf(fp2, "[%s] ", miCliente.domicilioCliente.alcaldia);

fprintf(fp2, "[%s] ", miCliente.domicilioCliente.ciudad);

fprintf(fp, "%s ", miCliente.contrasena);

fputs("\n", fp);

fputs("\n", fp2);

fclose(fp);

fclose(fp2);

printf("Cliente registrado exitosamente. ID: %d", miCliente.numCliente1);

system("pause");

menuAdmin();

}

void registrarCuenta()

{

FILE \*fp;

fp = fopen("Cuentas.txt", "r+t");

bool primerRegistro = true;

if(fp == NULL)

{

fp = fopen("Cuentas.txt", "w+t");

//miCuenta.numCuenta = 1;

//fprintf(fp, "%d ", miCuenta.numCuenta);

fclose(fp);

}

else

{

primerRegistro = false;

fclose(fp);

fp = fopen("Cuentas.txt", "a+t");

while(!feof(fp))

{

fscanf(fp, "%d %d %d\n", &miCuenta.numCuenta, &miCuenta.numCliente2, &miCuenta.saldoDeudor);

}

fclose(fp);

miCuenta.numCuenta++;

}

system("cls");

printf("Ingrese el ID del cliente que desea obtener una cuenta: ");

scanf("%d", &miCuenta.numCliente2);

//----------Buscando si el cliente existe

if(existeCliente(miCuenta.numCliente2))

{

fp = fopen("Cuentas.txt", "a+t");

if(primerRegistro)

fprintf(fp, "%d ", 1);

else

fprintf(fp, "%d ", miCuenta.numCuenta);

fprintf(fp, "%d ", miCuenta.numCliente2);

fprintf(fp, "%d ", 5000);

fputs("\n", fp);

fclose(fp);

printf("La cuenta fue otorgada con exito al cliente No. %d", miCuenta.numCliente2);

}

else

{

printf("ERROR. No existe un cliente con ese ID.");

}

fclose(fp);

system("pause");

menuAdmin();

}

bool existeCliente(int numCliente)

{

FILE \*fp;

fp = fopen("Clientes.txt", "r+t");

bool existe = false;

if(fp == NULL)

{

fp = fopen("Clientes.txt", "w+t");

fprintf(fp, "%d ", 1);

}

else

{

fclose(fp);

fp = fopen("Clientes.txt", "a+t");

while(!feof(fp))

{

fscanf(fp, "%d %s %s %s %s\n", &miCliente.numCliente1, miCliente.nombre, miCliente.apePat, miCliente.apeMat, miCliente.contrasena);

if(miCliente.numCliente1 == numCliente)

{

existe = true;

}

}

}

fclose(fp);

return existe;

}

void registrarDeposito ()

{

system("cls");

FILE \*fp = fopen( "Cuentas.txt" , "r+t" );

if (fp == NULL) //Si no hay un archivo es porque aún no tiene una cuenta

{

fclose(fp);

printf("No tienes una cuenta todavía. ¿Deseas registrar una? \n");

scanf("%s", &registro);

if (registro == 's') registrarCuenta();

else menuCliente();

}

printf("Cuentas en existencia\n\n");

while (!feof(fp))

{

fscanf( fp, "%d %d %d\n", &miCuenta.numCuenta, &miCuenta.numCliente2, &miCuenta.saldoDeudor);

if(numClienteSesion == miCuenta.numCliente2)printf("Cuenta: %d\tAdeudo: %d\n\n", miCuenta.numCuenta, miCuenta.saldoDeudor);

}

fclose(fp);

fp = fopen( "Cuentas.txt" , "r+t" );

bool depositar = false;

int cuentaDepositar;

printf("\n\nSeleccione la cuenta a depositar: ");

scanf("%d", &cuentaDepositar);

int deposito;

validacion:

printf("Monto del Deposito: ");

scanf("%d", &deposito);

if (deposito<=0) {

printf("Ingrese una cantidad positiva. Inténtelo nuevamente\n");

goto validacion;

}

while (!feof(fp))

{

fscanf(fp, "%d %d %d\n", &miCuenta.numCuenta, &miCuenta.numCliente2, &miCuenta.saldoDeudor);

if(numClienteSesion == miCuenta.numCliente2 && cuentaDepositar == miCuenta.numCuenta)

depositar = true; //la variable depositar nos permite saber si el cliente metió una cuenta que le corresponde

}

if(depositar)

{

//modificar saldo actual

FILE \*fpTemporal;

fpTemporal = fopen("CuentasTemporal.txt", "a+t");

rewind(fp);

while (!feof(fp))

{

fscanf(fp, "%d %d %d\n", &miCuenta.numCuenta, &miCuenta.numCliente2, &miCuenta.saldoDeudor);

fprintf(fpTemporal, "%d ", miCuenta.numCuenta);

fprintf(fpTemporal, "%d ", miCuenta.numCliente2);

if(numClienteSesion == miCuenta.numCliente2 && cuentaDepositar == miCuenta.numCuenta)

fprintf(fpTemporal, "%d", miCuenta.saldoDeudor-deposito);

else

fprintf(fpTemporal, "%d", miCuenta.saldoDeudor);

fprintf(fpTemporal, "\n");

}

rewind(fp);

rewind(fpTemporal);

while (!feof(fpTemporal))

{

fscanf(fpTemporal, "%d %d %d\n", &miCuenta.numCuenta, &miCuenta.numCliente2, &miCuenta.saldoDeudor);

fprintf(fp, "%d ", miCuenta.numCuenta);

fprintf(fp, "%d ", miCuenta.numCliente2);

fprintf(fp, "%d ", miCuenta.saldoDeudor);

fprintf(fp, "\n");

}

fclose(fpTemporal);

remove("CuentasTemporal.txt");

printf("Deposito hecho.");

getch();

}

else

{

printf("Datos incorrectos.");

getch();

}

fclose(fp);

menuCliente();

}

void registrarPrestamo()

{

system("cls");

FILE \*fp = fopen( "Cuentas.txt" , "r+t" );

if (fp == NULL) //Si no hay un archivo es porque aún no tiene una cuenta

{

fclose(fp);

printf("No tienes una cuenta todavía. ¿Deseas registrar una? \n");

scanf("%c", &registro);

if (registro == 's') registrarCuenta();

else menuCliente();

}

printf("Cuentas en existencia\n\n");

while (!feof(fp))

{

fscanf( fp, "%d %d %d\n", &miCuenta.numCuenta, &miCuenta.numCliente2, &miCuenta.saldoDeudor);

if(numClienteSesion == miCuenta.numCliente2)printf("Cuenta: %d\tAdeudo: %d\n\n", miCuenta.numCuenta, miCuenta.saldoDeudor);

}

fclose(fp);

fp = fopen( "Cuentas.txt" , "r+t" );

bool prestar = false;

int cuentaPrestamo;

printf("\n\nSeleccione la cuenta para realizar el prestamo: ");

scanf("%d", &cuentaPrestamo);

double prestamo;

printf("Monto del Prestamo: ");

scanf("%ld", &prestamo);

while (!feof(fp))

{

fscanf(fp, "%d %d %ld\n", &miCuenta.numCuenta, &miCuenta.numCliente2, &miCuenta.saldoDeudor);

if(numClienteSesion == miCuenta.numCliente2 && cuentaPrestamo == miCuenta.numCuenta)

prestar = true; //la variable prestar nos permite saber si el cliente metió una cuenta que le corresponde

}

if(prestar)

{

//modificar saldo actual

FILE \*fpTemporal;

fpTemporal = fopen("CuentasTemporal.txt", "a+t");

rewind(fp);

while (!feof(fp))

{

fscanf(fp, "%d %d %ld\n", &miCuenta.numCuenta, &miCuenta.numCliente2, &miCuenta.saldoDeudor);

fprintf(fpTemporal, "%d ", miCuenta.numCuenta);

fprintf(fpTemporal, "%d ", miCuenta.numCliente2);

if(numClienteSesion == miCuenta.numCliente2 && cuentaPrestamo == miCuenta.numCuenta)

fprintf(fpTemporal, "%d", miCuenta.saldoDeudor+prestamo);

else

fprintf(fpTemporal, "%d", miCuenta.saldoDeudor);

fprintf(fpTemporal, "\n");

}

rewind(fp);

rewind(fpTemporal);

while (!feof(fpTemporal))

{

fscanf(fpTemporal, "%d %d %d\n", &miCuenta.numCuenta, &miCuenta.numCliente2, &miCuenta.saldoDeudor);

fprintf(fp, "%d ", miCuenta.numCuenta);

fprintf(fp, "%d ", miCuenta.numCliente2);

fprintf(fp, "%d ", miCuenta.saldoDeudor);

fprintf(fp, "\n");

}

fclose(fpTemporal);

remove("CuentasTemporal.txt");

printf("Prestamo hecho. Gracias por confiar en nosotros.");

getch();

}

else

{

printf("Datos incorrectos.");

getch();

}

fclose(fp);

menuCliente();

}

void verClientes(){

system("cls");

FILE \*fp=fopen("Clientes.txt","r+");

if (fp==NULL)

{

fclose(fp);

printf("No hay clientes registrados. ¿Registrar uno? (s/n)\n");

scanf("%s", &registro);

if (registro == 's') registrarCliente();

else menuAdmin();

}

printf("ID:\tNombre:\t\t\t\tContrasenia:\n");

while (!feof(fp))

{

fscanf( fp, "%d %s %s %s %s\n", &miCliente.numCliente1, &miCliente.nombre, &miCliente.apePat, &miCliente.apeMat, &miCliente.contrasena);

printf("\n%d\t%s %s %s\t\t%s\n",miCliente.numCliente1,miCliente.nombre,miCliente.apePat,miCliente.apeMat,miCliente.contrasena);

}

system("pause");

fclose(fp);

menuAdmin();

}

void verCuentas()

{

system("cls");

FILE \*fp = fopen( "Cuentas.txt" , "r+" );

if (fp == NULL) { //Si no hay un archivo es porque aún no se han registrado Cuentas

fclose(fp);

printf("No hay cuentas registradas. ¿Registrar una? (s/n)\n");

scanf("%s", &registro);

if (registro == 's') registrarCuenta();

else menuAdmin();

}

while (!feof(fp)) { //cada dato que esté en el archivo se muestra con el ciclo

fscanf( fp, "%d %d %d\n", &miCuenta.numCuenta, &miCuenta.numCliente2, &miCuenta.saldoDeudor);

printf("Cliente %d:\n\n\tNo. Cuenta: %d\tDeuda: %d\n\n", miCuenta.numCliente2, miCuenta.numCuenta, miCuenta.saldoDeudor);

}

system("pause");

fclose(fp);

menuAdmin();

}

void verSaldoCuentas(int cliente) {

system("cls");

FILE \*fp = fopen( "Cuentas.txt" , "r+" );

if (fp == NULL) { //Si no hay un archivo es porque aún no tiene una cuenta

fclose(fp);

printf("No tienes una cuenta. ¿Registrar una con un asesor? (s/n)\n");

scanf("%s", &registro);

if (registro == 's') registrarCuenta();

else menuCliente();

}

printf("Usted tiene los siguentes numeros de cuenta:\n\n");

while (!feof(fp)) {

fscanf( fp, "%d %d %d\n", &miCuenta.numCuenta, &miCuenta.numCliente2, &miCuenta.saldoDeudor);

if (cliente == miCuenta.numCliente2) {

printf("No. Cuenta: %d\tDeuda: %d\n\n", miCuenta.numCuenta, miCuenta.saldoDeudor);

}

}

system("pause");

fclose(fp);

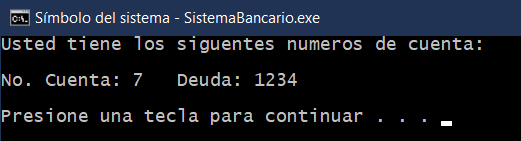
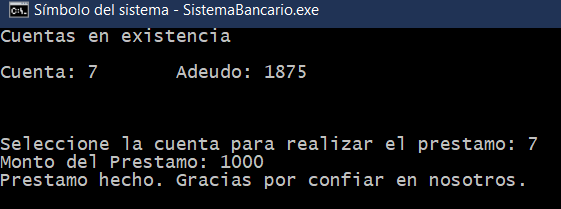
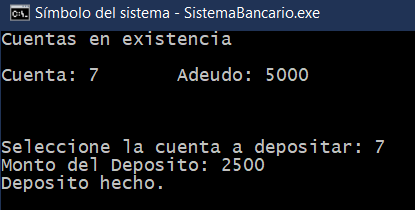
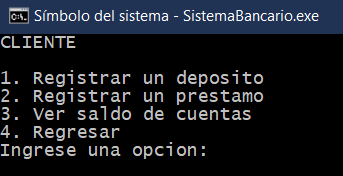
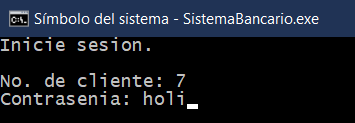
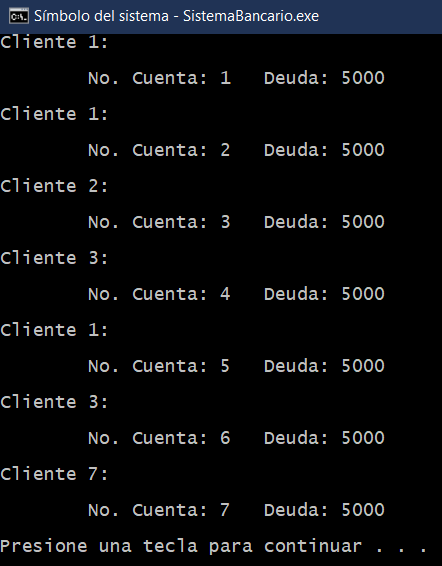
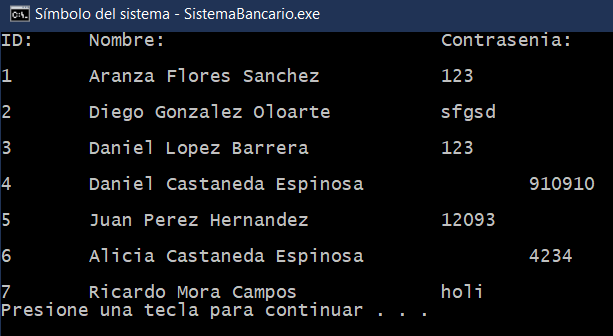
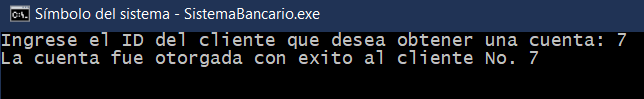
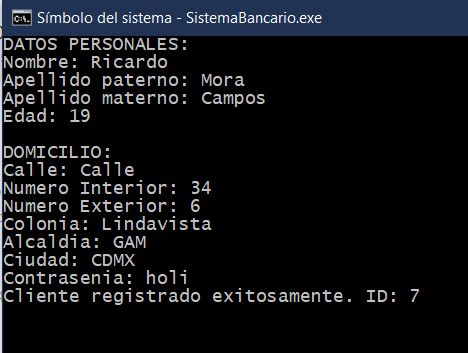
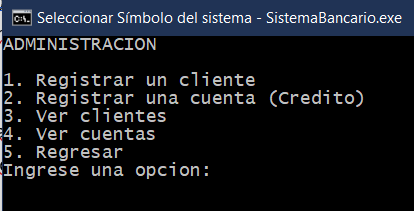
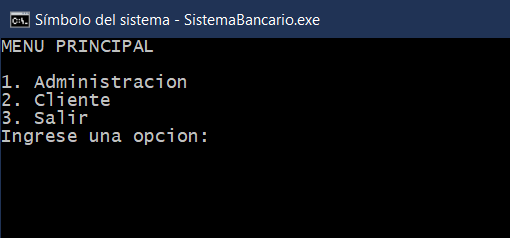
menuCliente();

}

void salir(){

exit(-1);

}

**Funcionamiento **

**Conclusiones**

El resultado del programa puede ser optimizado en nuestra opinión, pero consideramos que funciona de manera muy eficiente.

En cuanto al conocimiento, la mayoría reforzamos los conocimientos que hemos obtenido en el curso. Aunque hubo cosas que de manera de prueba y error, así como comparaciones entre versiones que cada uno tenía, pudimos ver cómo debía de resolverse algún problema.

Todos aprendimos a trabajar en equipop, pues nunca habíamos trabajado con más de una persona en el desarrollo de un programa. Aprendimos a usar nuevas herramientas de desarrollo de software en equipo, obtuvimos más confianza en preguntarnos dudas surgían a lo largo del trabajo.