|  |  |  |
| --- | --- | --- |
| **CARRERA:**  Ingeniería de Software | **GUÍA**  No. 01 | **TIEMPO ESTIMADO:**  1h y 20 min. |
| **ASIGNATURA:**  Estructura de Datos NRC: 4851 | **FECHA DE ELABORACION:** 12/01/2021  **SEMESTRE**: Noviembre 2020-Abril 2021 | |
| **TÍTULO:**  Calculo de la edad | **DOCENTE:** Ing. Fernando Solís | |

**OBJETIVO**

Calcular la edad al registrar nuevos usuarios en una lista de cuentas bancaria.

**INSTRUCCIONES**

**p.e.**

1. Utilice como material principal, aquel indicado en clase por el docente.
2. Utilice información consultada en Internet y conocimiento adquirido en clase.

**ACTIVIDADES**

1. **Ubicación de recursos**

**p.e.**

1. Formar grupos de máximo 2 personas por computador
2. Instalar la herramienta Dev C++
3. **Planteamiento del problema**

**p.e. Ejercicio No 1**

Se crea una clase con métodos para calcular la edad de un usuario dado dentro de nuestro programa minibank el cual nos permite a través de un tipo de dato fecha de una clase, ingresada por consola para generar la edad actual de la persona en el dia en que se digite dicha fecha, esta funcionalidad toma en cuenta los siguientes parámetros.

1.- El ingreso de la fecha de nacimiento debe estar dentro de las fechas permitidas, la función no admitirá que se ingresen fechas de nacimiento de años que aún no pasan en la realidad ejemplo (12/12/2025).

2.- La clase para el calculo de la edad tendrá un tipo de dato fecha actual y de nacimiento para obtener la edad del usuario.

3.- La clase para calcular la edad tendrá un método que devuelva un booleano para que sea implementado en un ciclo repetitivo, hasta que el usuario ingrese una fecha de nacimiento valida.

1. **Entregable (s)**

**p.e.**

1. Cada grupo deberá entregar (impresos) el archivo con el código fuente (.cpp)

**Archivo CalculoEdad.h**

#include"Fecha.h"

#include<iostream>

class CalculoEdad

{

public:

CalculoEdad();

Fecha DiferenciaEntreFechas();

void setNacimiento(Fecha);

private:

Fecha actual;

Fecha nacimiento;

};

**Archivo CalculoEdad.cpp**

#include "CalculoEdad.h"

#include <ctime>

using namespace std;

CalculoEdad::CalculoEdad() {

time\_t today = time(NULL);

struct tm now = \*localtime(&today);

this->actual.setAnio(now.tm\_year+1900);

this->actual.setMes(now.tm\_mon + 1);

this->actual.setDia(now.tm\_mday);

}

void CalculoEdad::setNacimiento(Fecha nacimiento) {

this->nacimiento = nacimiento;

}

Fecha CalculoEdad::DiferenciaEntreFechas() {

Fecha aux;

if (this->actual.get\_dia() < this->nacimiento.get\_dia() ) {

this->actual.setDia(actual.get\_dia() + 30 );

this->actual.setMes(actual.get\_mes()-1);

aux.setDia(actual.get\_dia() - nacimiento.get\_dia());

}

else

{

aux.setDia(actual.get\_dia() - nacimiento.get\_dia());

}

if (this->actual.get\_mes() < this->nacimiento.get\_mes())

{

this->actual.setMes(actual.get\_mes() + 12);

this->actual.setAnio(actual.get\_anio() - 1);

aux.setMes(actual.get\_mes() - nacimiento.get\_mes());

}

else {

aux.setMes(actual.get\_mes() - nacimiento.get\_mes());

}

aux.setAnio(actual.get\_anio() - nacimiento.get\_anio());

return aux;

}

**Archivo Fecha.h**

#include <stdio.h>

#include <string>

#include <time.h>

class Fecha

{

public:

Fecha();

Fecha(int, int, int);

void generar\_fecha();

int get\_anio();

int get\_mes();

int get\_dia();

std::string get\_fecha();

void setDia(int);

void setMes(int);

void setAnio(int);

private:

int anio;

int mes;

int dia;

};

**Archivo Fecha.cpp**

#include "Fecha.h"

Fecha::Fecha() {

this->anio = 0;

this->mes = 0;

this->dia = 0;

}

Fecha::Fecha(int dia, int mes, int anio) {

this->anio = anio;

this->mes = mes;

this->dia = dia;

}

void Fecha::generar\_fecha() {

time\_t current\_time;

struct tm local\_time;

time(&current\_time);

localtime\_s(&local\_time, &current\_time);

anio = local\_time.tm\_year + 1900;

mes = local\_time.tm\_mon + 1;

dia = local\_time.tm\_mday;

}

int Fecha::get\_anio() {

return anio;

}

int Fecha::get\_mes() {

return mes;

}

int Fecha::get\_dia() {

return dia;

}

void Fecha::setDia(int newDia) {

dia = newDia;

}

void Fecha::setMes(int newMes) {

mes = newMes;

}

void Fecha::setAnio(int newAnio) {

anio = newAnio;

}

std::string Fecha::get\_fecha() {

std::string s\_anio(std::to\_string(anio));

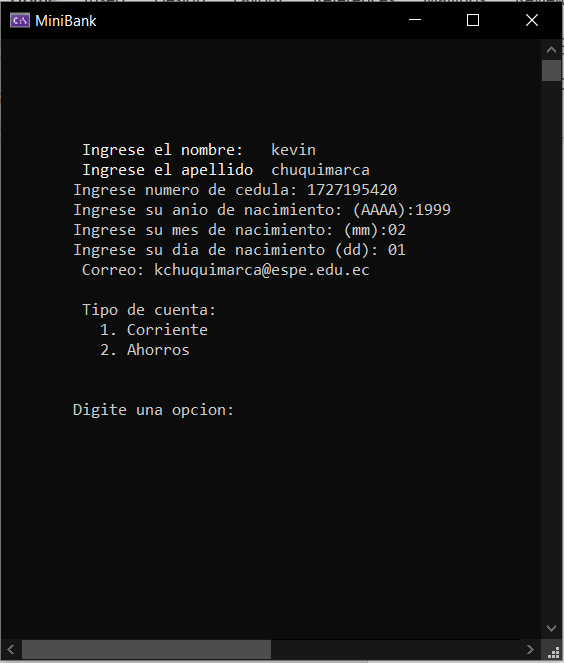
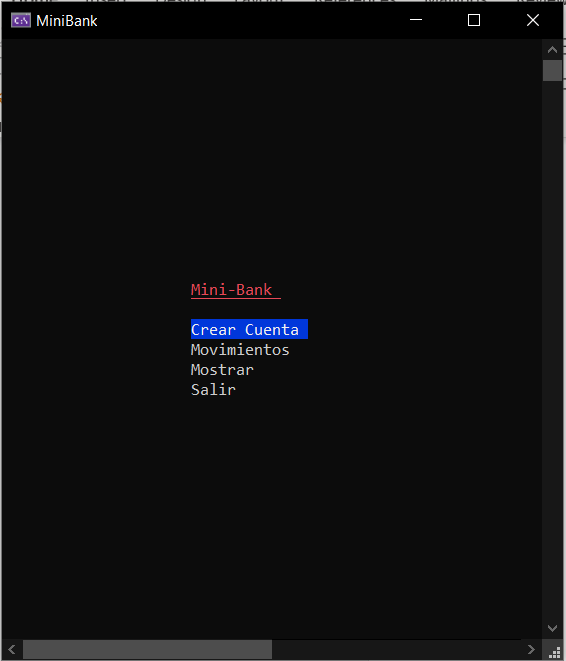
std::string s\_mes(std::to\_string(mes));

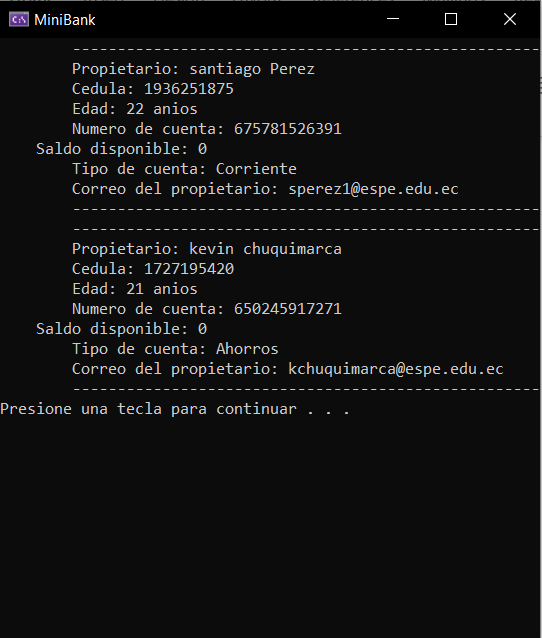
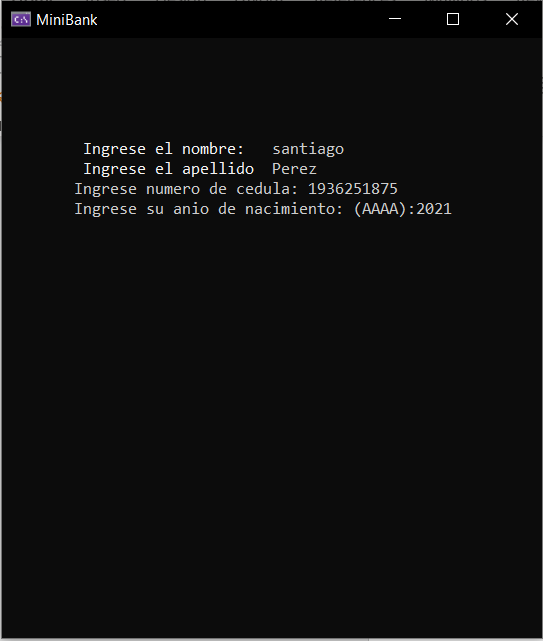
std::string s\_dia(std::to\_string(dia));

return s\_dia + "/" + s\_mes + "/" + s\_anio + "\n";

}

1. También se deberá presentar un informe con el marco teórico y las conclusiones correspondientes a la ejecución de la práctica.





\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DOCENTE RESPONSABLE COORDINADOR DE ÁREA

Ing. Fernando Solis. MsC. Ing. Silvia Arévalo.