

Actividad 2

Escribir, buscar, modificar correo por identificador

Alan Jahir Martínez Sepúlveda

216569127

05/09/2021

SEMINARIO DE SOLUCIÓN DE  
.  
PROBLEMAS DE ESTRUCTURA DE DATOS II

Clave del curso: I5889

NRC: 59623

Calendario: 2021-B

Sección: D10



**Descripción de la actividad:**

El estudiante deberá entregar el diagrama de secuencia (UML 2.0) y el código portátil para implementar los requerimientos 1,2, 3, 4, 4.1.1, 5, 5.1 y 5.1.1.

# ESPECIFICACIÓN DE REQUERIMIENTOS FUNCIONALES

Diseñe y codifique un programa simulador de lector de correo electrónico que cumpla con los siguientes requerimientos funcionales:

1. El programa no se conectará a un servidor de correo electrónico real, sino que trabajará en modo fuera de línea.
2. El software almacenará la información de los correos electrónicos en archivos de acceso directo con registros de longitud fija en formato binario.
3. Escribir Correo Electrónico: el programa debe permitir al usuario escribir un correo electrónico nuevo y almacenarlo en un archivo binario de datos.

Cada mensaje de correo electrónico será almacenado como un registro de longitud fija con los siguientes campos:

1. Identificador de registro de hasta 10 dígitos (asignado por el usuario).
   1. Fecha de Envío.
   2. Hora de Envío.
   3. Remitente (From: ).
   4. Destinatario (To: ).
   5. Copia Carbón (CC: ).
   6. Copia Carbón Ciega (BCC: ).
   7. Asunto (Subject: ).
   8. Contenido (debe soportar saltos de línea, es decir, diferentes párrafos).

La fecha y hora de envío no son ingresados por el usuario, sino que son almacenados directamente de la fecha y hora del sistema.

1. Leer Correo Electrónico: el software debe proveer un mecanismo para que el usuario pueda leer el correo electrónico almacenado.
   * 1. Acceso Directo: el usuario puede buscar y leer un correo electrónico introduciendo directamente el identificador de 10 dígitos del registro asignado al correo electrónico por el sistema.
2. Modificar Correo Electrónico: el programa debe admitir la modificación de cualquier correo electrónico que el usuario solicite.
   1. Para seleccionar el correo electrónico a modificar, el usuario podrá:
      1. Buscar el correo electrónico a modificar por identificador (**§** 4.1.1).

Código:

Main.cpp

#include "menu.h"

int main()

{

    Menu m;

    m.eje();

    return 0;

}

Menu.cpp

#include "menu.h"

*//#include "archivo.h"*

*//#include "correo.h"*

#include <iostream>

#include <ctime>

#include <cstring>

using namespace std;

Menu::Menu()

{

    opcion = 0;

}

void Menu::eje()

{

    do{

        cout << "1. Escribir correo" << endl;

        cout << "2. Leer correo" << endl;

        cout << "3. Modificar correo" << endl;

        cout << "4. Salir" << endl;

        cout << "Ingrese su opcion: " << endl;

        cin >> opcion;

        menuOpciones();

        cout << endl;

    }

    while (opcion != 4);

    cout << "Saliste!" << endl;

}

void Menu::menuOpciones()

{

    char \*puntero;

    char temp[200];

    puntero = temp;

    unsigned long int posicion;

    unsigned long int buscar;

    switch(opcion){

        case 1:

            {

                cout << endl;

                cout << "Datos del nuevo correo" << endl;

                cout << "ID: " << endl;

                cin >> posicion;

                correo.setId(posicion);

                cin.ignore();

                cout << "Remitente: " << endl;

                cin.getline(temp,20);

                correo.setRemitente(puntero);

                cout << "Destinatario: " << endl;

                cin.getline(temp,50);

                correo.setDestinatario(puntero);

                cout << "Copia carbono: " << endl;

                cin.getline(temp,50);

                correo.setCopiaCarbono(puntero);

                cout << "Copia carbono ciega: " << endl;

                cin.getline(temp,20);

                correo.setCopiaCarbonoCiega(puntero);

                cout << "Asunto: " << endl;

                cin.getline(temp,50);

                correo.setAsunto(puntero);

                cout << "Contenido:" << endl;

                correo.setContenido();

                cout << endl << endl;

                archivo.escribir(correo, posicion);

                break;

            }

        case 2:

            {

                cout << "Ingresa el ID del correo a buscar: " << endl;

                cin >> posicion;

                archivo.leer(posicion);

                break;

            }

        case 3:

            {

                cout << "Ingresa el ID del correo a modificar: " << endl;

                cin >> posicion;

                archivo.modificar(posicion);

                break;

            }

        case 4:

            {

                break;

            }

        default:

            break;

    }

}

Menu.h

#ifndef MENU\_H

#define MENU\_H

#include "archivo.h"

class Menu{

    private:

        int opcion;

        Archivo archivo;

        Correo correo;

        void menuOpciones();

    public:

        Menu();

        void eje();

};

#endif

Correo.h

#ifndef CORREO\_H

#define CORREO\_H

#include <iostream>

#include <string>

#include <time.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

class Correo

{

    private:

        unsigned long int id;

        char remitente[20];

        char destinatario[50];

        char copiaCarbono[50];

        char copiaCarbonoCiega[20];

        char asunto[50];

        char contenido[200];

        char dia[10];

        char hora[5];

        void funcionDiaHora();

*//const string currentDateTime();*

    public:

        Correo();

        int getId() const;

        char\* getDia() const;

        char\* getHora() const;

        char\* getRemitente() const;

        char\* getDestinatario() const;

        char\* getCopiaCarbono() const;

        char\* getCopiaCarbonoCiega() const;

        char\* getAsunto() const;

        char\* getContenido() const;

        void setId(const int &);

        void setRemitente(char\* );

        void setDestinatario(char\* );

        void setCopiaCarbono(char\* );

        void setCopiaCarbonoCiega(char\* );

        void setAsunto(char\* );

        void setContenido();

};

#endif

Correo.cpp

#include <iostream>

#include <ctime>

*//#include <stdio.h>*

#include <cstring>

#include "correo.h"

using namespace std;

Correo::Correo(){}

void Correo::funcionDiaHora(){

    time\_t now = time(nullptr);

    struct tm tm = \*localtime(&now);

    sprintf(hora, "%d/%d/%d", tm.tm\_mday, tm.tm\_mon + 1, tm.tm\_year + 1900);

    sprintf(hora, "%d:%d", tm.tm\_hour, tm.tm\_min);

}

int Correo::getId() const{

    return id;

}

char\* Correo::getRemitente() const {

    return (char\* )remitente;

}

char\* Correo::getDestinatario() const {

    return (char\* )destinatario;

}

char\* Correo::getCopiaCarbono() const {

    return (char\* )copiaCarbono;

}

char\* Correo::getCopiaCarbonoCiega() const {

    return (char\* )copiaCarbonoCiega;

}

char\* Correo::getAsunto() const {

    return (char\* )asunto;

}

char\* Correo::getContenido() const{

    return (char\* ) contenido;

}

char\* Correo::getDia() const {

    return (char\* )dia;

}

char\* Correo::getHora() const {

    return (char\* )hora;

}

*//setters:*

void Correo::setId(const int &i){

    id = i;

}

void Correo::setRemitente(char\* r){

    strncpy(remitente, r, 20);

}

void Correo::setDestinatario(char\* d){

    strncpy(destinatario, d, 50);

}

void Correo::setCopiaCarbono(char\* cc){

    strncpy(copiaCarbono, cc, 50);

}

void Correo::setCopiaCarbonoCiega(char\* ccb){

    strncpy(copiaCarbonoCiega, ccb, 20);

}

void Correo::setAsunto(char\* a){

    strncpy(asunto, a, 50);

}

void Correo::setContenido(){

    char temp[200], final[200];

    bool bandera = true;

    int i = 0, j, contador = 0;

    int co1, co2;

    do{

        cin.getline(temp, 200);

        while (temp[i] != '\000'){

            i++;

        }

        temp[i] = '\n';

        contador += i+1;

        if (temp[0] == 'E' and temp[i - 1] == 'R'){

            bandera = false;

        }

        else {

            for (j = co1, co2 = 0; j < contador; j++, co2++){

                final[j] = temp[co2];

            }

            co1 = contador;

        }

        i = 0;

    }

    while (bandera);

*//funcionDiaHora();*

    strncpy(contenido, final, 200);

*// currentDateTime();*

    cout << endl;

}

Archivo.h

#include <fstream>

#include "correo.h"

using namespace std;

class Archivo

{

    private:

        fstream archivo;

        unsigned long apuntador(unsigned long int&);

    public:

        Archivo();

        void escribir(const Correo&,unsigned long int&);

        void leer(unsigned long int&);

        void modificar(unsigned long int&);

};

Archivo.cpp

#include "archivo.h"

#include <iostream>

using namespace std;

Archivo::Archivo(){

}

unsigned long int Archivo::apuntador(unsigned long int& i){

    return (i-1) \* sizeof(Correo);

}

void Archivo::escribir(const Correo& c,unsigned long int& i)

{

    archivo.open("emails.dat", ios::in | ios::out | ios::binary);

    if (archivo.fail()){

        cout << "Ocurrio un error en el archivo" << endl;

        exit(1);

    }

    else {

        archivo.seekp(apuntador(i), ios\_base::beg);

        archivo.write((char\*)&c, sizeof(c));

        archivo.close();

        cout << "Registro exitoso" << endl;

    }

}

void Archivo::leer(unsigned long int& i)

{

    Correo c;

    archivo.open("emails.dat", ios::in | ios::binary);

    if (archivo.fail()){

        cout << "Ocurrio un error al abrir el archivo" << endl;

        exit(1);

    }

    else {

        archivo.seekp(apuntador(i), ios\_base::beg);

        archivo.read((char\* )&c, sizeof(Correo));

        archivo.close();

        cout << "Correo encontrado!" << endl;

        cout << "ID: " << c.getId() << endl;

        cout << "Remitente: " << c.getRemitente() << endl;

        cout << "Destinatario: " << c.getDestinatario() << endl;

        cout << "Copia carbono: " << c.getCopiaCarbono() << endl;

        cout << "Copia carbono ciega: " << c.getCopiaCarbonoCiega() << endl;

        cout << "Asunto: " << c.getAsunto() << endl;

        cout << "Contenido: " << c.getContenido() << endl;

    }

}

void Archivo::modificar(unsigned long int& i)

{

    int opcion, bandera;

    int find;

    char auxRemitente[20];

    char auxDestinatario[50];

    char auxCopiaCarbono[50];

    char auxCopiaCarbonoCiega[20];

    char auxAsunto[50];

    Correo correo;

    archivo.open("emails.dat", ios::in | ios::binary);

    fstream temp("temporal.dat",ios::out | ios::binary);

    if (!archivo.good()){

        cout << "Error no se encontro el archivo...!" << endl;

        system("PAUSE");

    }

    else {

            archivo.seekp(apuntador(i), ios\_base::beg);

            archivo.read((char\* )&correo, sizeof(Correo));

            if (i == correo.getId())

            {

                cout << "ID: " << correo.getId() << endl;

                cout << "Remitente: " << correo.getRemitente() << endl;

                cout << "Destinatario: " << correo.getDestinatario() << endl;

                cout << "Copia carbono: " << correo.getCopiaCarbono() << endl;

                cout << "Copia carbono ciega: " << correo.getCopiaCarbonoCiega() << endl;

                cout << "Asunto: " << correo.getAsunto() << endl;

                cout << "Contenido: " << correo.getContenido() << endl;

                cout << "-----------------------------------------" << endl;

                    cout << "\t\tMenu modificar\n";

                cout << "-----------------------------------------" << endl;

                    cout << "1.- Modificar remitente\n";

                    cout << "2.- Modificar destinatario\n";

                    cout << "3.- Modificar copia carbono\n";

                    cout << "4.- Modificar copia carbono ciega\n";

                    cout << "5.- Modificar asunto\n";

                    cout << "6.- Modificar contenido\n";

                    cout << "0.- SALIR\n";

                    cout << "Introduce la opcion: >" << endl;

                    cin >> opcion;

                    switch(opcion)

                    {

                        case 1:{

                                cout << "Este es el remitente: " << correo.getRemitente() << endl;

                                cout << "Nuevo Remitente: " << endl;

                                cin >> auxRemitente;

                                char \*cp;

                                cp = auxRemitente;

                                temp << cp;

                                correo.setRemitente(cp);

                                temp.seekp(apuntador(i), ios\_base::beg);

                                temp.write((char\* )&correo, sizeof(correo));

                                cout << endl;

                                cout << "Contenido cambiado" << endl;

                                break;

                            }

                        case 2:

                                {

                                    cout << "Este es el destinatario: " << correo.getDestinatario() << endl;

                                    cout << "Nuevo destinatario: " << endl;

                                    cin >> auxDestinatario;

                                    char \*cp;

                                    cp = auxDestinatario;

                                    temp << cp;

                                    correo.setDestinatario(cp);

                                    temp.seekp(apuntador(i), ios\_base::beg);

                                    temp.write((char\* )&correo, sizeof(correo));

                                    cout << endl;

                                    cout << "Contenido cambiado con exito!" << endl;

                                    break;

                                }

                            case 3:

                            {

                                cout << "Este es la copia carbono: " << correo.getCopiaCarbono() << endl;

                                cout << "Nueva copia carbono: " << endl;

                                cin >> auxCopiaCarbono;

                                char \*cp;

                                cp = auxCopiaCarbono;

                                temp << cp;

                                correo.setCopiaCarbono(cp);

                                temp.seekp(apuntador(i), ios\_base::beg);

                                temp.write((char\* )&correo, sizeof(correo));

                                cout << endl;

                                cout << "Contenido cambiado con exito!" << endl;

                                break;

                            }

                            case 4:

                            {

                                cout << "Este es la copia carbono ciega: " << correo.getCopiaCarbonoCiega() << endl;

                                cout << "Nueva copia carbono ciega: " << endl;

                                cin >> auxCopiaCarbonoCiega;

                                char \*cp;

                                cp = auxCopiaCarbonoCiega;

                                temp << cp;

                                correo.setCopiaCarbonoCiega(cp);

                                temp.seekp(apuntador(i), ios\_base::beg);

                                temp.write((char\* )&correo, sizeof(correo));

                                cout << endl;

                                cout << "Contenido cambiado con exito!" << endl;

                                break;

                            }

                            case 5:

                            {

                                cout << "Este es el asunto: " << correo.getAsunto() << endl;

                                cout << "Nuevo asunto: " << endl;

                                cin >> auxAsunto;

                                char \*cp;

                                cp = auxAsunto;

                                temp << cp;

                                correo.setAsunto(cp);

                                temp.seekp(apuntador(i), ios\_base::beg);

                                temp.write((char\* )&correo, sizeof(correo));

                                cout << endl;

                                cout << "Contenido cambiado con exito!" << endl;

                                break;

                            }

                            case 6:

                            {

                                cout << "Este es el contenido: " << correo.getContenido() << endl;

                                cout << "Nuevo contenido: " << endl;

*// cin >> auxAsunto;*

*// char \*cp;*

*// cp = auxAsunto;*

*// temp << cp;*

                                correo.setContenido();

                                temp.seekp(apuntador(i), ios\_base::beg);

                                temp.write((char\* )&correo, sizeof(correo));

                                cout << endl;

                                cout << "Contenido cambiado con exito!" << endl;

                                break;

                            }

                            case 0:

                                break;

                            default:

                                system("cls");

                                cout << "Opcion invalida!\n" << "Ingrese una opcion correcta\n";

                                system("PAUSE");

                                system("cls");

                                break;

                }

                bandera = 1;

                if (bandera != 1)

                {

                    cout << "Codigo no encontrado" << endl;

                    system("PAUSE");

                }

            archivo.close();

            cout << "Finalizado" << endl;

            temp.close();

            remove("emails.dat");

            rename("temporal.dat","emails.dat");

        }

    }

}

Diagrama de secuencia (UML 2.0):





