



PROYECTO FINAL BASE DE DATOS DE UNIVERSIDAD

INTEGRANTES DEL GRUPO:

- ❑ Carpio Peña, Josue
- ❑ León Zarate, Ariana
- ❑ Rodriguez Pinto, Anthony

Docente: DSc. Manuel Eduardo Loaiza
Fernández

Departamento de de ciencia de la computación
Universidad católica San Pablo
Semestre 2021-II
Arequipa-Perú

PASOS

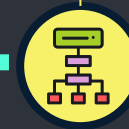
Llamar a las
bibliotecas

Step 1



Crear los atributos
y métodos de cada
clase

Step 3



Step 2



Crear las clases que
usaremos

Step 4

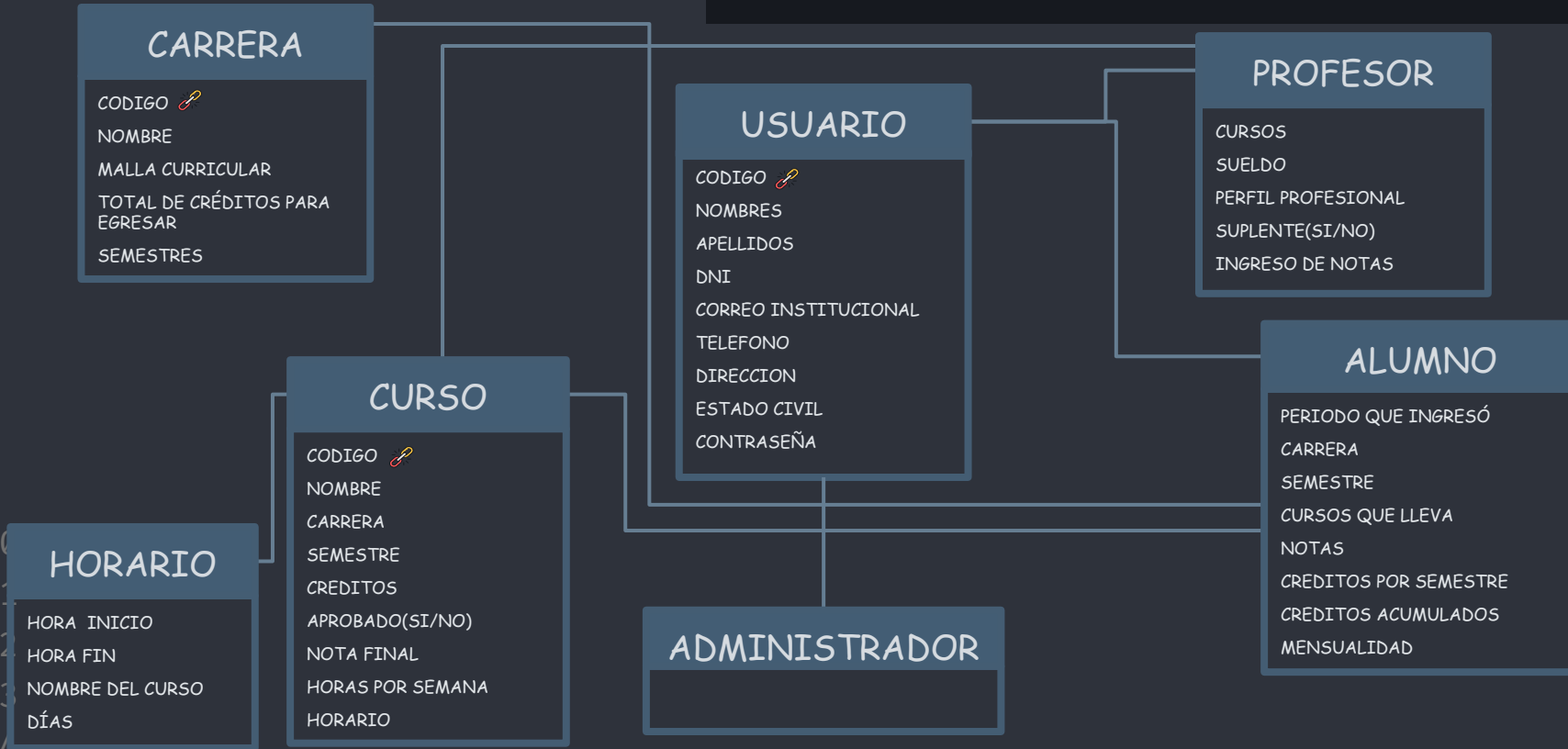


Definir los métodos

PROYECTO FINAL: BASE DE DATOS

Estructura

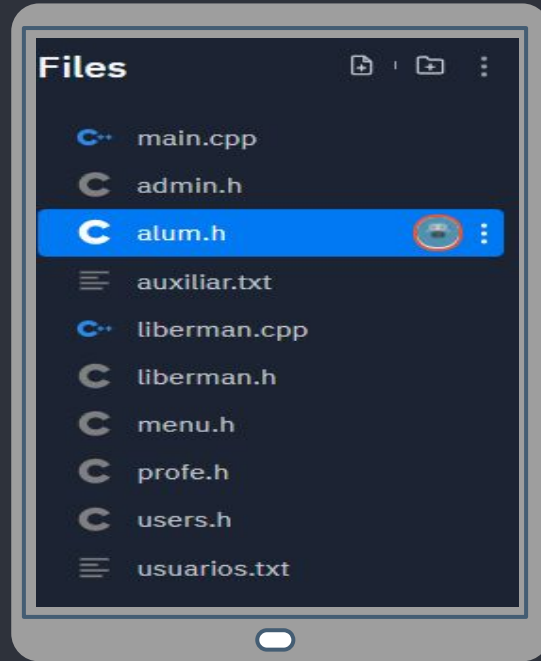
1
2
3
4
5
6
7
8
9
10
11
12
13
14



Archivos{

1
2
3
4
5
6
7
8
9
10
11
12
13
14

}



| CRONOGRAMA | SEMANA 2 | | |
|---|----------|---------|--------|
| | JOSUE | ANTHONY | ARIANA |
| MÉTODO ADMINISTRADOR (MODIFICAR DATOS) | | X | |
| MÉTODO ADMINISTRADOR (MODIFICAR DATOS PERSONALES) | | X | |
| MÉTODO PROFESOR (MODIFICAR DATOS PERSONALES) | | | X |
| MÉTODO PROFESOR (MODIFICAR NOTAS) | | | X |
| MÉTODO ALUMNO (MODIFICAR DATOS PERSONALES) | X | | |
| MÉTODO ALUMNO (VER NOTAS) | X | | |
| MÉTODOS MOSTRAR DATOS OTROS USUARIOS | X | X | X |

Main.cpp {

```
//Realizado en Replit
/*Grupo: Las chicas super poderosas          CIENCIA DE LA COMPUTACIÓN I  CCOMP2-1
INTEGRANTES:
- Josue Carpio
- Ariana Leon
- Anthony Rodriguez */

#include "liberman.h"

int main()
{
    Menu inicio;
    inicio.menuPrincipal();
    return 0;
}
```

}

Class Usuario {

```
1 |  
2 class Usuario  
3 {  
4     public:  
5     string contrasena;  
6     string codigo;  
7     string nombres;  
8     string apellidos;  
9     string dni;  
10    string correoInsti;  
11    string celular;  
12    string direccion;  
13    string estdCiv;  
14  
15    Usuario();  
16 };
```

```
13 Usuario :: Usuario()  
14 {  
15     contrasena = " ";  
16     codigo = " ";  
17     nombres = " ";  
18     apellidos = " ";  
19     dni = " ";  
20     correoInsti = " ";  
21     celular = " ";  
22     direccion = " ";  
23     estdCiv = " ";  
24  
25 }
```

Class Profesor {

```
class Profesor : Usuario
{
    public:
        string  cursosEnsenas;
        string  sueldo;
        string  perfilProf;
        string  suplente;
        Profesor();
        //void crearProf();
        //void menuProf();
        //void modificarNotas();
        //void modificarDatosP();
        //void mostrarDatosP();

        //friend class Admin;
        //friend class Alumno;
};
```

```
32  Profesor :: Profesor()
33  {
34      cursosEnsenas = " ";
35      sueldo = " ";
36      perfilProf = " ";
37      suplente = " ";
38  }
```


Class Alumno {

```
1
2  class Alumno : Usuario
3  {
4      public:
5          string carrera;
6          string cursos;
7          string periodoIngreso;
8          int semestre;
9          int creditosSemestre;
10         int creditosAcumulados;
11         double notas;
12         float mensualidad;
13
14         Alumno();
15         void crearAlumno();
16         //void menuAlumno();
17         //void modificarDatosA();
18         //void mostrarDatosA();
19
20     };
```

```
40  Alumno :: Alumno()
41  {
42      carrera = " ";
43      cursos = " ";
44      periodoIngreso = " ";
45      semestre = 0;
46      creditosSemestre = 0;
47      creditosAcumulados = 0;
48      notas = 0;
49      mensualidad = 0;
50  }
```

Class Administrador {

```
1  class Admin : Usuario
2
3
4
5
6  {
7
8      public:
9      Admin();
10
11  };
12
13
14 }
```

```
27  Admin :: Admin()
28  {
29      Usuario();
30  }
```

Menu Principal {

```
817 void Menu :: menuPrincipal()
818 {
819     int opc;
820     do
821     {
822         cout << "\t\t***Universidad las chicas superpoderosas***\t\t"<<endl;
823         cout << "[1] Administrador" << endl;
824         cout << "[2] Profesor" << endl;
825         cout << "[3] Alumno" << endl;
826         cout << "[4] Salir" << endl;
827         cout << "Elija una opción..." << endl;
828         cin >> opc;
829         std::cout << "\033[H\033[2J\033[3J";
830         switch (opc)
831         {
832             case 1:
833             {
834                 menuAdmin();
835                 break;
836             }
```

```
837         case 2:
838         {
839             menuProf();
840             break;
841         }
842         case 3:
843         {
844             menuAlumno();
845             break;
846         }
847         case 4:
848         {
849             break;
850         }
851         default:
852         {
853             cout << "Opción no válida" << endl;
854             break;
855         }
856     }
857 }
858 while(opc != 4 );
859 }
```

Menú Profesor {

```
722 void Menu :: menuProf()  
723 {  
724     int opc;  
725     do  
726     {  
727         cout << "\t\t***PROFESOR***\t\t" << endl;  
728         cout << "[1] Modificar datos personales" << endl;  
729         cout << "[2] Modificar notas" << endl;  
730         cout << "[3] Mostrar Usuario" << endl;  
731         cout << "[4] Mostrar datos Propios" << endl;  
732         cout << "[5] Salir" << endl;  
733         cout << "Elija una opción..." << endl;  
734         cin >> opc;  
735         std::cout << "\033[H\033[2J\033[3J";  
736         switch (opc)  
737         {  
738             case 1:  
739             {  
740                 //modificarDatosPer();  
741                 break;  
742             }
```

```
743         case 2:  
744         {  
745             //modificarNotas();  
746             break;  
747         }  
748         case 3:  
749         {  
750             //mostrarUsuario();  
751             break;  
752         }  
753         case 4:  
754         {  
755             //mostrarProfesor();  
756             break;  
757         }  
758         case 5:  
759         {  
760             break;  
761         }  
762         default:  
763         {  
764             cout << "Opción no válida" << endl;  
765             break;  
766         }  
767     }  
768 }  
769 while(opc != 5 );  
770 }
```

Menú Alumno {

```
772 void Menu :: menuAlumno()
773 {
774     int opc;
775     do
776     {
777         cout << "\t\t***ALUMNO***\t\t" << endl;
778         cout << "[1] Modificar datos personales" << endl;
779         cout << "[2] Mostrar Datos" << endl;
780         cout << "[3] Mostrar Usuario" << endl;
781         cout << "[4] Salir" << endl;
782         cout << "Elija una opción..." << endl;
783         cin >> opc;
784         std::cout << "\033[H\033[2J\033[3J";
785         switch (opc)
786         {
787             case 1:
788             {
789                 //modificarDatosPer();
790                 break;
791             }
```

```
792         case 2:
793         {
794             //mostrarAlumno();
795             break;
796         }
797         case 3:
798         {
799             //mostrarUsuario();
800             break;
801         }
802         case 4:
803         {
804             break;
805         }
806         default:
807         {
808             cout << "Opción no válida" << endl;
809             break;
810         }
811     }
812 }
813 }
814 while(opc != 4 );
815 }
```

Menú Administrador {

```
672 void Menu :: menuAdmin()
673 {
674     int op;
675     do
676     {
677         cout << "\t\t***ADMINISTRADOR***\t\t" << endl;
678         cout << "[1] Crear Usuario" << endl;
679         cout << "[2] Modificar Usuario" << endl;
680         cout << "[3] Mostrar Datos de Usuario" << endl;
681         cout << "[4] Eliminar Usuario" << endl;
682         cout << "[5] Salir" << endl;
683         cout << "Elija una opción..." << endl;
684         cin >> op;
685         std::cout << "\033[H\033[2J\033[3J";
686         switch (op)
687         {
688             case 1:
689             {
690                 crearUsuario();
691                 break;
692             }
```

```
693         case 2:
694         {
695             modificarUsuario();
696             break;
697         }
698         case 3:
699         {
700             verUsuarioAux();
701             break;
702         }
703         case 4:
704         {
705             //eliminarUsuario();
706             break;
707         }
708         case 5:
709         {
710             break;
711         }
712         default:
713         {
714             cout << "Opción no válida" << endl;
715             break;
716         }
717     }
718 }
719 while(op != 5 );
720 }
```

Función Error{

```
14 void error()  
15 {  
16     cout<<"No se pudo abrir el archivo de registros, asegurese que el archivo se  
17     encuentre en\n";  
18     cout<<"la misma ubicación que el programa o que el archivo de texto se  
19     llame: \n";  
20     cout<<"usuarios.txt, si el archivo tiene otro nombre renómbrelo al ya  
21     mencionado\n\n";  
22 }
```

}

Class Menu{

```
163 class Menu
164 {
165     public:
166     Menu();
167     class Usuario usuario;
168     class Admin admin;
169     class Profesor profesor;
170     class Alumno alumno;
171
172     void menuPrincipal();
173     void crearAdmin();
174     void crearProf();
175     void crearAlumno();
176
177     //USUARIO
178     void crearUsuarioAux();
179     void verUsuarioAux();
```

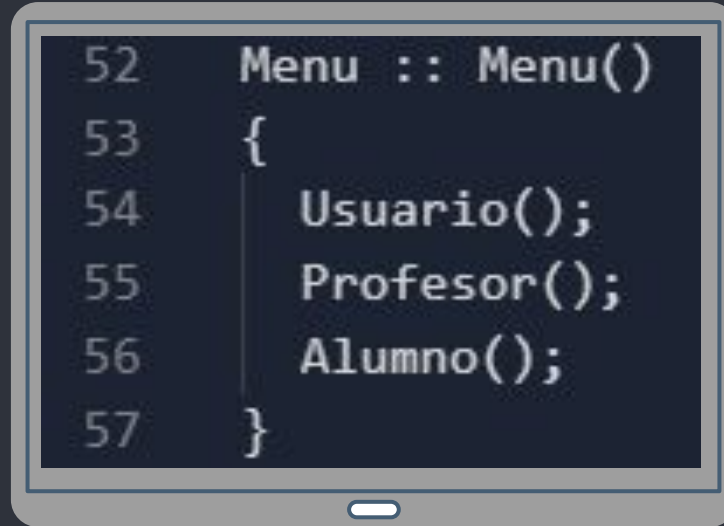
```
}
```

```
181 //ADMIN
182 void menuAdmin();
183 void crearUsuario();
184 void eliminarUsuario();
185 void verUsuario();
186 void verUsuarioAd();
187 void modificarUsuario();
188
189 //PROF
190 void menuProf();
191 void modificarNotas();
192 void modificarDatosP();
193 void mostrarDatosP();
194 void verUsuarioP();
195
196 //ALUMNO
197 void menuAlumno();
198 void modificarDatosA();
199 void mostrarDatosA();
200 void verUsuarioA();
201 };
```



```
1  Class Menu{
```

```
14 }
```



Método CrearUsuario{

```
327 void Menu :: crearUsuario()
328 {
329     cout << endl << "\t\t\t\t\t*****Crear un Usuario*****\t\t\t\t\t";
330     char opc;
331     cout << endl << "Administrador(D), Profesor(P), Alumno(A): " ;
332     cin >> opc;
333     switch (opc)
334     {
335         case 'D':
336         {
337             crearAdmin();
338             break;
339         }
340         case 'P':
341         {
342             crearProf();
343             break;
344         }
345         case 'A':
346         {
347             crearAlumno();
348             break;
349         }
350     }
351 }
```

}

```
1 Class CrearAdmin{
```

```
2  
3  
4 void Menu :: crearAdmin()  
5 {  
6     ofstream escritura;  
7     ifstream verificador;  
8     string auxcodigo;  
9     bool coincidencia=false;  
10    verificador.open("admin.txt",ios::in);  
11    escritura.open("admin.txt",ios::app);  
12    if(escritura.is_open() && verificador.is_open())
```

```
13  
14 }
```

Class CrearAdmin{

```
12     if(escritura.is_open() && verificador.is_open())
13     {
14         cin.ignore(1000000, '\n');
15         cout<<"Ingresa el código del usuario: ";
16         getline(cin, auxcodigo);
17         if(auxcodigo==" ")
18         {
19             do
20             {
21                 cout<<"código de usuario no válido!, intentalo nuevamente: ";
22                 getline(cin, usuario.codigo);
23             }
24             while(auxcodigo==" ");
25         }
```

```
14 }
```

Class CrearAdmin{

```
2
3
4
5
6
7
8
9
10
11
12
13
14 }
```



```
26         do
27         {
28             verificador.seekg(0);
29             getline(verificador, admin.codigo);
30             while(!verificador.eof())
31             {
32                 getline(verificador, admin.contrasena);
33                 getline(verificador, admin.nombres);
34                 getline(verificador, admin.apellidos);
35                 getline(verificador, admin.dni);
36                 getline(verificador, admin.correoInsti);
37                 getline(verificador, admin.celular);
38                 getline(verificador, admin.direccion);
39                 getline(verificador, admin.estdCiv);
```

Class CrearAdmin{

```
41     if(admin.codigo==auxcodigo)
42     {
43         coincidencia=true;
44         cout<<"\n\nYa existe un usuario con ese código!\n\n";
45         cout<<"El usuario con ese código es: "<<admin.nombres<<"\n\n";
46         cout<<"Ingresa un código válido: ";
47         getline(cin,auxcodigo);
48         if(auxcodigo=="")
49         {
50             do
51             {
52                 cout<<"\ncódigo de usuario no válido!, intentalo nuevamente: ";
53                 getline(cin,auxcodigo);
54             }
55             while(auxcodigo=="");
```

```
14 }
```

Class CrearAdmin{

2
3
4
5
6
7
8
9
10
11
12
13
14

```
56         break;  
57     }  
58 }  
59     getline(verificador, admin.codigo);  
60 }  
61     if(verificador.eof() && auxcodigo != admin.codigo)  
62     coincidencia = false;  
63 }  
64 while(coincidencia == true);  
65
```

}

Class CrearAdmin{

```
66     cout << "\\033[H\\033[2J\\033[3J";
67     cout<<"\\t\\t\\t***Crear un usuario***\\t\\t\\t\\n\\n";
68     admin.codigo=auxcodigo;
69     cout << endl <<"Ingrese los siguientes datos: ";
70     cin.ignore(1000000, '\\n');
71     cout<<endl << "Contrasena: ";
72     getline(cin,admin.contrasena);
73     cout << endl << "Nombres: ";
74     getline(cin, admin.nombres);
75     cout << endl << "Apellidos: ";
76     getline(cin, admin.apellidos);
77     cout << endl << "DNI: ";
78     getline(cin,admin.dni);
79     cout << endl << "Correo institucional: ";
80     getline(cin,admin.correoInsti) ;
81     cout << endl << "Celular: ";
82     getline(cin, admin.celular);
83     cout << endl << "Dirección: ";
84     getline(cin,admin.direccion);
85     cout << endl << "Estado civil: ";
86     getline(cin,admin.estdCiv);
```

}

Class CrearAdmin{

```
88      escritura<<admin.codigo<<endl<<admin.contrasena<<endl<<admin.nombres<<endl<< admin.apellidos  
89      <<endl << admin.dni<<endl << admin.correoInsti<<endl<<admin.celular<<  
90      endl<<admin.direccion<<endl<<admin.estdCiv<<endl;  
91  
92      cout<<"El registro se ha completado correctamente.\n\n";  
93      cout << "\033[H\033[2J\033[3J";  
94  }  
95  else  
96  {  
97      error();  
98  }  
99  escritura.close();  
100  verificador.close();  
101  }
```

}

Método VerUsuario{

```

59 void Menu :: verUsuarioD()
60 {
61     string auxCodigo;
62     ifstream mostrar;
63     bool encontrado=false;
64     mostrar.open("admin.txt",ios::in);
65     if(mostrar.is_open())
66     {
67         fflush(stdin);
68         cout<<"\t\t\t***Ver datos de un Administrador***\t\t\t\t\n\n";
69         cout<<"Ingresa el código del Administrador que deseas consultar detalles:
70 ";
71         cin.ignore(1000000, '\n');
72         getline(cin,auxCodigo);
73         getline(mostrar,admin.codigo);
74         while(!mostrar.eof())
75         {
76             getline(mostrar,admin.contrasena);
77             getline(mostrar,admin.nombres);
78             getline(mostrar,admin.apellidos);
79             getline(mostrar,admin.dni);
80             getline(mostrar,admin.correoInsti);
81             getline(mostrar,admin.celular);
82             getline(mostrar,admin.direccion);
83             getline(mostrar,admin.estdCiv);

```

```

84         if(auxCodigo== admin.codigo)
85         {
86             encontrado=true;
87             cout<<"\n\nRegistro Encontrado\n\n";
88             cout<<"Código: "<<admin.codigo<<endl;
89             cout<<"Nombres: "<<admin.nombres<<endl;
90             cout<<"Apellidos: "<<admin.apellidos<<endl;
91             cout<<"DNI: "<<admin.dni<<endl;
92             cout<<"Correo Institucional: "<<admin.correoInsti<<endl;
93             cout<<"Celular: "<<admin.celular<<endl;
94             cout<<"Direccion: "<<admin.direccion<<endl;
95             cout<<"Estado Civil: "<<admin.estdCiv<<endl;
96             cout<<"\n\n";
97         }
98         getline(mostrar,admin.codigo);
99     }
100     if(encontrado==false)
101     {
102         cout<<"\n\nNo se encontro el registro: "<<auxCodigo<<"\n\n";
103     }
104 }
105 else
106 {
107     error();
108 }
109 mostrar.close();
110 }
111 }

```

Método ModificarUsuario{

```

67 void Menu::modificarUsuarioD()
68 {
69     ifstream lectura;
70     ifstream verificador;
71     ofstream auxiliar;
72     string auxCodigo;
73     string codigoModif;
74     string auxContra;
75     string auxNombres;
76     string auxApellidos;
77     string auxDNI;
78     string auxCorreoInsti;
79     string auxCelular;
80     string auxDireccion;
81     string auxEstdCiv;
82     bool encontrado=false;
83     bool coincidencia=false;
84     bool mismoCodigo=false;
85     lectura.open("admin.txt",ios::in);
86     verificador.open("admin.txt",ios::in);
87     auxiliar.open("auxiliar.txt",ios::out);
88     cout<<"\t\t\t\t***Modificar los datos de un usuario***\t\t\t\t\n\n";
89     if(lectura.is_open() && verificador.is_open() && auxiliar.is_open())
90     {
91         fflush(stdin);
92         cout<<"Ingresa el codigo del usuario que deseas modificar: ";
93         cin.ignore(1000000, '\n');
94         getline(cin,auxCodigo);

```

```

14 }

```

```

96     if(auxCodigo==" ")
97     {
98         do
99         {
100             cout<<"codigo de usuario no valido!, intentalo nuevamente: ";
101             getline(cin,auxCodigo);
102         }
103         while(auxCodigo==" ");
104     }
105     codigoModif=auxCodigo;
106
107     getline(lectura,admin.codigo);
108     while(!lectura.eof())
109     {
110
111         getline(lectura,admin.contrasena);
112         getline(lectura,admin.nombres);
113         getline(lectura,admin.apellidos);
114         getline(lectura,admin.dni);
115         getline(lectura,admin.correoInsti);
116         getline(lectura,admin.celular);
117         getline(lectura,admin.direccion);
118         getline(lectura,admin.estdCiv);
119
120         if(auxCodigo==admin.codigo)
121         {
122             encontrado=true;
123             mostrarRegistroD(codigoModif);
124             cout<<"*****";
125             cout<<"\n\n";
126             cout<<"Ingresa la nueva informacion para el usuario\n\n";
127             fflush(stdin);
128             cout<<"Ingresa el codigo del usuario: ";
129             getline(cin,auxCodigo);

```

Método ModificarUsuario{

```
130     if(auxCodigo==codigoModif)
131     {
132         mismoCodigo=true;
133     }
134     if(mismoCodigo==false)
135     {
136         do
137         {
138             if(auxCodigo==codigoModif)
139             {
140                 coincidencia=false;
141                 break;
142             }
143             verificador.seekg(0);
144             getline(verificador,admin.codigo);
145             while(!verificador.eof())
146             {
147                 getline(verificador,admin.contrasena);
148                 getline(verificador,admin.nombres);
149                 getline(verificador,admin.apellidos);
150                 getline(verificador,admin.dni);
151                 getline(verificador,admin.correoInsti);
152                 getline(verificador,admin.celular);
153                 getline(verificador,admin.direccion);
154                 getline(verificador,admin.estdCiv);
155             }
```

```
156     if(auxCodigo==admin.codigo)
157     {
158         coincidencia=true;
159         cout<<"\n\nYa existe un usuario con ese codigo!\n\n";
160         cout<<"El usuario con ese codigo es: "<<admin.nombres<<"\n\n";
161         cout<<"Ingresa un codigo valido! ";
162         getline(cin,auxCodigo);
163     }
164     if(auxCodigo==" ")
165     {
166         do
167         {
168             cout<<"\ncodigo de usuario no valido! ";
169             cout<<"intentalo nuevamente: ";
170             getline(cin,auxCodigo);
171         }
172         while(auxCodigo=="");
173     }
174     break;
175 }
176
177     getline(verificador,admin.codigo);
178 }
179
180     if(verificador.eof() && auxCodigo!=admin.codigo)
181     {
182         coincidencia=false;
183     }
184
185 }
186 while(coincidencia==true);
187 }
```

Método ModificarUsuario{

```
188 cout << "\033[H\033[2J\033[3J";
189 cout<< "***Modificar los datos de un usuario***\n\n";
190 cout<<"Ingresa el codigo del administrador que deseas modificar: ";
191 cout<<codigoModif;
192 mostarRegistroD(codigoModif);
193 cout<<"*****";
194 cout<<endl;
195 cout<<"Ingresa la nueva informacion para el usuario\n";
196 cout<<"Ingresa el codigo del usuario: ";
197 cout<<auxCodigo;
198 fflush(stdin);
199 cout<<endl << "Contrasena: ";
200 getline(cin,auxContra);
201 fflush(stdin);
202 cout<<"Nombres: ";
203 getline(cin,auxNombres);
204 fflush(stdin);
205 cout << endl << "Apellidos: ";
206 getline(cin, auxApellidos);
207 fflush(stdin);
208 cout << endl << "DNI: ";
209 getline(cin,auxDNI);
210 fflush(stdin);
```

```
211 cout << endl << "Correo institucional: ";
212 getline(cin,auxCorreoInsti) ;
213 fflush(stdin);
214 cout << endl << "Celular: ";
215 getline(cin, auxCelular);
216 fflush(stdin);
217 cout << endl << "Dirección: ";
218 getline(cin,auxDireccion);
219 fflush(stdin);
220 cout << endl << "Estado Civil: ";
221 getline(cin,auxEstdCiv);
```

}

Método ModificarUsuario{

```
223     auxiliar<<auxCodigo<<endl<<auxContra<<endl<<auxNombres<<endl<<  
224     auxApellidos<<auxDNI<<endl<<auxCorreoInsti<<endl<<auxCelular<<  
225     endl<<auxDireccion<<endl<<auxEstdCiv<<endl;  
226     cout<<"El Registro se ha modificado correctamente.\n\n";  
227     }  
228     else  
229     {  
230     auxiliar<<admin.codigo<<endl<<admin.contrasena<<endl<<admin.  
231     nombres<<endl<<admin.apellidos<<endl<<admin.  
232     dni<<endl<<endl<<admin.correoInsti<<endl<<endl<<admin.  
233     celular<<endl<<endl<<admin.direccion<<endl<<admin.  
234     estdCiv<<endl;  
235     }  
236     getline(lectura,admin.codigo);  
237     }  
238     }
```

```
240     else  
241     {  
242     error();  
243     }  
244     if(encontrado==false)  
245     {  
246     cout<<"\n\nNo se encontro ningun registro con ese codigo!\n\n";  
247     }  
248     lectura.close();  
249     verificador.close();  
250     auxiliar.close();  
251     remove("admin.txt");  
252     rename("auxiliar.txt","admin.txt");  
253     }
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14



GIT HUB {

<https://github.com/Anthony-Rodriguez18/PROYECTO-BASE-DE-DATOS>



}



1
2
3
4
5
6
7
8
9
10
11
12
13
14



```
int main {  
    COUT<<"GRACIAS";  
}
```



}

