



PROYECTO FINAL BASE DE DATOS DE UNIVERSIDAD

INTEGRANTES DEL GRUPO:

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

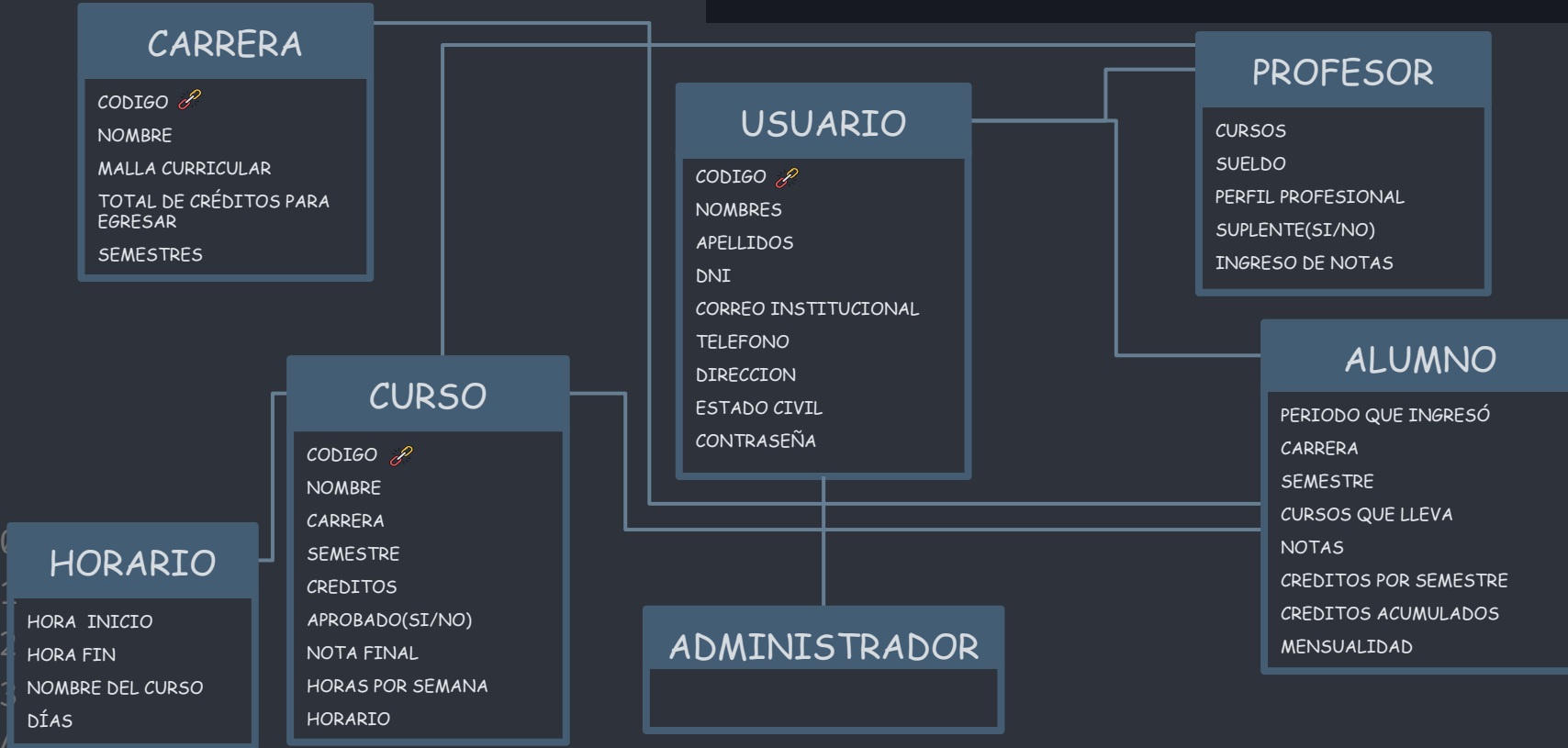
Docente: DSc. Manuel Eduardo Loaiza
Fernández

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

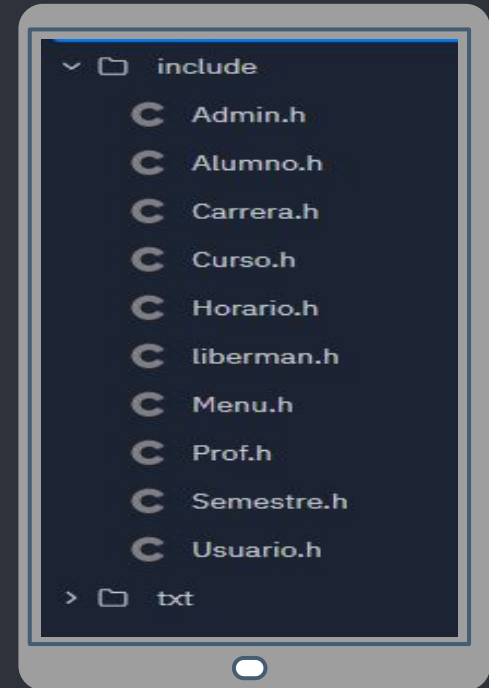
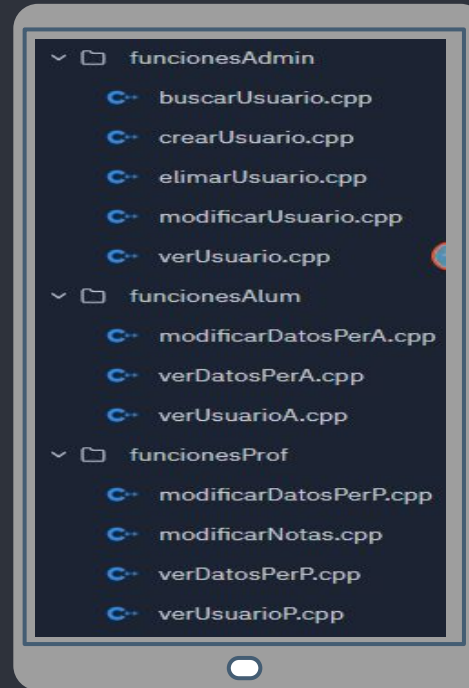
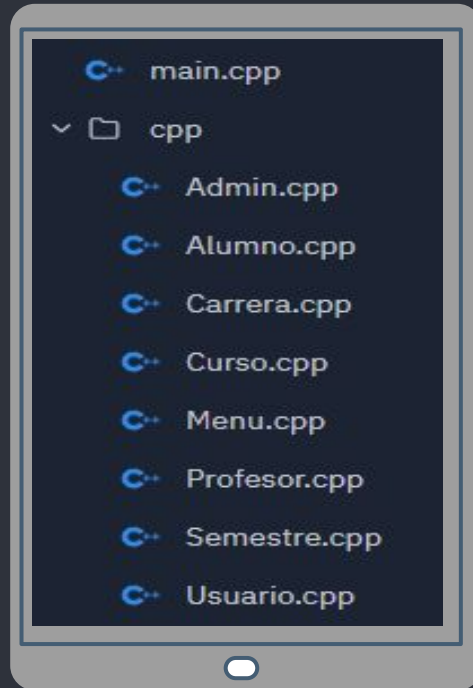
PROYECTO FINAL: BASE DE DATOS

Estructura

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



Archivos{



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  #ifndef USUARIO_H_
2  #define USUARIO_H_
3
4  class Usuario
5  {
6      protected:
7          string contrasena;
8          int codigo;
9          string nombres;
10         string apellidos;
11         string dni;
12         string correoInsti;
13         string celular;
14         string direccion;
15         string estdCiv;
16
17     public:
18         Usuario( );
19         Usuario(int ,string ,string ,string ,string ,string ,string ,string );
20         ~Usuario();
21         void datos();
```

```
22     //Getters
23     int getCodigo(){return this->codigo;}
24     string getContrasena(){return this->contrasena;}
25     string getNombres(){return this->nombres;}
26     string getApellidos(){return this->apellidos;}
27     string getCorreoInsti(){return this->correoInsti;}
28     string getDNI(){return this->dni;}
29     string getCelular(){return this->celular;}
30     string getDireccion(){return this->direccion;}
31     string getEstadoCiv(){return this->estdCiv;}
32
33     //Setters
34     void setUsuario(string ,string ,string ,string ,string ,string ,string ,string
35     ,string);
36
37 #endif
```

Class Usuario {

```
1  #include "../include/liberman.h"
2
3  Usuario :: Usuario()
4  {
5      this->contrasena = " ";
6      this->codigo = 0;
7      this->nombres = " ";
8      this->apellidos = " ";
9      this->dni = " ";
10     this->correoInsti = " ";
11     this->celular = " ";
12     this->direccion = " ";
13     this->estdCiv = " ";
14 }
15
16 Usuario :: Usuario(int codigo,string nombres,string apellidos,string dni,string
17 correoInsti,string celular,string direccion,string estdCiv)
18 {
19     this->codigo = codigo;
20     this->nombres = nombres;
21     this->apellidos = apellidos;
22     this->dni = dni;
23     this->correoInsti = correoInsti;
24     this->celular = celular;
25     this->direccion = direccion;
26     this->estdCiv = estdCiv;
27 }
```

```
28 Usuario :: ~Usuario()
29 {
30 }
31
32 void Usuario::datos()
33 {
34     cout << "Código: " << codigo<<endl;
35     cout << "Contraseña: " << contrasena <<endl;
36     cout << "Nombres: " << nombres <<endl;
37     cout << "Apellidos: " << apellidos <<endl;
38     cout << "DNI: " << dni <<endl;
39     cout << "Correo Institucional: " << correoInsti<<endl;
40     cout << "Celular: " << celular <<endl;
41     cout << "Dirección: " << direccion <<endl;
42     cout << "Estado Civil: " << estdCiv<<endl;
43 }
44
45
46 void setUsuario(string contrasena,string codigo,string nombres,string apellidos,
47 string dni,string correoInsti,string celular,string direccion,string estdCiv)
48 {
49     contrasena = contrasena;
50     codigo = codigo;
51     nombres =nombres;
52     apellidos = apellidos;
53     dni =dni;
54     correoInsti = correoInsti ;
55     celular = celular;
56     direccion = direccion;
57     estdCiv = estdCiv;
58 }
```

Class Profesor {

```
1  #ifndef PROFESOR_H_
2  #define PROFESOR_H_
3
4  class Profesor : public Usuario
5  {
6  private:
7      string  cursosEnsenas;
8      string  sueldo;
9      string  perfilProf;
10     string  suplente;
11
12 public:
13     Profesor();
14     Profesor(int, string,string ,string ,string ,string ,string ,string ,string ,string ,
15             string, string, string, string);
16     ~Profesor();
17
18     void datos();
19     void setProfesor(string, string ,string ,string ,string ,string ,string ,string, string,
20                     string, string, string);
21     void setDatosPer( string ,string ,string ,string ,string ,string ,string);
22     void setCodigo(int);
23 };
24
25 #endif
```

```
3  Profesor :: Profesor():Usuario()
4  {
5      cursosEnsenas = " ";
6      sueldo = " ";
7      perfilProf = " ";
8      suplente = " ";
9  }
10
11 Profesor :: Profesor(int codigo,string contrasena,string nombres,string apellidos,
12                     string dni,string correoInsti,string celular,string direccion,string estdCiv,
13                     string cursosEnsenas, string sueldo, string perfilProf, string suplente): Usuario
14                     (codigo, nombres, apellidos, dni, correoInsti, celular, direccion, estdCiv)
15 {
16     this->contrasena = contrasena;
17     this->cursosEnsenas = cursosEnsenas;
18     this->sueldo = sueldo;
19     this->perfilProf = perfilProf;
20     this->suplente = suplente;
21 }
22
23 Profesor :: ~Profesor(){
24 }
```

Class Profesor {

```
23 void Profesor :: setProfesor(string nombres,string apellidos,string dni,string
24 correoInsti,string celular,string direccion,string estdCiv, string cursosEnsen
25 a, string sueldo, string perfilProf, string suplente )
26 {
27     this->nombres = nombres;
28     this->apellidos = apellidos;
29     this->dni = dni;
30     this->correoInsti = correoInsti ;
31     this->celular = celular;
32     this->direccion = direccion;
33     this->estdCiv = estdCiv;
34     this->cursosEnsen = cursosEnsen;
35     this->sueldo = sueldo;
36     this->perfilProf = perfilProf;
37     this->suplente = suplente;
38 }
39
40 void Profesor :: setDatosPer( string nombres,string apellidos,string dni,string
41 celular,string direccion,string estdCiv)
42 {
43     this->nombres =nombres;
44     this->apellidos = apellidos;
45     this->dni =dni;
46     this->celular = celular;
47     this->direccion = direccion;
48     this->estdCiv = estdCiv;
49 }
```

```
49 void Profesor:: setCodigo(int nuevoCodigo)
50 {
51     this->codigo = nuevoCodigo;
52 }
53
54 void Profesor :: datos()
55 {
56     cout << "Código: " << codigo<<endl;
57     cout << "Contraseña: " << contrasena <<endl;
58     cout << "Nombres: " << nombres <<endl;
59     cout << "Apellidos: " << apellidos <<endl;
60     cout << "DNI: " << dni <<endl;
61     cout << "Correo Institucional: " << correoInsti<<endl;
62     cout << "Celular: " << celular <<endl;
63     cout << "Dirección: " << direccion <<endl;
64     cout << "Estado Civil: " << estdCiv<<endl;
65     cout << "Número de cursos que enseña: " << cursosEnsen << endl;
66     cout << "Sueldo: " << sueldo<< endl;
67     cout << "Perfil profesional: " << perfilProf<<endl;
68     cout << "Suplente: " << suplente<<endl;
69 }
70 }
```


Class Alumno {

```

1  #ifndef ALUM_H_
2  #define ALUM_H_
3
4  class Alumno : public Usuario
5  {
6  private:
7      string carrera;
8      string cursos;
9      string periodoIngreso;
10     string semestre;
11     string creditosSemestre;
12     string creditosAcumulados;
13     string notas;
14     string mensualidad;
15
16 public:
17     Alumno();
18     Alumno(int, string ,string ,string ,string ,string ,string ,string ,string ,string ,string ,
19         string , string , string ,string ,string ,string );
20     ~Alumno();
21     void datos();
22     void setDatosPer( string ,string ,string ,string ,string ,string ,string );
23     void setAlumno( string ,string ,string ,string ,string ,string ,string ,string ,string ,string ,
24         string , string , string , string , string , string ,string );
25     void setCodigo(int);
26 #endif

```

```

3  Alumno :: Alumno():Usuario()
4  {
5      carrera = " ";
6      cursos = " ";
7      periodoIngreso = " ";
8      semestre = " ";
9      creditosSemestre = " ";
10     creditosAcumulados = " ";
11     mensualidad = " ";
12 }
13
14 Alumno :: Alumno(int codigo, string contrasena, string nombres, string apellidos,
15     string dni, string correoInsti, string celular, string direccion, string estdCiv, string
16     carrera , string cursos, string periodoIngreso, string semestre, string
17     creditosSemestre, string creditosAcumulados, string mensualidad ) : Usuario(codigo,
18     nombres, apellidos, dni, correoInsti, celular, direccion, estdCiv)
19 {
20     this->contrasena = contrasena;
21     this->carrera = carrera;
22     this->cursos = cursos;
23     this->periodoIngreso = periodoIngreso;
24     this->semestre = semestre;
25     this->creditosSemestre = creditosSemestre;
26     this->creditosAcumulados = creditosAcumulados;
27     this->mensualidad = mensualidad;
28 }
29
30 Alumno :: ~Alumno(){
31 }

```

Class Alumno {

```
29 void Alumno :: setAlumno(string contrasena, string nombres,string apellidos,string
dni,string correoInsti,string celular,string direccion,string estdCiv,string
carrera, string cursos, string periodoIngreso, string semestre,string
creditosSemestre, string creditosAcumulados,string mensualidad)
30 {
31     this->nombres = nombres;
32     this->apellidos = apellidos;
33     this->dni = dni;
34     this->correoInsti = correoInsti ;
35     this->celular = celular;
36     this->direccion = direccion;
37     this->estdCiv = estdCiv;
38     this->carrera = carrera;
39     this->cursos = cursos;
40     this->periodoIngreso = periodoIngreso;
41     this->semestre = semestre;
42     this->creditosSemestre = creditosSemestre;
43     this->creditosAcumulados = creditosAcumulados;
44
45     this->mensualidad = mensualidad;
46 }
47
48 void Alumno:: setCodigo(int nuevoCodigo)
49 {
50     this->codigo = nuevoCodigo;
51 }
```

```
53 void Alumno :: setDatosPer( string nombres,string apellidos,string dni,string
celular,string direccion,string estdCiv)
54 {
55     this->nombres = nombres;
56     this->apellidos = apellidos;
57     this->dni = dni;
58     this->celular = celular;
59     this->direccion = direccion;
60     this->estdCiv = estdCiv;
61 }
62
63 void Alumno::datos()
64 {
65     cout << "Código: " << codigo<<endl;
66     cout << "Contraseña: " << contrasena <<endl;
67     cout << "Nombres: " << nombres <<endl;
68     cout << "Apellidos: " << apellidos <<endl;
69     cout << "DNI: " << dni <<endl;
70     cout << "Correo Institucional: " << correoInsti<<endl;
71     cout << "Celular: " << celular <<endl;
72     cout << "Dirección: " << direccion <<endl;
73     cout << "Estado Civil: " << estdCiv<<endl;
74     cout << "Cursos: " <<cursos<<endl;
75     cout << "Periodo de ingreso: " <<periodoIngreso<<endl;
76     cout << "Semestre: " <<semestre<<endl;
77     cout << "credito del semestre: " <<creditosSemestre<<endl;
78     cout << "Creditos acumulados: " <<creditosAcumulados<<endl;
79     cout << "Notas: " <<notas<<endl;
80     cout << "Mesualidad: " <<mensualidad<<endl;
81 }
```

Class Administrador {

```
1  #ifndef ADMIN_H_
2  #define ADMIN_H_
3
4  class Admin: public Usuario
5  {
6  public:
7      Admin();
8      Admin(int ,string ,string ,string ,string ,string ,string ,string );
9      ~Admin();
10     void datos();
11     void setAdmin(string ,string ,string ,string ,string ,string ,string ,string);
12     void setCodigo(int);
13 };
14
15 #endif
```

```
3  Admin :: Admin()
4  {
5      contrasena = "ADMIN123";
6  }
7
8  Admin :: Admin(int _codigo,string _nombres,string _apellidos,string _dni,string
9  _correoInsti,string _celular,string _direccion,string _estdCiv)
10 {
11     contrasena = "ADMIN123";
12     this->codigo = _codigo;
13     this->nombres = _nombres;
14     this->apellidos = _apellidos;
15     this->dni = _dni;
16     this->correoInsti = _correoInsti ;
17     this->celular = _celular;
18     this->direccion = _direccion;
19     this->estdCiv = _estdCiv;
20 }
21
22 Admin :: ~Admin()
23 { }
```

Class Administrador {

```
24 void Admin :: datos()
25 {
26     cout << "Código: " << codigo<<endl;
27     cout << "Contraseña: " << contrasena <<endl;
28     cout << "Nombres: " << nombres <<endl;
29     cout << "Apellidos: " << apellidos <<endl;
30     cout << "DNI: " << dni <<endl;
31     cout << "Correo Institucional: " << correoInsti<<endl;
32     cout << "Celular: " << celular <<endl;
33     cout << "Dirección: " << direccion <<endl;
34     cout << "Estado Civil: " << estdCiv<<endl;
35 }
```

```
37 void Admin :: setAdmin(string nombres,string apellidos,string dni,string
correoInsti,string celular,string direccion,string estdCiv)
38 {
39     this->nombres = nombres;
40     this->apellidos = apellidos;
41     this->dni = dni;
42     this->correoInsti = correoInsti ;
43     this->celular = celular;
44     this->direccion = direccion;
45     this->estdCiv = estdCiv;
46 }
47
48 void Admin :: setCodigo(int nuevoCodigo)
49 {
50     this->codigo = nuevoCodigo;
51 }
```

```
}
14
```

Menu Principal {

```
236 void Menu :: menuPrincipal()
237 {
238     int opc;
239     do
240     {
241         cout << "\t**Universidad las chicas superpoderosas**\t" << endl;
242         cout << "[1] Administrador" << endl;
243         cout << "[2] Profesor" << endl;
244         cout << "[3] Alumno" << endl;
245         cout << "[4] Salir" << endl;
246         cout << "Elija una opción..." << endl;
247         cin >> opc;
248         std::cout << "\033[H\033[2J\033[3J";
249
250         switch (opc)
251         {
252             case 1:
253             {
254                 menuAdmin();
255                 break;
256             }
```

```
257         case 2:
258         {
259             menuProf();
260             break;
261         }
262         case 3:
263         {
264             menuAlumno();
265             break;
266         }
267         case 4:
268         {
269             cout << "\t\t**BASE DE DATOS FINALIZADA\t\t**";
270             //cargarDatos();
271             break;
272         }
273         default:
274         {
275             cout << "Opción no válida" << endl;
276             break;
277         }
278     }
279     while(opc != 4 );
280 }
281 }
```

Menú Profesor {

```

122 void Menu :: menuProf()
123 {
124     int auxcodigo; string auxcontra;
125     cout<<endl<<"Ingrese su código: ";
126     cin>>auxcodigo;
127     cout<<endl<<"ingrese su contraseña: ";
128     cin>>auxcontra;
129
130     if(loginProf(auxcodigo,auxcontra ))
131     {
132         int opc;
133         do
134         {
135             cout << "\t\t***PROFESOR***\t\t"<<endl;
136             cout << "[1] Modificar datos personales" << endl;
137             cout << "[2] Modificar notas" << endl;
138             cout << "[3] Mostrar Usuario" << endl;
139             cout << "[4] Mostrar datos Propios" << endl;
140             cout << "[5] Salir" << endl;
141             cout << "Elija una opción..." << endl;
142             cin >> opc;
143             std::cout << "\033[H\033[2J\033[3J";
144
145             switch (opc)
146             {
147                 case 1:
148                 {
149                     modificarDatosP(auxcodigo);
150                     break;
151                 }
152             }
153         }
154     }
155 }
    
```

```

152     case 2:
153     {
154         //modificarNotas();
155         break;
156     }
157     case 3:
158     {
159         mostrarDatosUsu();
160         break;
161     }
162     case 4:
163     {
164         mostrarDatosP(auxcodigo);
165     }
166     case 5:
167     {
168         break;
169     }
170     default:
171     {
172         cout << "Opción no válida" << endl;
173         break;
174     }
175 }
176 }
177 while(opc != 5 );
178 }
179 }
180 }
181 }
    
```

Menú Alumno {

```

183 void Menu :: menuAlumno()
184 {
185     int auxcodigo; string auxcontra;
186     cout<<endl<<"Ingrese su código: ";
187     cin >> auxcodigo;
188     cout<<endl<<"Ingrese su contraseña: " ;
189     cin>>auxcontra;
190     if(loginAlumno(auxcodigo, auxcontra))
191     {
192         int opc;
193         do
194         {
195             cout << "\t\t***ALUMNO***\t\t"<<endl;
196             cout << "[1] Modificar datos personales" << endl;
197             cout << "[2] Mostrar Datos Personales" << endl;
198             cout << "[3] Mostrar Usuario" << endl;
199             cout << "[4] Salir" << endl;
200             cout << "Elija una opción..." << endl;
201             cin >> opc;
202             std::cout << "\033[H\033[2J\033[3J";
203             switch (opc)
204             {
205                 case 1:
206                 {
207                     modificarDatosA(auxcodigo);
208                     break;
209                 }

```

```

210             case 2:
211             {
212                 mostrarDatosA(auxcodigo);
213                 break;
214             }
215             case 3:
216             {
217                 mostrarDatosUsu();
218                 break;
219             }
220             case 4:
221             {
222                 break;
223             }
224             default:
225             {
226                 cout << "Opción no válida" << endl;
227                 break;
228             }
229         }
230     }
231 }
232 while(opc != 4 );
233 }
234 }

```


Menú Administrador {

```

51 void Menu :: menuAdmin()
52 {
53     int auxcodigo; string auxcontra;
54     cout <<endl << "Ingrese su código: " ;
55     cin >> auxcodigo;
56     cout <<endl << "Ingrese su contraseña: " ;
57     cin >> auxcontra;
58
59     if(loginAdmin(auxcodigo,auxcontra))
60     {
61         int op;
62         do
63         {
64             cout << "\t\t***ADMINISTRADOR***\t\t"<<endl;
65             cout << "Número de admin: " << contD << endl;
66             cout << "Número de alumnos: " << contA << endl;
67             cout << "Número de profes: " << contP<< endl;
68             cout << "[1] Crear Usuario" << endl;
69             cout << "[2] Modificar Usuario" << endl;
70             cout << "[3] Mostrar Lista de Usuarios" << endl;
71             cout << "[4] Eliminar Usuario" << endl;
72             cout << "[5] Buscar Usuario" << endl;
73             cout << "[6] Salir" << endl;
74             cout << "Elija una opción..." << endl;
75             cin >> op;
76
77             std::cout << "\033[H\033[2J\033[3J";
78
79             switch (op)
80             {
81                 case 1:
82                 {
83                     crearUsuario();
84                     break;
85                 }
86             }

```

```

86             case 2:
87             {
88                 modificarUsuario();
89                 break;
90             }
91             case 3:
92             {
93                 verUsuario();
94                 break;
95             }
96             case 4:
97             {
98                 eliminarUsuario();
99                 break;
100            }
101            case 5:
102            {
103                buscarUsuario();
104                break;
105            }
106            case 6:
107            {
108                break;
109            }
110            default:
111            {
112                cout << "Opción no válida" << endl;
113                break;
114            }
115        }
116        while(op != 6 );
117    }
118 }
119
120 }

```


Class Menu{

```
4  class Menu
5  {
6      private:
7          Admin* ptr_admins;
8          Alumno* ptr_alumns;
9          Profesor* ptr_profes;
10         int contD, contA, contP;
11
12     public:
13         Menu();
14         ~Menu();
15         bool loginAdmin(int, string);
16         bool loginAlumno(int, string);
17         bool loginProf(int, string);
18         void menuPrincipal();
19         void menuAdmin();
20         void menuProf();
21         void menuAlumno();
22         void cargarDatos();
23         void guardarDatos();
24
```

```
25     //ADMIN
26     //CREAR_NUEVO_USUARIO
27     void crearAdmin();
28     void crearProf();
29     void crearAlumno();
30     void crearUsuario();
31     //ELIMINAR_USUARIO
32     void eliminarUsuario();
33     void eliminarAdmin();
34     void eliminarAlumno();
35     void eliminarProf();
36     //VER_USUARIO
37     void verUsuario();
38     void verAdmin();
39     void verAlumno();
40     void verProf();
41     //MODIFICAR_DATOS_TODO_USUARIO
42     void modificarUsuario();
43     void modificarAdmin();
44     void modificarAlumno();
45     void modificarProf();
46     //BUSCAR_USUARIO
47     void buscarAdmin();
48     void buscarAlumno();
49     void buscarProf();
50     void buscarUsuario();
```

```
52     //PROF
53     //MODIFICAR_DATOS_PER
54     void modificarDatosP(int);
55     //MODIFICAR_NOTAS
56     void modificarNotas();
57     //VER_DATOS_PER
58     void mostrarDatosP(int);
59
60     //ALUMNO
61     //MODIFICAR_DATOS_PER
62     void modificarDatosA(int);
63     //VER_DATOS_PER
64     void mostrarDatosA(int);
65
66     //VER_DATOS_USUARIO_ALUM_PROFE
67     void mostrarDatosD();
68     void mostrarDatosA();
69     void mostrarDatosP();
70     void mostrarDatosUsu();
```

```
14 }
```

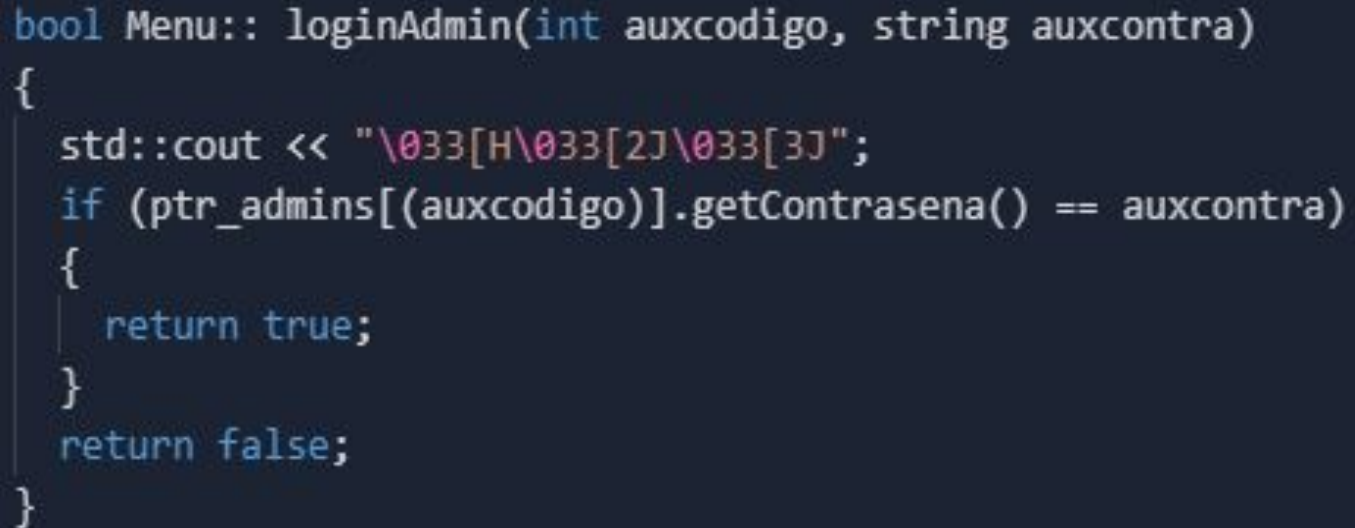
Class Menu{

```
3  Menu :: Menu()
4  {
5      ptr_admins = new Admin[30];
6      ptr_profes = new Profesor[30];
7      ptr_alumns = new Alumno[100];
8      contD = 1;
9      contA = 0;
10     contP = 0;
11     ptr_admins[0] = Admin(0, "Admin god", "admingoddd", "563746298",
12         "admin.god@ucsp.edu.pe", "976453433", "urb el cielo", "Solterosky");
13 }
14 Menu :: ~Menu()
15 {
16     delete ptr_admins;
17     delete ptr_alumns;
18     delete ptr_profes;
19 }
```

}

Método Login{

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



```
bool Menu:: loginAdmin(int auxcodigo, string auxcontra)
{
    std::cout << "\033[H\033[2J\033[3J";
    if (ptr_admins[(auxcodigo)].getContrasena() == auxcontra)
    {
        return true;
    }
    return false;
}
```

}

Método CrearUsuario{

```
120 void Menu :: crearUsuario()
121 {
122     cout << endl << "\t\t\t\t\t*****Crear un Usuario*****\t\t\t\t\t";
123     char opc;
124     cout << endl << "Administrador(D), Profesor(P), Alumno(A): " ;
125     cin >> opc;
126     switch (opc)
127     {
128         case 'D':
129         {
130             crearAdmin();
131             break;
132         }
133         case 'P':
134         {
135             crearProf();
136             break;
137         }
138         case 'A':
139         {
140             crearAlumno();
141             break;
142         }
143     }
144 }
```

}

Class CrearAdmin{

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

```
3 void Menu :: crearAdmin()
4 {
5     int auxcodigo;
6     string auxnombres, auxapellidos,auxdni, auxcorreoInsti,auxcelular,auxdireccion,
       auxestdCiv;
7
8     cout << "\033[H\033[2J\033[3J";
9     cout << "\t\t\t**CREAR UN ADMINISTRADOR**\t\t\t\n\n";
10    cout << endl << "Ingrese los siguientes datos: ";
11    cin.ignore(1000000, '\n');
12    cout << endl << "Nombres: ";
13    getline(cin, auxnombres);
14    cout << endl << "Apellidos: ";
15    getline(cin, auxapellidos);
16    cout << endl << "DNI: ";
17    getline(cin,auxdni);
18    cout << endl << "Correo institucional: ";
19    getline(cin,auxcorreoInsti) ;
20    cout << endl << "Celular: ";
21    getline(cin, auxcelular);
22    cout << endl << "Dirección: ";
23    getline(cin,auxdireccion);
24    cout << endl << "Estado civil: ";
25    getline(cin,auxestdCiv);
```

```
27     auxcodigo = contD;
28     ptr_admins[contD] = Admin(auxcodigo,auxnombres, auxapellidos, auxdni,
       auxcorreoInsti,auxcelular,auxdireccion, auxestdCiv);
29
30     cout << endl << "Usuario creado con éxito" << endl << endl;
31     contD++;
32 }
```

Método EliminarUsuario{

```
void Menu :: eliminarAdmin()
{
    int aux;
    cout << endl << "Ingrese el código del usuario que desea eliminar: ";
    cin >> aux;

    for (int i = (aux-1); i < contD; i++)
    {
        ptr_admins[i] = ptr_admins[(i+1)];
        ptr_admins[i].setCodigo(i+1);
    }
    contD --;
    cout << endl << "Usuario Eliminado con éxito" << endl;
}
```

}

Método BuscarUsuario{

```
3 void Menu :: buscarAdmin()  
4 {  
5     int aux;  
6     cout << "Ingrese el código del usuario que desea buscar: " ;  
7     cin >> aux;  
8  
9     ptr_admins[(aux)].datos();  
10 }
```

```
14 }
```

Método VerUsuarios{

```
void Menu :: verAdmin()
{
    for (int i = 0; i < contD; i++)
    {
        if (ptr_admins[i].getNombres() != " ")
        {
            ptr_admins[i].datos();
        }
        std::cout << std::endl;
    }
}
```

```
39 void Menu :: verUsuario()
40 {
41     cout << endl << "\t\t\t\t\t*****Mostrar Datos un Usuario*****\t\t\t\t\t";
42     char opc;
43     cout << endl << "Administrador(D), Profesor(P), Alumno(A): " ;
44     cin >> opc;
45     switch (opc)
46     {
47         case 'D':
48         {
49             verAdmin();
50             break;
51         }
52         case 'P':
53         {
54             verProf();
55             break;
56         }
57         case 'A':
58         {
59             verAlumno();
60             break;
61         }
62     }
63 }
```


Método ModificarUsuario{

```
3 void Menu :: modificarAdmin()
4 {
5     int aux;
6     string auxnombres, auxapellidos,auxdni, auxcorreoInsti,auxcelular,auxdireccion,
7     auxstdCiv;
8
9     cout <<endl<< "Ingrese el código del Usuario que desea modificar: ";
10    cin >> aux;
11
12    cout <<endl << "Datos actuales del usuario: " << endl;
13    ptr_admins[aux].datos();
14 }
```

```
11    cout <<endl << "Datos actuales del usuario: " << endl;
12    ptr_admins[aux].datos();
13
14    cout << endl << "\t-----\t";
15    cout <<endl<< "Ingrese los nuevos datos";
16    cin.ignore(1000000, '\n');
17    cout << endl << "Nombres: ";
18    getline(cin, auxnombres);
19    cout << endl << "Apellidos: ";
20    getline(cin, auxapellidos);
21    cout << endl << "DNI: ";
22    getline(cin,auxdni);
23    cout << endl << "Correo institucional: ";
24    getline(cin,auxcorreoInsti) ;
25    cout << endl << "Celular: ";
26    getline(cin, auxcelular);
27    cout << endl << "Dirección: ";
28    getline(cin,auxdireccion);
29    cout << endl << "Estado civil: ";
30    getline(cin,auxstdCiv);
31
32    ptr_admins[(aux)].setAdmin(auxnombres, auxapellidos,auxdni, auxcorreoInsti,
33    auxcelular,auxdireccion,auxstdCiv);
34
35    cout <<endl << "Registro modificado con éxito" << endl;
36 }
```

Metodos Alumno{

```
void Menu::modificarDatosA(int auxcodigo)
{
    string auxnombres, auxapellidos,auxdni, auxcorreoInsti,auxcelular,auxdireccion,
    auxestdCiv;

    cout <<endl << "Datos actuales del usuario: " << endl;
    ptr_alumns[auxcodigo-1].datos();

    cout << endl << "\t-----\t";
    cout << endl << "Ingrese los nuevos datos";
    cin.ignore(1000000, '\n');
    cout << endl << "Nombres: ";
    getline(cin, auxnombres);
    cout << endl << "Apellidos: ";
    getline(cin, auxapellidos);
    cout << endl << "DNI: ";
    getline(cin,auxdni);
    cout << endl << "Correo institucional: ";
    getline(cin,auxcorreoInsti) ;
    cout << endl << "Celular: ";
    getline(cin, auxcelular);
    cout << endl << "Dirección: ";
    getline(cin,auxdireccion);
    cout << endl << "Estado civil: ";
    getline(cin,auxestdCiv);

    ptr_alumns[auxcodigo-1].setDatosPer(auxnombres, auxapellidos,auxdni,auxcelular,
    auxdireccion,auxestdCiv);
}
```

Metodos Alumno{

```
void Menu :: mostrarDatosD()
{
    int auxcodigo;
    cout << "Ingrese el código del usuario que desea ver: " ;
    cin>>auxcodigo;
    cout << "Nombres: " << ptr_admins[auxcodigo].getNombres() <<endl;
    cout << "Apellidos: " << ptr_admins[auxcodigo].getApellidos() <<endl;
    cout << "DNI: " << ptr_admins[auxcodigo].getDNI()<<endl;
    cout << "Correo Institucional: " << ptr_admins[auxcodigo-1].getCorreoInsti()
    <<endl;
    cout << "Celular: " << ptr_admins[auxcodigo].getCelular() <<endl;
    cout << "Dirección: " << ptr_admins[auxcodigo].getDireccion() <<endl;
    cout << "Estado Civil: " << ptr_admins[auxcodigo].getEstadoCiv()<<endl;
}
```

}

```
void Menu :: mostrarDatosA(int auxcod)
{
    ptr_alumns[auxcod-1].datos();
}
```

Metodos Alumno{

```
void Menu:: mostrarDatosUsu()
{
    cout << endl << "\t\t\t\t\t*****VER DATOS DE UN USUARIO*****\t\t\t\t\t";
    char opc;
    cout << endl << "Administrador(D), Profesor(P), Alumno(A): ";
    cin >> opc;
    switch (opc)
    {
        case 'D':
        {
            mostrarDatosD();
            break;
        }
        case 'P':
        {
            mostrarDatosP();
            break;
        }
        case 'A':
        {
            mostrarDatosA();
            break;
        }
    }
}
```

}

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



GIT HUB {

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



}



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



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



}

