



# 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

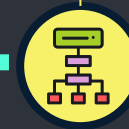
Llamar a las  
bibliotecas

Step 1



Crear los atributos  
y métodos de cada  
clase

Step 3



Step 2



Crear las clases que  
usaremos

Step 4

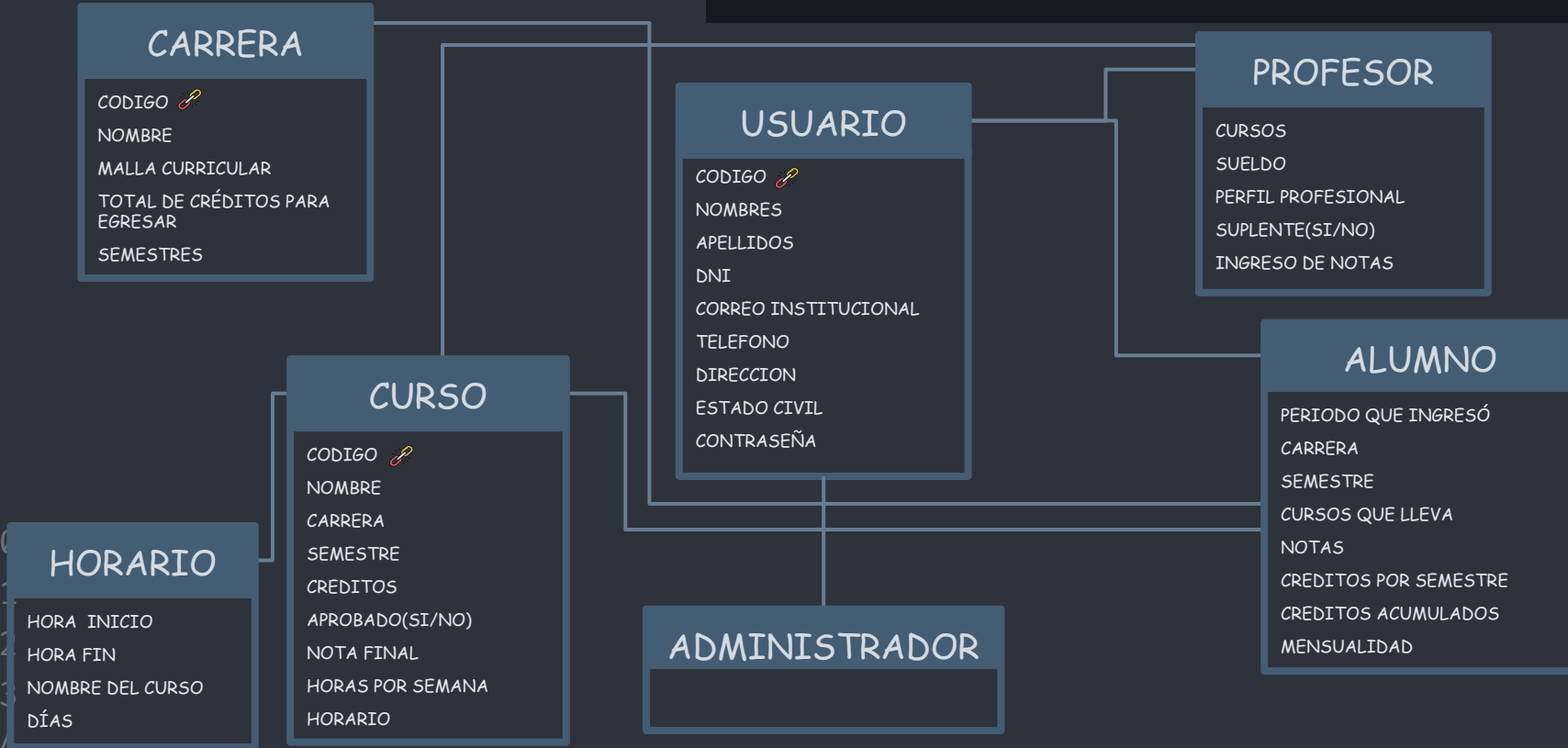


Definir los métodos

# PROYECTO FINAL: BASE DE DATOS

## Estructura

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



| CALENDARIO                                      | SEMANA 1 |         |        |
|---|----------|---------|--------|
|   | JOSUE    | ANTHONY | ARIANA |
| CLASE USUARIO                                   | X        | X       | X      |
| MÉTODO MENÚ PRINCIPAL                           | X        | X       | X      |
| CLASE ADMINISTRADOR                             |          | X       |        |
| MÉTODO ADMINISTRADOR (VER, MENÚ, CREAR USUARIO) |          | X       |        |
| CLASE PROFESOR                                  |          |         | X      |
| MÉTODO PROFESOR (VER, MENÚ)                     |          |         | X      |
| CLASE ALUMNO                                    | X        |         |        |
| MÉTODO ALUMNO (VER, MENÚ)                       | X        |         |        |

# Class Usuario {

```
14  class Usuario
15  {
16      public:
17          string contraseña;
18          string codigo;
19          string nombres;
20          string apellidos;
21          string dni;
22          string correoInsti;
23          string celular;
24          string direccion;
25          string estdCiv;
26
27          Usuario();
28
29  };
```

```
62  Usuario::Usuario()
63  {
64      contraseña = " ";
65      codigo = " ";
66      nombres = " ";
67      apellidos = " ";
68      dni = " ";
69      correoInsti = " ";
70      celular = " ";
71      direccion = " ";
72      estdCiv = " ";
73
74  }
```

# Class Profesor {

```
105  class Profesor : Usuario
106  {
107      public:
108          string  cursosEnseña;
109          string  sueldo;
110          string  perfilProf;
111          string  suplente;
112          Profesor();
113          //void crearProf();
114          //void menuProf();
115          //void modificarNotas();
116          //void modificarDatosP();
117          //void mostrarDatosP();
118
119          //friend class Admin;
120          //friend class Alumno;
121  };
```

```
122  Profesor :: Profesor()
123  {
124      cursosEnseña = " ";
125      sueldo = " ";
126      perfilProf = " ";
127      suplente = " ";
128  }
```

# Class Alumno {

```
130  class Alumno : Usuario
131  {
132      public:
133          string carrera;
134          string cursos;
135          string periodoIngreso;
136          int semestre;
137          int creditosSemestre;
138          int creditosAcumulados;
139          double notas;
140          float mensualidad;
141
142          Alumno();
143          void crearAlumno();
144          //void menuAlumno();
145          //void modificarDatosA();
146          //void mostrarDatosA();
147
148          friend class Admin;
149          friend class Profesor;
150  };
```

```
151  Alumno :: Alumno()
152  {
153      carrera = " ";
154      cursos = " ";
155      periodoIngreso = " ";
156      semestre = 0;
157      creditosSemestre = 0;
158      creditosAcumulados = 0;
159      notas = 0;
160      mensualidad = 0;
161  }
```

}

# Class Administrador {

```
76  class Admin : Usuario
77  {
78      public:
79      Admin();
80
81      //void crearAdmin();
82      //void menuAdmin();
83      //void crearUsuario();
84      //void eliminarUsuario();
85      //void verUsuario();
86      //void modificarUsuario();
87
88      //friend class Alumno;
89      //friend class Profesor;
90
91  };
```

```
93  Admin :: Admin()
94  {
95      Usuario();
96  }
```



# Menu Principal {

```
741 void Menu :: menuPrincipal()
742 {
743     int opc;
744     do
745     {
746         cout << "\t\t***Universidad las chicas superpoderosas***\t\t" << endl;
747         cout << "[1] Administrador" << endl;
748         cout << "[2] Profesor" << endl;
749         cout << "[3] Alumno" << endl;
750         cout << "[4] Salir" << endl;
751         cout << "Elija una opción..." << endl;
752         cin >> opc;
753         std::cout << "\033[H\033[2J\033[3J";
754         switch (opc)
755         {
756             case 1:
757             {
758                 menuAdmin();
759                 break;
760             }
```

```
761         case 2:
762         {
763             menuProf();
764             break;
765         }
766         case 3:
767         {
768             menuAlumno();
769             break;
770         }
771         case 4:
772         {
773             break;
774         }
775         default:
776         {
777             cout << "Opci\242n no v\240lida" << endl;
778             break;
779         }
780     }
781 }
782 while(opc != 4);
783 }
```

# Menú Profesor {

```
646 void Menu :: menuProf()  
647 {  
648     int opc;  
649     do  
650     {  
651         cout << "\t\t***PROFESOR***\t\t" << endl;  
652         cout << "[1] Modificar datos personales" << endl;  
653         cout << "[2] Modificar notas" << endl;  
654         cout << "[3] Mostrar Usuario" << endl;  
655         cout << "[4] Mostrar datos Propios" << endl;  
656         cout << "[5] Salir" << endl;  
657         cout << "Elija una opción..." << endl;  
658         cin >> opc;  
659         std::cout << "\033[H\033[2J\033[3J";  
660         switch (opc)  
661         {  
662             case 1:  
663             {  
664                 //modificarDatosPer();  
665                 break;  
666             }
```

```
667         case 2:  
668         {  
669             //modificarNotas();  
670             break;  
671         }  
672         case 3:  
673         {  
674             //mostrarUsuario();  
675             break;  
676         }  
677         case 4:  
678         {  
679             //mostrarProfesor();  
680             break;  
681         }  
682         case 5:  
683         {  
684             break;  
685         }  
686         default:  
687         {  
688             cout << "Opción no válida" << endl;  
689             break;  
690         }  
691     }  
692     while(opc != 5 );  
693 }  
694 }
```

# Menú Alumno {

```
696 void Menu :: menuAlumno()
697 {
698     int opc;
699     do
700     {
701         cout << "\t\t***ALUMNO***\t\t" << endl;
702         cout << "[1] Modificar datos personales" << endl;
703         cout << "[2] Mostrar Datos" << endl;
704         cout << "[3] Mostrar Usuario" << endl;
705         cout << "[4] Salir" << endl;
706         cout << "Elija una opción..." << endl;
707         cin >> opc;
708         std::cout << "\033[H\033[2J\033[3J";
709         switch (opc)
710         {
711             case 1:
712             {
713                 //modificarDatosPer();
714                 break;
715             }
```

```
716         case 2:
717         {
718             //mostrarAlumno();
719             break;
720         }
721         case 3:
722         {
723             //mostrarUsuario();
724             break;
725         }
726         case 4:
727         {
728             break;
729         }
730         default:
731         {
732             cout << "Opción no válida" << endl;
733             break;
734         }
735     }
736 }
737 while(opc != 4 );
738 }
739 }
```

# Menú Administrador {

```
596 void Menu :: menuAdmin()
597 {
598     int op;
599     do
600     {
601         cout << "\t\t***ADMINISTRADOR***\t\t"<<endl;
602         cout << "[1] Crear Usuario" << endl;
603         cout << "[2] Modificar Usuario" << endl;
604         cout << "[3] Mostrar Datos de Usuario" << endl;
605         cout << "[4] Eliminar Usuario" << endl;
606         cout << "[5] Salir" << endl;
607         cout << "Elija una opción..." << endl;
608         cin >> op;
609         std::cout << "\033[H\033[2J\033[3J";
610         switch (op)
611         {
612             case 1:
613             {
614                 crearUsuario();
615                 break;
616             }
```

```
617         case 2:
618         {
619             verUsuario();
620             break;
621         }
622         case 3:
623         {
624             //mostrarUsuarioComp();
625             break;
626         }
627         case 4:
628         {
629             //eliminarUsuario();
630             break;
631         }
632         case 5:
633         {
634             break;
635         }
636         default:
637         {
638             cout << "Opción no válida" << endl;
639             break;
640         }
641     }
642     while(op != 5 );
643 }
644 }
```

## 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{

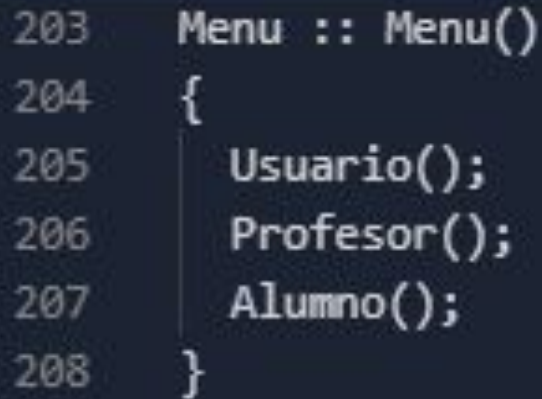
```
163 class Menu
164 {
165     public:
166     Menu();
167     class Usuario usuario;
168     class Admin admin;
169     class Profesor profesor;
170     class Alumno alumno;
171
172     void menuPrincipal();
173     void crearAdmin();
174     void crearProf();
175     void crearAlumno();
176
177 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



```
203 Menu :: Menu()  
204 {  
205     Usuario();  
206     Profesor();  
207     Alumno();  
208 }
```

## Método CrearUsuarioAux{

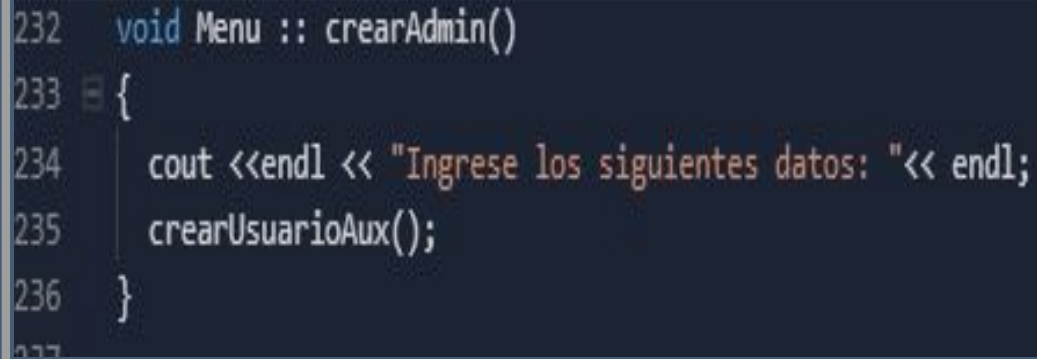
```
210 void Menu :: crearUsuarioAux()  
211 {  
212     cout << endl << "Código: ";  
213     getline(cin, usuario.codigo);  
214     cout << "Contraseña: " << endl ;  
215     getline(cin, usuario.contraseña);  
216     cout << endl << "Nombres: ";  
217     getline(cin, usuario.nombres);  
218     cout << endl << "Apellidos: ";  
219     getline(cin, usuario.apellidos);  
220     cout << endl << "DNI: ";  
221     getline(cin, usuario.dni);
```

```
222     cout << endl << "Correo institucional: ";  
223     getline(cin, usuario.correoInsti) ;  
224     cout << endl << "Celular: ";  
225     getline(cin, usuario.celular);  
226     cout << endl << "Dirección: ";  
227     getline(cin, usuario.direccion);  
228     cout << endl << "Estado civil: ";  
229     getline(cin, usuario.estdCiv);  
230     cout << endl;  
231 }
```

```
}
```



1  
2 **Class** **CrearAdmin**{  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14 }



```
232 void Menu :: crearAdmin()  
233 {  
234     cout <<endl << "Ingrese los siguientes datos: " << endl;  
235     crearUsuarioAux();  
236 }
```

# Método CrearProf{

```
238 void Menu :: crearProf()
239 {
240     cout << endl << "Ingrese los siguientes datos: ";
241     crearUsuarioAux();
242     cout << endl << "Cursos que enseñan: ";
243     getline(cin, profesor.cursosEnseña);
244     cout << endl << "Sueldo: " ;
245     cin >> profesor.sueldo;
246     fflush(stdin);
247     cout << endl << "perfil profesional: " ;
248     getline(cin, profesor.perfilProf);
249     fflush(stdin);
250     cout << "Suplente (si/no): ";
251     getline(cin, profesor.suplente);
252     fflush(stdin);
253 }
```

}

# Método CrearAlumno{

```
255 void Menu ::crearAlumno()  
256 {  
257     cout <<endl << "Ingrese los siguientes datos: ";  
258     crearUsuarioAux();  
259     cout<<"carrera: " <<endl;  
260     getline(cin, alumno.carrera);  
261     cout<<"cursos: " <<endl;  
262     getline(cin,alumno.cursos);  
263     cout<<"periodo de ingreso: " <<endl;  
264     getline(cin, alumno.periodoIngreso);  
265     cout<<"Semestre: " <<endl;  
266     cin>> alumno.semestre;  
267     cout<<"creditos del semestre: ";  
268     cin>> alumno.creditosSemestre;  
269     cout<<"creditos acumulados: ";  
270     cin>> alumno.creditosAcumulados;  
271     cout<<"Notas del alumno: ";  
272     cin>> alumno.notas;  
273     cout<<"Mensualidad del alumno: ";  
274     cin>> alumno.mensualidad;  
275 }
```

}

# Método CrearUsuario{

```
277 void Menu :: crearUsuario()
278 {
279     ofstream escritura;
280     ifstream verificador;
281     string auxCodigo;
282     bool coincidencia=false;
283     verificador.open("usuarios.txt",ios::in);
284     escritura.open("usuarios.txt",ios::app);
285     if(escritura.is_open() && verificador.is_open())
286     {
287         cout<<"\t\t\t\t***Crear Usuario***\t\t\t\t\n\n";
288         fflush(stdin);
289         cout<<"Ingresa el código del cliente: ";
290         getline(cin,auxCodigo);
291         if(auxCodigo=="")
292             do
293             {
294                 cout<<"código de cliente no válido!, intentalo nuevamente: ";
295                 getline(cin,auxCodigo);
296             }
297             while(auxCodigo=="");
```

```
298     do
299     {
300         verificador.seekg(0);
301         getline(verificador,usuario.codigo);
302         while(!verificador.eof())
303         {
304             getline(verificador,usuario.contraseña);
305             getline(verificador,usuario.nombres);
306             getline(verificador,usuario.apellidos);
307             getline(verificador,usuario.dni);
308             getline(verificador,usuario.correoInsti);
309             getline(verificador,usuario.celular);
310             getline(verificador,usuario.direccion);
311             getline(verificador,usuario.estdCiv);
312
313             if(usuario.codigo==auxCodigo)
314             {
315                 coincidencia=true;
316                 cout<<"\n\nYa existe un cliente con ese código!\n\n";
317                 cout<<"El cliente con ese código es: ";
318                 cout<<"usuario.nombres<<"\n\n";
319                 cout<<"Ingresa un código válido!: ";
320                 getline(cin,auxCodigo);
```

# Método CrearUsuario{

```

320         if(auxCodigo=="")
321         {
322             do
323             {
324                 cout<<"\ncódigo de cliente no válido!,
325                 intentalo nuevamente: ";
326                 getline(cin,auxCodigo);
327             }
328             while(auxCodigo=="");
329         }
330         break;
331     }
332     //getline(verificador,usuario.codigo);
333 }
334 if(verificador.eof())&&auxCodigo!=usuario.codigo
335     coincidencia=false;
336
337 }
338 while(coincidencia==true);
339
340 cout << "\033[H\033[2J\033[3J";
341 char opc;
342 cout << endl << "\t\t\t\t\t*****Crear un Usuario*****\t\t\t\t\t";
343 cout << endl << "Administrador(D), Profesor(P), Alumno(A): " ;
344 cin >> opc;

```

```

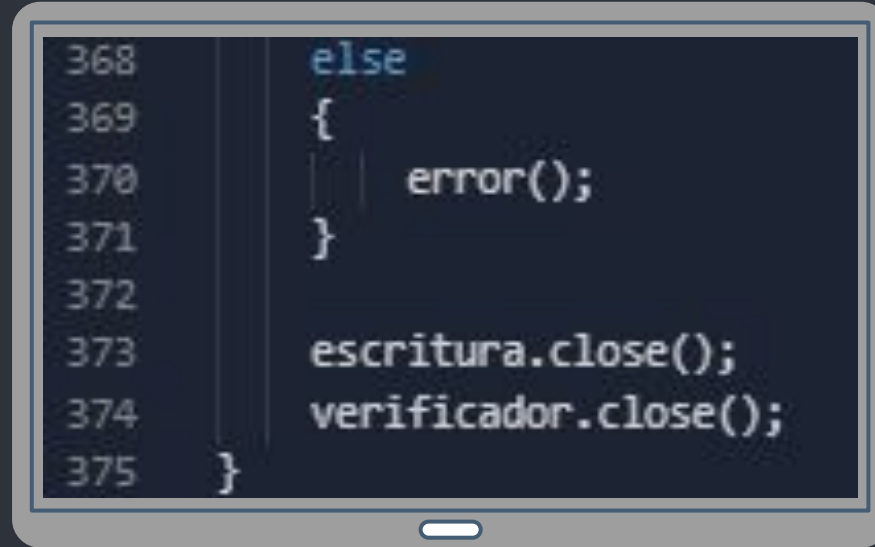
345     switch (opc)
346     {
347     case 'D':
348     {
349         crearAdmin();
350         escritura<<usuario.codigo<<endl<<usuario.contraseña<<endl<<usuario.nombres<<endl<<
351         usuario.apellidos <<endl << usuario.dni<<endl <<
352         usuario.correoInsti<<endl<<usuario.celular<<
353         endl<<usuario.direccion<<endl<<usuario.estdCiv<<endl;
354     }
355     case 'P':
356     {
357         crearProf();
358         escritura<<usuario.codigo<<endl<<usuario.contraseña<<endl<<usuario.nombres<<endl<<
359         usuario.apellidos <<endl << usuario.dni<<endl <<
360         usuario.correoInsti<<endl<<usuario.celular<<
361         endl<<usuario.direccion<<endl<<usuario.estdCiv<<endl<<alumno.carrera<<endl<<alumno.
362         cursos<<endl<<alumno.periodoIngreso<<endl<<alumno.semestre<<endl<<alumno.
363         creditsSemestre<<endl<<alumno.creditsAcumulados<<endl<<alumno.
364         notas<<endl<<alumno.mensualidad;
365     }
366     case 'A':
367     {
368         crearAlumno();
369         escritura<<usuario.codigo<<endl<<usuario.contraseña<<endl<<usuario.nombres<<endl<<
370         usuario.apellidos <<endl << usuario.dni<<endl <<
371         usuario.correoInsti<<endl<<usuario.celular<<
372         endl<<usuario.direccion<<endl<<usuario.estdCiv<<endl<<alumno.carrera<<endl<<alumno.
373         cursos<<endl<<alumno.periodoIngreso<<endl<<alumno.semestre<<endl<<alumno.
374         creditsSemestre<<endl<<alumno.creditsAcumulados<<endl<<alumno.
375         notas<<endl<<alumno.mensualidad;
376     }
377     }
378
379     cout<<"El registro se ha completado correctamente.\n\n";
380 }

```

# Método CrearUsuario{

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

}



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



# GIT HUB {

[https://github.com/Anthony-Rodriguez18/PROYECTO\\_FINAL\\_BASE\\_DE-DATOS](https://github.com/Anthony-Rodriguez18/PROYECTO_FINAL_BASE_DE-DATOS)



}



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



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



}

