



Universidad Católica
San Pablo

UNIVERSIDAD CATÓLICA SAN PABLO
ESC. DE CIENCIA DE LA COMPUTACIÓN

Ciencia de la Computación

Informe de código

Nombre de proyecto: Universidad de las Chicas Super Poderosas

Anthony Rodríguez Pinto

anthony.rodriguez.pinto@ucsp.edu.pe

Josue Carpio Peña

josue.carpio@ucsp.edu.pe

Ariana Jasmín León Zarate

ariana.leon@ucsp.edu.pe

Manuel Eduardo Loaiza Fernández

AREQUIPA - 2021

La actividades propuestas según cronograma de semana uno son las siguientes:

	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		

- **Proceso de actividades:**

- Todos:

- Empezó creando la clase usuario se utiliza el nivel de acceso public en ella se da la contraseña, el código, nombres, apellidos, DNI, correo institucional, celular, dirección y el estado civil del usuario, nuestras variables están puestas como String para facilitar la búsqueda en el archivo; luego se definió su constructor, decidimos hacer un constructor por defecto donde todos los atributos son nulos (“ ”). El propósito de crear la clase usuario es para que las clases administrador, profesor y alumno puedan heredar sus atributos.

- Ariana leon:

- La segunda clase que se realizó fue la del profesor, esta clase hereda los atributos de la clase usuario y a su vez tiene como atributos propios los cursos que enseña, su sueldo, el perfil del profesor y pregunta si es suplente o no, estos tienen un nivel de acceso public. Para esta clase también se creó un constructor por defecto, al igual que la clase padre iniciamos todos sus atributos como nulos.
 - También se creó el método menú profesor la que tenemos las siguientes opciones: modificar datos personales, modificar notas, mostrar usuario, mostrar datos propios y salir.

- Anthony Rodríguez:

- La siguiente clase que se realizó fue la de administrador, el propósito de esta clase es que tengamos un usuario-administrador ya que en unas instituciones se dividen los cargos y uno de ellos es el que administra todo, administrador tiene el poder de agregar datos sobre el profesor y el alumno (datos específicos como salario o pensión) esta clase no tiene atributos ni métodos propios, pero hereda los atributos de usuario.
- También creó la parte de menú administrador, dando las preguntas para llevar a cabo lo que se desea, tanto si queremos ingresar un usuario nuevo, modificar un usuario, mostrar datos de un usuario o eliminar el usuario. El administrador es el único que puede crear, eliminar usuarios y ver todos los datos de un usuario

- Josue Carpio:

- La clase que realizó fue la de alumno, como estamos haciendo una base de datos para una universidad es necesario tener esta clase, está hereda los atributos de usuario y sus atributos propios son: carrera, cursos, periodo en el que ingresó, el semestre, créditos del semestre, créditos acumulados, las notas, la mensualidad, al igual que las demás clases su constructor por defecto inicializa todos los atributos como nulos (“ “)

- **Cambios que hicimos:**

1. Al inicio estamos implementando los métodos en cada clase de los objetos pero decidimos que mejor la mayoría de los métodos están en una clase la cual creamos una class llamada “menú”
2. Queríamos implementar struct en el código pero por problemas decidimos mejor ponerlo como class.

- **Dificultades en las actividades:**

1. Uno de los problemas que se tuvo en el código fue por el compilador, a veces no se puede compilar o si se podía mandaba errores y teníamos que crear otro replit para que recién pueda ejecutarse
2. Otro problema que se tuvo al realizar el código fue tener dudas sobre utilizar struct o class para alumno y profesor, debido a que si utilizamos struct la lógica lo reconocía como si fueran uno mismo, y por otro lado en class no sabíamos cómo pasar métodos de distintas clases a objeto.

3. Otro problema que se tuvo en el proceso fue al crear el demo, debido a que en la función crearUsuario() pedimos distintos atributos, pero al momento que se debe ingresar los datos para contraseña o código, dependiendo en el orden que se encontraran , no se podía ingresar ningún dato y se salteaba al siguiente atributo.
4. Un problema que tuvimos es la implementación de los archivos porque no teníamos una idea clara de como ponerlas , como ponerlo en orden y no sabíamos cómo diferenciar cuando queríamos poner en profesor o cuando era alumno o cuando era administrador.

- **Conclusión:**

El avance de esta semana tuvimos un gran avance e investigación para poder realizar nuestro proyecto, claramente tuvimos problemas al no entender algunos conceptos que tuvimos, la cual nos costó mucho tiempo, pero ya pudimos solucionar la mayoría de los problemas que tuvimos y así tener el objetivo de mejorar nuestro código y poder pasar a la semana dos.

- **ANEXO:**

- ANEXO 1 (Menú Principal)

```
739 void Menu :: menuPrincipal()
740 {
741     int opc;
742     do
743     {
744         cout << "\t\t**** Universidad las chicas superpoderosas ****\t\t" << endl;
745         cout << "[1] Administrador" << endl;
746         cout << "[2] Profesor" << endl;
747         cout << "[3] Alumno" << endl;
748         cout << "[4] Salir" << endl;
749         cout << "Elija una opción..." << endl;
750         cin >> opc;
751         std::cout << "\033[H\033[2J\033[3J";
752     } while (opc != 4);
753 }
```

- ANEXO 2 (Menú Administrador)

```
594 void Menu :: menuAdmin()
595 {
596     int op;
597     do
598     {
599         cout << "**** Universidad las chicas superpoderosas ****" << endl;
600         cout << "[1] Crear Usuario" << endl;
601         cout << "[2] Modificar Usuario" << endl;
602         cout << "[3] Mostrar Datos de Usuario" << endl;
603         cout << "[4] Eliminar Usuario" << endl;
604         cout << "[5] Salir" << endl;
605         cout << "Elija una opción..." << endl;
606         cin >> op;
607         std::cout << "\033[H\033[2J\033[3J";
608     } while (op != 5);
609 }
```

- ANEXO 3(Menú Profesor)

```
644 void Menu :: menuProf()  
645 {  
646     int opc;  
647     do  
648     {  
649         cout << "**** Universidad las chicas superpoderosas ****" << endl;  
650         cout << "[1] Modificar datos personales" << endl;  
651         cout << "[2] Modificar notas" << endl;  
652         cout << "[3] Mostrar Usuario" << endl;  
653         cout << "[4] Mostrar datos Propios" << endl;  
654         cout << "[5] Salir" << endl;  
655         cout << "Elija una opción..." << endl;  
656         cin >> opc;  
657         std::cout << "\033[H\033[2J\033[3J";
```

- ANEXO 4 (Menú Alumno)

```
694 void Menu :: menuAlumno()  
695 {  
696     int opc;  
697     do  
698     {  
699         cout << "**** Universidad las chicas superpoderosas ****" << endl;  
700         cout << "[1] Modificar datos personales" << endl;  
701         cout << "[2] Mostrar Datos" << endl;  
702         cout << "[3] Mostrar Usuario" << endl;  
703         cout << "[4] Salir" << endl;  
704         cout << "Elija una opción..." << endl;  
705         cin >> opc;  
706         std::cout << "\033[H\033[2J\033[3J";
```