



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](mailto:anthony.rodriguez.pinto@ucsp.edu.pe)

Josue Carpio Peña

[josue.carpio@ucsp.edu.pe](mailto:josue.carpio@ucsp.edu.pe)

Ariana Jasmín León Zarate

[ariana.leon@ucsp.edu.pe](mailto:ariana.leon@ucsp.edu.pe)

Manuel Eduardo Loaiza Fernández

**AREQUIPA - 2021**

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

	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

- **Proceso de actividades:**

- Todos:

- Entre todos hicimos la división del código poniendo en archivos .h y incluyendo un .cpp a parte del main (anexo 1)
    - Terminar de corregir errores de la primera semana que quedaron pendientes.

- Ariana leon:

- Creó el método de mostrar datos de profesor, el cual consiste en ingresar el código del profesor que queremos visualizar y nos muestra todos sus atributos, tanto los que hereda de la clase usuario (nombres, apellidos, dni, celular, dirección, correo, entre otros) como sus propios atributos (cursos que enseña, sueldo, perfil profesional, entre otros)

- Corrigió errores y bugs en el código

- Anthony Rodríguez:

- Creó el método de mostrar datos de administrador, el cual consiste en ingresar el código del administrador que queremos visualizar y nos muestra todos sus atributos, que en este caso todos los hereda de la

clase usuario (nombres, apellidos, dni, celular, dirección, correo, entre otros).

- Corrigió errores y bugs en el código

- Josue Carpio:

- Creó el método de mostrar datos del alumno, el cual consiste en ingresar el código del alumno que queremos visualizar y nos muestra todos sus atributos, tanto los que hereda de la clase usuario (nombres, apellidos, dni, celular, dirección, correo, entre otros) como sus propios atributos(carrera, cursos, periodo ingreso, semestre, créditos del semestre, créditos acumulados, notas y mensualidad).
- Corrigió errores y bugs en el código

- **Cambios que hicimos:**

1. El primer cambio que hicimos fue poner archivos .h y .cpp ya no solo estamos trabajando con el main.cpp, esto para la formalidad y el entendimiento del código. Creamos un archivo .h para cada clase (usuario,alumno,administrador,profesor,manu), luego creamos un archivo .cpp para cada métodos de las clases (crearUsuario, verUsuario, modificarUsuario), por último creamos “liberman.h” que es donde llamaremos a todas las bibliotecas que utilizaremos, incluyendo los .h que creamos para cada clase.
2. Implementamos la función modificar datos de un Usuario, como su nombre lo dice se encarga de modificar los datos del Usuario que elijamos (Para mostrar los datos creamos funciones auxiliares tanto para el administrador como para el profesor y el alumno, recibe como parámetro el código del usuario que ingresamos), para esto primero pide que ingresemos el código del usuario que deseamos modificar, luego nos muestra todos sus datos actuales y por último nos pide que ingresemos los nuevos datos para actualizarlos. Internamente este método crea un archivo de texto auxiliar que es donde copiará los datos de todos los usuarios y los datos actualizados del usuario escogido, por último elimina el archivo de texto original y renombra el auxiliar para que se convierta en el original.

- **Dificultades en las actividades:**

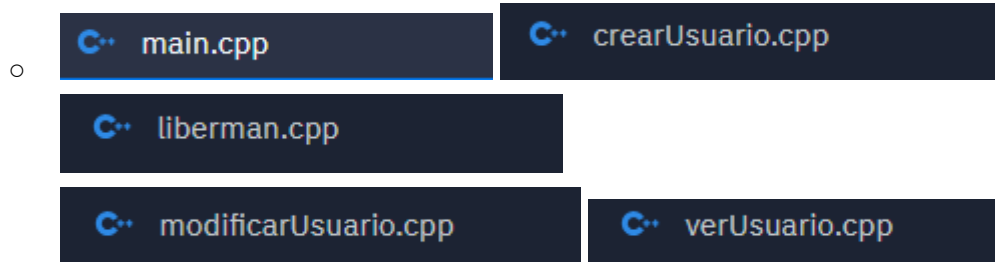
1. La dificultad que tuvimos en esta semana fue que no pudimos implementar los archivos con facilidad, tuvimos muchas dificultades tanto que ya no corría el código.
2. Dificultad en el método ver ya que en el momento de correrlo no detectaba bien los atributos, en algunos casos en el archivo que guardaba los datos creaba espacios.
3. Se tiene dificultad y errores para modificar los datos.

- **Conclusión:**

Tuvimos un gran avance esta semana, los únicos problemas fueron con los archivos y que algunos métodos no funcionan correctamente, debido a la reestructuración del proyecto ,hay métodos que debemos de terminar en la semana 3 ya que hay clases que se deben de crear y se crea una dependencia la cual no podemos crearlo en este momento, pudimos arreglar los problemas de la semana 1, y podemos seguir con el avance del código.

- **ANEXO:**

- ANEXO 1 .cpp (Librerías y archivos)

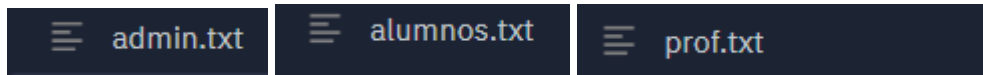


- ANEXO 2 .h (librerías y archivos)

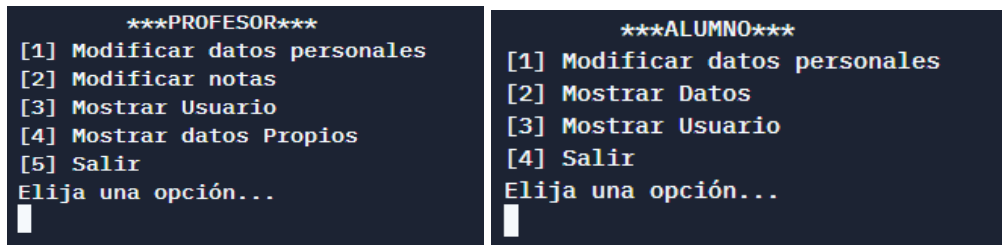
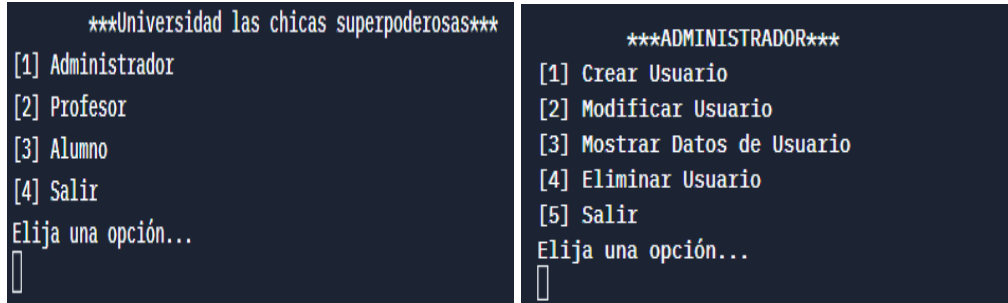


- ANEXO 3 .txt( archivos)

○



○ ANEXO 4(Menús)



○