



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 4		
	JOSUE	ANTHONY	ARIANA
REESTRUCTURACIÓN DE DATOS	X	X	x
FUNCIÓN LOGIN		X	
MODIFICAR DATOS PERSONALES(ALUMNO)			X
MODIFICAR DATOS (ADMINISTRADOR)	X		
MODIFICAR DATOS PERSONALES(PROFESOR)			X
MOSTRAR DATOS (ALUMNO, PROFESOR, ADMINISTRADOR)	X	X	
IMPLEMENTAR MEMORIA DINÁMICA	X	X	X

- **Proceso de actividades:**

- Reestructuración del código.
- Creamos la función login para que se pueda registrar cada usuario.
- Se arreglo el ver usuario,
- Perfeccionamos los métodos de Menú.
- También comenzamos con la implementación de las clases “Carrera”, “Curso” y por último “Horario”.

- **Cambios que hicimos:**

1. Los principales cambios que se hicieron fue la reconstrucción total del código por errores que nos marcó en clase, se implementó las herencias en usuarios, administrador, profesor y alumno. Se realizaron las funciones de administrador estas son: buscar usuario, crear usuario, eliminar usuario, modificar usuario y ver usuario. Se implementan funciones de alumno: modificación de datos personales, mostrar los datos personales y mostrar usuario. Y por último se implementaron las funciones del profesor estas son: modificar datos personales, modificar notas, mostrar los datos personales y ver los datos de otro Usuario.

2. Se redefinieron los atributos de la clase Menu, pasando una lista de administradores, alumnos y profesores como punteros del tipo de cada clase, redefinimos los métodos para que las modificaciones se realizarán en la lista y no de frente en el archivo de texto. Aún nos falta implementar el método que recorra cada lista y lo pase al txt.

Le pasamos 3 contadores, de alumnos, administradores y profesores, esto con el fin de que al final pase el contador a la primera línea del txt y adicionalmente para que las funciones que necesitan recorrer las listas solo las recorran hasta el contador y no hasta el final de la lista.

3. Se creó la clase horario dando como atributos la hora inicial, hora final y el día.
4. La clase curso tiene como atributos código, nombre del curso, créditos que aporta, horas por semana, la cantidad de notas que va a tener, la nota final, un puntero de tipo Horario para asignar qué días y a qué hora inicia y termina y por últimos si aprobó o no.
5. La clase semestre tiene como atributos el número de semestre, la cantidad de cursos y un puntero de tipo Cursos.
6. Por último la clase Carrera que tiene como atributos el código de la carrera, el nombre, los créditos que se necesita para ingresar, el número de semestres, y un puntero de tipo Semestres.
7. Estas últimas clases aún no están bien implementadas, nos falta definir algunos métodos para poder utilizarlas de una manera correcta.

- **Dificultades en las actividades:**

1. En el entendimiento del uso de punteros, exactamente en qué momento aplicar este tema o cuando no usarlo, por lo que nos creó cierto tiempo para comprenderlo de mejor manera.
2. Otra dificultad que tuvimos es implementar las clases secundarias (horario, curso, semestre y carrera) se podría decir que son clases anidadas, una recibe a la otra como atributo, tenemos los atributos ya definidos así como el constructor por defecto y el constructor que pide parámetros.

- **Conclusión:**

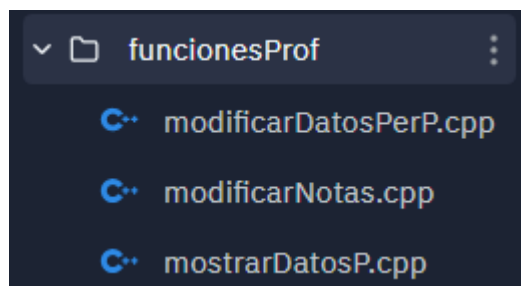
Esta semana nos fue mejor que la anterior, pudimos avanzar más cosas y corregir los errores que nos salieron la anterior semana, logramos implementar la mayoría de funciones con menos errores que antes, pudimos reestructurar todo el código de manera que utilice memoria dinámica.

- **ANEXO:**

- ANEXO 1 Funciones Alumno



- ANEXO 2 Funciones profesor



- ANEXO 3 Funciones admin

📁 funcionesAdmin

🔗 buscarUsuario.cpp

🔗 crearUsuario.cpp

🔗 eliminarUsuario.cpp

🔗 modificarUsuario.cpp

🔗 verUsuario.cpp