

CS1112: Programación II

Unidad 5: POO

Sesión de Laboratorio - 9B

Profesores:

María Hilda Bermejo mbermejo@utec.edu.pe

Estanislao Contreras econtreras@utec.edu.pe

Jorge Villavicencio jvillavicencio@utec.edu.pe

Edson Mendiola emendiola@utec.edu.pe

Ian Paul Brossard ibrossard@utec.edu.pe

Jose Chavez jchaveza@utec.edu.pe

Julio Yarasca jyarascam@utec.edu.pe

Percy Quevedo pquevedo@utec.edu.pe

Wilder Nina wnina@utec.edu.pe

José Fiestas jfiestas@utec.edu.pe

Material elaborado por:

Maria Hilda Bermejo



Logro de la sesión:

Al finalizar la sesión, los alumnos se familiarizan con el paradigma de la programación orientada a Objetos.

- Clase – Objeto
- Métodos de acceso (setter y getters)
- Constructores y destructores
- Objetos dinámicos
- Array de objetos

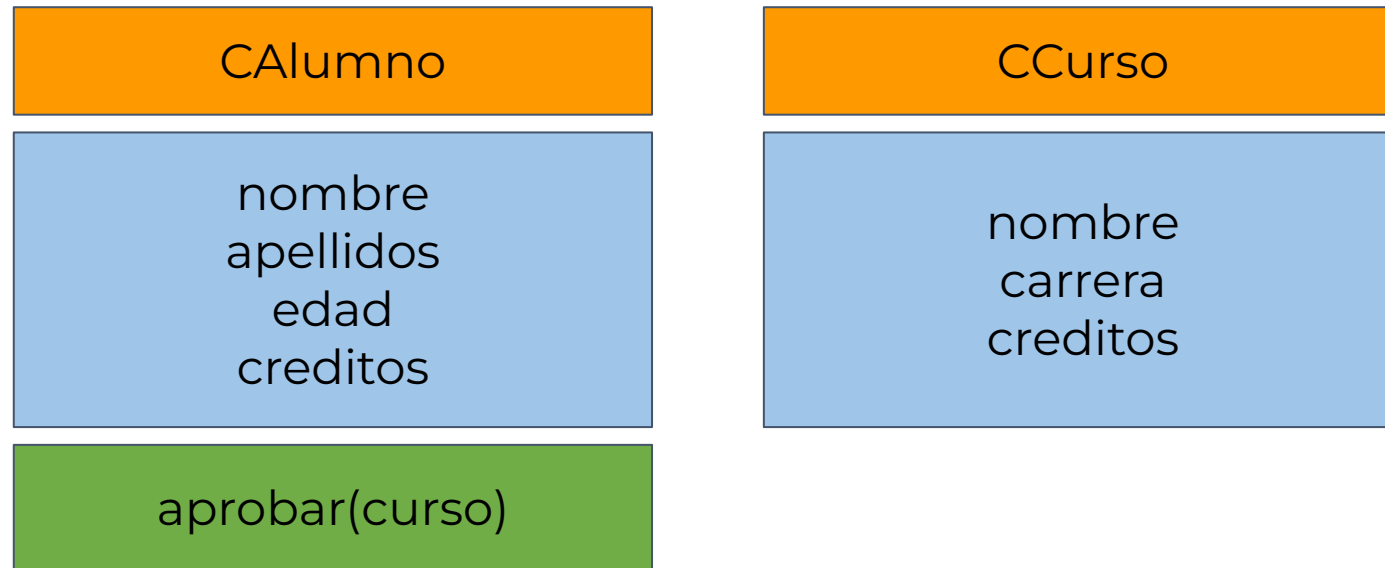
5.4

Unidad 5: POO
arreglos de objetos

UTEC

Ejercicio 1

Implementar la clase CCurso con los atributos que se muestra en la imagen y sus respectivos métodos de acceso setter y getter.



Luego implementar en la clase CAumno el método “aprobar” que recibe como parámetro un objeto de tipo curso y procede a agregar los créditos respectivos al alumno. Asimismo, implementar un método que encuentre al alumno con la mayor cantidad de créditos

Ejercicio 1

Main.cpp

```
#include "CAumno.h"
#include "CCurso.h"

int main()
{
    CAumno* pAlumno = new CAumno("Jorge", 50);
    CCurso* pCurso = new CCurso("POO1", 4);

    cout<<"Creditos antes de llevar "<<pCurso->getNombre()
        <<": "<<pAlumno->getCreditos()<<endl;

    pAlumno->aprobar(pCurso);

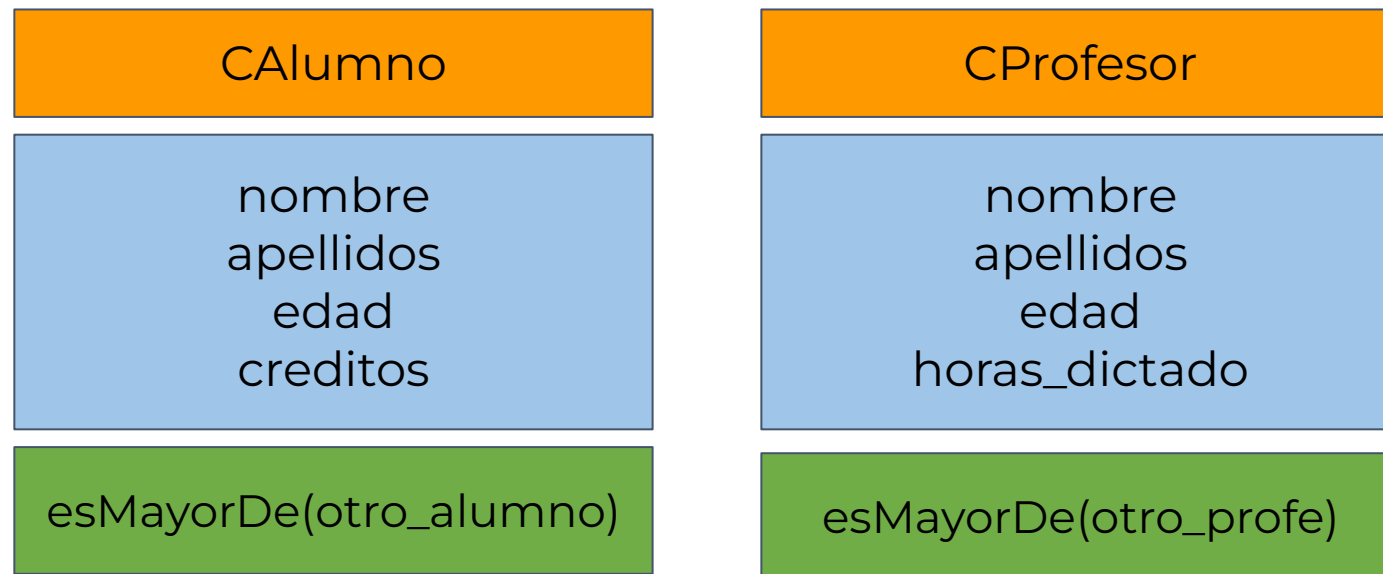
    cout<<"Creditos despues de aprobar "<<pCurso->getNombre()
        <<": "<<pAlumno->getCreditos()<<endl;

    delete pAlumno; delete pCurso;
    return 0;
}
```

Creditos antes de llevar POO1: 50
Creditos despues de aprobar POO1: 54

Ejercicio 1

Implementar un programa para registrar N objetos de cierta clase, para luego encontrar el mayor de ellos en base al criterio que se aplique en la función “esMayorDe” el cual recibe como parámetro otro objeto de la misma clase.



- Diseñe una función que aplique la búsqueda del mayor elemento sobre cualquier array de objetos usando la función “esMayorDe”.
- Siguiendo la misma estrategia, ahora ordene el array de objetos de mayor a menor.

Ejercicio 2

Diseñe una clase **Viaje** con las siguientes propiedades:

- **destino** (Berlin, Madrid, Sidney, Seul)
- **tiempo de viaje**
- **temporada alta** (si o no)
- **clase** (business, economy)

Defina 2 constructores y un destructor para objetos de clase Viaje



Ejercicio 2

Defina una función (método) que permita calcular el **costo** del viaje:

- multiplicando el tiempo de viaje por 50
- multiplicando este resultado por 2 si es temporada alta
- multiplicando este resultado por 4 si es clase business
- si el destino es Madrid, se agregan 500; si es Berlín, 1000; si es Sidney, 1500; y si es Seúl, 2000

El código debe crear objetos de clase Viaje para 5 personas y guardarlos en un vector (definir los valores iniciales en el código), calcular sus precios e imprimirlos.

Ejercicio 3

Calcule las fechas de entrega de **proyectos** en varias materias si el tiempo de desarrollo es igual a la dificultad del proyecto (decimal entre 0 y 1), multiplicado por el **tiempo máximo** de 60 días

Calcule el **tiempo necesario para desarrollar todos los proyectos** si se desarrollan uno después del otro



¡Nos vemos en la siguiente
clase!

