

**UNIVERSIDAD CATÓLICA DE SANTIAGO DEL ESTERO**

Departamento Académico Rafaela

Trabajo práctico N° 3

Carrera: Ing. en Informática

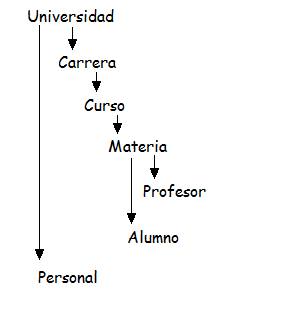
Materia: Programación I

Profesor: Sebastian Henzenn, SebastianDaniele, Andres Amherdt.

Fecha: 12/11/2014

Alumno: Camila Kopech, Wendy Sclerandi.

El código contiene 8 **entidades**, 7 de ellas especificadas en el siguiente diagrama. Algunas se relacionan con otra clase mediante **colecciones** (listas).



Por ejemplo:

publicclass Universidad

{

privateList<Carrera> carrera;

}

Para cargar datos en las listas, creamos distintos **métodos**, tales como “*AgregarCarrera()*”, “*AgregarAlumno()*”, etc :

publicvoidAgregarCarrera(Carrera c)

{

Carrera.Add(c);

}

Para aplicar el concepto de **herencia**, utilizamos la clase Persona. De la misma heredamos Profesor, Personal y Alumno.

publicclass Profesor : Persona {…}

publicclass Personal : Persona {…}

publicclass Alumno : Persona {…}

Para instanciar un **atributo** (definido con *get* y *set*) se utilizan **constructores**:

public Universidad()

{

carrera = new List<Carrera>();

}

Al instanciar la misma clase, con parámetros, aplicamos **sobre-escritura** de métodos:

public Universidad(stringnom)

{

carrera = new List<Carrera>();

nombre = nom;

}

El ejemplo de **polimorfismo** que aplicamos fue el método “Mostrar datos”, declarado virtualmente en la clase Persona:

public virtual stringMostrarDatos()

{

return "";

}

El mismo lo utilizamos en las clases Alumno y Profesor, mostrando diferentes datos por medio del “*override*”:

En alumnos:

private string datos;

public override string MostrarDatos()

{

datos = Nombre + " - Notas:";

foreach (int i in nota)

{

datos = datos + " " + i;

}

return datos;

}

En profesores:

public override string MostrarDatos()

{

return (Nombre + " - " + Titulo);

}

Finalmente, recorrimos cada lista con “*foreach*” para mostrar los atributos de las diferentes clases en pantalla, utilizando “*Console.Writeline()*”.

La resolución de las **consignas** d) y e) se encuentran en el método *ProfesoresYPromedios()*, el cual retorna la cantidad de profesores menores a 35 años, y crea una lista con los alumnos de mejor promedio.