

MAIN

```
internal class Program
{
    static void Main(string[] args)
    {
        Persona per1 = new Persona("Luis", "Gomez", "44525674", 2002);
        per1.Imprimir();

        Estudiante est1 = new Estudiante("Lucas", "Gomez", "44342344", 2002, 2, 0089, 5, "Sistemas");
        est1.Imprimir();
        Console.ReadKey();
    }
}
```

CLASS PERSONA

```
internal class Persona //Clase Madre
{
    //Atributos
    private string nombre, apellido, dni;
    private int año;

    //Constructor
    public Persona(string pnombre, string papellido, string pdni, int paño)
    {
        nombre = pnombre;
        apellido = papellido;
        dni = pdni;
        año = paño;
    }

    //Propiedades lectura y escritura

    public string Nombre
    {
        get { return nombre; }
        set { nombre = value; }
    }
    public string Apellido
    {
        get { return apellido; }
        set { apellido = value.ToUpper(); }
    }
    public string Dni
    {
        get { return dni; }
        set { dni = value; }
    }
}
```

```

//propiedad que me traiga concatenado el nombre y apellido

public string ApeNom
{
    get {
        return string.Concat(Nombre, "-", Apellido);
    }
}

//propiedad que me traiga la Edad
public int Edad
{
    get { return DateTime.Now.Year - año; } //Calcular edad || funciones fecha y hora
    set { año = value; }
}

//Metodo que se llame Imprimir
//polimorfismo virtual
//para que no se redefinen hay que poner virtual en la clase madre y override en la clase hija
public virtual void Imprimir()
{
    Console.WriteLine($"Nombre: {Nombre} Apellido: {Apellido} apellido concatenado: {ApeNom} , y su edad es: {Edad}, con dni {Dni}");
}
}

```

CLASS ESTUDIANTE

```

internal class Estudiante : Persona
{
    //Atributos propios del estudiante
    private int legajo;
    private int añoCarr;
    private int materiasACursar;
    private string carrera;

    //constructor La palabra clave base se utiliza para obtener acceso a los miembros de la clase base desde una clase derivada.
    //base hay que ponerlo con atributos del constructor madre.
    //los primeros atributos tienen que coincidir con los ultimos despues del :base

    public Estudiante(string nombre, string apellido, string dni, int año, int pleg, int pañoCar, int pmat, string pcarr):base(nombre,apellido, dni, año)
    {
        legajo = pleg;
        añoCarr = pañoCar;
        materiasACursar = pmat;
        carrera = pcarr;
    }

    //Propiedades
    public int Legajo
    {
        get { return legajo; }
        set { legajo = value; }
    }
    public string Carrera
    {
        get { return carrera; }
        set { carrera = value; }
    }
    public int AñoCar
    {
        get { return añoCarr; }
        set { añoCarr = value; }
    }
    public int MateriasACursar
    {
        get { return materiasACursar; }
        set { materiasACursar = value; }
    }

    //Metodo imprimir me traiga los siguientes campos(nombre, apellido, año, carrera,
    //polimorfismo override
    //para que no se redefinen hay que poner override en la clase hija y en la clase madre hay que poner virtual
    public override void Imprimir()
    {
        Console.WriteLine($"Nombre: {Apellido}, apellido concatenado: {ApeNom} , edad: {Edad}, dni: {Dni}, legajo: {Legajo}, carrera: {Carrera}");
    }
}
}

```