```
using System;
public class Persona
{
  public string Nombre { get; }
 public string Apellido { get; }
    public DateTime FechaNacimiento { get; }
public string Telefono { get; }
    public string Direccion { get; }
 public Persona(string nombre, string apellido, DateTime fechaNacimiento,
string telefono, string direccion)
    {
        if (string.IsNullOrWhiteSpace(nombre))
        {
            throw new ArgumentException("El nombre que ingresono puede estar
vacío.", nameof(nombre));  }
            if (string.IsNullOrWhiteSpace(apellido))
        {
            throw new ArgumentException("El apellido ingresado no puede
estar vacío.", nameof(apellido));   }
           if (fechaNacimiento >= DateTime.Today)
          {
            throw new ArgumentException("La fecha de nacimiento debe ser de
antes a la fecha de la actualidad.", nameof(fechaNacimiento));  }
             if (string.IsNullOrWhiteSpace(telefono))
           {
            throw new ArgumentException("El teléfono no puede estar en
blanco.", nameof(telefono)); }
            if (string.IsNullOrWhiteSpace(direccion))
          {
            throw new ArgumentException("La dirección no puede dwjarla
vacía.", nameof(direccion));  }
         Nombre = nombre;
          Apellido = apellido;
           FechaNacimiento = fechaNacimiento;
            Telefono = telefono;
              Direccion = direccion;
    } }
    public class Alumno : Persona{
    public string Carnet { get; }
    public DateTime FechaIngreso { get; }
    public Alumno(string nombre, string apellido, DateTime fechaNacimiento,
string telefono, string direccion, string carnet, DateTime fechaIngreso)
        : base(nombre, apellido, fechaNacimiento, telefono, direccion)
    {
        if (string.IsNullOrWhiteSpace(carnet))
```

```
        {       throw new ArgumentException("El carnet no puede dejarlo
vacío.", nameof(carnet));
        }  if (fechaIngreso >= DateTime.Today)
        {
            throw new ArgumentException("La fecha de ingreso debe ser
anterior a la fecha en la que estamos.", nameof(fechaIngreso));}
        Carnet = carnet;
         FechaIngreso = fechaIngreso;
     }
    }
public class Profesor : Persona
{ public string Especialidad { get; }
    public string Departamento { get; }
    public decimal Salario { get;} }
    public Profesor(string nombre, string apellido, DateTime
fechaNacimiento, string telefono, string direccion, string especialidad,
string departamento, decimal salario)
        : base(nombre, apellido, fechaNacimiento, telefono, direccion)
    {

        if (string.IsNullOrWhiteSpace(especialidad))
        {
              throw new ArgumentException("La especialidad no puede deharlo
como vacía.", nameof(especialidad))
        if (string.IsNullOrWhiteSpace(departamento)) {
                 throw new ArgumentException("El departamento no puede
dejarlo vacío.", nameof(departamento)); }}
          if (salario <0)
            { throw new ArgumentException("El salario no puede ser
registrado como negativo negativo.", nameof(salario));  }
        Especialidad = especialidad;
         Departamento =departamento;
          Salario= salario;
    }
    public decimal CalcularSalarioAnual()
    {   return Salario * 12;  }
    using System;

public class Alumno : Persona
{
    public string Clave { get; }
     public int Grado { get; }
       public string Seccion { get; }
```

```
       public Alumno(string nombre, string apellido, DateTime
fechaNacimiento, string telefono, string direccion, string clave, int grado,
string seccion)
        : base(nombre, apellido, fechaNacimiento, telefono, direccion)
          {
        if (string.IsNullOrWhiteSpace(clave))
        {     throw new ArgumentException("La clave que ingreso puede estar
vacía.", nameof(clave));  }
        if (grado <= 0)
        {
            throw new ArgumentException("El grado que ingrese debe ser mayor
que cero.", nameof(grado)); }

        if (string.IsNullOrWhiteSpace(seccion)) {
            throw new ArgumentException("La sección no puede dejarla sin
nada, vacía.", nameof(seccion));  }
        Clave = clave;
          Grado = grado;
           Seccion = seccion;
        }
        }
```