**CASO 1**

**Miguel Palacios**

**Vicent Sornosa**

**Javier Ruano**

**Jesús Pico**

# EJERCICIO 2

public class Grupo

{

private string nombre { get; set; }

private string acronimo { get; set; }

private DateTime fechaAlta { get; set; }

private int numDocumentos() { }

private ICollection<Profesor> profesorado { get; set; }

private ICollection<Alumno> alumado { get; set; }

}

public abstract class Miembro

{

private string nombre;

private string email;

}

public class Alumno : Miembro

{

private int numExpediente { get; set; }

private ICollection<Grupo> alumnado { get; set; }

}

public class Profesor : Miembro

{

private ICollection<Conversacion> creador { get; set; }

}

public class Conversacion

{

private string texto { get; set; }

private bool modificado { get; set; }

private Profesor creador;

private ICollection<Conversacion> inicial { get; set; }

private ICollection<Conversacion> subconversaciones { get; set; }

}

# EJERCICIO 3

public Grupo(string nombre, string acronimo, DateTime fechaAlta, Profesor profesorado)

{

}

public Miembro(string nombre, string email)

{

}

public Alumno(int numExpediente, Grupo alumnado) : base(nombre, email)

{

}

// RELAJAR: Conversacion creador

public Profesor() : base(nombre, email)

{

}

// RELAJAR: Conversacion inicial

public Conversacion(string texto, bool modificiado, Profesor creador)

{

}

# EJERCICIO 4

Profesor creador = new Profesor();

Conversacion c = new Conversacion(texto, modificado, creador);

creador.addConversacion(c);

Conversacion inicial = new Conversacion(texto, modificado, creador);

c.addConversacion(inicial);