

Relaciones entre objetos

Los objetos pueden relacionarse entre sí.
Existen relaciones **jerárquicas** y relaciones de **contención**.

Relación jerárquica

Esta relación está dada por la herencia. Cuando decimos, por ejemplo, que la clase Empleado hereda de Persona estamos estableciendo una relación jerárquica entre ambos objetos.

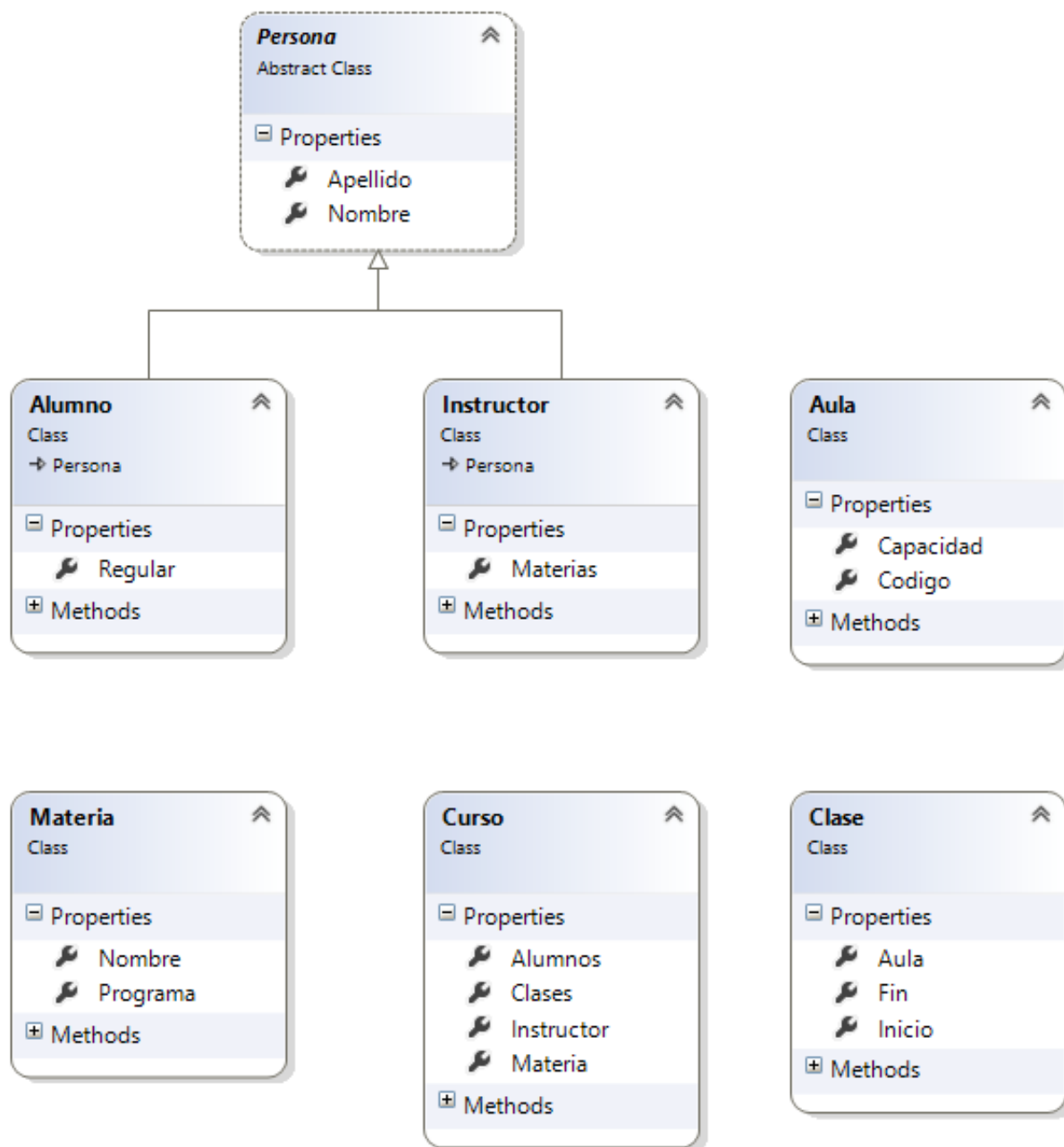
Relaciones de contención

Esta relación se establece cuando un objeto tiene uno o más atributos que hacen referencia a otros objetos. Estas relaciones de contención puede darse de uno a uno o de uno a muchos.

Por ejemplo:

- Los automóviles tienen un motor: esta es una relación **uno a uno**.
- Los camiones tienen (n) ruedas: esta relación es de **uno a muchos**.

En el **diagrama de clases** que se muestra a continuación se pueden apreciar ambos tipos de relaciones:



Comenzemos por la clase **Persona**: es una clase abstracta con 2 atributos simples. No hay ningún tipo de relación en esta clase.

C#

```

public abstract class Persona
{
    public string Nombre {get; set;}
    public string Apellido { get; set; }
}
  
```

VB.NET

```

Public MustInherit Class Persona
    Public Property Nombre As String
    Public Property Apellido As String
End Class

```

Alumno hereda de Persona y tiene un atributo booleano. En este caso, tampoco hay relaciones.

C#

```

public class Alumno:Persona
{
    public Alumno(string pNombre, string pApellido, bool pRegular)
    {
        this.Nombre = pNombre;
        this.Apellido = pApellido;
        this.Regular = pRegular;
    }
    public bool Regular { get; set; }
}

```

VB.NET

```

Public Class Alumno
    Inherits Persona

    Public Sub New(pNombre As String, pApellido As String, pRegular As Boolean)
        Me.Nombre = pNombre
        Me.Apellido = pApellido
        Me.Regular = pRegular
    End Sub

    Public Property Regular As Boolean
End Class

```

Sin embargo, [Instructor](#), que también hereda de persona, tiene un atributo [Materias](#) que está compuesto por una colección de objetos del tipo [Materia](#). Esta es una relación del tipo uno a muchos.

C#

```

public class Instructor:Persona
{
    public Instructor(string pNombre, string pApellido, List<Materia> pMaterias)
    {
        this.Nombre = pNombre;
        this.Apellido = pApellido;
        this.Materias = pMaterias;
    }
    public List<Materia> Materias { get; set; } //Relacion muchos a muchos
}

```

VB.NET

```

Public Class Instructor
    Inherits Persona

    Public Sub New(pNombre As String, pApellido As String, pMaterias As List(Of Materia))
        Me.Nombre = pNombre
        Me.Apellido = pApellido
        Me.Materias = pMaterias
    End Sub

    Public Property Materias As List(Of Materia) 'Relacion muchos a muchos

End Class

```

Materia tiene 2 atributos simples: no existen relaciones con otros objetos, al igual que la clase **Aula**.

C#

```

public class Materia
{
    public Materia(string pNombre, string pPrograma)
    {
        this.Nombre = pNombre;
        this.Programa = pPrograma;
    }
    public string Nombre { get; set; }
    public string Programa { get; set; }
}

public class Aula
{
    public Aula(string pCodigo, int pCapacidad)
    {
        this.Codigo = pCodigo;
        this.Capacidad = pCapacidad;
    }
    public string Codigo { get; set; }
    public int Capacidad { get; set; }
}

```

VB.NET

```

Public Class Materia

    Public Sub New(pNombre As String, pPrograma As String)
        Me.Nombre = pNombre
        Me.Programa = pPrograma
    End Sub

    Public Property Nombre As String
    Public Property Programa As String

End Class

```

```

Public Class Aula
    Public Sub New(pCodigo As String, pCapacidad As Integer)
        Me.Codigo = pCodigo
        Me.Capacidad = pCapacidad
    End Sub

    Public Property Codigo As String
    Public Property Capacidad As Integer

End Class

```

Sin embargo la clase [Clase](#) tiene establece una relación uno a uno con la clase [Aula](#).

C#

```

public class Clase
{
    public Clase(DateTime pInicio, DateTime pFin, Aula pAula)
    {
        this.Inicio = pInicio;
        this.Fin = pFin;
        this.Aula = pAula;
    }
    public DateTime Inicio { get; set; }
    public DateTime Fin { get; set; }
    public Aula Aula { get; set; } //Relacion 1 a muchos
}

```

VB.NET

```

Public Class Clase
    Public Sub New(pInicio As Date, pFin As Date, pAula As Aula)
        Me.Inicio = pInicio
        Me.Fin = pFin
        Me.Aula = pAula
    End Sub

    Public Property Inicio As Date
    Public Property Fin As Date
    Public Property Aula As Aula

End Class

```

Finalmente, la clase [Curso](#) posee las siguientes relaciones:

- Alumnos es una colección de [Alumno](#) (relación uno a muchos).
- Clases es una colección de [Clase](#) (relación uno a muchos).
- Instructor es del tipo [Instructor](#) (relación uno a uno).
- Materia es del tipo [Materia](#) (relación uno a uno).

C#

```

public class Curso
{
    public Curso(Instructor pInstructor, Materia pMateria,
                List<Alumno> pAlumnos, List<Clase> pClases)
    {
        this.Instructor = pInstructor;
        this.Materia = pMateria;
        this.Alumnos = pAlumnos;
        this.Clases = pClases;
    }
    public Instructor Instructor { get; set; } //Relacion 1 a muchos
    public Materia Materia { get; set; } //Relacion 1 a muchos
    public List<Alumno> Alumnos { get; set; } //Relacion muchos a muchos
    public List<Clase> Clases { get; set; } //Relacion muchos a muchos
}

```

VB.NET

```

Public Class Curso
    Public Sub New(pInstructor As Instructor, pMateria As Materia,
                pAlumnos As List(Of Alumno), pClases As List(Of Clase))

        Me.Instructor = pInstructor
        Me.Materia = pMateria
        Me.Alumnos = pAlumnos
        Me.Clases = pClases

    End Sub

    Public Property Instructor As Instructor 'Relacion 1 a muchos
    Public Property Materia As Materia 'Relacion 1 a muchos
    Public Property Alumnos As List(Of Alumno) 'Relacion muchos a muchos
    Public Property Clases As List(Of Clase) 'Relacion Muchos a muchos

End Class

```