

Seminari

SET5- 1

Enginyeria del
Programari

ETS Enginyeria
Informàtica

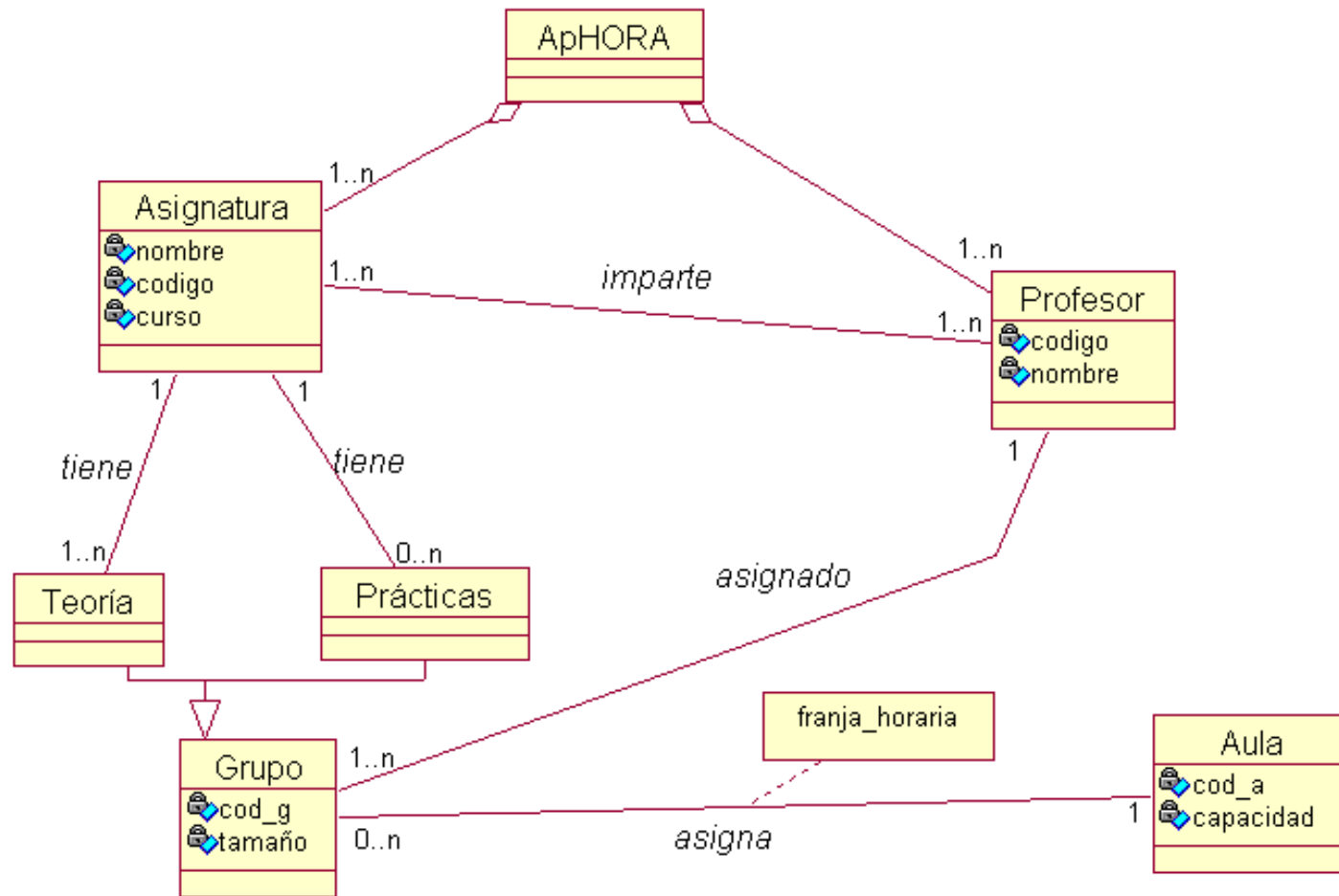
DSIC – UPV

Curs 2024-2025

Tema 5. Disseny de la Capa Lògica

(Exercici 1 - ApHora)

Butlletí – Exercici 1 (ApHORA)



Butlletí – Exercici 1 (ApHora)

```
public class ApHora
{
    private ICollection<Profesor> profesores;
    private ICollection<Asignatura> asignaturas;
    public ApHora()
    {
        this.profesores = new List<Profesor>();
        this.asignaturas = new List<Asignatura>();
    }
}

public class Aula
{
    private int cod_a;
    private int capacidad;
    private ICollection<Grupo> grupos;
    public Aula(int cod_a, int capacidad)
    {
        this.cod_a = cod_a;
        this.capacidad = capacidad;
        grupos = new List<Grupo>();
    }
}
```

```
public class Profesor
{
    private int codigo;
    private string nombre;
    private ApHora enApHora;
    private ICollection<Asignatura> asignaturas;
    private ICollection<Grupo> grupos;
    public Profesor(int codigo, string nombre,
                    ApHora apHora)
    {
        this.codigo = codigo;
        this.nombre = nombre;
        this.enApHora = apHora;
        this.asignaturas = new List<Asignatura>();
        this.grupos = new List<Grupo>();
    }
}
```

Butlletí – Exercici 1 (ApHora)

```
public class Asignatura
{
    private string nombre;
    private int codigo;
    private string curso;
    private ApHora enApHora;
    private ICollection<Profesor> profesores;
    private ICollection<Teoria> gruposTeoria;
    private ICollection<Practica> gruposPracticas;

    public Asignatura(String nombre, int codigo, String curso, Profesor profesor, ApHora apHora)
    {
        this.nombre = nombre;
        this.codigo = codigo;
        this.curso = curso;
        this.enApHora = apHora;
        this.profesores = new List<Profesor>();
        this.profesores.Add(profesor);
        this.gruposTeoria = new List<Teoria>();
        this.gruposPracticas = new List<Practica>();
    }
}
```

Butlletí – Exercici 1 (ApHora)

```
public abstract class Grupo
{
    private int cod_g;
    private int tamanyo;
    private DateTime hora_desde;
    private DateTime hora_hasta;
    private Aula aula;
    private Profesor profesor;
    public Grupo(int cod_g, int tamanyo,
DateTime hora_desde, DateTime hora_hasta, Aula
aula, Profesor profesor)
    {
        this.cod_g = cod_g;
        this.tamanyo = tamanyo;
        this.hora_desde = hora_desde;
        this.hora_hasta = hora_hasta;
        this.aula = aula;
        this.profesor = profesor;
    }
}
```

```
public class Teoria : Grupo
{
    private Asignatura asignaturaT;

    public Teoria(int cod_g, int tamanyo, DateTime hora_desde,
DateTime hora_hasta, Aula aula,
        Profesor profesor, Asignatura asignatura) : base(cod_g,
tamanyo, hora_desde, hora_hasta, aula, profesor)
    {
        asignaturaT= asignatura;
    }
}

public class Practica : Grupo
{
    private Asignatura asignaturaP;

    public Practica(int cod_g, int tamanyo, DateTime hora_desde,
DateTime hora_hasta, Aula aula,
        Profesor profesor, Asignatura asignatura) : base(cod_g,
tamanyo, hora_desde, hora_hasta, aula, profesor)
    {
        asignaturaP= asignatura;
    }
}
```

Butlletí – Exercici 1 (ApHora)

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
    {
```

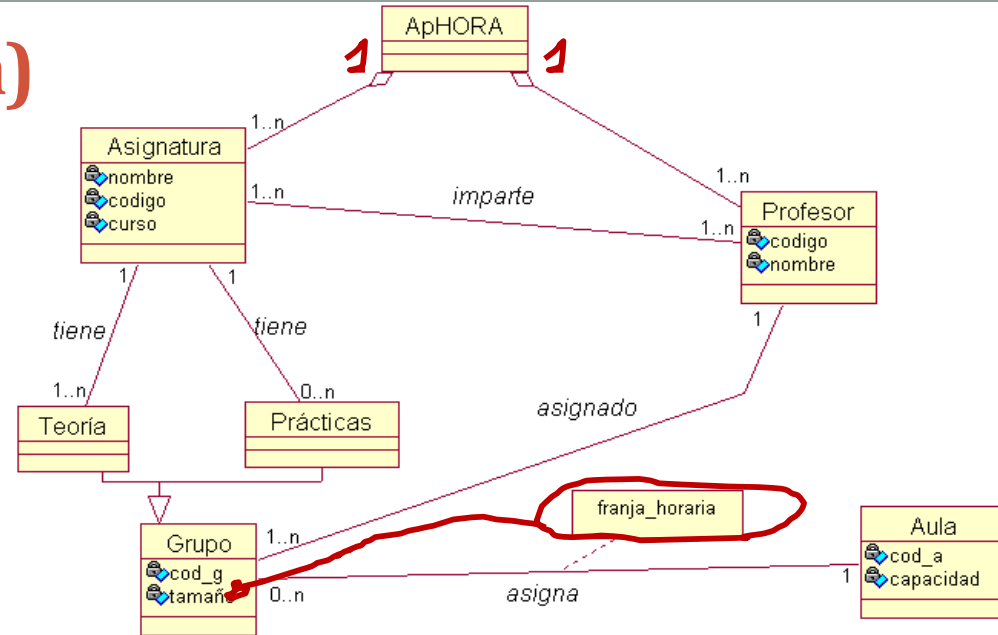
```
        ApHora miApHora = new ApHora();
```

```
        Aula aula11 = new Aula(11,75);
```

```
        Profesor mc = new Profesor(1, "mc", miApHora);
        miApHora.AddProfesores(mc);
```

```
        Asignatura isw = new Asignatura("ISW", 1, "tercero", mc, miApHora);
        mc.AddAsignatura(isw);
        miApHora.AddAsignaturas(isw);
```

```
        Teoria teo_isw = new Teoria(11, 55, new DateTime(2021, 9, 14, 11, 30, 0), new DateTime(2021, 9, 14, 13, 0, 0), aula11, mc, isw);
        isw.AddTeoria(teo_isw);
        mc.AddGrupos(teo_isw);
        aula11.AddGrupo(teo_isw); //dejamos consistente el modelo, la navegación de la asociación es bidireccional
```



// en una inicialització mínima, no necessitem tenir grups de pràctiques . Si ens demanen crear una instància de cada classe, llavors s'invocaria el constructor de Pràctiques i es deixaria consistent el model (respecte a la navegabilitat), de forma similar a Teoria.