

# DOCUMENTACIÓN SOBRE RESTRICCIONES OCL DEL METAMODELO PDA

**Autor:** Sergio Martín Ledesma

**Última actualización:** 22/03/2024

**Name:** actividad1\_PDA\_23\_24

**NS Prefix:** actividad1\_PDA\_23\_24

**NS URI:** [http://www.example.org/actividad1\\_PDA\\_23\\_24](http://www.example.org/actividad1_PDA_23_24)

## Contenido

DOCUMENTACIÓN SOBRE RESTRICCIONES OCL DEL METAMODELO PDA.....	1
A. FIGURA DEL METAMODELO .....	3
A. EJERCICIO 1 .....	3
a) ATRIBUTOS DERIVADOS .....	3
b) INVARIANTES .....	4
A. EJERCICIO 3.....	5
a) Código completo del archivo .ecore .....	6

## A. FIGURA DEL METAMODELO



### a) ATRIBUTOS DERIVADOS

- **D01. Añade el atributo `sexenio_vivo` al concepto de Sexenio. Éste estará a true si hace menos de seis años (desde el año actual, 2024) que se dicho sexenio al profesorado.**
  - `(2024-self.AniolInicio)<6;`
- **D2. Añade el atributo `descuento_docencia_tfe` al concepto Profesor que calcule el descuento que procede para el profesor a partir de sus méritos de docencia. Cada TFE dirigido o co-dirigido se contabiliza como 4 horas. La participación en TFE como tribunal se contabiliza como 1 hora.**
  - `self.tribunal_de->size() + (self.tfe->size()*4 + self.meritos->selectByType(TFE)->size()*4);`
- **D3. Añade el atributo `descuento_docencia_evaluaciones_docentes` al concepto Profesor que calcule el descuento que procede para el profesor a partir de sus méritos por evaluaciones docentes. Cada evaluación docente con una calificación superior o igual a 7 se contabilizará como 5 horas de descuento.**
  - `self.meritos->selectByType(Evaluacion)->select(Calificacion >= 7)->size()* 5`

- **D4. Añade un atributo descuentos\_docencia al concepto Profesor que se calculará a partir de la suma de los atributos derivados descuento\_docencia\_tfe y descuento\_docencia\_evaluaciones\_docentes. El valor resultante no podrá ser superior a 40 horas.**
  - `descuento_docencia_tfe +  
descuento_docencia_evaluaciones_docentes;`
- **D5. Añade el atributo descuento\_investigación\_final al concepto Profesor que calcule el descuento que procede para el profesor a partir de sus méritos de investigación. No podrá superar 40 horas. Si el profesor tiene algún sexenio, cada sexenio se multiplica por 5 horas, en caso contrario el número de publicaciones en JCR se multiplicará por 1 hora. (0.75 puntos)**
  - `if(self.meritos->selectByType(Sexenio)->size() >0){  
    derivation: self.meritos->selectByType(Sexenio)->size();  
}  
else{  
    derivation: self.meritos->selectByType(Publicacion)->select(e |  
    e.Indexacion = TipoIndexacion::JCR)->size()  
}`
- **D6. Añade el atributo descuento\_gestión\_final al concepto Profesor que calcule el descuento que procede para el profesor a partir de sus méritos de gestión. Por cada comisión a la que pertenezca el profesor se le sumará 1 hora de descuento. Si además tiene un CargoNoEstatutario de Secretario se le sumará un descuento de 4 horas y si dicho cargo es de Subdirector se le sumará un descuento de 6 horas. No podrá superar 40 horas.**
  - `self.meritos->selectByType(Comision)->size() + (self.meritos->selectByType(Secretario)->size() * 4) + (self.meritos->selectByType(Subdirector)->size() * 6);`
- **D7. Añade el atributo descuento\_final al concepto Profesor teniendo en cuenta que la suma de los descuentos finales por investigación, docencia y gestión no podrá superar 80 horas**
  - `self.descuento_docencias + self.descuento_investigacion_final +  
self.descuento_gestion_final`
- **D8. Referencia derivada: Un área de conocimiento imparte docencia en aquellos campus donde sus profesores (los profesores que están adscritos al área están impartiendo docencia.**
  - `self.profesores.imparte_en->asSet()`

## **b) INVARIANTES**

- **I1. Un profesor no puede tener más de 6 sexenios de investigación.**
  - (Contexto Profesor)
  - `invariant self.meritos->selectByType(Sexenio)->size()<6;`
- **I2. Los sexenios siempre estarán comprendidos entre seis años, siendo el año de inicio menor que el año final. (0,25 puntos).**
  - (Contexto Sexenio)

- invariant self.AniolInicio < self.AnioFinal
- **I3. El director y co-director de un TFE no pueden ser la misma persona.**
  - (Contexto TFE)
  - invariant co\_director\_distinto: self.co\_director <> self.oclContainer;
- **I4. Al menos un profesor del tribunal debe pertenecer al mismo área de conocimiento que el profesor tutor del TFE. (0,75 puntos).**
  - (Contexto TFE)
  - invariant self.tribunal.oclContainer->includes(self.co\_director.oclContainer);
- **I5. En el tribunal de un TFE no puede aparecer ni el director ni el co-director del TFE.**
  - (Contexto TFE)
  - invariant tribunal\_sin\_director\_o\_codirector: self.tribunal->excludes(self.oclContainer) and self.tribunal->excludes(self.co\_director);
- **I6. Un profesor no podrá tener dos méritos de CargoNoEstatutario del mismo tipo (secretario / subdirector) (0,75 puntos).**
  - (Contexto Profesor)
  - invariant self.meritos->selectByType(Secretario)->size() <= 1 and self.meritos->selectByType(Subdirector->size() <= 1;
- **I7. No pueden solaparse los años correspondientes a distintos sexenios de un mismo profesor. (1 puntos).**
  - (Contexto Profesor)
  - invariant self.meritos->selectByType(Sexenio)->forAll( e1, e2 | e1 <> e2 implies (not((e1.AniolInicio <= e2.AniolInicio and e2.AniolInicio<=e1.AnioFinal) or (e1.AniolInicio <= e2.AnioFinal and e2.AnioFinal<=e1.AnioFinal) )) );

## A. EJERCICIO 3

Consulta N°	Contexto	Texto de la consulta	Resultados de la consulta
Q1	Campus	Self.profesores	
Q2			
Q3			
Q4	Universidad	self.departamentos->select(Name = 'DEP').areaconocimientos.profesores.meritos->selectByType(TFE)	
Q5	Departamento	self.areaconocimientos.profesores.meritos->selectByType(Sexenio)->size()	
Q6			

## a) Código completo del archivo .ecore

```
import ecore : 'http://www.eclipse.org/emf/2002/Ecore';

package actividad1_PDA_23_24 : actividad1_PDA_23_24 =
'http://www.example.org/actividad1_PDA_23_24'
{
    class Universidad
    {
        attribute Nom_universidad : String[1]{ id };
        property campus : Campus[*|1]{ ordered composes };
        property departamentos : Departamento[*|1]{ ordered composes };
    }
    class Campus
    {
        attribute Name : String[1]{ id };
        property profesores_campus#imparte_en : Profesor[*|1];
    }
    class Departamento
    {
        attribute Name : String[1]{ id };
        property areaconocimientos : AreaConocimiento[*|1]{ ordered
composes };
    }
    class AreaConocimiento
    {
        attribute Name : String[1]{ id };
        property profesores : Profesor[*|1]{ ordered composes };
        property imparte_docencia_en : Campus[*|1]{ ordered derived }
    }
    _____{
    _____initial: self.profesores.imparte_en->oclAsSet();
    _____
```

```

    }
}

class Profesor
{
    attribute Name : String[1] { id };
    attribute CategoriaProfesional : CategoriaProfesional[?];
    attribute Doctorado : Boolean[1];
    attribute descuento_docencia_tfe : ecore::EInt[1] { derived transient
volatile }
    {
        initial: self.tribunal_de->size() + (self.tfe->size()*4 + self.meritos-
>selectByType(TFE)->size()*4);
    }
    attribute descuento_docencia_evaluaciones_docentes :
ecore::EInt[1] { derived transient volatile }
    {
        initial: self.meritos->selectByType(Evaluacion)-
>select(Calificacion >= 7)->size() * 5;
    }
    attribute descuento_docencias : ecore::EInt[1] { derived
transient volatile }
    {
        initial:
        let suma_descuento:Integer =
self.descuento_docencia_evaluaciones_docentes + self.descuento_docencia_tfe in
        if(suma_descuento < 40) then
            suma_descuento
        else
            40
        endif;
    }
    attribute descuento_investigacion_final :
ecore::EInt[1] { derived transient volatile }
    {

```

```

initial:

    let suma_investigacion:Integer = self.meritos-
>selectByType(Sexenio)->size() * 5 in

        if(self.meritos->selectByType(Sexenio)->size() >0) then

            if(suma_investigacion<40) then

                suma_investigacion

            else

                40

            endif

        else

            if (self.meritos->selectByType(Publicacion)->select(e |
e.Indexacion = TipoIndexacion::JCR)->size() < 40) then

                self.meritos->selectByType(Publicacion)->select(e |
e.Indexacion = TipoIndexacion::JCR)->size()

            else

                40

            endif

        endif;

    }

    attribute descuento_gestion_final :
ecore::EInt[1] { derived transient volatile }

    {

        initial:

            let suma_descuento:Integer = self.meritos-
>selectByType(Comision)->size() + (self.meritos->selectByType(Secretario)->size() * 4) +
(self.meritos->selectByType(Subdirector)->size() * 6) in

                if(suma_descuento < 40) then

                    suma_descuento

                else

                    40

                endif;

            }

            attribute descuento_final :
ecore::EInt[1] { derived transient volatile }

```



```

{
    initial:

        let suma_descuento:Integer = self.descuento_docencias +
self.descuento_investigacion_final + self.descuento_gestion_final in

        if(suma_descuento < 80) then

            suma_descuento

        else

            80

        endif;

    }

    property meritos : Merito[*|1]

    attribute

    property tfe#co_director :

    property tribunal_de#tribunal

    property

    invariant menos_6_sexenios:

    invariant

        dos_meritos_diferentes_noEstatuarios: self.meritos->selectByType(Secretario)-
>size() <= 1 and self.meritos->selectByType(Subdirector)->size() <= 1;

        invariant

            solapamiento_sexenios:

self.meritos->selectByType(Sexenio)->forAll( e1, e2 | e1 <> e2 implies (not((e1.AniolInicio
<= e2.AniolInicio and e2.AniolInicio<=e1.AnioFinal) or (e1.AniolInicio <= e2.AnioFinal and
e2.AnioFinal<=e1.AnioFinal) )) );

    }

    enum CategoriaProfesional {

    {

        literal CU;

```

```

serializable }

```

```

        literal TU = 1;
        literal TEU = 2;
        literal CD = 3;
        literal AYD = 4;
        literal PC = 5;
        literal AY = 6;
    }
    abstract class Merito
    {
        attribute Name : String[1] { id
};

    }
    abstract class MeritoInvestigacion

    abstract class MeritoGestion

    abstract class MeritoDocencia

    class TFE extends MeritoDocencia
    {
        attribute Titulo : String[1];
        attribute Autor : String[1];
        attribute Calificacion :

        property co_director#tfe :

        property tribunal#tribunal_de

        invariant

        co_director_distinto: self.co_director <> self.oclContainer;

        invariant

        mismo_area_conocimiento_codirector_tribunal: self.tribunal.oclContainer-
>includes(self.co_director.oclContainer);

        invariant

```

tribunal\_sin\_director\_o\_codirector: **self.tribunal->excludes(self.oclContainer)**  
**and self.tribunal->excludes(self.co\_director);**

}

**class** Evaluacion **extends**

*MeritoDocencia*

{

**attribute** Calificacion :

*ecore::EFloat[1];*

**attribute** AnioEvaluacion :

*ecore::EInt[1];*

}

**abstract class** CargoNOEstatuario

**extends** *MeritoGestion;*

**class** Secretario **extends**

*CargoNOEstatuario;*

**class** Subdirector **extends**

*CargoNOEstatuario;*

**class** Sexenio **extends**

*MeritoInvestigacion*

{

**attribute** AnioInicio :

*ecore::EInt[1];*

**attribute** AnioFinal :

*ecore::EInt[1];*

**attribute** sexenio\_vivo :

**Boolean[1] { derived transient volatile }**

{

**initial:** (2024-

**self.AnioInicio)<6;**

}

**invariant**

inicio\_menor\_final: **self.AnioInicio < self.AnioFinal;**

**invariant**

compendio\_6\_anios: 0< (**self.AnioFinal - self.AnioInicio**)**and (self.AnioFinal - self.AnioInicio)<=6;**

}

*MeritoInvestigacion*

**String**[1];

Nombre\_revista : **String**[1];

*CuartilIndexacion*[1];

*TipoIndexacion*[1];

**serializable** }

**serializable** }

*MeritoGestion*;

**class** Publicacion **extends**

{

**attribute** Titulo :

**attribute**

**attribute** Cuartil :

**attribute** Indexacion :

}

**enum** CuartilIndexacion {

{

**literal** C1;

**literal** C2 = 1;

**literal** C3 = 2;

**literal** C4 = 3;

}

**enum** TipoIndexacion {

{

**literal** SJR;

**literal** JCR = 1;

}

**class** Comision **extends**

}