

Enunciado:

La transcripción que aparece a continuación corresponde a una entrevista a una ingeniera de software que necesita un sistema de información para ayudarle en la gestión de sus proyectos.

Pregunta: Bien, coménteme cuál sería el principal objetivo del sistema que usted necesita.

Respuesta: Bueno, básicamente lo que quiero es un gestor de proyectos sencillo, que me permita gestionar las tareas asociadas y asignárselas a los empleados.

P: Empecemos entonces por los proyectos, ¿qué información quiere que gestione el sistema sobre ellos?

R: Básicamente, el nombre del proyecto, una descripción y el presupuesto que tiene. Puede que luego sea interesante añadirle más información, pero de momento me valdría con eso.

P: De acuerdo, ha hablado antes de tareas. ¿Me puede explicar ese concepto?

R: Sí, claro. Nosotros descomponemos los proyectos en tareas, lo que se conoce como WBS (*Work Breakdown Structure*). Cada tarea tiene un identificador que usamos para referirnos a ella, p.e. la tarea T-28F. También tienen una descripción y una estimación del coste de su realización. En el caso de que una tarea sea muy compleja, puede descomponerse en subtareas, que también pueden descomponerse recursivamente.

P: Entiendo, ¿esas tareas tienen algún orden especial?

R: El que le vayamos dando al crearlas, aunque luego podríamos cambiarlo.

P: Entiendo, un proyecto tiene una secuencia de tareas que a su vez pueden tener una secuencia de subtareas y así recursivamente. Debemos conocer el orden. ¿Qué más hay que saber sobre las tareas?

R: Pues que una vez que creamos la WBS, vamos asignando las tareas a los empleados.

P: ¿En qué consiste una asignación de una tarea a un empleado?

R: Básicamente, se encarga a un empleado la realización de una tarea con una fecha de inicio y otra de fin.

P: ¿A cuántos empleados se le asigna una misma tarea?

R: Cuando decidimos a quién asignársela, a uno sólo. Nuestras tareas simples están pensadas para que las pueda realizar un solo empleado.

P: ¿Qué es una tarea simple?

R: Una tarea que no está descompuesta en tareas más simples.

P: Entiendo, sólo se asignan tareas simples, ¿no es así?

R: Bueno, no necesariamente. Podemos asignar tareas complejas para su supervisión.

P: Entiendo, se puede asignar cualquier tarea (compleja o simple) a cualquier empleado.

R: Sí, aunque eso depende del rol que tenga el empleado en el proyecto.

P: Explíqueme eso.

R: Bueno, en cada proyecto se definen una serie de roles y cada empleado puede desempeñar distintos roles en distintos proyectos. Por ejemplo, uno puede ser el director de un proyecto y a la vez ser el responsable de pruebas de otro proyecto.

P: Entiendo, ¿un mismo empleado puede jugar más de un rol en un mismo proyecto?

R: No es frecuente, pero podría ocurrir, especialmente en proyectos pequeños.

P: Una vez establecido un rol de un empleado en un proyecto, ¿puede cambiar?

R: Sí, reasignamos los empleados según los proyectos que van saliendo.

P: Entonces, le interesa saber desde qué fecha hasta qué fecha un empleado ha desempeñado un rol en un proyecto ¿no?

R: Sí, quiero saber qué empleados han ido pasando por cada proyecto y en qué fechas.

P: Entiendo que cada rol está asociado a un cargo predefinido, ¿es así?

R: Sí, claro. Los nombres de los roles coinciden con los cargos habituales: director, analista, responsable de pruebas, etc. Tenemos definidos los roles según la norma ISO-9001.

P: Muy bien, creo que con eso tengo para una primera versión. Gracias.

Ejercicio:

Teniendo en cuenta el enunciado anterior, elabore un modelo conceptual usando diagramas de clases UML. Si identifica algún problema en el enunciado, indíquelo junto con el modelo desarrollado.

Criterios de corrección (partiendo de 10 puntos)

- Cada clase errónea o inexistente: -2 puntos
- Cada asociación errónea o inexistente: -1.5 puntos
- Cada multiplicidad errónea: -1 punto
- Cada atributo erróneo: -0,5 punto

Solución 1 (con composición):

@startuml

```
' Opciones gráficas
hide circle
hide methods
```

' Entidades

```
class Proyecto {
  nombre
  descripción
  presupuesto
}
```

```
class Empleado {
  dni
  nombre
  ?
}
```

```
class Tarea {
  id
  descripción
  estimación
}
```

```
class Rol {
  nombre : Cargo
  fecha_inicio [0..1]
  fecha_fin [0..1]
}
```

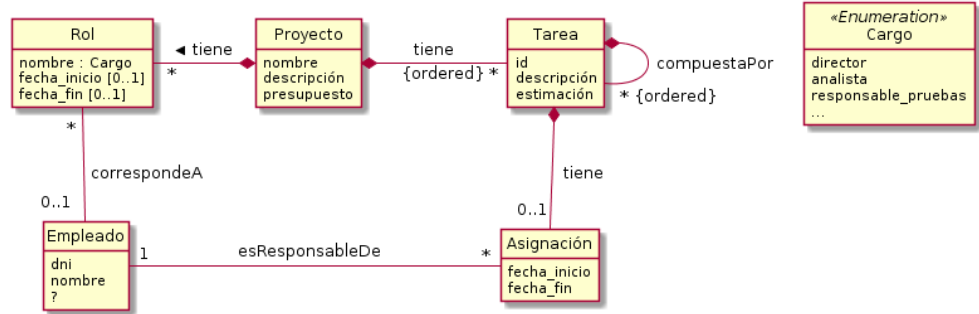
```
class Cargo <<Enumeration>> {
  director
  analista
  responsable_pruebas
  ...
}
```

```
class Asignación {
  fecha_inicio
  fecha_fin
}
```

' Asociaciones

```
Proyecto *--right- "{ordered} *" Tarea : " tiene "
Proyecto *--left- " * " Rol : " tiene ">
Tarea *-- " * {ordered}" Tarea : "compuestaPor"
Tarea *--down- "0..1" Asignación : " tiene"
Asignación "*" --left- "1" Empleado : " esResponsableDe "
Rol " * " --down- "0..1" Empleado : " correspondeA "
```

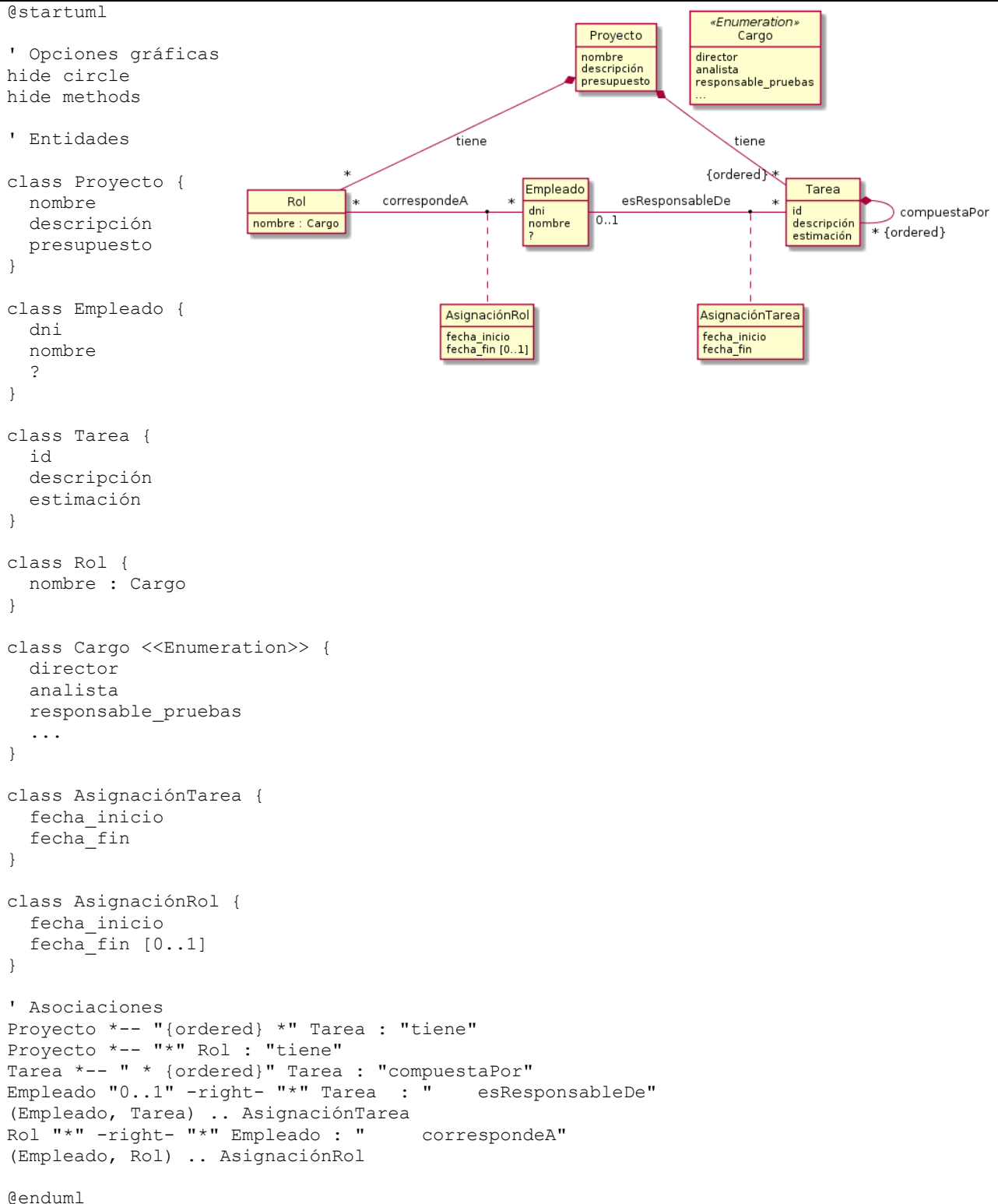
@enduml



Prueba de modelado conceptual

febrero 2018

Solución 2 (con clase asociación):



Código PlantUML (probar en <https://www.planttext.com/>)

SOLUCIÓN ALTERNATIVA

```
@startuml
' Opciones gráficas
hide circle
hide methods
skinparam shadowing false
' Entidades
class Proyecto {
    nombre
    descripción
    presupuesto
}
class Empleado {
    dni
    nombre
    ?
}
class Tarea {
    id
    descripción
    estimación
    fecha_inicio
    fecha_fin [0..1]
}
class E_Rol <<Enumeration>> {
    director
    analista
    responsable_pruebas
    ...
}
class AsignaciónRol {
    rol:E_Rol
    fecha_inicio
    fecha_fin [0..1]
}
class AsignaciónTarea {
    fecha_inicio
    fecha_fin [0..1]
}
' Asociaciones
Proyecto --> "*" Tarea : "tiene"
Tarea --> "*" Tarea : "compuestaPor"
Proyecto --> "*" AsignaciónRol : "P-Rol"
Empleado --> "*" AsignaciónRol : "E-Rol"
Tarea --> "*" AsignaciónTarea : "T-AsigTarea"
AsignaciónRol --> "*" AsignaciónTarea : "R-AsigTarea"
note right
    RN-01: Fechas dentro de rango
    El intervalo de fechas de AsignaciónTarea
    debe estar dentro del intervalo de AsignaciónRol,
    y del intervalo de ejecución de la Tarea
end note
enduml
```

