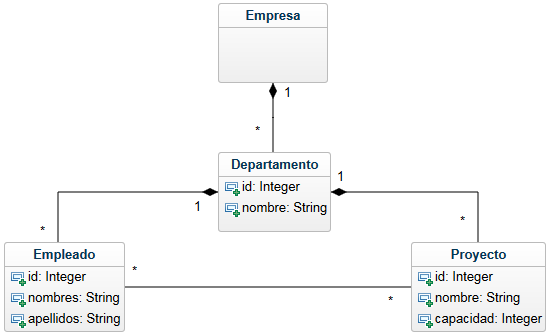
**Enunciado del problema**

Se ha realizado un modelo para representar la información de una Empresa descrito en el siguiente modelo de clases:



Para todas las clases se supone la creación de getters y setters.

**Departamento**

Un departamento tiene un identificador y nombre. Además posee Proyectos y Empleados.

**Proyecto**

Un proyecto tiene un identificador, nombre, capacidad. Ademas posee empleados asociados.

* La capacidad de un proyecto determina la cantidad máxima de empleados que puede aceptar.
* Asociar un empleado consiste en agregarlo en la lista de empleados asociados.
* Desasociar un empleado consiste en quitarlo de la lista de empleados asociados.
* El listado de empleados asociados consiste en informar (imprimir en la salida de pantalla) Nombre Proyecto, [Salto de línea], Nombre y apellidos del Empleado, [Salto de línea]

**Empleado**

Un empleado tiene un identificador, nombres, apellidos. Ademas posee proyectos asociados.

* Un empleado puede asociar proyectos.
  + Al asociar un proyecto si el número de empleados asociados es mayor o igual a la capacidad en el proyecto debe crear una excepción (DCapacidadRebasadaException) que informe que no es posible hacer la asociación (“No es posible asociar a ”, Nombres y apellidos del empleado, “ porque no hay más capacidad en ”, Nombre del proyecto).
  + De igual forma si el Empleado ya está asociado debe crear una excepción (DEmpleadoAsociadoException) que informe que no es posible hacer la asociación (“Ya está asociado ”, Nombres y apellidos del Empleado, “ en ”, Nombre del proyecto)
  + De lo contrario realizar la asociación para lo cual debe proceder a agregar el proyecto como proyecto asociado y asociar al empleado en el proyecto e informar que se hizo la asociación (Nombres y apellidos del Empleado, “ fue asociado en ”, Nombre del proyecto).
* Un empleado puede desasociar proyectos que consiste en quitar (desasociar) el proyecto como proyecto asociado y desasociar al empleado en el proyecto.
* El listado de proyectos asociados consiste en informar (imprimir en la salida de pantalla) Nombres y apellidos del empleado, [Salto de línea], Identificador proyecto, [Tab], Nombre proyecto, [Salto de línea]

**Empresa**

Una empresa debe tener Identificador y nombre. Ademas posee departamentos

**Operaciones requeridas en Empresa**

Se desea realizar las siguientes operaciones en la Empresa (DEmpresa):

* Inicializar: consiste en crear los siguientes departamentos:
  + “Tecnologias de la Informacion” con identificador 1.

En el departamento creado se deben crear los proyectos:

* + - “Ingenieria de Software" con capacidad 2 e identificador 1.
    - "Soporte usuarios" con capacidad 1 e identificador 2.
    - "Infraestructura" con capacidad 2 e identificador 3.

En el departamento creado se deben crear los empleados:

* + - "María del Tránsito Mojica" con identificador 1.
    - "Petronila Landazábal" con identificador 2.
    - "Epaminondas Tuta" con identificador 3.
* RallyAsociacionProyecto3: consiste en asociar a los tres empleados, en orden ascendente de identificador, en el proyecto con identificador 3.
* ListarProyecto3: consiste en listar los empleados en el proyecto con identificador 3.
* RallyAsociacionProyecto1: Asociar a los empleados con identificador 1 y 2 en el proyecto con identificador 1.
* ListarProyecto1: consiste en listar los Empleados en el proyecto con identificador 1.
* RallyAsociacionProyecto1Pos: Asociar al Empleado con identificador 1 en el proyecto con identificador 1.
* RallyDesasociacionProyecto1: Desasociar al Empleado con identificador 2 en el proyecto con identificador 1.
* ListarTodosLosEmpleados: Listar los proyectos asociados por cada uno los empleados teniendo en cuenta que si el número de proyectos asociados es cero debe informar ("SIN PROYECTOS ASOCIADOS").

El proyecto tiene el siguiente contenido:

* Carpeta ***src***:
  + Paquete ***org.example.empresa.to***: Contiene las clases requeridas para representar los elementos del negocio. Se sugiere complementarlas.
  + Paquete ***org.example.empresa.to.exception***: Contiene las clases requeridas para representar los elementos de excepción del negocio. Se sugiere complementarlas.
  + Paquete ***org.example.empresa***: Deberá contener la clase que ejecuta el programa requerido. Se sugiere complementarlas.

**Notas sobre el ambiente requerido**:

Ambiente con jdk y Eclipse.

**Trabajo requerido:**

El objetivo del ejercicio es desarrollar una aplicación que realice las operaciones especificadas en el título “Operaciones requeridas” usando las clases especificadas con las restricciones especificadas

Se sugiere realizar las siguientes actividades:

1. Importar el proyecto en Eclipse.

2. Completar las clases del paquete ***org.example.empresa.to.***

3. Completar las clases del paquete ***org.example.empresa.exception***

4. Completar las clases del paquete ***org.example.empresa***