



| Apellidos, Nombre | Username | Expediente | Hora |
|-------------------|----------|------------|------|
|                   |          |            |      |

Una empresa de software desea realizar una aplicación a través de la que gestionar todos los proyectos que lleva a cabo. Para ello, además de llevar un registro de los proyectos que se están realizando, también se lleva otro registro del personal que trabaja en la empresa. Para la implementación de esta aplicación hay que tener en cuenta los siguientes requisitos:

- Por cada Programador de la empresa es necesario conocer: su dni, nombre y antigüedad en la empresa medido en años
- Por cada Proyecto es preciso conocer: su referencia (cadena de caracteres que actúa como identificador de forma que no puede haber dos proyectos con la misma referencia) nombre, cliente (cadena de caracteres que representa el nombre del cliente) y líder (programador encargado de liderar el equipo de programadores que realizan el proyecto).
- Todo Proyecto está formado por una serie de Tareas. Una Tarea representa algo que debe realizarse mientras se realiza el proyecto. Hay distintos tipos de Tareas: Análisis, Diseño e Implementación. Toda tarea, independientemente de su tipo, posee una serie de características comunes: *código* (identificador de la tarea, se debe garantizar que en un mismo proyecto no puede haber dos tareas con el mismo identificador) *descripción* (cadena de caracteres que indica en qué consiste la tarea) *miembros* (programadores encargados de realizar la tarea teniendo en cuenta que una tarea puede tener asignados varios programadores distintos) *duración* (tiempo que dura la tarea expresado en días).

Por su parte, una tarea de tipo Análisis tiene además una propiedad metodología que describe la metodología utilizada en el análisis de la tarea. Los posibles valores de esta propiedad son SCRUM, OOP, EXTREME\_PROGRAMMING.

Una tarea de tipo Diseño tiene adicionalmente una propiedad patrones que indica el conjunto de patrones de diseño utilizados en el desarrollo de la tarea (en el diseño de una tarea pueden emplearse varios patrones de diseño). Los valores de los patrones de diseño son: SINGLETON, FACTORY, DAO, OBSERVER

Por último, toda tarea de tipo Implementación posee además la propiedad lenguaje que indica el lenguaje de programación utilizado para la implementación de la tarea. Los posibles valores de esta propiedad son JAVA, C, DOT\_NET

A continuación se indica las funcionalidades que habría que implementar.

#### **void addProgramador(String dni, String nombre, int antigüedad) throws ProgramadorRepetidoException**

Esta funcionalidad registrará un nuevo Programador en la empresa. Se disparará la excepción ProgramadorRepetidoException si ya se hubiera registrado previamente otro programador con el mismo dni.

#### **void addProyecto(String referencia, String nombre, String cliente, String dniLider) throws ProyectoRepetidoException, ProgramadorDesconocidoException**

Esta funcionalidad registra un nuevo Proyecto en la empresa. Se disparará la excepción ProyectoRepetidoException si ya se hubiera registrado previamente otro proyecto con la misma referencia. Se propagará la excepción ProgramadorDesconocidoException si no se tuviera registrado un programador con el dni que describe el líder del proyecto.

#### **void addTarea(String referencia, Tarea tarea) throws ProyectoDesconocidoException, TareaRepetidaException**

Esta funcionalidad permite agregar una nueva tarea al proyecto. Como parámetros se recibe la referencia que identifica al proyecto y el objeto Tarea a añadir. Este objeto Tarea en realidad será un objeto Analisis, Diseño o Implementación. Para la creación de estos objetos Tarea es preciso que diseñes adicionalmente métodos newAnalisisInstance(), newDiseñoInstance() y newImplementacionInstance() que devuelvan respectivamente objetos Analisis, Diseño e Implementación. A estos métodos se les tendrá que pasar como parámetro los valores correspondientes a las propiedades incluidas en los objetos.

El método addTarea() propagará la excepción ProyectoDesconocidoException si no existiera un proyecto identificado por la referencia suministrada. Además se propagará la excepción TareaRepetidaException si se intentase añadir una tarea cuyo código coincidiese con el de otra tarea añadida al proyecto con anterioridad.

#### **int getDuracionProyecto(String referencia) throws ProyectoDesconocidoException**

Esta funcionalidad calcula la duración (expresada en días) de un proyecto identificado por su referencia. Este cálculo consiste simplemente en sumar la duración de cada una de las tareas de que consta el proyecto. El método propagará la excepción ProyectoDesconocidoException si no existe un proyecto identificado por la referencia.

Set<Programador> getProgramadoresProyecto(String referencia) throws ProyectoDesconocidoException

A través de esta funcionalidad se obtiene los Programadores que trabajan en el proyecto. Se considera como programadores del proyecto al líder así como a los miembros de cada una de las tareas de que consta el proyecto. Por supuesto un mismo programador no puede aparecer varias veces. Se considera que un programador es igual a otro si tiene el mismo dni que aquél.

Esta funcionalidad puede propagar la excepción ProyectoDesconocidoException ya que es posible que como parámetro se suministre una referencia que no identifique a ninguno de los proyectos registrados.

Para comprobar el correcto funcionamiento de esta funcionalidad, será preciso que una vez invocado a este método, se muestren por pantalla los datos de cada uno de los programadores devueltos. Los programadores aparecerán ordenados por su nombre.

void showProyectos()

Esta funcionalidad mostrará en pantalla un listado con todos los proyectos registrados en la empresa ordenado por la referencia del proyecto. Por cada proyecto se mostrará una línea con los datos del proyecto incluyendo: referencia, nombre, cliente y nombre del líder del proyecto. Además por cada proyecto se incluirá un sub-listado con las tarea de que consta. En este sub-listado se mostrará una línea cuyo formato dependerá del tipo de la tarea a listar. A continuación se indican los tres formatos de líneas existentes:

- ANALISIS: <descripción> <metodología>
- DISEÑO: <descripción>  
          <patron<sub>1</sub>>  
          <patron<sub>2</sub>>  
          ...  
          <patron<sub>n</sub>>
- IMPLEMENTACIÓN: <descripción> <lenguaje>

## Calificación

- Aprobado:** Debe implementarse correctamente las funcionalidades addProgramador(),addProyecto(),addTarea(), newAnalisisInstance(),newDisenoInstance(), newImplementacionInstance() y getDuracionProyecto()
- Notable:** además de lo anterior, hay que implementar correctamente la funcionalidad getProgramadoresProyecto()
- Sobresaliente:** además de lo anterior, hay que implementar correctamente la funcionalidad showProyectos()

Se entenderá que algo funciona cuando la ejecución del programa lo demuestre