

Final Objetos 1 – Febrero 2018

Ejercicio 1

Sea un sistema TopProgrammer para el desarrollo colaborativo de pequeños proyectos de software. Cada proyecto que se carga en el sistema (con nombre de proyecto, cliente y breve descripción) está compuesto de un conjunto de tareas que es necesario completar.

El sistema tiene un registro de cada usuario habilitado para contribuir en los proyectos (con nickname y correo electrónico). Cada usuario puede desempeñar en TopProgrammer uno de los siguientes roles: Manager, Desarrollador o Tester. Los managers crean los proyectos y pueden agregar tareas a ellos, los desarrolladores generan soluciones para las tareas del proyecto y los testers generan tests para esas soluciones. Cada usuario acumula el puntaje de las tareas que resuelve.

Las tareas de un proyecto tienen un número, una descripción, una fecha límite y una cantidad de puntos que otorga cuando se resuelve. Una tarea es marcada como completada por el manager del proyecto cuando él considera que una solución resuelve la tarea en cuestión (puede ser antes o después de la fecha límite que tenía cargada la tarea). Cuando se marca como completada, se registra la fecha en que se completó y se reparte el puntaje que asignaba la tarea con el siguiente esquema: un 10% para el manager del proyecto, un 50% para el desarrollador que construyó la solución y un 40% para el Tester que hizo el test correspondiente. Cuando todas las tareas de un proyecto están completadas, el proyecto se marca como completado y no se le pueden agregar nuevas tareas (este control debe efectuarse cada vez que se quiera agregar una tarea). Mientras un proyecto no está completado, se le pueden agregar nuevas tareas.

Tareas:

- 1- Modele con un diagrama UML una solución al problema que se describe. Incluya todas las clases necesarias, las relaciones entre ellas con su cardinalidad y el protocolo de cada una.
- 2- Implemente todo el código SmallTalk necesario. Incluya los constructores de todas las clases y los siguientes mensajes (más todos los necesarios para que éstos funcionen):

TopProgrammer>>iniciarProyecto: unProyecto conManager: unManager,
"habilita el proyecto con fecha del día asociando unManager al mismo"

TopProgrammer>>proyectoMásAvanzado
"devuelve el proyecto que tiene el mayor porcentaje de tareas completadas"

TopProgrammer>>proyectosCompletadosDesde: una Fecha hasta: otraFecha
"devuelve la colección de proyectos completados entre las fechas indicadas ordenados por cantidad de tareas en orden descendente"

UsuarioManager>>cerrarTarea: unaTarea
"marca unaTarea como completada y distribuye el puntaje correspondiente"

Proyecto>>agregarTarea: unaTarea

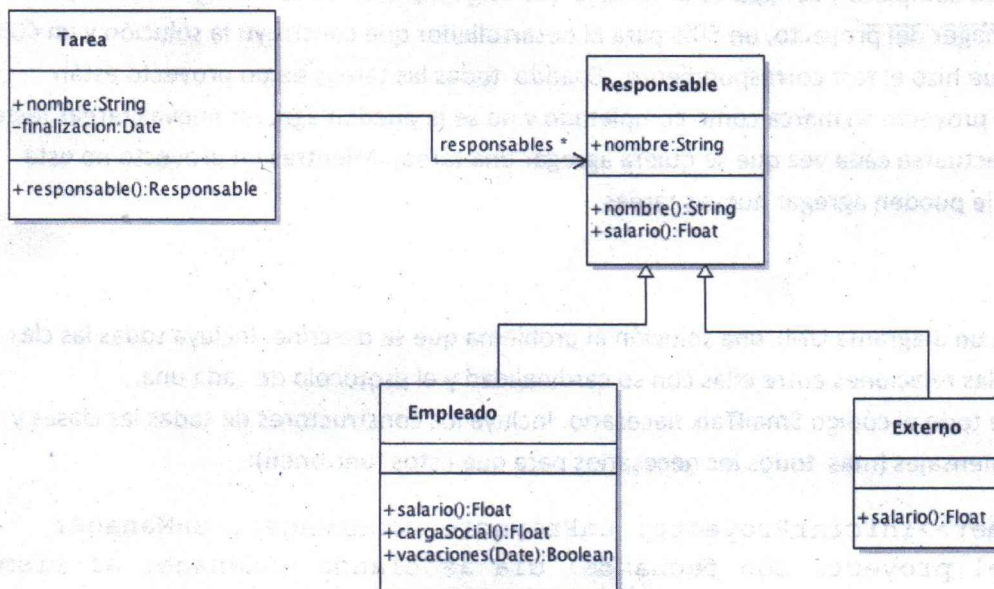
"Si el proyecto está abierto agrega unaTarea su lista. Si está cerrado devuelve el mensaje 'Proyecto cerrado, no se pueden agregar tareas' "

Proyecto>>cerrar

"verifica que todas las tareas del proyecto están completadas y marca el proyecto como cerrado"

Ejercicio 2

Dado el siguiente diagrama de clases, responda las preguntas que figuran abajo:



- ¿Bajo qué conjunto-de mensajes son polimórficas las clases Responsable, Empleado y Externo?
- Si tiene que implementar la clase Tarea en Pharo. ¿qué variable o variables de instancia debería tener?
- ¿Qué tipo de **relación UML** existen entre las clases Tarea y Responsable?
- ¿Qué tipo de **relación UML** existe entre las clases Responsable y Externo?