**😃 Leé por lo menos dos veces el enunciado antes de resolver.**

# Enunciado

Una consultora IT nos pide diseñar un sistema que permita realizar la gestión de tareas a realizar.

El gestor de tareas poseerá una colección de Tareas que luego podrán ser asignadas a los usuarios que después las cumplirán. Las tareas serán asignadas en estricto orden de llegada.

Nos indican que hay tres tipos de tareas. Todas las tareas tienen el nombre del proyecto al que pertenecen, el número de versión afectada, el número de versión donde se estima quedará la corrección, un título de tarea y el usuario que la reportó.

Todas las tareas deben poder calcular su prioridad y además son Estimables, por lo que deben poder indicar su tiempo de resolución estimado en horas (entero).

Las tareas pueden ser de:

* Desarrollo, que posee una descripción funcional del desarrollo y un tiempo estimado en horas (entero).
* Error interno, que indica el nombre del ambiente donde se produjo, una colección de casos de prueba (cada caso de prueba es un String) y el nombre del tester que detectó el error (String). El tiempo estimado de resolución siempre será de 40 horas.
* Error en producción, que además de describir el ambiente y poseer una colección de casos de prueba (como los errores internos) poseen el nombre del ambiente del cliente donde se detectó y un flag que indica si es un error del tipo *stopper* o no. El tiempo de resolución estimado será siempre de 8 horas si el error es *stopper* y de 16 horas si no lo es.

La prioridad de las tareas puede ser alta, media o baja. Se calculará de la siguiente forma:

* Para los errores de producción si el tiempo de resolución es de 8 horas tendrá prioridad ALTA, si es de 16 horas tendrá prioridad MEDIA.
* Para el resto de las tareas que no sean errores de producción (desarrollos o errores internos) se calculará:
  + ALTA si tiene tiempo de resolución de hasta 16 horas;
  + MEDIA si tiene tiempo de resolución menor o igual a 24 horas y mayor a 16 horas;
  + BAJA para el resto de los casos.

El gestor de tareas tiene además dos funcionalidades claves:

* Realiza la asignación de tareas a los usuarios. Esta asignación se hará en forma rotativa, tarea por tarea, una a cada usuario hasta haber sido asignadas todas las tareas, pudiendo ser que algunos usuarios tengan asignadas más tareas que otros.
* Genera y retorna un reporte de tiempos promedio de tarea por cada prioridad.

De los usuarios sabemos su nombre y apellido y el nombre del proyecto base en el cual trabaja. Los usuarios deberán resolver las tareas que les sean asignadas basados en la prioridad de las mismas, primero las de prioridad ALTA, luego las de prioridad MEDIA y por último las de prioridad BAJA, sin importar el orden en que le fueron asignadas.

Basado en el enunciado descripto, realizá:

1. El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
2. El método **reporteTiempoPromedioTareasPorPrioridad()**, de la clase que corresponda, que debe **devolver** (no mostrar por consola) el tiempo promedio de las tareas por cada prioridad.
3. El método **asignarTareas()** de la clase que corresponda, que debe asignar a los usuarios todas las tareas disponibles (la estructura inicial debe quedar vacía tras la asignación). La asignación debe hacerse tal como se indicó anteriormente.

# Criterios

Para considerar aprobado el examen, el mismo debe demostrar la correcta aplicación de los siguientes conceptos de la programación orientada a objetos:

* Correcta definición de clases y asignación adecuada de sus responsabilidades.
* Encapsulamiento, ocultamiento de información y uso de getters y setters sólo cuando corresponda.
* Modularización reutilizable y mantenible con uso de métodos con correcta parametrización.
* Correcta aplicación de miembros de instancia y de clase.
* Correcta aplicación de herencia y polimorfismo.
* Correcta aplicación conceptual de las relaciones entre clases.
* Correcta aplicación de TADs vistas en clase

