

ISW1-SW-Project

Nuestra empresa de desarrollo de Software está empezando a crecer mucho y se está complicando controlar los proyectos en desarrollo, las tareas de cada proyecto, los desarrolladores, equipos que trabajan en ellos, etc. Es por eso que hemos decidido realizar un sistema para trackear los proyectos y así poder obtener entre otras cosas, un worksheet (planilla de trabajo) de los desarrolladores para saber si hay sobretrabajo y determinar cuánto cuesta hacerlo.

Un **proyecto**, además de conocer un nombre, está formado de subproyectos y/o tareas concretas. El comienzo de un proyecto se debe calcular teniendo en cuenta que debe empezar en la fecha más temprana de inicio de sus subproyectos/tareas. La fecha de finalización también se debe calcular y será la última fecha de finalización de sus subproyectos/tareas.

Las **tareas concretas** también conocen un nombre, un responsable, una fecha de inicio deseada, el esfuerzo en horas que lleva realizarla y de qué tareas depende (puede no depender de ninguna). Una tarea concreta solo puede empezar cuando todas las tareas de las que depende terminaron y no antes de la fecha de inicio deseada. Una tarea concreta va terminar dependiendo de la fecha de inicio, el esfuerzo que demanda y la dedicación de quienes la realizan. Por ejemplo, SS A (Sub System A) empieza en July/1/2024, tiene un esfuerzo de 8*hour y su responsable, Dan Ingalls, tiene una dedicación de 8*hour/day, entonces la tarea terminará en July/1/2024. Si la hiciera Alan Kay que tiene una dedicación de 6*hour/day, entonces terminaría en July/2/2024.

Ejemplo de Proyecto y sus subproyectos y tareas:

Nombre	Inicio Des.	Esfuerzo	Dependientes	Subtareas	Responsable
Sistema ERP			-	{Modelo,UI }	-
Modelo			-	{SS A, SSB, SS C}	-
SS A	July/1/2024	8*hour	{}	-	Dan Ingalls
SS B	July/1/2024	16*hour	{}	-	Parc MobTeam
SS C	July/2/2024	16*hour	{SS A, SS B}	-	Alan Kay
UI	July/2/2024	6*hour	{Modelo}	-	Adele Goldberg

(En **negrita** los proyectos, en *itálica* las tareas concretas)

El **responsable** de realizar una tarea concreta puede ser un **desarrollador** o un **equipo**. El desarrollador además de conocer un nombre tiene una dedicación horaria diaria (ej. 6*hour/day) y un costo por hora (ej: 60*dollar/hour).

Ejemplo de Desarrolladores:

Desarrollador	Dedicación	Costo
Dan Ingalls	8*hour/day	60*dollar/hour
Alan Kay	6*hour/day	80*dollar/hour
Adele Goldberg	10*hour/day	65*dollar/hour

Un **equipo** de trabajo está formado por desarrolladores y subequipos y trabaja estilo mob programming. El tiempo para realizar una tarea depende del desarrollador más lento, por ej. si una tarea lleva 16*hour y el más lento tiene una dedicación de 6*hour/day, la tarea será realizada en 3*day.

Los desarrolladores no trabajan horas extra y si terminan una tarea, no empiezan una nueva hasta el día siguiente.

No se tiene en cuenta feriados ni fines de semana para calcular los tiempos de desarrollo.

Ejemplo de Equipo:

Parc Team
Dan Ingalls
Alan Kay

Una vez conformado el proyecto se pide poder obtener un **worksheet** para:

- Saber si el proyecto tiene **sobreasignaciones**, o sea desarrolladores que deberían trabajar más que su dedicación algún día
- Saber los **días de sobreasignación** de los desarrolladores. Si un desarrollador no está sobreasignado, será una colección vacía.
- El **costo total** del proyecto (haya o no sobreasignación). El costo total se calcula en base a la cantidad de días a trabajar de cada desarrollador y su costo asociado. Por ejemplo, si un desarrollador cobra 1000*peso/hour, tiene una dedicación de 6*hour/day y trabajó 2*day, el costo será:
 $(1000*\text{peso/hour}) * (6*\text{hour/day}) * (2*\text{day}) = 12000*\text{peso}$

Ejemplo de worksheet:

Dan Ingalls	July 1, 2024->2/ July 2, 2024->1/ July 3, 2024->1
Alan Kay	July 1, 2024->1/ July 2, 2024->1/ July 3, 2024->1/ July 4, 2024->1/ July 5, 2024->1/ July 6, 2024->1
Adele Goldberg	July 7, 2024->1

Costo total: 5450 * dollars .

En esta versión se debe asumir que los objetos siempre serán creados válidos, o sea que los nombres son correctos, que no hay tareas repetidas en los proyectos, etc.

Implementar por medio de TDD y siguiendo todas las heurísticas de diseño vistas durante el cuatrimestre, lo necesario para resolver este problema.

RECOMENDACIÓN: Empezar modelando una tarea concreta, en particular cuándo debe terminar para un caso simple (un desarrollador que tiene misma dedicación que el esfuerzo de la tarea) y continuar a partir de ahí :-)

Entrega:

- Entregar por mail el fileout de la categoría de clase **ISW1-2024-1C-2doParcial** que debe incluir toda la solución (modelo y tests). El archivo de fileout se debe llamar: **ISW1-2024-1C-2doParcial.st**
- Entregar también por mail el archivo que se llama **CuisUniversity-nnnn.user.changes**

3. Probar que el archivo generado en 1) se cargue correctamente en una imagen “limpia” (o sea, sin la solución que crearon. Usen otra instalación de CuisUniversity/imagen si es necesario) y que todo funcione correctamente. Esto es fundamental para que no haya problemas de que falten clases/métodos/objetos en la entrega.
4. Realizar la entrega enviando mail a la lista de Docentes: docentes@isw2.com.ar con el **Subject: LU nnn-aa - Solución 2do Parcial 1c2024. En caso de rebotar el envío, intentar comprimiendo los adjuntos y mandarlo nuevamente copiando esta vez a hernan.wilkinson@gmail.com**
5. **RECOMENDACIÓN IMPORTANTE: Salvar la imagen de manera frecuente o con el autosave**
6. Se asume que a esta altura de la cursada saben trabajar con la imagen, recuperarla, recuperar código fuente, revertir cambios y demás incidencias que pudieran ocurrir durante el exámen.

Revisen bien los puntos de arriba. Cualquier error en los nombres o formato podrían ser penalizados en la nota.

IMPORTANTE: No retirarse sin tener el ok de los docentes de haber recibido el mail con la resolución.

CERRAR EL TRABAJAJO A LAS 21:45.

LAS ENTREGAS RECIBIDAS DESPUÉS DE LAS 22:00 HRS NO SERÁN TENIDAS EN CUENTA