

# ISW1-SW-Project 2.0

El sistema de tracking de proyectos fue todo un éxito y ahora hay que mejorarlo. En base al uso que tuvo, se detectaron errores comunes que realizan los usuarios y nueva funcionalidad que se desea agregar.

Se decidió realizar los cambios en tres iteraciones. La *primera* iteración consta de arreglar el código existente teniendo en cuenta que:

- No puede haber nombres vacíos.
- No puede usarse como esfuerzo, dedicación o costo una medida no estrictamente positiva. Por suerte **no** nos piden verificar que las unidades sean correctas.
- Los equipos deben tener como mínimo dos desarrolladores.
- Los equipos no pueden tener integrantes repetidos.
- Las tareas concretas no pueden tener dependientes directos repetidos. Si pueden haber dependientes indirectos repetidos.
- Todos los proyectos deben tener subtareas/subproyectos si o si.
- Las tareas concretas de los proyectos no pueden estar repetidas.

La *segunda* iteración consta de implementar las siguiente funcionalidad.

- Es necesario agregar dos tipos de equipos nuevos:
  - El **equipo de trabajo paralelo**, que divide el esfuerzo en partes iguales entre todos los desarrolladores.
  - El **equipo rápido**, que utiliza el desarrollador de más dedicación para determinar los días que lleva realizar una tarea, al contrario del equipo mob que usa el que menos dedicación tiene.
  - Es importante asegurar que si un equipo está compuesto por otro, debe ser del mismo tipo porque sino es imposible definir cuánto lleva realizar un esfuerzo. O sea, un equipo de mob programming que esté compuesto por otros equipos, solo podrá hacerlo si los otros equipos son también de mob programming.
- Se desea poder hacer una representación visual de un proyecto. Por ejemplo, para el siguiente proyecto:

Nombre	Inicio Des.	Esfuerzo	Dependientes	Subtareas	Responsable
<b>Sistema ERP</b>			-	{Modelo,UI }	-
<b>Modelo</b>			-	{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

Se debería obtener un colección de strings con la fecha de inicio, de fin, la duración en días, la cantidad de desarrolladores y el nombre de la tarea indentando de acuerdo a la profundidad del árbol de tareas. El formato debe ser exactamente el especificado en este ejemplo.

```
2024/07/01 - 2024/07/07 - 7*days - 3 devs |-> Sistema ERP
2024/07/01 - 2024/07/06 - 6*days - 2 devs |-> Modelo
2024/07/01 - 2024/07/01 - 1*day - 1 dev |-> SS A
2024/07/01 - 2024/07/03 - 3*days - 2 devs |-> SS B
2024/07/04 - 2024/07/06 - 3*days - 1 dev |-> SS C
2024/07/07 - 2024/07/07 - 1*day - 1 dev |-> UI
```

La *tercera* iteración consta de desarrollar lo siguiente:

- Introducir la posibilidad de indicarle a un proyecto que es el proyecto principal, que es el que contiene a todos los subproyectos y subtareas. Un proyecto principal debe cumplir debe:
  - Asegurar que todos los dependientes de las tareas concretas forman parte directa o indirectamente del proyecto. Por ejemplo, no puede haber una tarea que dependa de otra que no está incluida en el proyecto principal.
  - Un proyecto principal no puede estar compuesto de otro proyecto principal

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

Los que deben recuperar solo el 1er parcial, deben implementar la primera iteración.

Los que deben recuperar solo el 2do parcial, deben implementar la primera y segunda iteración.

Los que deben recuperar el 1er y 2do parcial, deben implementar la primera, segunda y tercera iteración.

#### **PROTOCOLO A TENER EN CUENTA:**

- fromDate distanceTo: toDate: Devuelve la diferencia en días entre fromDate y toDate.
- aMeasure strictlyPositive: Devuelve true si la medida es mayor a 0 con su unidad. Ej: (1\*day) strictlyPositive → true
- aString padded: leftOrRight to: length with: char: Agrega a #left o #right, char a aString para que tenga longitud lenght. Ejemplo:  
'ABC' padded: #right to: 6 with: \$ . → 'ABC ' (notar el espacio entre \$ y . )

## Entrega:

1. Entregar el fileout de la categoría de clase **ISW1-2024-1C-Recuperatorio** que debe incluir toda la solución (modelo y tests). El archivo de fileout se debe llamar: **ISW1-2024-1C-Recuperatorio.st**
2. Entregar también 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. Deben entregar usando el siguiente form: <https://tinyurl.com/isw1-1c2024-recu>
5. De forma alternativa si no pudiste entregar con el form, realizar la entrega enviando mail a: **entregas@isw2.com.ar** con el **Subject: LU nnn-aa - Solución Recuperatorio 1c2024.**  
En caso de rebotar el envío, reintentar comprimiendo los adjuntos.
6. **RECOMENDACIÓN IMPORTANTE: Salvar la imagen de manera frecuente o con el autosave**
7. 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 la resolución por algún medio.

**CERRAR EL TRABAJO A LAS 21:45.**

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