TPE - INFORME

Integrantes:

- Baader, Juan; Legajo: 58647
- Battilana, Joaquín; Legajo: 57683
- Lombardía, Maximiliano; Legajo: 56276

Nuestro programa se compone de tres clases fundamentales: **Task**, **TaskHolder** y **Input.**

Sobre el manejo de tareas:
Al comenzar tuvimos muchas ideas de cómo implementar la list, empezamos por una sorted list pero luego al observar las diferentes maneras en las que se podía llamar al conjunto de objetos concluimos que sería muy complicado implementarlo de esta manera. De ese punto en adelante dudamos entre utilizar un Sortedset o un array, la ventaja del array siendo que seria mas simple implementar el complete pero se sacrifica eficiencia tener que aplicar un sort cada vez que lo quisiéramos listar, en cambio el sortedset seria mucha mas eficiente en esos aspectos pero requería de que al aplicar un complete a una de las tareas esta tenga que ser eliminada del set y luego vuelta a agregar. Cuando llegó la hora de tener que decidir escogimos utilizar el sortedset por ser el más eficiente en términos generales, al igual que traer otros beneficios como la velocidad de un hash.

Sobre Task:
Task es la clase donde está definido todo lo relacionado a los datos de las tarea(en forma de atributos) y comparación de tareas. Pertenece, junto a la clase TaskHolder, al "back-end".
Existen tres métodos que son los más importantes, estos son el **⇔**, el **==** y el **eql?**, son los que se encargan de comparar el estado de completado de una tarea en el caso del primero y en los dos últimos para ver si dos tareas referencian a lo mismo o si representan lo mismo, respectivamente.
El resto de los métodos sirven para dar formato de impresión a la tarea (**to_s** y **to_s_without_group**), indicar que una tarea se ha completado (**completed**), generar una string a partir de una fecha para el caso de "tomorrow" y "today"(**date_to_str**) y el método para generar el hash(**hash**).

Sobre TaskHolder:
Creamos esta clase para poder manejar los datos del Sortedset fuera de la clase Input, ya que esto permite una lectura más clara del código. Como esta clase la consideramos parte del back end, la mayoría de sus funciones crean un array sobre el cual el input puede operar, como en el caso de los "list", o simplemente recibe datos del input y los maneja para agregarlos al Sortedset con los atributos requeridos. Todas sus funciones podrían haber sido llevadas a cabo por el la clase input pero tener esta clase ayudó mucho a simplificar el código y poder separar los problemas que fuimos encontrando.
Además de guardar el Sortedset, está clase contiene el id, el cual va iterando a medida que se agregan clases o se resume una vez que se carga un archivo, set_date y set_group que se crean al usar set date_task y set group y el sortedset que contiene todas las tareas de tipo task.

Sobre Input:

La clase Input es, en síntesis, la encargada de todo lo relacionado al “front-end”.
El método más importante de la clase es input_checker, que es el que decide qué método de TaskHolder se debe usar dado un comando de consola. Todo esto es posible ya que Input tiene un atributo que es una instancia de TaskHolder.
El resto de métodos son auxiliares de input_check, donde lo que hacen es generar los datos necesarios para pasarle al holder(**generate_desc** y **to_date**), mensajes que muestra la consola(**show_error** y **show_add_message**) y, por supuesto, los métodos esenciales(**initialize** y **to_s**).
En el caso del Input, inicialmente se pensó implementarlo como un módulo, debido a que inicialmente solo tenía un método, que era input_checker. Sin embargo, a medida que fuimos desarrollando la aplicación, nos dimos cuenta de que, para facilitar la comunicación entre el input y el holder, podíamos agregar una instancia de holder como atributo y que llame a los métodos de la clase mediante esa instancia. Otro factor que se tuvo en cuenta es que existían un par de problemáticas, como pasar la fecha en formato Date, que implementando más métodos de forma auxiliar al input checker se solucionarían. Cabe destacar que el factor crucial en este caso fue el atributo, ya que agregar más métodos no cambia en nada respecto del cambio de módulo a clase.