

Práctica 3 - ADSOF

Víctor de Juan Sanz
Guillermo Julián Moreno

Marzo 2013

El proyecto se ha creado usando una clase Proyecto, que se encarga de todo el procesamiento del fichero y del guardado de los datos en memoria. La función `main` se limita a leer el nombre de archivo de los argumentos y a llamar a las funciones necesarias de Proyecto.

Leemos el archivo línea a línea, procesando pesos, tareas y conexiones. Si el archivo no contiene los datos en ese orden el comportamiento del programa está indefinido, y probablemente falle con una excepción. Sin embargo, dado que en el enunciado se nos especifica que los datos de entrada serán siempre correctos, no hemos creado ningún tipo de control de errores.

Una vez cargado todo el archivo en memoria, se puede llamar a las funciones para imprimir en pantalla. No es necesario llamar a ninguna función para procesar los tiempos.

La razón es que las tareas calculan sus valores usando una estrategia *lazy*. Los valores se calculan sólo una vez y sólo cuando se pide el valor. La estrategia es similar en todos los campos que requieren un cálculo, así que veremos sólo el ejemplo del comienzo optimista de una tarea.

Cuando se llama a `Tarea.getComienzoOptimista`, verificamos si no se ha calculado antes el valor. Dado que las duraciones, comienzos y demás parámetros son siempre positivos, usamos un valor de `-1` como marca de "no calculado". El valor se calcula usando las tareas antecedentes tal y como se nos explica en el enunciado, y se almacena en el campo privado correspondiente.

De esta forma, no es necesario usar una función externa a Tarea que calcule los valores necesarios de las tareas, mantenemos la simplicidad de las funciones de cálculo y obtenemos un rendimiento muy bueno al calcular sólo una vez cada valor de cada tarea.

Hay dos tareas específicas, la inicial y la final, que están representadas como subclases de Tarea. En el caso de `TareaInicio` nos limitamos a llamar al constructor de la clase base con todos los parámetros 0 para que nos devuelva los valores necesarios, y el nombre "Inicio". Además, sobrescribimos el método `getDuracionEstimada` para que devuelva 0 (el cálculo normal de Tarea devuelve `NaN`).

La tarea final, `TareaFinal`, sigue la misma estrategia salvo que además sobrescribe `getFinPesimista` y `getComienzoPesimista` para que valga lo mismo que el comienzo optimista de la tarea final.

Como último detalle, para facilitar la lectura de los datos, nuestra función mira la longitud del nombre de cada tarea para añadir un tabulador adicional o no, y que de esta forma salgan todos los campos alineados.

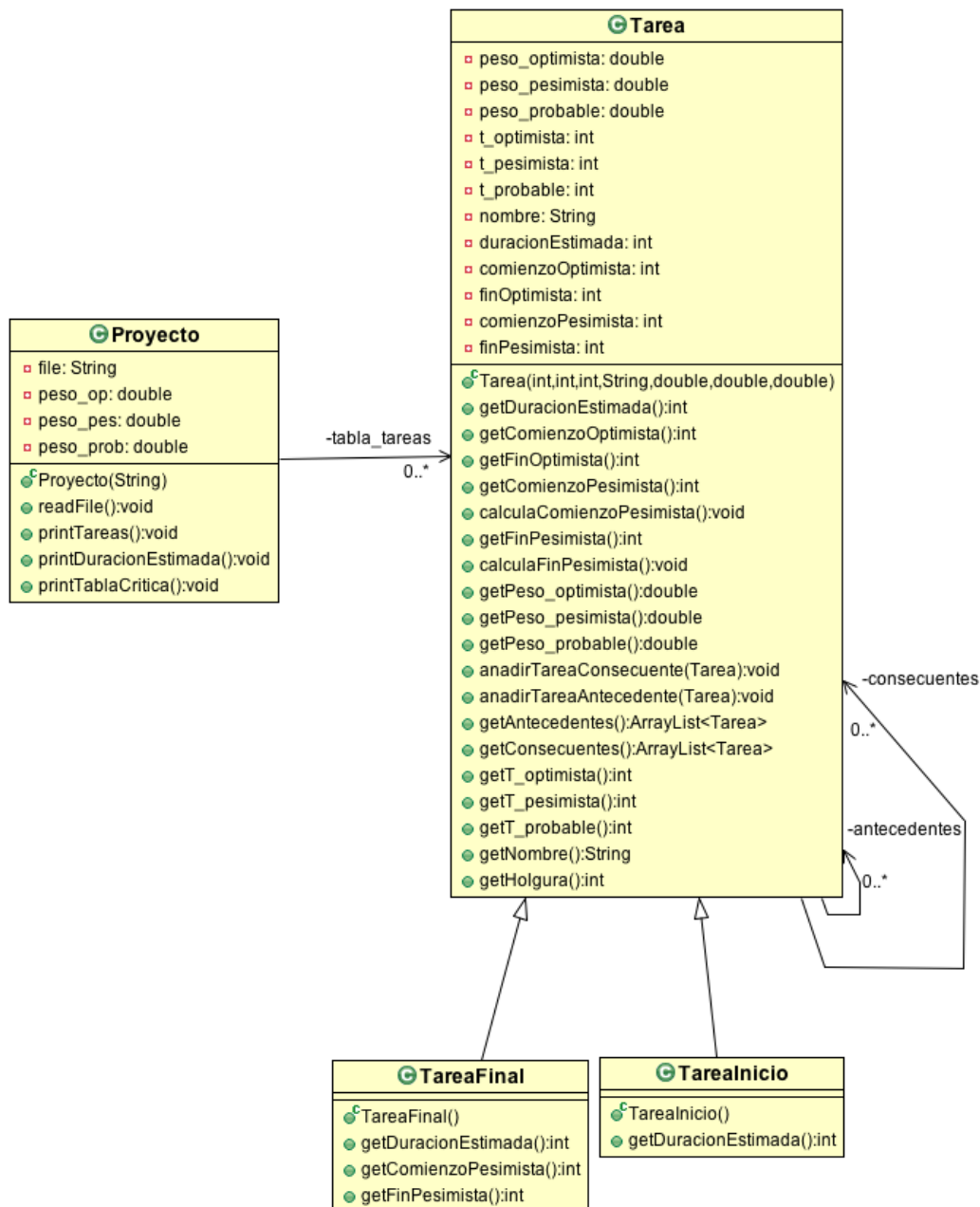


Figura 1: Diagrama de clases