

Algoritmos y Programación II (75.411)	Trabajo Práctico Individual N° 2
Cátedra Lic. Gustavo Carolo	Asana.com Project Reporter
1° Cuatrimestre 2014	Fecha de entrega: 7 y 6 de mayo de 2014

Trabajo práctico grupal N° 1

Reporte proyectos con Asana API

Introducción

En el trabajo práctico individual N° 2 logramos crear distintos TDA's obteniendo la información desde archivos con formato Json, luego mostramos la información obtenida dentro de esos TDA's, ahora bien, los datos que se obtuvieron fueron datos "crudos", sin ningún análisis acerca de los mismos, que pasa si por ejemplo, queremos saber que tan atrasada está una tarea en curso, o que tanto se atrasó una tarea finalizada, o bien, cuantos "bugs" fueron reportados y cuantos fueron solucionados. Para lograr un análisis mas profundo del estado del proyecto, necesitamos contar con toda la información centralizada para poder realizar dicho análisis.

Objetivo

Dado un identificador de proyecto de Asana, queremos obtener toda su información con el fin de generar un reporte cuantificado que nos provea de un panorama general del estado del proyecto.

Definición de TDA's

TDA_ProjectDetail

El **TDA_ProjectDetail** del TP2 Individual, deberá ser modificado, de manera tal que la cantidad máxima de tareas sea indeterminada (eliminar el MAX = 50), mediante el uso del **TDA_ListaSimple** que provee la cátedra. De la misma manera, el TDA_ProjectReport, deberá contener una cantidad indefinida de TDA_Task, esto también deberá solucionarse utilizando el TDA_ListaSimple.

El mismo TDA_ProjectDetail, deberá también agregar distintas primitivas que calculen métricas respecto a una tarea, por ejemplo:

```
int obtenerDiasRetraso(TDA_Task task, int* diasRetraso);
```

Se calcula como la diferencia en dias de la fecha de finalización y la fecha de estimación, en caso de que la fecha de finalización sea nula, se estima usando la fecha del día actual.

```
int obtenerSprintDelInicio(TDA_Task task, int* numeroSprint);
```

Algoritmos y Programación II 173.411	Trabajo Práctico Individual N° 2
Catedra Lic. Gustavo Carolo	Asana.com Project Reporter
1° Cuatrimestre 2014	Fecha de entrega: 7 y 6 de mayo de 2014

int obtenerSprintDeFinalizacion(TDA_Task task, int numeroSprint);*

Las tareas tienen un conjunto de etiquetas, las etiquetas referentes a sprints, son de la forma "sprint1", "sprint2", "sprint15". Una etiqueta puede tener desde una a varias etiquetas de sprint, la de número inferior, establece el sprint de inicio, y la de mayor número es la de finalización, para el ejemplo actual, la primitiva debe devolver "1" para inicio, y "15" para finalización.

*int obtenerTipo(TDA_task task, char** tipo)*

El tipo de tarea, se obtiene leyendo el conjunto de etiquetas (tags) de la tarea, los tags que definen el tipo de tarea son "bug", "milestone", "feature", de estos tres, dado una tarea, solo puede existir uno, o ninguno, en caso de que no exista ninguno, el tipo de tarea es "regular".

TDA_ProjectReport

Para el trabajo práctico actual, vamos a crear un nuevo tipo de dato abstracto que denominaremos **TDA_ProjectReport**, éste será el encargado de la obtención de la información del proyecto, procesado y exportación de informes. Todo esto lo llevaría a cabo usando distintos TDA's para cada tarea, entre ellos, **los TDA's que se han desarrollado para el TP2 Individual**.

El TDA_ProjectReport se debe crear recibiendo una clave de autenticación y un identificador de proyecto (ambos datos serán proveídos), con estos datos, ya puede hacer uso del **TDA_AsanaClient** (que provee la cátedra, ver apéndice A), éste TDA posee tres primitivas con las cuales se podrán obtener los archivos que procesamos en el trabajo práctico anterior, archivos de proyecto (tema 1), archivos de detalle de proyecto (tema 2), archivos de tareas (tema 3), cabe destacar que los archivos de tareas tienen cada uno su propio identificador, estos identificadores se obtienen luego de haber procesado el archivo de detalle de proyecto, que contiene todos los identificadores de tareas.

Por el lado del TDA_ProjectReport, deberá proveer las primitivas necesarias para la obtención de los siguientes datos:

- Cantidad de tareas finalizadas
- Cantidad de tareas finalizadas con atraso (aquellas tareas con fecha de finalización y días de atras mayor estricto a cero)
- Cantidad total de tareas (misma definición que tema 2 de TP2 individual)
- Cantidad total de sprints (misma definición que tema 2 de TP2 individual)
- Promedio de atraso (sumatoria de días de atraso sobre cantidad de tareas finalizadas con atraso)
- Cantidad de bugs reportados (es la cantidad de tareas de tipo "bug" con fecha de estimación nula)
- Cantidad de hitos alcanzados (es la cantidad de tareas de tipo "milestone" con fecha de finalización no nula)

Algoritmos y Programación II (75.411)	Trabajo Practico Individual N° 2
Catedra Lic. Gustavo Carolo	Asana.com Project Reporter
1° Cuatrimestre 2014	Fecha de entrega: 7 y 6 de mayo de 2014

- Cantidad de hitos pendientes (es la cantidad de tareas de tipo “milestone” con fecha de finalización nula)

TDA_CsvFile

El TDA_CsvFile es un tipo de dato que será utilizado por el TDA_ProjectReport para llevar a cabo la tarea de exportación del listado de tareas con los distintos datos. Dicho TDA debe proveer las siguientes primitivas.

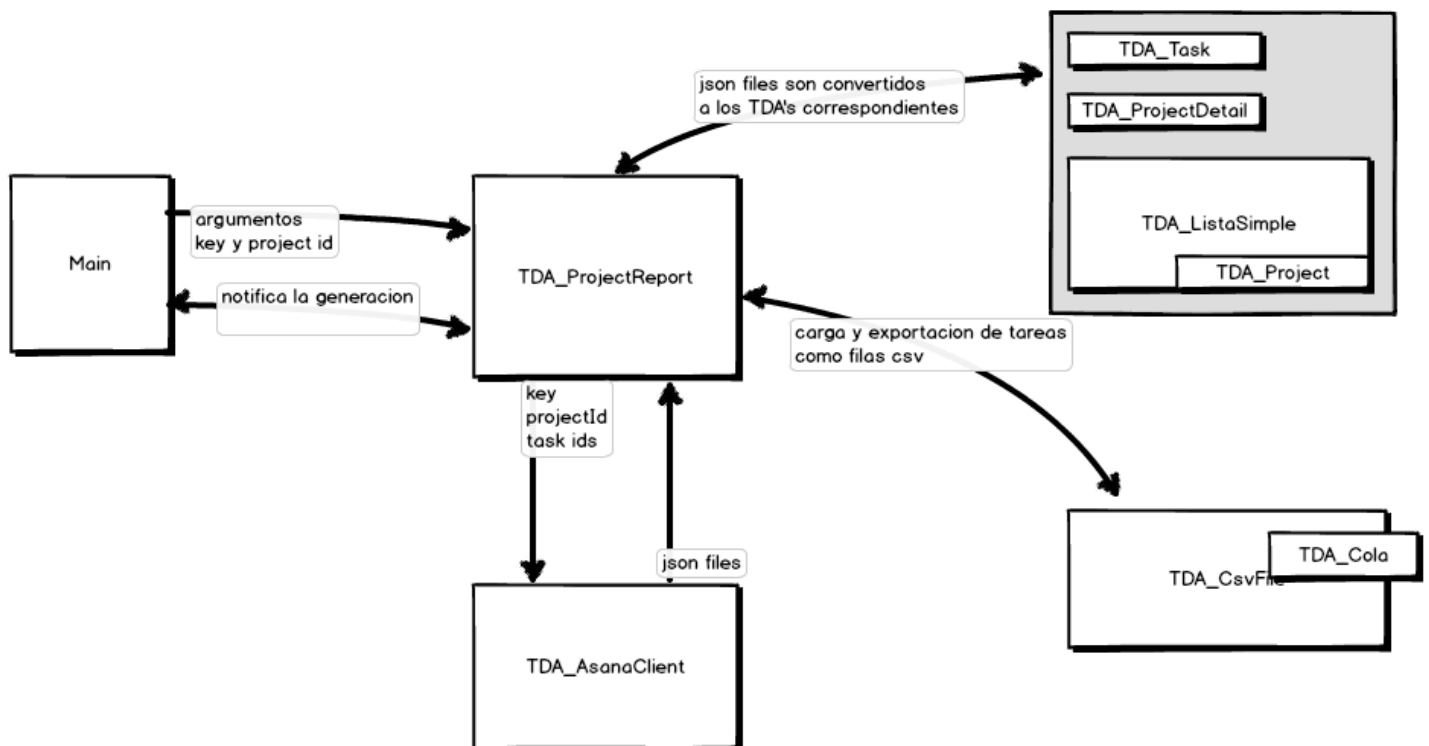
int agregarFila(TDA_CsvFile csvFile, char* fila);*

Recibe el TDA y agrega la fila recibida en su estructura.

int exportarDatos(TDA_CsvFile csvFile, char* pathArchivo);*

Recorre todas las filas recibidas y las exporta con formato CSV y política FIFO al archivo destino solicitado en el argumento “pathArchivo”.

Todos los TDA's mencionados, deberán proveer primitivas de creación y destrucción en caso de que no hayan sido especificadas.



Archivos de salida esperados

Algoritmos y Programación II (75.411)	Trabajo Practico Individual N° 2
Catedra Lic. Gustavo Carolo	Asana.com Project Reporter
1° Cuatrimestre 2014	Fecha de entrega: 7 y 8 de mayo de 2014

La ejecución del TP, con los argumentos adecuados, debe dar como salida, dos archivos. El primero, será el que describe el proyecto y sus métricas cuantificables y el segundo el archivo csv con toda la información de las tareas.

Los archivos “project.txt” y “tasks.csv” estan disponibles para descargarse en este mismo directorio de google drive.

Ejecución

tp1g key projectid

Apéndice A

TDA_AsanaClient

El TDA que se utiliza para traer desde la API de asana los archivos correspondientes a un proyecto, será proporcionada por la cátedra, una vez que se descarguen los archivos correspondientes, es recomendable que corran el “main.c” que se provee en el archivo comprimido, para que puedan probar la funcionalidad independientemente del proyecto. Dicho TDA tiene una dependencia que debe ser configurada, esto es el programa [cUrl](#), el cual deberán bajar e instalar, ya sea para windows o linux.

Para los usuarios de linux será tan simple como ejecutar “sudo apt-get install curl”.

Los usuarios de windows, deberán descargar el instalador correspondiente desde [aquí](#). Una vez elegido el sistema operativo e instalado el programa, se debe agregar la ubicación de instalación a la variable “PATH” del sistema. Windows para ello, provee una página de ayuda que pueden visitar en el siguiente [enlace](#) donde se muestra un ejemplo, o la explicación general se puede encontrar [aquí](#).

Para probar que “curl” se encuentra en la variable PATH configurado exitosamente, abrir la linea de comandos (cmd o terminal) y tipear el comando “curl --help” si el mensaje es una lista de comandos, entonces la instalación fue satisfactoria, si el mensaje dice que no se encontró el comando requerido, se deberá verificar los pasos anteriores.