



Universidad de Buenos Aires

Facultad de Ingeniería

75.61 – Taller de Programacion III

Trabajo Práctico

Load Test Console

1º Cuat. - 2017

Docentes :

- Andres Veiga
- Pablo Roca

Alumno:

- Pablo Méndez 88908 pablo.guillermo.mendez@gmail.com

Índice

1. Introducción.....	3
2. Diagramas.....	4
2.1. Diagrama de actividades.....	4
2.2. Diagrama de robustez.....	5
3. Código fuente.....	6

1. Introducción

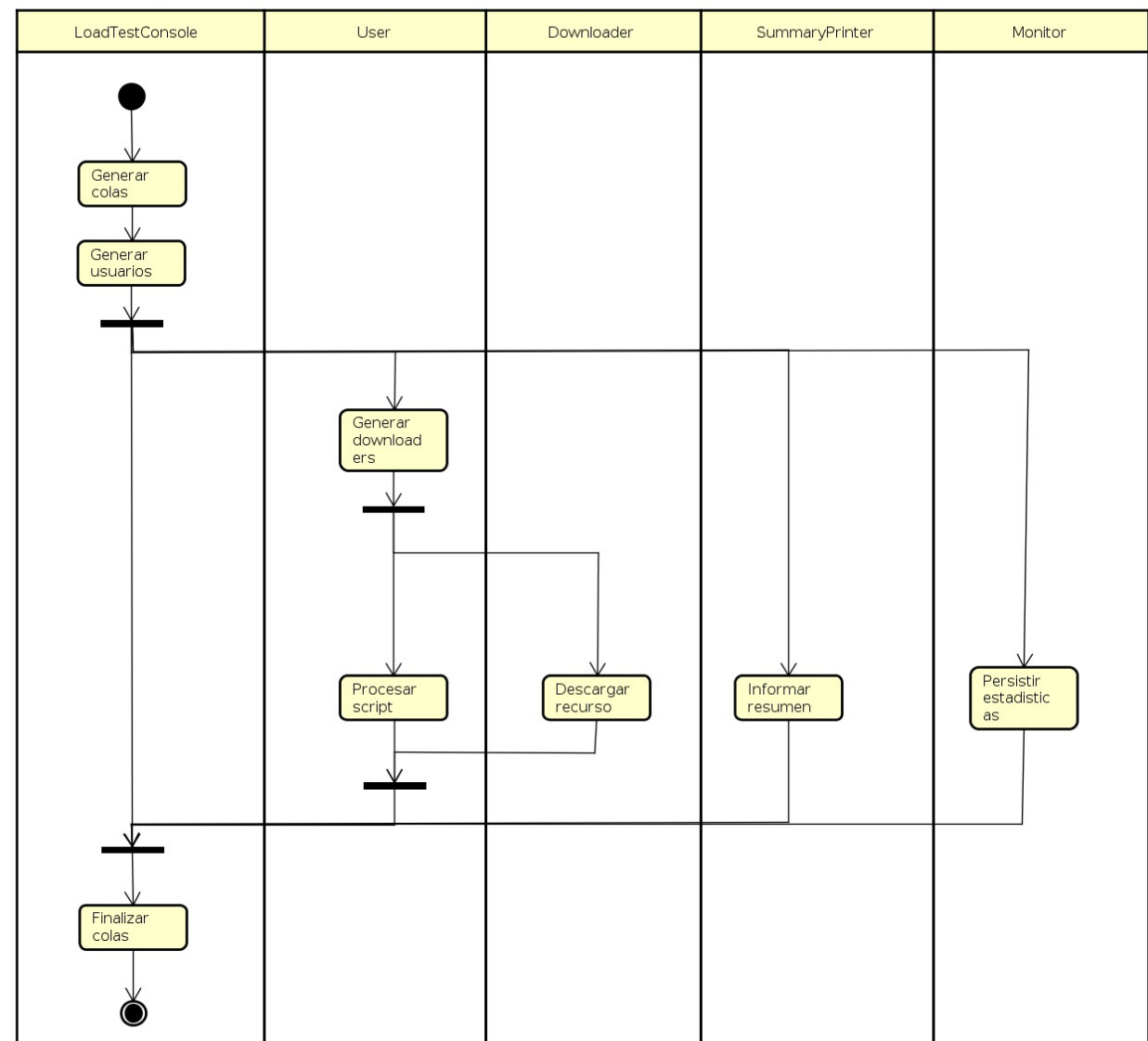
El presente trabajo consta en desarrollar una aplicación para simular tests de carga mediante la ejecución concurrende de requests http (GET, PUT y POST) y la descarga de algunos de los regusrsos asociados a las páginas obtenidas a estos métodos (SCRIPT, IMG y LINK). Para esto se desarrolló un programa java que hace uso de muchos conceptos acerca del manejo multithreading aprendidos en materias anteriores como Técnicas de Programación Concurrentes; de manera de, poder hacer un manejo adecuado y controlado de los hilos mediante la utilización de pool de threads y su correspondiente sincronización a través de colas de mensajes y locks.

2. Diagramas

2.1. Diagrama de actividades

A continuación se presenta el diagrama de actividades; el cual, compone la Vista de procesos del modelo 4+1 y mediante el cual se presenta la forma en la que se comunican los procesos (hilos) del sistema.

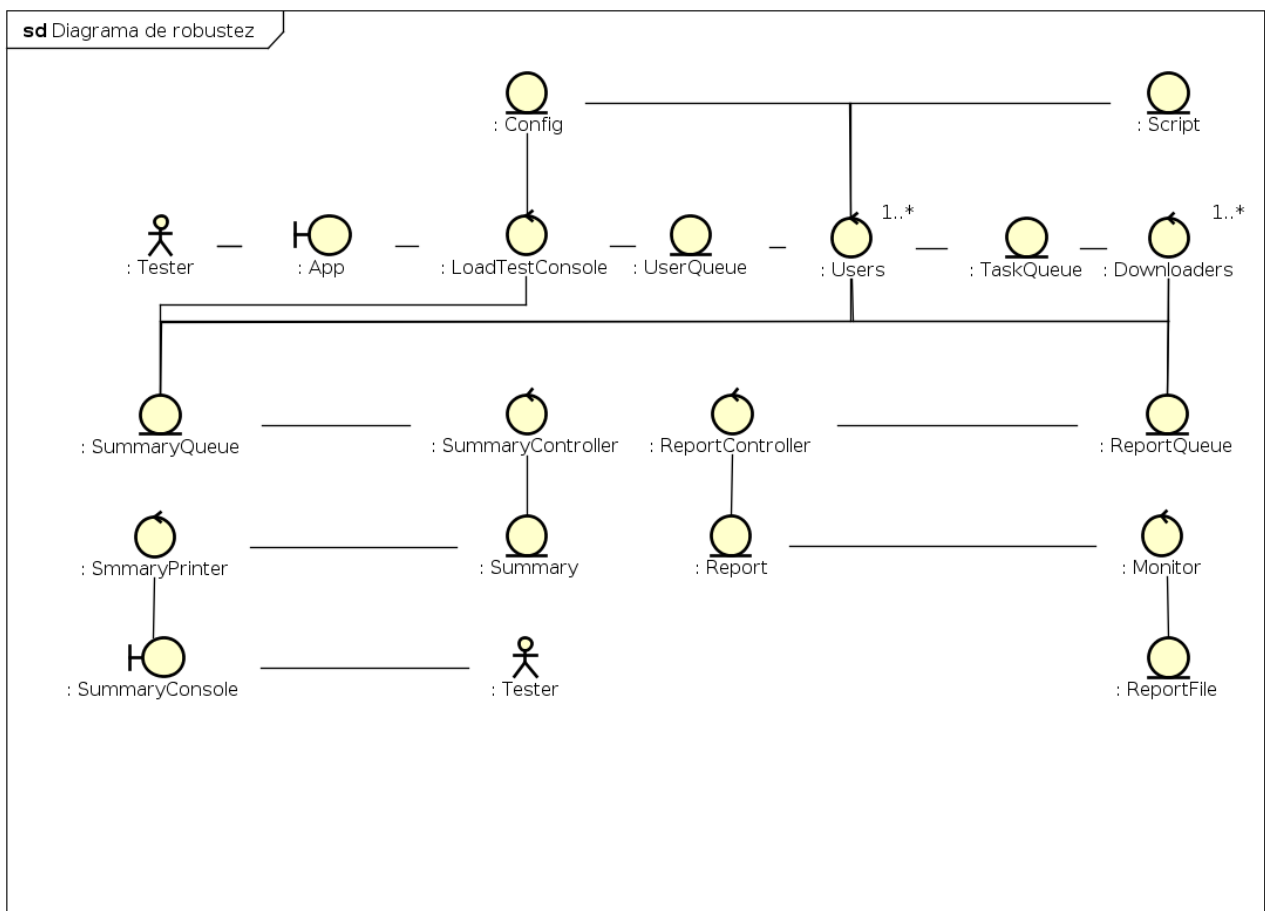
En este caso se presentan los procesos fundamentales del mismo, los cuales se encuentran en un escenario en el que el proceso principal genera al monitor, la impresora de resúmenes y a los usuarios. Estos últimos a su vez leen el script de url's y luego de un análisis de las páginas que se obtienen de ellos, generan tareas para que los downloaders descarguen los recursos asociados a esta última; esto es, imágenes, links y scripts.



2.2. Diagrama de robustez

El diagrama de robustez presentado a continuación pertenece a la vista lógica del modelo 4+1 y tiene por objetivo presentar cual es la funcionalidad del sistema, así como tambien las entidades y los controladores que lo conformarán paa poder llevarlas a cabo.

En este caso se muestra el flujo de información desde que el sistema es disparado por un usuario y como los distintos controladores van interactuando entre si a través de distintas entidades del mismo; esto es, colas de mensajes y entidades del dominio del problema protegidas con locks.



3. Código fuente

A continuación se presenta el código fuente de la solución desarrollada para este problema.