

# Trabajo Práctico N°: 1 (grupal) - Logging

## Enunciado

Se deberá implementar una herramienta que permita facilitar los diversos procesos de “Logging” que podrían requerir una aplicación.

## Funcionalidad a Implementar

### Definir la interfaz del usuario (programador)

- Proveer una API capaz de registrar mensajes/eventos con algún “nivel” determinado de prioridad.
- Proveer una API para configurar la herramienta.

### Niveles de Logging

- Proveer los siguientes Niveles (ordenados por prioridad):
  - DEBUG > INFO > WARN > ERROR > FATAL > OFF
- Permitir configurar a la herramienta para trabajar en algún nivel particular. Si por ejemplo activamos el nivel INFO deberíamos registrar todos los eventos que se encuentran a su derecha: INFO, WARN, ERROR, FATAL.

### Destinos

- Permitir loguear a un archivo particular.
- Permitir loguear por consola.
- Permitir loguear a más de un archivo y/o consola en simultáneo.

### Formato Mensajes

Permitir si se desea definir el formato de todos los mensajes logueados por algun patron:

- %d{xxxxxxx} debería aceptar cualquier formato válido de [SimpleDateFormat](#).
- %p debería mostrar el *Nivel* del mensaje.
- %t debería mostrar el nombre del thread actual.
- %m debería mostrar el contenido del mensaje logueado por el usuario.
- %% debería mostrar un % (es el escape de %).
- %n debería mostrar el “separador” con el que el usuario configuró la herramienta o un default a elección.
- %L line number.
- %F filename.
- %M method name.

Ejemplo:

```
%d{HH:mm:ss} - %p - %t - %m
```

```
21:12:09 - INFO - main - Se registró un nuevo usuario.
```

## Configuración desde un archivo de Properties

Permitir levantar la configuración de la herramienta desde un archivo de [properties](#).

## Objetivos

- Implementar una versión inicial de la herramienta con funcionalidad mínima.
- Aplicar las técnicas vistas en la teoría y en la práctica.

## Herramientas a utilizar

- Java >= 1.7
- Maven >= 3
- JUnit >= 4.11
- Git

## Restricciones

- Trabajo Práctico grupal implementado en java.
- Se deben utilizar las mismas herramientas que en el TP0 (git + maven + junit4).
- Todas las clases del sistema deben estar justificadas.
- Todas las clases deben llevar un comentario con las responsabilidades de la misma.
- El uso de herencia debe estar justificado. Se debe explicar claramente el porqué de su conveniencia por sobre otras opciones.
- Se debe tener una cobertura completa del código por tests.

## Criterios de Corrección

- Cumplimiento de las restricciones
- Documentación entregada
- Diseño del modelo
- Diseño del código
- Test Unitarios

Se tendrán en cuenta también la completitud del tp, la correctitud, distribución de responsabilidades, aplicación y uso de criterios y principios de buen diseño, buen uso del repositorio y uso de buenas prácticas en gral.

## Entrega

- Un archivo comprimido del repositorio local git en el que trabajaron (incluyendo el directorio .git).
- Se tendrá que subir el archivo al campus digital utilizando la tarea “Entrega TP N°: 1” de la Semana 7. La tarea estará disponible desde el 08/05/2014 a las 19:00 hasta el 15/05/2014 a las 22:00. No se aceptaran entregas fuera de término.

## Calendario

Jue 8/5	Presentación del TP
Jue 15/5	Entrega TP, via campus