

# Trabajo Práctico N° 2: Monitoreo de Eventos

## Enunciado

Se desea desarrollar un dashboard que permita a los usuarios monitorear, en tiempo real, los eventos que van sucediendo.

Este dashboard deberá permitir visualizar valores y resultados de consultas que, a su vez, se podrán armar dinámicamente sobre la información que se está recibiendo.

Ej: Cantidad de XX reportados por hora



En este momento, se está pensando en monitorear diferentes sistemas de tickets (estilo Redmine, Jira, etc) que soporten integrarse a un servicio de monitoreo (mediante una interfaz de integración), y de esta forma, ante ciertos eventos en el ciclo de vida de los tickets, se nos notifique la novedad en nuestro dashboard.

Por ejemplo, ante cada cambio de estado de un ticket, los sistemas de tickets van a informar a nuestro sistema dicho suceso, enviándole toda la información de dicho ticket.

El sistema de monitoreo, deberá ser capaz de integrarse en los diferentes sistemas de tickets para poder recibir la información de los sucesos que ocurran en el mismo, y capaz de procesar las consultas definidas en ese momento, visualizando el resultado de la misma en el dashboard. Por ejemplo, nuestro usuario podría querer monitorear la cantidad de tickets abiertos por hora, y la cantidad de tickets cerrados por hora, desde este momento. Ingresará dicha consulta, y a medida que ingresan novedades se va actualizando el valor correspondiente a cada consulta. Como herramienta de prueba se desea contar con un sistema de tickets simple junto con la herramienta.

### Como usuario administrador quiero:

- Poder definir consultas (al estilo: cantidad de tickets abiertos por hora, promedio de tiempo de resolución de un ticket)
- Poder crear, editar y borrar dashboards con diferentes consultas a visualizar en el mismo

- Poder habilitar o deshabilitar dashboards y consultas dentro del dashboard
- Ver el histórico que se generó para el intervalo de cada consulta, pudiendo ver la evolución de la consulta en el tiempo (Ejemplo: si la consulta es 'tickets cerrados por hora', quisiera poder ver, para cada hora que pasó, el total del mismo, y poder con ellos armar un gráfico de línea o de barras de cómo evoluciona con el tiempo, incluso poder tener el promedio según la información que tenemos)

### Como usuario guest quiero:

- Poder ver un dashboard dado su nombre, con las consultas configuradas por el administrador

Se desea a futuro incorporar aplicaciones móviles para ver los dashboards existentes.

### Sistema de Tickets

Se desea contar con un sistema de manejo de tickets **simple y genérico**.

El sistema deberá permitir crear proyectos y los usuarios asociados al mismo, configurar roles de usuario dentro del proyecto, estados para los tickets, campos obligatorios y opcionales para cada tipo de tickets y transición de estados definida según el rol del usuario.

Organizados dentro de proyectos, los tickets manejados tendrán un núcleo muy simple de información: Título, Descripción, Tipo de Ticket. A cada uno, además, se le podrá asociar comentarios, que los usuarios podrán hacer para cada ticket, con texto y autor del comentario. Cada ticket puede ir cambiando de estado.

Los estados por los que puede pasar un ticket, se definirán por cada proyecto, siendo el dueño del proyecto, quien determine dicho flujo.

Lo único obligatorio para todo flujo, será determinar cuál será el estado inicial, y cuál el final, para cada incidencia.

Se desea a futuro incorporar aplicaciones móviles para el sistema de tickets.

Se busca una solución escalable, que permita integrar distintos tipos de "fuentes de datos" para contar con una variedad de dashboards de distinta información, y que permita trabajar con grandes cantidades de datos.

## Iteración 1

- Analizar el sistema solicitado.
- Diseñar y documentar una arquitectura propuesta para el sistema.

## Iteración 2

Con el objetivo de validar diseño y arquitectura, y a modo de prueba de concepto se pide:

- Hacer uso de un mock del sistema de tickets para generar datos de entrada.
- Usar el motor de reglas para recibir datos del sistema de tickets y producir datos para los dashboard
- Implementar dashboards para cargar reglas y mostrar datos en formato programático.

## Iteración 3

- Implementar sistema de tickets que nutra al sistema de monitoreo.
- Implementar dashboards para cargar reglas y mostrar datos.

## Restricciones

- Se debe utilizar la API implementada en el TP1

## Criterios de Corrección

- Diseño del modelo
- Diseño del código
- Documentación entregada (informe, diagramas y tests)
- Correctitud
- Completitud
- Buen uso de las herramientas indicadas
- Uso de buenas prácticas en general

## Calendario

Jueves 19/4	Presentación del TP
Jueves 26/4	Entrega Iteracion 1. Validación y aceptación de la entrega en clase.
Jueves 10/5	Entrega Iteracion 2. Validación y aceptación de la entrega en clase.
Jueves 7/6	Entrega Iteracion 3. Validación y aceptación de la entrega en clase.

Referencias: <https://github.com/7510-tecnicas-de-disenio/material-clases>