75.10 Técnicas de Diseño

Trabajo Práctico Monitoreo de Eventos
Primer Cuatrimestre 2018

Grupo: 8

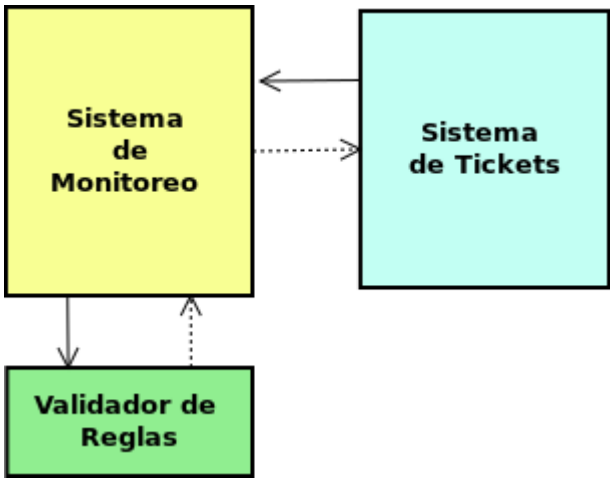
Integrantes:

Alumno	Padrón
Masrian Viken	96438

INDICE

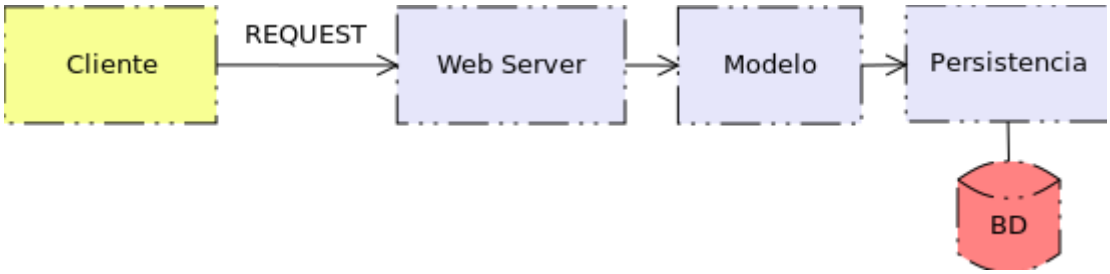
Introducción General

En este informe se describe distintos aspectos de la resolución del Sistema de Monitoreo. El esquema general que se planteo fue:



Se cuenta con tres sistemas, cada una de las implementaciones es independiente de las demás. Sin embargo hay que aclarar que cuentan con bastante interacción. Esa interacción se lleva a cabo de distintas formas para cada caso. La interacción entre el Sistema de Monitoreo y el Validador de Reglas se realiza por medio de la interoperabilidad mientras que la interacción del Sistema de Monitoreo con el Sistema de Tickets es a través de la asociación de objetos.

Tanto el Sistema de Monitoreo como el Sistema de Tickets cuentan con un sistema web para que el usuario pueda obtener una mejor usabilidad. En consecuencia se decidió hacer uso del patrón de arquitectura EA.



Nota: Por falta de tiempo no se llego a persistir los datos

Sistema de Tickets

El Sistema de Tickets permite las siguientes funcionalidades:

- Creación de proyectos
- Registro de Usuarios
- Creación de Tickets
- Creación de Tipos de Tickets
- Asignación de Usuarios a Proyectos
- Tomar Tickets
- Crear Comentarios

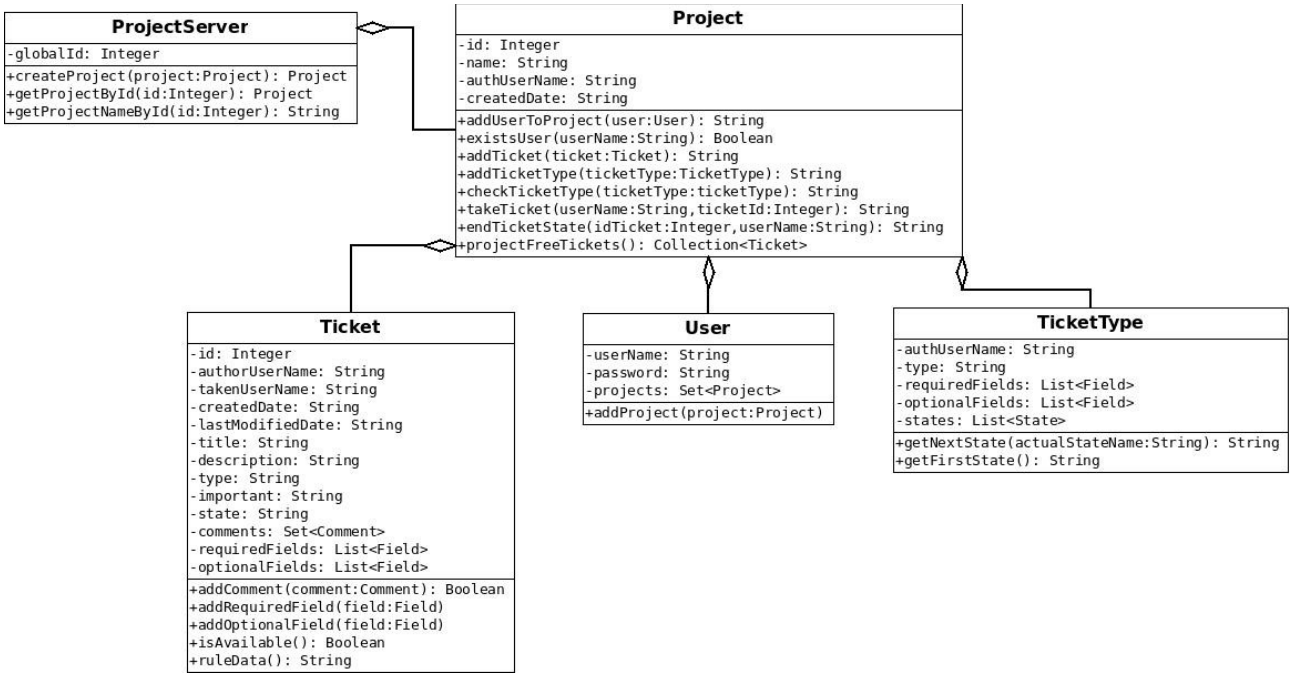
Para brindar las funcionalidades anteriores y algunas otras que no fueron nombradas, el modelo provee los siguientes servicios:

GET	/tickets/projects/{idProject}/tickets/all
GET	/tickets/projects/{idProject}/tickets/free
GET	/tickets/projects/{idProject}/ticketsCreated/{userName}
GET	/tickets/projects/{idProject}/ticketsTaken/{userName}
GET	/tickets/projects/{id}
POST	/tickets/projects/create
POST	/tickets/projects/{idProject}/addTicket
POST	/tickets/projects/{idProject}/addTicketType
POST	/tickets/projects/{idProject}/addUser
POST	/tickets/projects/{idProject}/endState/{idTicket}
POST	/tickets/projects/{idProject}/take/{idTicket}
POST	/tickets/projects/{idProject}/ticket/{idTicket}/addComment
GET	/tickets/users/all
GET	/tickets/users/{userName}/projects
POST	/tickets/users/create
POST	/tickets/users/login

Nota: La mayoría de los servicios son consumidos por el sistema web.

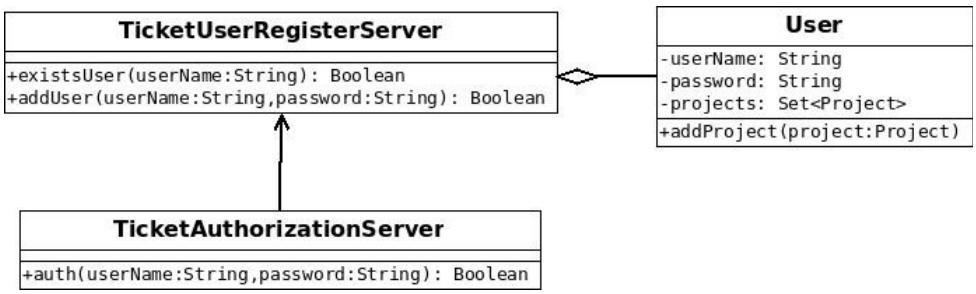
Diagrama de Clases

Las entidades que permiten el funcionamiento esencial del Sistema de Tickets se pueden observar en el siguiente diagrama:



En términos generales se puede notar que se cuenta con un ProjectServer el cual se encarga de administrar todos los proyectos del sistema. Cada uno de los proyectos cuenta con un conjunto de usuarios, tipos de tickets y tickets creados.

Para el registro y autenticación de usuarios se consideraron las siguientes entidades:



Sistema de Monitoreo

El Sistema de Monitoreo permite las siguientes funcionalidades:

- Creación de Dashboards
- Registro de Usuarios
- Creación de Consultas
- Asignación de Usuarios a Dashboards
- Relaciónar Dashboard a un Proyecto
- Consultar Resultados de las Consultas
- Ver Consultas

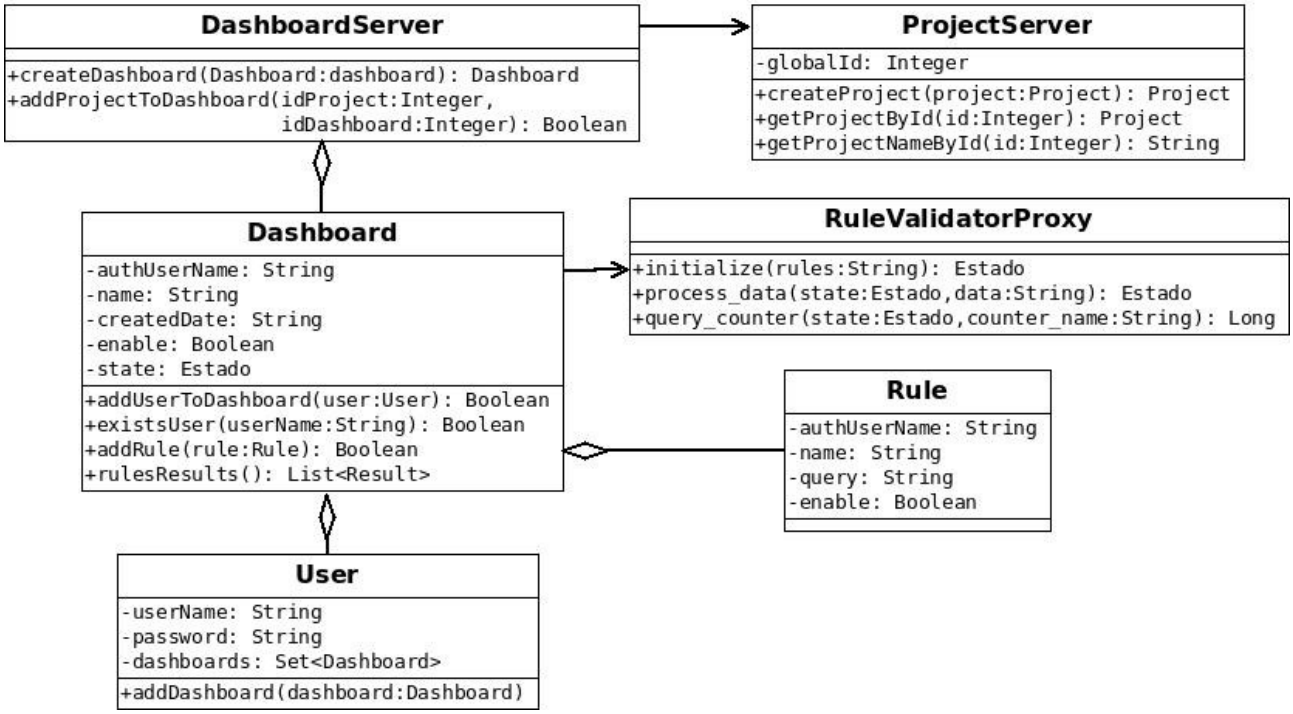
Para brindar las funcionalidades, el modelo provee los siguientes servicios:

GET	/dashboards/{idDashboard}/results
GET	/dashboards/{idDashboard}/rules/all
GET	/dashboards/{idDashboard}/users/all
POST	/dashboards/create
POST	/dashboards/{idDashboard}/addProject
POST	/dashboards/{idDashboard}/addRule
POST	/dashboards/{idDashboard}/addUser
GET	/dashboards/users/all
GET	/dashboards/users/{userName}
POST	/dashboards/users/create
POST	/dashboards/users/login

Nota: La mayoría de los servicios son consumidos por el sistema web.

Diagrama de Clases

Las entidades que permiten el funcionamiento esencial del Sistema de Monitoreo se pueden observar en el siguiente diagrama:



En términos generales se cuenta con un DashboardServer que se encarga de administrar todos los dashboards del sistema. Este ademas cuenta con una relación de asociación con el ProjectServer para poder llevar a cabo la asignación de un proyecto a un determinado dashboard. Luego cada dashboard cuenta con un conjunto de usuarios, reglas. También tiene asociada una entidad llamada RuleValidatorProxy mediante la cual interactuá con el Sistema Validador de Reglas implementado en Clojure.

Sistema Validador de Reglas

El Sistema Validador de Reglas permite las siguientes funcionalidades:

- Creación de Reglas
- Carga de Datos
- Consultar Resultados de Reglas

Para brindar las funcionalidades, el modelo provee los siguientes servicios:

initialize_processor
process_data
drop_signals
query_counter