

## **Trabajo Práctico N°2: Monitoreo de eventos**

**\_ Grupo N° 07.**

**\_ Integrantes:**

**Fontana, Lucas            - Padrón: 95814 -**

**Rodríguez, Rómulo       - Padrón: 86816 -**

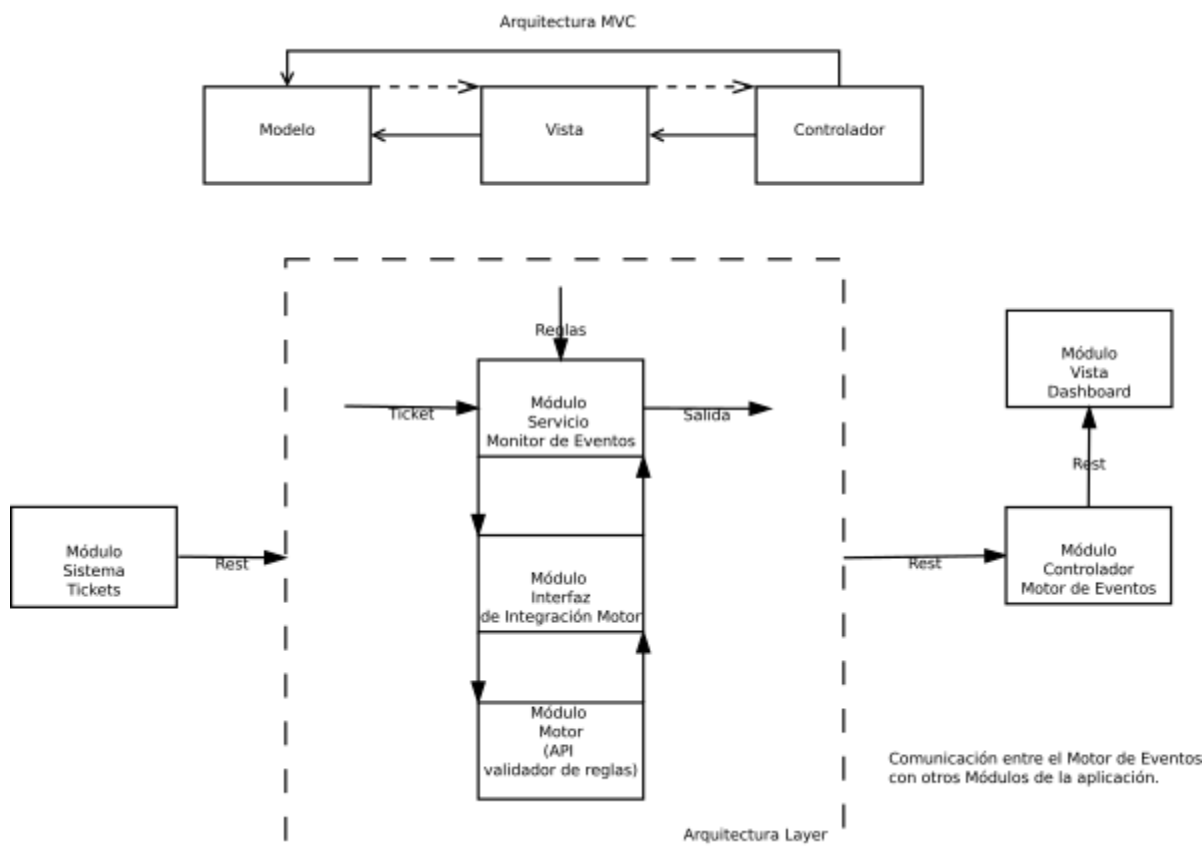
**Iteración 2**

### Comentarios iniciales:

(Adjuntando a diseño de Lucas, detallo aspectos particulares de enfoque de diseño.

Se propone para el Monitoreo de eventos una arquitectura principal MVC.

Enfocándose en el “motor” de la aplicación (La API desarrollada en la primer entrega, una microarquitectura Layer.



En base a un breve respuesta del Profesor Diego, Montaldo, hace mención de la palabra “REST”.

Investigando al respecto, algunas definiciones:

- “Transferencia de estado representacional”.
- “REST is not really a technology in itself, but more an architectural pattern. REST is very simple and just involves using plain XML or JSON as a communication medium, combined with URL patterns that are "representational" of the underlying system, and HTTP methods such as GET, PUT, POST and DELETE.”
- “REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON.”

Se entiende que se trata de una interfaz de comunicación entre módulos de una misma aplicación y entre aplicaciones para el intercambio de datos. O sea que se pretende este tipo de comunicación, en particular entre los Ticket ---Motor---Dashboard.

En particular haciendo referencia al esquema presentado:

Una capa de servicio encapsula la lógica del negocio, la API validador de reglas (el Motor).

Un módulo intermedio , en primera instancia consistente en una API de Clojure que encapsula los plugins Cheshire y DataJason. El propósito es la conversión de los mensajes en formato JSON en estructuras de datos que entiende el Motor y viceversa para la respuesta.

Renombrando a Martin Fowler en su libro UML gota a gota, hace mención a riesgos, en particular riesgos de habilidades y riesgos tecnológicos.

Como herramientas :

Clojure

Grails : *Grails* es un framework para aplicaciones web libre desarrollado sobre el lenguaje de programación Groovy (el cual a su vez se basa en la Java platform).

Actualmente:

Grails Version: 3.3.1

Groovy Version: 2.4.11

JVM Version: 1.8.0\_111

Con el propósito de “rememorar y solucionar las dificultades debido a la utilización de esta herramienta” (conocimiento básico), se hacen algunas pruebas básicas de integración de la API (el Motor).

Se procede entonces a convertir los tests del TP1 en test unitarios en Grails, e ir evaluando su respuesta.

En este sentido se incorpora un plugin en Clojure para utilizar el mismo en el desarrollo, es decir directamente colocar en una sub-carpeta “clj” del proyecto “Monitor de Eventos “ de Grails, lo desarrollado en Clojure. Lamentablemente se encontraron limitaciones de uso del plugin:

Solo namespace grails y no puede haber dependencias (require).

Solo puede ser encapsulado en clases del dominio y no en la capa de servicio.

Puede verse esta tentativa de tests inicial en el repositorio en la rama\_prototipo...

Siguiente paso de intento de integración para capturar la funcionalidad es encapsular en JAR la API (el Motor). Y continuar con las pruebas.

De éxito implementarlo según el diseño planteado.

## SEGUNDA ITERACIÓN

### **CURL:**

**cURL** es un proyecto de software consistente en una [biblioteca](#) (libcurl) y un [intérprete de comandos](#) (curl) orientado a la transferencia de archivos. Soporta los protocolos [FTP](#), [FTPS](#), [HTTP](#), [HTTPS](#), [TFTP](#), [SCP](#), [SFTP](#), [Telnet](#), [DICT](#), [FILE](#) y [LDAP](#), entre otros. La primera versión se publicó en 1997 y se basó en una pequeña herramienta llamada *httpget* escrita por el brasileño Rafael Sagula.<sup>4</sup>

cURL soporta certificados HTTPS, HTTP POST, HTTP PUT, subidas FTP, Kerberos, subidas mediante formulario HTTP, proxies, cookies, autenticación mediante usuario y contraseña (Basic, Digest, NTLM y Negotiate para HTTP y [kerberos](#) 4 para FTP), continuación de transferencia de archivos, tunneling de proxy HTTP y otras prestaciones. cURL es [Open Source](#), software libre distribuido bajo la [Licencia MIT](#).

*El principal propósito y uso para cURL es automatizar transferencias de archivos o secuencias de operaciones no supervisadas. Es por ejemplo, una herramienta válida para simular las acciones de usuarios en un [navegador web](#).*

**GET** (sin ID retorna listado de instancias)

- `curl -i -H "Accept: application/json" localhost:8080/books/1`
- Ejemplo retorno
- `{"id":1,"title":"The Stand"}`
- 

**POST**

- `curl -i -X POST -H "Content-Type: application/json" -d '{"title":"Along Came A Spider"}' localhost:8080/books`

**PUT**

- `curl -i -X PUT -H "Content-Type: application/json" -d '{"title":"Along Came A Spider"}' localhost:8080/books/1`

**DELETE**

- `curl -i -X DELETE localhost:8080/books/1`

## Aplicación REST

### Aplicación a TP (pre-creados algunos datos):

#### GET

```
curl -i -H "Accept: application/json" localhost:8080/ticketsmock/1
```

#### HTTP/1.1 200

X-Application-Context: application:development

Content-Type: application/json;charset=UTF-8

Transfer-Encoding: chunked

Date: Thu, 17 May 2018 03:01:58 GMT

```
{"id":1,"titulo":"A","dateCreated":"2018-05-17T02:37:27Z","caracteristica":"ESTE ES UN TICKET MOCK","lastUpdated":"2018-05-17T02:37:27Z","estado":{"id":1},"flujo":{"id":1},"codigo":1,"propietario":"AA","descripcion":"MockA"}
```

#### GET (lista)

```
curl -i -H "Accept: application/json" localhost:8080/ticketsmock/
```

#### HTTP/1.1 200

X-Application-Context: application:development

Content-Type: application/json;charset=UTF-8

Transfer-Encoding: chunked

Date: Thu, 17 May 2018 03:04:44 GMT

```
[{"id":1,"titulo":"A","dateCreated":"2018-05-17T02:37:27Z","caracteristica":"ESTE ES UN TICKET MOCK","lastUpdated":"2018-05-17T02:37:27Z","estado":{"id":1},"flujo":{"id":1},"codigo":1,"propietario":"AA","descripcion":"MockA"}, {"id":2,"titulo":"B","dateCreated":"2018-05-17T02:37:27Z","caracteristica":"ESTE ES UN TICKET MOCK","lastUpdated":"2018-05-17T02:37:27Z","estado":{"id":2},"flujo":{"id":1},"codigo":2,"propietario":"BB","descripcion":"MockB"}, {"id":3,"titulo":"C","dateCreated":"2018-05-17T02:37:27Z","caracteristica":"ESTE ES UN TICKET MOCK","lastUpdated":"2018-05-17T02:37:27Z","estado":{"id":3},"flujo":{"id":1},"codigo":3,"propietario":"CC","descripcion":"MockC"}, {"id":4,"titulo":"D","dateCreated":"2018-05-17T02:37:27Z","caracteristica":"ESTE ES UN TICKET MOCK","lastUpdated":"2018-05-17T02:37:27Z","estado":{"id":4},"flujo":{"id":2},"codigo":4,"propietario":"DD","descripcion":"MockD"}, {"id":5,"titulo":"E","dateCreated":"2018-05-17T02:37:27Z","caracteristica":"ESTE ES UN TICKET MOCK","lastUpdated":"2018-05-17T02:37:27Z","estado":{"id":5},"flujo":{"id":2},"codigo":5,"propietario":"EE","descripcion":"MockE"}, {"id":6,"titulo":"F","dateCreated":"2018-05-17T02:37:27Z","caracteristica":"ESTE ES UN TICKET MOCK","lastUpdated":"2018-05-17T02:37:27Z","estado":{"id":6},"flujo":{"id":2},"codigo":6,"propietario":"FF","descripcion":"MockF"}]
```

## POST

```
curl -i -X POST -H "Content-Type: application/json" -d '{"titulo": "NUEVO-TICKET","caracteristica":"creado con CURL","estado":{"id":1},"flujo":{"id":1},"codigo":333,"propietario":"CURL","descripcion":"EMPRESA CURL"}' localhost:8080/ticketsmock/
```

### HTTP/1.1 201

X-Application-Context: application:development

**Location: http://localhost:8080/ticketsmock/7**

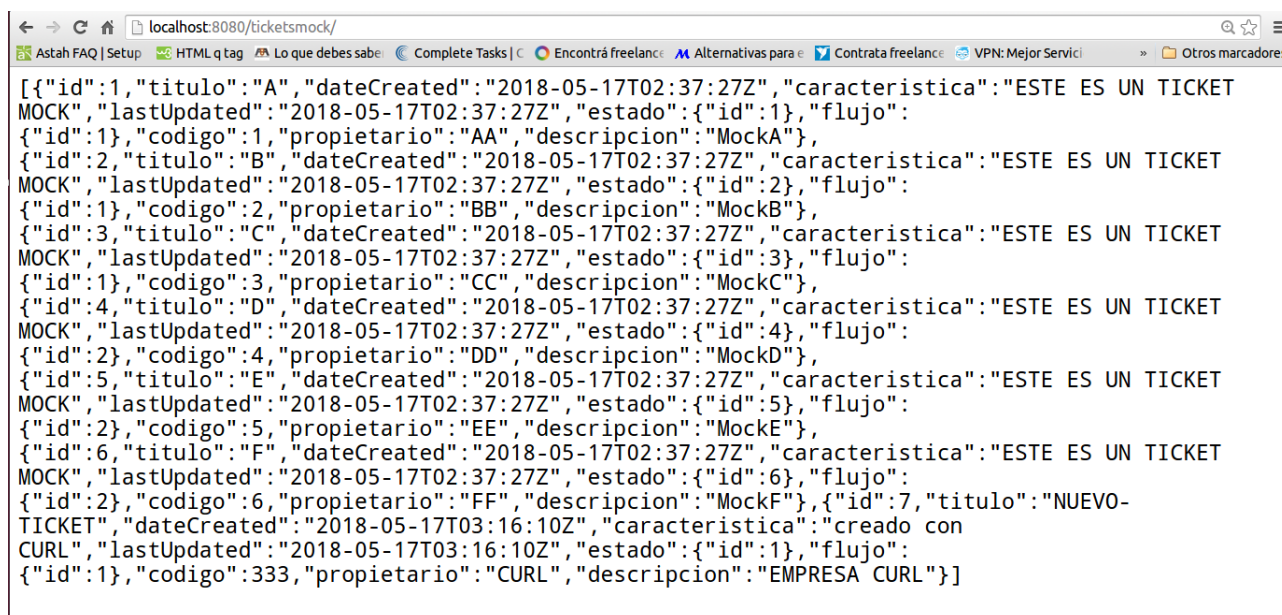
Content-Type: application/json; charset=UTF-8

Transfer-Encoding: chunked

Date: Thu, 17 May 2018 03:16:10 GMT

```
{"id":7,"titulo":"NUEVOTICKET",  
"dateCreated":"20180517T03:16:10Z",  
"caracteristica":"creado con CURL",  
"lastUpdated":"2018-05-17T03:16:10Z",  
"estado":{"id":1},  
"flujo":{"id":1},  
"codigo":333,  
"propietario":"CURL",  
"descripcion":"EMPRESA CURL"}
```


En el navegador (captura de pantalla):





### Creación de ticket modelo básico:

localhost:8080/ticket/create

Astah FAQ | Setup HTML q tag Lo que debes saber Complete Tasks | C Encontrá freelance Alternativas para e Co

 Grails

 Principal  Ticket Lista

Crear Ticket

Estado \* fiuba.materia7510.aplicacion.proveedor.Estado : 1 ▾


Codigo \* 137

Propietario \* Perito Rubio

Titulo \* Empresa Kariopolis

Descripcion \* Tratamiento Torticolis

Flujo \* fiuba.materia7510.aplicacion.proveedor.Flujo : 1 ▾

 Crear