My Event

Ricardo Acedo de Talavera
Alberto López Castilla
Gabriel G. Valenzuela Camacho

1. Introducción

En el ámbito de la cultura, estos últimos años se ha dejado de invertir dinero por parte de las instituciones gubernamentales debido a la crisis económica. Cada vez podemos ver menos eventos de ocio relacionados con el arte, el espectáculo y la música entre otros.

Con mucha mas intensidad se nota esta decadencia cultural para los artistas emergentes. Los nuevos artistas que intentan insertarse en el mundo del espectáculo lo tienen muy difícil debido a que cada vez hay menos sitios para demostrar lo que valen y enseñar sus habilidades a la gente. Ya no hablemos del hecho de poder vivir de lo que mas les gusta hacer.

Con la aplicación que hemos desarrollado, nos hemos propuesto poner nuestro granito de arena para mejorar esta situación.

Procuramos el acercamiento entre personas, artistas y recintos donde realizar eventos. De esta manera los artistas tendrán mas fácil encontrar lugares donde trabajar y las salas de eventos tendrán mas facilidad para publicitarse y conseguir clientes gracias a estos nuevos artistas. La aplicación permitirá que los artistas puedan crear eventos en una fecha concreta. Los usuarios podrán ver los eventos creados por los artistas para posteriormente apuntarse en el mismo si es de su interés.

Con la plataforma para artistas emergentes esperamos realzar este sector, ya que hay mucho talento escondido del que no podemos disfrutar debido a que estos artistas no son famosos y no están subvencionados por las grandes discográficas o promotoras de eventos.

2. Requisitos

- Alta de usuarios
- Alta de artistas
- Alta de eventos
- Baja de usuarios
- Baja de artistas
- Baja de eventos
- Consulta de usuarios
- Consulta de artistas
- Consulta de eventos
- Edición de usuarios
- Edición de artistas
- Edición de eventos
- Identificación de usuarios y artistas
- Los artistas crean eventos
- Los usuarios pueden apuntarse a los eventos creados por los artistas.

3. Esquema funcional

La aplicacion tiene una arquitectura MVC, para mostrar un esquema funcional lo haremos de la siguiente forma:

- En primer lugar, tendremos el paquete de las vistas que seran creadas con Vaadin, y las peticiones seran enviadas al controlador.
- Tendremos un paquete Controlador, que su funcion sera atender las peticiones de las vistas y creacion de los objetos, para posteriormente realizar las peticiones al modelo. Tambien sera el encargado de avisar a las vistas o vista, los datos modificados.
- Nuestro paquete Modelo, sera donde se recogen las peticiones del controlador, se realizan las consultas a la base de datos y se devuelven los resultados al controlador.

Algunas funcionalidades de nuestro MVC no son las convencionales ya que en Vaadin no se utiliza esa arquitectura, la que recomienda la documentación oficial es Modelo-Vista-Presentacion.

4. Diagrama UML

A continuación se presenta el diagrama de clases UML, donde se refleja las relaciones entre las distintas clases de nuestro sistema. Figura 1:

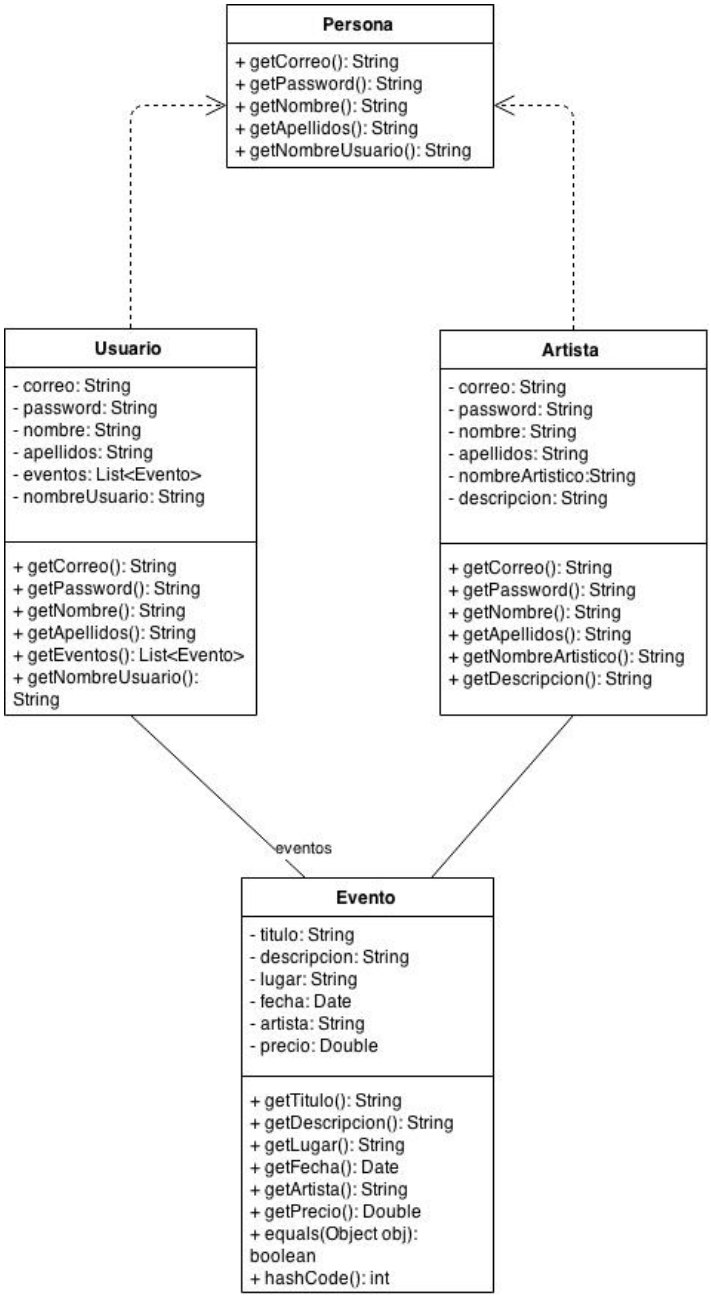

Figura 1

5. Esquema Maven

A continuación exponemos el esquema Maven generado a través de la terminal. Para ello hemos utilizado, estando en el directorio del proyecto, el comando: mvn dependency:tree

```
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building MyEvent 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-dependency-plugin:2.8:tree (default-cli) @ MyEvent ---
[INFO] com.grupo1:MyEvent:war:1.0-SNAPSHOT
```

```
[INFO] +- com.vaadin.addon:vaadin-charts:jar:1.0.0:compile
[INFO] |  +- com.google.code.gson:gson:jar:2.2:compile
[INFO] |  \- commons-io:commons-io:jar:2.2:compile
[INFO] +- org.mongodb:mongo-java-driver:jar:2.11.3:compile
[INFO] +- javax.servlet:javax.servlet-api:jar:3.0.1:provided
[INFO] +- com.vaadin:vaadin-server:jar:7.4.1:compile
[INFO] |  +- com.vaadin:vaadin-sass-compiler:jar:0.9.12:compile
[INFO] |  |  \- com.vaadin.external.flute:flute:jar:1.3.0.gg2:compile
[INFO] |  +- com.vaadin:vaadin-shared:jar:7.4.1:compile
[INFO] |  |  +- com.vaadin.external.streamhtmlparser:streamhtmlparser-jsilver:ja
r:0.0.10.vaadin1:compile
[INFO] |  |  \- com.vaadin.external.google:guava:jar:16.0.1.vaadin1:compile
[INFO] |  \- org.jsoup:jsoup:jar:1.8.1:compile
[INFO] +- com.vaadin:vaadin-push:jar:7.4.1:compile
[INFO] |  \- com.vaadin.external.atmosphere:atmosphere-runtime:jar:2.2.4.vaadin4
:compile
[INFO] |     \- com.vaadin.external.slf4j:vaadin-slf4j-jdk14:jar:1.6.1:compile
[INFO] +- com.vaadin:vaadin-client:jar:7.4.1:provided
[INFO] |  +- org.w3c.css:sac:jar:1.3:compile
[INFO] |  +- javax.validation:validation-api:jar:1.0.0.GA:provided
[INFO] |  \- javax.validation:validation-api:jar:sources:1.0.0.GA:provided
[INFO] +- com.vaadin:vaadin-client-compiler:jar:7.4.1:provided
[INFO] |  +- commons-collections:commons-collections:jar:3.1:provided
[INFO] |  +- commons-logging:commons-logging:jar:1.1.3:provided
[INFO] |  +- ant:ant:jar:1.6.5:provided
[INFO] |  +- net.sourceforge.cssparser:cssparser:jar:0.9.11:provided
[INFO] |  +- ant:ant-launcher:jar:1.6.5:provided
[INFO] |  +- org.ow2.asm:asm:jar:5.0.3:provided
[INFO] |  +- org.ow2.asm:asm-util:jar:5.0.3:provided
[INFO] |  |  \- org.ow2.asm:asm-tree:jar:5.0.3:provided
[INFO] |  +- org.ow2.asm:asm-commons:jar:5.0.3:provided
[INFO] |  +- org.eclipse.jetty:jetty-annotations:jar:8.1.12.v20130726:provided
[INFO] |  |  +- org.eclipse.jetty:jetty-plus:jar:8.1.12.v20130726:provided
[INFO] |  |  |  +- org.eclipse.jetty.orbit:javax.transaction:jar:1.1.1.v20110521
0645:provided
[INFO] |  |  |  \- org.eclipse.jetty:jetty-jndi:jar:8.1.12.v20130726:provided
[INFO] |  |  |     +- org.eclipse.jetty:jetty-server:jar:8.1.12.v20130726:provid
ed
[INFO] |  |  |     |  \- org.eclipse.jetty.orbit:javax.servlet:jar:3.0.0.v201112
011016:provided
[INFO] |  |  |     \- org.eclipse.jetty.orbit:javax.mail.glassfish:jar:1.4.1.v20
1005082020:provided
[INFO] |  |  |        \- org.eclipse.jetty.orbit:javax.activation:jar:1.1.0.v201
105071233:provided
[INFO] |  |  +- org.eclipse.jetty:jetty-webapp:jar:8.1.12.v20130726:provided
[INFO] |  |  |  +- org.eclipse.jetty:jetty-xml:jar:8.1.12.v20130726:provided
[INFO] |  |  |  \- org.eclipse.jetty:jetty-servlet:jar:8.1.12.v20130726:provided
[INFO] |  |  |     \- org.eclipse.jetty:jetty-security:jar:8.1.12.v20130726:prov
ided
[INFO] |  |  +- org.eclipse.jetty.orbit:javax.annotation:jar:1.1.0.v201108011116
:provided
[INFO] |  |  \- org.eclipse.jetty.orbit:org.objectweb.asm:jar:3.1.0.v20080306191
0:provided
[INFO] |  +- org.eclipse.jetty:jetty-servlets:jar:8.1.12.v20130726:provided
[INFO] |  |  +- org.eclipse.jetty:jetty-continuation:jar:8.1.12.v20130726:provid
ed
[INFO] |  |  \- org.eclipse.jetty:jetty-client:jar:8.1.12.v20130726:provided
[INFO] |  |     \- org.eclipse.jetty:jetty-http:jar:8.1.12.v20130726:provided
[INFO] |  |        \- org.eclipse.jetty:jetty-io:jar:8.1.12.v20130726:provided
```

```
[INFO] | +- org.eclipse.jetty:jetty-util:jar:8.1.12.v20130726:provided
[INFO] | +- org.jdesktop:swing-worker:jar:1.1:provided
[INFO] | +- commons-codec:commons-codec:jar:1.8:provided
[INFO] | +- org.apache.commons:commons-lang3:jar:3.1:provided
[INFO] | +- org.apache.james:apache-mime4j:jar:0.6:provided
[INFO] | +- org.apache.httpcomponents:httpclient:jar:4.3.1:provided
[INFO] | +- org.apache.httpcomponents:httpcore:jar:4.3:provided
[INFO] | +- org.apache.httpcomponents:httpmime:jar:4.3.1:provided
[INFO] | +- net.sourceforge.nekohtml:nekohtml:jar:1.9.19:provided
[INFO] | +- xalan:serializer:jar:2.7.1:provided
[INFO] | +- xerces:xercesImpl:jar:2.11.0:provided
[INFO] | +- xml-apis:xml-apis:jar:1.4.01:provided
[INFO] | +- com.ibm.icu:icu4j:jar:50.1.1:provided
[INFO] | \- com.vaadin:vaadin-client-compiler-deps:jar:1.2.0:provided
[INFO] \- com.vaadin:vaadin-themes:jar:7.4.1:compile
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 3.287 s
[INFO] Finished at: 2015-05-11T19:09:48+02:00
[INFO] Final Memory: 11M/26M
[INFO] ------------------------------------------------------------------------
```

6. Diseño de la base de datos

A continuación presentamos el esquema de la base de datos y las colecciones correspondientes. Figura 2

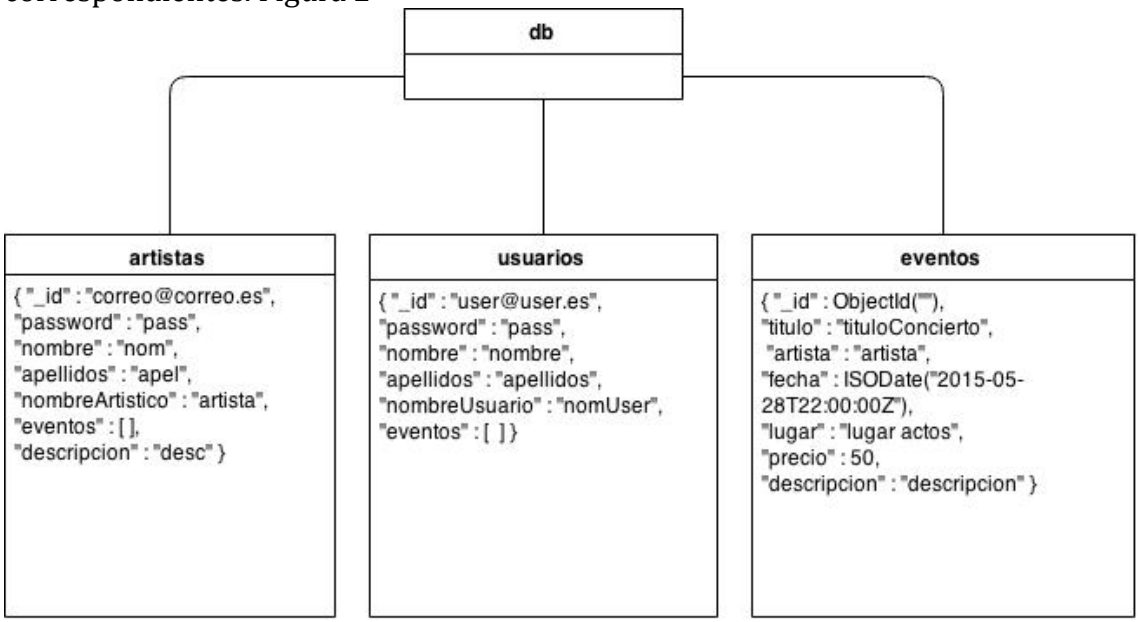

Figura 2

1. **Artistas**: Utilizaremos esta colección para guardar los documentos de cada artista registrado en la aplicación:
2. **Usuarios**: Utilizaremos esta colección para guardar los documentos de cada usuario registrado en la aplicación.
3. **Eventos**: Utilizaremos esta colección para guardar los documentos asociados a cada evento, eventos creados por parte de los artistas.

7. Extras

En este punto se describirán todos los extras añadidos en nuestro proyecto.
Principalmente son características de Vaadin que no se han visto en clase.
A continuación se enumeraran estos extras:

- Navigator: Nos proporciona la capacidad de cambiar de vistas dentro de la misma UI. De esta forma hemos realizado una aplicación que cambie entre las vistas de login, registro y la vista principal.

- Validator: Utilizamos el validador para validar las cuentas de correo y otros datos de importancia para guardarlos en la base de datos correctamente.

- Views: En las vistas principales, el panel principal se genera a partir de clases que implementan la interfaz vista de Vaadin. De esta forma las vistas del panel principal no se ocultan, si no que se carga el objeto de la clase vista correspondiente.

- Drag and Drop: Utilizamos esta técnica para intercambiar eventos entre dos tablas. De esta forma que los usuarios pueden guardar en una tabla los eventos a los que se haya apuntado.