Aplicación Tellmetweet

Memoria de proyecto final de grado

Grado de multimedia

Desarrollo de aplicaciones interactivas

Autor: Angel Contreras Picazo

Consultor: Kenneth Capseta Nieto

Profesor: Carlos Casado Martinez

28 de noviembre de 2015

CREDITOS / COPYRIGHT

Tellmetweet by Angel Contreras Picazo is licensed under a Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional License.

Licencia de Creative Commons:

Reconocimiento - No comercial - CompartirIgual (by-nc-sa)

- No está permitido un uso comercial ni de la obra original ni de sus posibles obras derivadas.
- Su distribución debe realizarse con la misma licencia que regula la obra original.
- Puedes copiar, y distribuir los contenidos de esta obra en estas condiciones:
    - Utilizar la misma licencia que regula esta obra
    - No hacerlo con fines comerciales
    - Reconocer la autoría del autor original de la obra original

Twitter4J tiene licencia Apache License 2.0, incluye software de JSON.org basado en la JSON License, que permite utilizar el software sin ninguna restricción siempre y cuando sea para bien y se incluya este aviso.

ExtJs y GeoExt están licenciados bajo la GNU General Public License 3(GPLv3), que permite su uso gratuito a cambio de liberar el código de cualquier programa realizado con ella, es posible no liberar el código pagando unas tasas.

Eclipse está licenciado bajo la Licencia Pública Eclipse (EPL) que permite usar, modificar y distribuir los programas, estando obligados a liberar sus códigos a cambio en algunos casos.

Paypal utiliza “Paypal Developer Agreement” que es un acuerdo entre el usuario y Paypal. https://www.paypal.com/webapps/mpp/ua/xdeveloper-full

DEDICATORIA

A mi mujer y a mi hija por toda la paciencia que han tenido conmigo durante mis años de estudio.

EXTRACTO

Hoy en día toda empresa que quiera estar en internet y vender sus productos tiene que tener una estrategia de Marketing Online, dicha estrategia puede ser distinta en función de las necesidades de cada empresa y puede incluir desde la optimización del sitio web y su servidor, pasando por una interfaz usable, campañas de posicionamiento SEO y SEM, promociones de email-marketing, advertising, publicación de contenidos y gestión de las redes sociales entre las más importantes y conocidas.

Las redes sociales como Twitter forman parte de la estrategia de marketing online de las empresas que están en internet. Hay muchos profesionales que se dedican a diseñar y llevar a cabo estas estrategias con el fin de mejorar las ventas y rentabilidad de la empresa para la que trabajan, pero todos estos profesionales no disponen de las herramientas adecuadas para trabajar sus estrategias.

Por este motivo mi trabajo de fin de grado, pretende ser una herramienta de referencia en el mundo del marketing online, la aplicación desarrollada en este proyecto, permite saber de que está hablando la gente en Twitter en un lugar concreto.

Conocer esta información nos da una gran ventaja respecto a la competencia, saber de que está hablando la gente en un lugar permite ajustar las campañas de marketing para que lleguen a esos miles de personas que están hablando de un tema en concreto.

AGRADECIMIENTOS

A todos los compañeros, consultores, profesores y tutores que me han acompañado durante el transcurso de mis estudios en el grado de multimedia.

INDICE

Imágenes empleadas en el video 11

1. Introducción 1

1.1 Antecedentes 1
1.2 Elección De La Propuesta 1
1.3 Motivación 2
1.4 Interés General De La Propuesta 2

2. Descripción / definición / hipótesis.....3

2.1 ¿Que Importancia Tienen Las Redes Sociales En El Marketing Online?.....3
2.2 ¿Que Puede Hacer Tellmetweet Por Las Redes Sociales? 3
2.3 Tecnologías Utilizadas 5

3. Objetivos 6

3.1 Objetivos Principales 6
3.1 Objetivos Secundarios 6

4. Marco teórico práctico 7

4.1 Situación Actual 7
4.2 Publico Potencial 8
4.3 La Competencia 8

5. Contenidos 10

5.1 Generales 10
5.2 Otras Opciones 10

6. Metodología 12

7. Arquitectura de la aplicación.....13

7.1 Cliente 13
7.2 Servidor 13
7.3 Base De Datos 13

7.4 Aplicación 14
7.5 Tecnologías 17

8. Plataforma de desarrollo 18

8.1 Hardware 18
8.2 Software 18
8.3 Objetivo Del Entorno De Desarrollo 19
8.4 Descripción Del Entorno De Desarrollo 20
8.5 Funcionamiento Del Entorno De Desarrollo 21

9. Planificación 23

10. Procesos de trabajo / desarrollo 25

10.1 Definición Del Proyecto 25
10.2 Puesta En Marcha Del Entorno De Desarrollo 25
10.3 Puesta En Marcha Del Servidor De Pruebas 25
10.4 Creación Y Puesta En Marcha De La Base De Datos 25
10.5 Puesta En Marcha De La Aplicación 26
10.6 Despliegue De La Aplicación 26
10.7 Primeras Pantallas Del Frontend 26
10.8 Replica En El Vps 26
10.9 Entregas 27

11. Apis utilizadas 28

11.1 Twitter4J 28
11.2 Paypal Sdk 28

12. Diagramas uml 30

13. Prototipos 31

13.1 Home 31
13.2 Trends 32
13.3 Map 33
13.4 Chart 34
13.5 Register 35

13.6 Login 36
15. Perfiles de usuarios 37
16. Usabilidad/ux 38
17. Seguridad 39
17.1 Servidor 39
17.2 Backend 39
17.1 Fronted 40
18. Tests 41
18.1 Aplicación 41
18.2 Usuarios 41
19. Versiones de la aplicación 42
20. Requisitos de instalación 43
21. Instrucciones de instalación 44
21.1 Instalación Del Servidor 44
21.2.1 Instalación De La Base De Datos 44
21.2.2 Creación De La Base De Datos 44
21.2.3 Importación De Scripts 45
21.3.1 Instalación Del Servidor De Aplicaciones 45
21.3.2 Configuración De Tomcat 46
21.4 Desplegar La Aplicación 46
22. Instrucciones de USO 49
22.1 Partes De La Aplicación 49
22.2 Home 50
22.3 Trending Topics 51
22.4 Map 53
22.5 Chart 54
22.6 Register 56
22.6 Login 56

22.6 Logout 57
22.7 Configuración 57
22.8 Premium 58

23. Bugs 59

23.1 Api Twitter 59

25. Presupuesto 60

26. Análisis de mercado 61

Annexo 1. Entregables del proyecto 63

Annexo 2. Código fuente (extractos) 64

2.1 Servidor 64
2.2.1 Aplicación - Controller 65
2.2.2 Aplicación - Bo 66
2.2.3 Aplicación - Interfaz De Un Dao 67
2.2.4 Aplicación - Implementación De Un Dao 68

Annexo 3. Librerías/Código externo 69

3.1 Twitter4J 69
3.2 Paypal Sdk 70

Annexo 4. Capturas de pantalla 71

Anexo 4.1 Puesta En Marcha De La Aplicación 71
Anexo 4.2 Puesta En Marcha Del Frontend 76

Annexo 5. Guia de usuario 79

Anexo 5.1 Perfiles De Usuario 79
Anexo 5.2 Validación De Campos 79
Anexo 5.3 Número De Consultas Diarias 80
Anexo 5.4 Trend List 80
Anexo 5.5 Mapa 81
Anexo 5.6 Chart 82
Anexo 5.7 Localizaciones Favoritas 83

Annexo 6 Libro de estilo 84
Icono Y Logotipo 84
Paleta De Colores 84
Paleta Tipográfica 85
Margenes 85
Iconos 85
Annexo 8. Glosario / índice analítico 86
Annexo 9. Bibliografía 87

IMÁGENES EMPLEADAS EN EL VIDEO

AppleTv - http://images.apple.com/es/tv/images/overview/hero_large.png

Extraer datos - http://findicons.com/search/data-mining

Underscores - http://www.sitepoint.com/wordpress-theme-development-getting-started-with-underscores/

Marketing Online - http://makecommunication.es/marketing-online-en-tu-negocio/

Eclipse Mars - https://projects.eclipse.org/releases/mars

Java Jdk - https://www.timetoast.com/timelines/historia-lenguaje-java

Apache Tomcat - https://vulnerabilityteam.files.wordpress.com/2009/06/apache-tomcat_logo_nomatte.jpg

MySql - https://upload.wikimedia.org/wikipedia/en/thumb/6/62/MySQL.svg/640px-MySQL.svg.png

Maven - http://4.bp.blogspot.com/-slDduww_6BE/U78BCXAdfVI/AAAAAAAAAM0/1A1GAFpQ188/s1600/mavenCaratula.png

Spring - http://www.javatpoint.com/images/spimages/spring1.png

Spring Roo - https://lh4.googleusercontent.com/-_DpgkWvc3bQ/UUwmwkLNdII/AAAAAAAAAhU/kG3QSpLOhtw/s301/Logo_SpringRoo.png

Hibernate - https://upload.wikimedia.org/wikipedia/commons/2/22/Hibernate_logo_a.png

ExtJs - http://www.professionalsofttech.com/wp-content/uploads/2014/04/extjs-300x160.jpg

Sencha Cmd - http://blog.ibuildings.com/?attachment_id=4207

Geo Ext - http://www.terrestris.de/wp-content/uploads/2012/02/geoext-logo_transp_11.png

Ajax - https://upload.wikimedia.org/wikipedia/commons/thumb/a/a1/AJAX_logo_by_gengns.svg/300px-AJAX_logo_by_gengns.svg.png

Junio - http://junit.org/images/junit-logo.png

1. INTRODUCCIÓN

La aplicación desarrollada permite consultar las tendencias o Trending Topics de Twitter, permite obtener las tendencias de un lugar determinado de entre los disponibles por la aplicación.

1.1 Antecedentes

A veces cuando alguien pregunta que es un Grado de Multimedia, es difícil de explicar ya que es un grado donde se tocan muchos palos, desde diseño hasta analítica web, pasando por programación, estándares Html, diseño de bases de datos, fotografía, audio y otros entre sí.

El Grado de Multimedia es un grado que da pinceladas de muchas cosas, pero realmente no se especializa en nada, es por esto que al explicar que lo mismo se estudia programación web que Fotografía o Narrativa, las reacciones no suelen ser positivas.

1.2 Elección de la propuesta

Durante el transcurso por el Grado de Multimedia, las asignaturas que más me han gustado han sido las relacionadas con la programación, Lenguajes y estándares web, Arquitectura de la información y Comportamiento de usuarios, el proyecto realizado me ayuda a reforzar estas áreas.

Las propuestas iniciales para mi trabajo de fin de grado fueron:

- Una aplicación para el nuevo AppleTv.
- Recoger datos de bases de datos publicas.
- Realizar a partir de un tema base “Underscore”, un tema para Wordpress.
- Realizar una aplicación web enfocada al Marketing online.

En un principio me hubiera gustado realizar la aplicación para el AppleTv, pero tuve que descartarla al comprobar que no había disponibilidad en ese momento y que seguramente sería difícil conseguir una unidad del nuevo AppleTv antes de finalizar el trabajo, con lo que cabría la posibilidad de que no se pudiera probar realmente el resultado final en el producto real.

Tanto recoger datos de otras bases de datos, como realizar un tema para Wordpress los descarté, porque me parecieron trabajos demasiado sencillos, de hecho hay una asignatura del grado multimedia “Sistemas de gestión de contenidos” donde ya se realiza un tema para Wordpress partiendo del tema base “Underscore”, por tanto estas dos opciones tienen poco mérito para mí, y no aportan nada nuevo.

1.3 Motivación

La propuesta seleccionada, realizar una aplicación web enfocada al Marketing Online, es un proyecto bastante ambicioso, que me va a permitir desarrollarme y mejorar mis competencias adquiridas durante el transcurso del grado, consolidando mis conocimientos como diseñador y programador al mismo tiempo, confirmando y reforzando todo lo aprendido en el Grado de Multimedia.

1.4 Interés general de la propuesta

El mundo del Marketing Online es muy interesante, y el disponer de una herramienta que pueda ser de ayuda en la estrategia de Marketing online planteada en una empresa creo que no tiene precio, el proyecto realizado permitirá disponer de una herramienta que nos indique de que se está hablando en Twitter en un determinado lugar, teniendo esta información se pueden ajustar las campañas de marketing y mejorar su rentabilidad.

2. DESCRIPCIÓN / DEFINICIÓN / HIPÓTESIS

2.1 ¿Que importancia tienen las redes sociales en el Marketing online?

Las redes sociales también conocidas como “Social media”, se han convertido durante los últimos años en un medio de comunicación de masas, revolucionando los tradicionales medios de comunicación.

Esta nueva forma de comunicación ha cambiado la forma de acceder a la información, mediante las redes sociales no es necesario esperar a poder ver las noticias de la televisión para estar informado, o a que salga el periódico, este nuevo medio de comunicación permite estar informado continuamente, además son los propios usuarios los que crean la información y contenidos que circulan por estos medios.

Cada día las redes sociales cobran más importancia en las estrategias de marketing online, pues aportan un tráfico importante a las webs, está demostrado y comprobado que las empresas con presencia en las redes sociales obtienen más tráfico, ingresos y beneficios que las que no están presentes.

2.2 ¿Que puede hacer Tellmetweet por las redes sociales?

La aplicación desarrollada en el presente proyecto, tiene una fase de recogida de datos, en la que recoge las tendencias por localizaciones y las almacena en una base de datos.

Conocer las tendencias por localizaciones puede ayudar a enfocar las campañas de marketing online realizadas en las redes sociales como Twitter, reduciendo su coste en publicidad e incrementando su número de conversiones.

Por ejemplo para mejorar los resultados de una campaña de TwitterAds (sistema de propaganda de Twitter), en España, se puede comprobar que a día de hoy es tendencia #eleccionescatalanas.

En la figura 1.1, se puede apreciar una lista de tendencias o “Trending Topics”, estas tendencias son temas populares sobre los que están hablando miles de usuarios en el momento actual.

Figura 1.1 Extraída de mi cuenta de Twitter

Por tanto conocer las últimas tendencias del país o localización al que va enfocado el negocio del cliente, puede ser muy interesante, ya que permite llegar a un mayor número de usuarios y conseguir generar más tráfico hacia la web del cliente.

Esta aplicación debería formar parte de la estrategia de marketing online de cualquier empresa que quiera aumentar el tráfico hacia su sitio web y conseguir un mayor número de clientes.

2.3 Tecnologías utilizadas

En este proyecto no se ha buscado reinventar la rueda, por ello hace uso de tecnologías que se usan habitualmente en las grandes compañías que se dedican a desarrollar software y simplifican la puesta en marcha o desarrollo, a continuación se detallan:

- Eclipse (Mars 1 Release 4.5.1): IDE de desarrollo.
- Java (Jdk1.7.0_79): Lenguaje de programación.
- Apache Tomcat (7.0.65): Servidor de aplicaciones.
- MySql (5.1.73): Base de datos.
- Apache Maven 3(.2.3): Herramienta para gestión y construcción del proyecto y dependencias.
- Spring (3.2.8): Framework que facilita el desarrollo generando los aspectos y la inyección de dependencias del proyecto.
- Spring Roo 2.0.0.M1: Herramienta para el desarrollo rápido de aplicaciones.
- Hibernate 4.3.6.Final: API de persistencia que permite tener a nivel de aplicación objetos(entidades) que representan la base de datos.
- JUnit: Framework para realizar pruebas.
- ExtJs (6.0.1): Framework javascript basado en el patrón MVC.
- Sencha cmd (6.0.1): Herramienta para optimizar ExtJs.
- GeoExt (3.0): Framework JavaScript que añade las funcionalidades de la librería OpenLayers, para crear mapas en una aplicación ExtJs.

3. OBJETIVOS

3.1 Objetivos principales

El objetivo principal es desarrollar una aplicación web, accesible desde internet mediante distintos dispositivos, en la que se puedan consultar las tendencias y aplicarlas en las estrategias de marketing online, para mejorar los ingresos de los negocio y aumentar su número de clientes.

3.1 Objetivos secundarios

A la aplicación podrán acceder usuarios sin registrarse que podrán acceder a algunas pantallas de la aplicación y usuarios registrados que podrán acceder a toda la aplicación y realizar cinco consultas diarias.

Los usuarios que paguen el plan “Premium” podrán realizar 20 consultas diarias.

La aplicación permitirá consultar tendencias en fechas anteriores. y ver durante cuantas horas ha sido Trending Topic un Tweet. Poder consultar tendencias actuales y anteriores nos permite evaluar el comportamiento de los usuarios, la empresa que sea capaz de analizar como evolucionan las tendencias podrá anticiparse al resto y obtener mejores resultados, pero para ello primero tendrá que poder obtenerlas y eso es lo que hace esta aplicación.

Poder conocer cuantas horas ha sido Trending Topic un Tweet, permite saber si el Community Manager de la empresa ha realizado correctamente su trabajo y si está obteniendo resultados o tiene que cambiar de estrategia.

4. MARCO TEÓRICO PRÁCTICO

4.1 Situación actual

Antiguamente las empresas necesitaban disponer de una red de comerciales en su plantilla si querían que sus productos se dieran a conocer y poder tener alguna oportunidad de venta en el mercado.

A partir del siglo XIX se introdujeron numerosas innovaciones tecnológicas que permitían que los contenidos viajaran velozmente alcanzando cada vez a más personas dando lugar a la comunicación de masas, que permiten crear relaciones e intercambios a nivel mundial, esto ha permitido que los nuevos medios de comunicación se introduzcan en las vidas cotidianas de las personas, formando parte se sus culturas y valores.

Con la llegada de internet y la comunicación de masas es posible hacer llegar la información y también los productos de una empresa a cualquier parte del mundo en un tiempo relativamente corto, por lo que ya no es imprescindible para las empresas disponer de una costosa red de comerciales, ahora es más económico y efectivo disponer de una buena estrategia de Marketing Online, para poder llegar a clientes de todo el mundo.

Las redes sociales, han revolucionado sin duda alguna la comunicación de masas y por ello son un punto clave en cualquier estrategia de marketing online.

Twitter es una de las redes sociales que más seguidores tiene, además dispone de un sistema de anuncios “Twitter Ads”, que permite a las empresas llegar hasta su público objetivo, por ello es fundamental disponer de una herramienta que permita conocer los trending topics de Twitter en un lugar concreto, ya que no tienen porque ser los mismos en Madrid o en Barcelona. Conocer los Trending Topics de un lugar permite optimizar la campaña en Twitter Ads de los negocios de una zona concreta.

4.2 Publico potencial

Cualquier persona que se dedique al mundo del marketing online, necesita una herramienta como la presentada en el presente proyecto, así mismo también es ideal para cualquier estudiante de marketing, personas que se dedican al SEO, Community managers, incluso webmaster o los propios dueños de su empresa que necesitan saber de que se está hablando en las redes sociales.

4.3 La competencia

Después de buscar en internet sobre herramientas similares, esto es lo que he encontrado:

Nombre aplicación	Url	Comentarios
Twitter	**twitter.com**	Permite ver los 10 últimos trending topics de la zona del usuario, es la fuente más fiable por ser la original.
Twirus	**es.twirus.com**	Su filtro no funciona bien, a veces siempre sale lo mismo, y si se busca una fecha anterior solo muestra artículos. “los, por..”
Trends24	**trends24.in**	Permite ver los twitters cada hora, los de España no coinciden con los de Twirus España.
Trendsmap	**trendsmap.com**	Algo confuso, ya que se mezclan en el mapa, los mostrados en Valencia no coinciden con los visualizados en mi cuenta.
Hashtags	**hashtags.org**	No especifica de donde son ni cuando se han obtenido.
Talltweets	**talltweets.com**	A veces no funciona y da error 404, los trending topics no coinciden con los visualizados en mi cuenta.

Lo primero que se ha podido comprobar, es que son pocas las herramientas que hay disponibles en el mercado actual que permiten conocer los trending topics de un lugar.

Lo segundo es que al contrastar los datos que ofrecen dichas herramientas con los que ofrece directamente una cuenta de Twitter, algunos trending topics coinciden, pero no lo hacen todos, por tanto no se puede decir que estas herramientas ofrezcan datos fiables y si no son fiables pueden llegar a arruinar la estrategia de Marketing de una empresa.

Lo tercero es que algunas herramientas como “Talltweets” solo permiten consultar una ciudad de un país, en el caso de España solo permite consultar Madrid, esto puede llegar a ser poco útil si el mercado es el Español, ya que si se quiere llevar a cabo una estrategia de marketing online de una empresa situada en Valencia, de muy poco servirá la herramienta, y lo mismo ocurre para el resto de países.

Por todo esto la competencia, no es realmente competencia, ya que ofrece información imprecisa e incluso poco útil, además de que algunas herramientas son confusas tanto en la presentación de sus datos como en su manejo.

5. CONTENIDOS

5.1 Generales

* Pantalla de inicio o Home:
    - Muestra los últimos Trend Topics de algunas ciudades de las disponibles por la aplicación.
    - Muestra la información del usuario registrado
    - Muestra un botón para registrarse en la aplicación, si el usuario no está conectado.
    - Muestra un botón para comprar el plan “Premium”, en caso de estar conectado.
    - Muestra un botón para hacer Login, si el usuario no está conectado.
    - Muestra un botón para desconectarse, en caso de estar conectado.

5.2 Otras opciones

* Pantalla Trending Topics:
    - Permite realizar consultas sobre los “Trends" almacenados en la base de datos aplicando filtros, por localización y fecha, solamente disponible para usuarios registrados.

* Pantalla Map:
    - Muestra un mapa geolocalizado que ubica las principales localizaciones de Twitter.

* Pantalla Chart:
    - Muestra una gráfica de barras, en la que se puede apreciar el número de horas tanto por día, como por semana y mes que ha sido Trending Topic un Tweet, solamente disponible para usuarios registrados.

- Popup Register:
  - Muestra el formulario de registro de un nuevo usuario.

- Popup Premium:
  - Muestra el formulario para comprar el plan “Premium”.

- Popup User Locations
  - Muestra el popup para cambiar las localizaciones favoritas del usuario, que son las que aparecen en la Home.

- Popup Login:
  - Muestra el formulario de login, para acceder a la aplicación como usuario registrado.

6. METODOLOGÍA

Para este proyecto se ha optado por el método de trabajo Scrum, que permite dividir las fases grandes del proyecto en fases más pequeñas y realizar entregas parciales en lugar de entregas completas que prolongan los tiempos y hacen más difícil detectar los errores.

Realizando entregas parciales es más fácil comprobar que se entrega lo que se había pedido o definido, y en caso de tener que realizar correcciones, es menos costoso hacerlas.

De esta forma con las primeras entregas, aunque no se hayan implementado todas las funcionalidades de la aplicación y no se haya finalizado la maquetación, tanto el consultor como el desarrollador podrán comprobar si lo entregado se adapta a lo acordado, si se detectaran errores o hubiera que modificar cosas, es mejor hacerlo ahora que cuando la aplicación ya está finalizada, es más fácil reparar los pequeños errores que de no hacerlo ahora podrían derivar en graves errores en la aplicación final.

Además cada entrega parcial, puede producir un feedback que puede tener repercusión en las siguientes entregas. Es muy habitual en cualquier proyecto que el cliente cambie de opinión, y utilizar un método de trabajo Scrum permite adaptarnos al cliente rápidamente.

7. ARQUITECTURA DE LA APLICACIÓN

7.1 Cliente

El cliente puede ser un navegador o dispositivo móvil, en esta capa se realiza el procesamiento de cliente y es externa al sistema.

El cliente accede mediante un navegador, el navegador realiza llamadas Rest a los servicios de la aplicación, dichas llamadas reciben por respuesta un Json que interpreta el Front de la aplicación y con los datos que trae dicho Json renderiza la información en pantalla.

Para la parte Front se ha empleado el framework Javascript **ExtJs**, la ventaja de usar javascript en la capa de la vista, es que el código javascript se ejecuta en el equipo del cliente, por tanto si la aplicación produce una excepción lo hace en el cliente y no en el servidor como lo hacen otras tecnologías como JSF que se ejecutan en el servidor y lo dejan fuera de servicio. Otra ventaja de usar javascript es que si aumenta el número de usuarios considerablemente, como este se ejecuta en el equipo del cliente no es necesario realizar inversiones astronómicas en servidores para poder atender la carga de usuarios, pues lo que se carga es el equipo del cliente y no el servidor.

7.2 Servidor

El servidor final se alojará en un servidor virtual (VPS) de la empresa OVH(www.ovh.es), su sistema operativo será Centos 6 a 64bits, en el servidor se instalará la máquina virtual de java, el servidor de aplicaciones Tomcat y la base de datos MySql.

7.3 Base de datos

Como base de datos se usará la última versión de MySql disponible en el momento de instalación del servidor.

7.4 Aplicación

La arquitectura de la aplicación, está basada en el patrón MVC (Modelo - Vista - Controlador), este patrón separa el código en tres capas para desacoplar el código y que varios perfiles puedan trabajar en capas distintas al mismo tiempo, permitiendo trabajar al diseñador en la vista a la vez que el programador trabaja en la capa Controlador.

- El modelo representa la información almacenada en la base de datos, se encarga de acceder a los datos y trabajar con ellos.

- El controlador responde a las acciones o peticiones realizadas por el usuario se comunica con el modelo para obtener la información y decide que vista devuelve para pasarle los datos.

- La vista se encarga de renderizar el código encargado de generar la visualización de los datos, normalmente HTML, PHP,,,etc.

En la aplicación además se ha incluido una capa de negocio (Business Object o BO), esta capa se encarga de realizar todas las validaciones necesarias.

También se ha incluido una capa DTO (Data Transfer Object), los DTO son los objetos que devuelven los servicios al front, estos objetos se devuelven en formato JSON que es el lenguaje que entienden las capas de visualización de datos (Vista). Resumiendo los DTO son objetos que transportan o transfieren datos entre capas.

Además se ha incluido un paquete Exception, que contiene las clases que ayudan a propagar las excepciones por la aplicación.

Y el paquete Utils, que contiene clases de ayuda y utilidades en la aplicación.

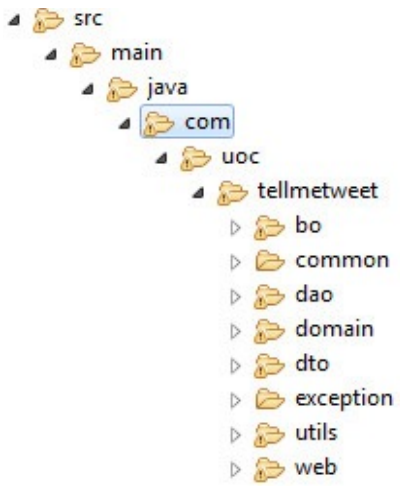


Figura 7.1

Explicación de la estructura del backend de la aplicación:

- web: Contiene los controladores de los servicios, reciben peticiones Rest y llaman a una clase de la lógica de negocio (paquete bo - Business Object)
- bo: realiza todas las comprobaciones necesarias y finalmente accede a una clase del paquete dao (Data Access Object) mediante su interfaz
- dao: Este paquete contiene los DAO y sus interfaces, solamente contiene los métodos que se encargan de realizar las operaciones CRUD con la base de datos.
- domain: Contiene la representación de las tablas de la base de datos, este ha sido generado automáticamente por Spring Roo, por tanto en caso de tener que modificar alguna tabla es mejor modificar el archivo “dbre.xml” (DataBase Reverse Engineer). Si está funcionando Spring roo, y se realiza una modificación en el dbre.xml, al guardar los cambios, Roo actualiza automáticamente las clases del dominio afectadas.
- dto: (Data Transfer Object) la capa “bo” recibe objetos de dominio de los DAO y los transforma en DTO, los objetos DTO se devuelven a la capa web transformados en JSon.
- common: contiene clases genéricas que implementan los DTO para no repetir los mismos métodos en todos los DAO, como los de las operaciones CRUD, también

contiene las constantes de la aplicación, usando constantes, cualquier parámetro que se utilice en la aplicación será referenciado mediante su correspondiente constante, de forma que si hay que modificar algo se modificará en el archivo de constantes y será fácil localizarlo y realizar dicha modificación.

* exception: contiene excepciones con métodos generales que propagan las excepciones de la aplicación, con la ayuda de estas clases un error de la base datos puede llegar a ser mostrado en la vista.
* utils: clases de utilidades en la aplicación, como conversores de fechas,,,etc

Emplear una arquitectura separada en capas como esta, permite desacoplar el código, de forma que la lógica de negocio no tiene porque saber si los DAO se comunican con la base de datos mediante JDBC o señales de humo, no le importa. De la misma forma la capa de presentación no tiene porque saber nada de la lógica de negocio ni de los DAO.

La forma en que la vista obtiene la información es a través de servicios Rest, la vista realiza peticiones que recibe el backend y este le devuelve objetos Json. Esto quiere decir que el Backend es totalmente independiente del Frontend.

Esta aplicación se ha desarrollado empleando ExtJs como Framework para la capa Frontend, pero si el día de mañana se quiere sustituir este por otro framework de presentación u otro lenguaje como PHP, no se tendrá que tocar ni una sola línea de código de la parte backend, solamente habrá que programar el nuevo Frontend, llamar a los servicios rest del backend y mostrar los Json recibidos en pantalla.

7.5 Tecnologías

A continuación de muestra un listado de los productos o tecnologías empleados.

Aplicación	Versión
Eclipse	Mars.1 Release (4.5.1)
Spring	3.2.8
Spring Roo	2.0.0.M1
Apache Maven	3.2.3
Apache Tomcat	7.0.65
Java	Jdk 1.7.0_79
MySql	5.1.73
Hibernate	4.3.6.Final
ExtJs	6.0.1
Sancha Cmd	6.0.1
GeoExt	3.0

8. PLATAFORMA DE DESARROLLO

8.1 Hardware

Características del equipo usado durante el desarrollo:

Componente	Descripción
CPU	Intel Core I5-2310 2,90GHz
Ram	8,00 GB
Sistema operativo	Windows 7 Professional, Service Pack 1

8.2 Software

Listado de software y versiones empleadas en el desarrollo:

Aplicación	Versión
Eclipse	Mars.1 Release (4.5.1)
Spring	3.2.8
Spring Roo	2.0.0.M1
Apache Maven	3.2.3
Apache Tomcat	7.0.65
Java	Jdk 1.7.0_79
MySql	5.1.73
Hibernate	4.3.6.Final
ExtJs	6.0.1
Sancha Cmd	6.0.1
GeoExt	3.0

8.3 Objetivo del entorno de desarrollo

La principal peculiaridad del entorno empleado es que una vez instalado se convierte en un entorno portable, lo cual permite llevarlo a otro ordenador simplemente haciendo una copia de la carpeta que lo contiene.

Es muy frustrante cuando no se tiene la suficiente experiencia y se intenta poner en marcha una aplicación web Java en un entorno de desarrollo como Eclipse, al principio pueden aparecer errores de que falta el runtime de java, o no se puede compilar porque falta una librería que no puede descargar Maven, bien porque no está correctamente configurado o no se han definido las variables de entorno.

La mayoría de estos problemas se resuelven estableciendo correctamente las variables de entorno y el path en el administrador del sistema, pero esto es algo que se desconoce cuando se empieza en el mundo de la programación y no es habitual saber hacerlo, y por si fuera poco a veces también aparecen conflictos con las versiones de Java que están instaladas en el sistema.

Si por ejemplo un instituto o universidad, quiere dar una asignatura en le que se enseñe a desarrollar aplicaciones web java, normalmente empleará un entorno de desarrollo como Netbeans, porque este ya viene preconfigurado, pero si se quiere enseñar a programar en Android se tendrá que emplear un entorno de desarrollo como Eclipse, además lo más lógico es que si en el mundo empresarial y sobre todo en las grandes empresas se trabaja con Eclipse, el curso se desarrolle en Eclipse para que así el alumno esté familiarizado con el entorno y esto le ayude en su inserción laboral.

El principal problema que puede tener una universidad que imparta un curso online con Eclipse, es que probablemente gran parte de sus alumnos pierdan la primera y puede que incluso la segunda semana intentando montar el entorno de desarrollo, volviéndose locos porque no configuran correctamente las variables de entorno de las aplicaciones o tecnologías empleadas.

Mediante el sistema empleado en este proyecto, todos estos problemas desaparecen, pues bastaría con colgar en los materiales del aula un zip con la carpeta que contiene el

entorno de desarrollo, el alumno solo tendría que descomprimirlo en su ordenador y empezar a programar.

8.4 Descripción del entorno de desarrollo

El entorno contiene tres carpetas y dos scripts.

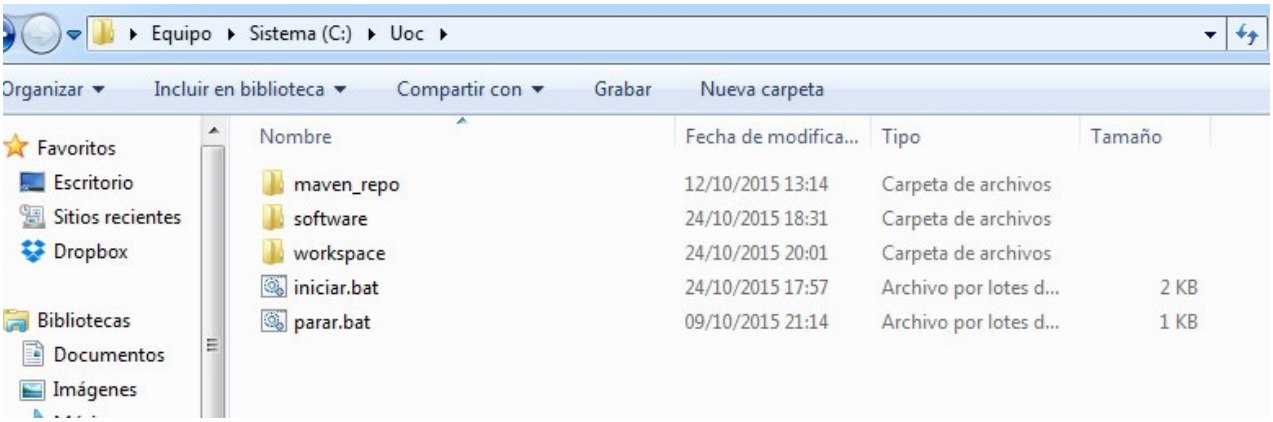


Figura 8.1

- Carpetas:
  - maven-repo: Esta carpeta almacenará todas las dependencias de la aplicación que descargue Maven.
  - software: Contiene todo el software necesario para el desarrollo.
  - workspace: Es la carpeta donde se almacenan los proyectos cada uno en su correspondiente carpeta, en este caso solamente habrá uno proyecto y por tanto una carpeta dentro “tellmetweet”.
- Scripts:
  - iniciar.bat: Ejecutable que inicia el entorno de desarrollo.
  - parar.bat : ejecutable que para el entorno de desarrollo.

8.5 Funcionamiento del entorno de desarrollo

Para iniciar el entorno de desarrollo, hay que abrir una consola de windows, podemos hacerlo o bien a través del menú de Inicio en “Accesorios - Símbolo del sistema” o mediante el buscador de programas escribiendo “cmd”

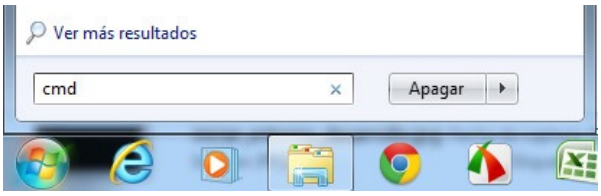


Figura 8.2

Nos situamos en la carpeta donde se encuentra el entorno, escribimos “iniciar” y pulsamos la tecla “Enter”

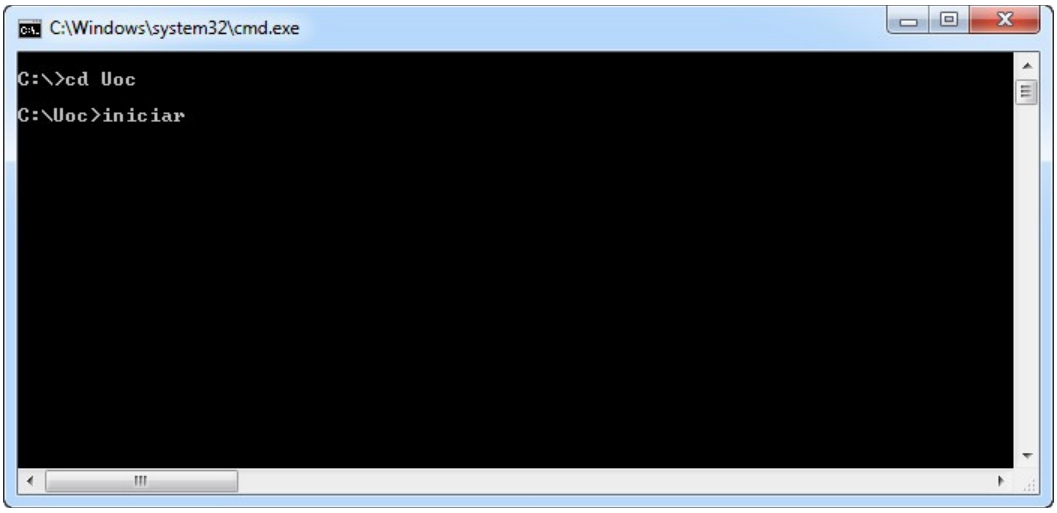


Figura 8.3

Al iniciar el entorno, se crea una nueva unidad en nuestro equipo, por defecto crea la unidad “M”, pero esta se puede cambiar por otra en el archivo “iniciar.bat”, se configuran las variables de entorno, en realidad todo el software necesario esta ubicado en la carpeta “software” del entorno y lo que hace el script es apuntar todas las variables de entorno a su software correspondiente, por lo que no hay que configurar variables del sistema, ni machacar ninguna instalación de Java, incluso se puede descomprimir una

nueva versión de la máquina virtual de Java y en el script hacer que apunte a esta, finalmente iniciará el ide “Eclipse”, y se situará en el workspace del proyecto.

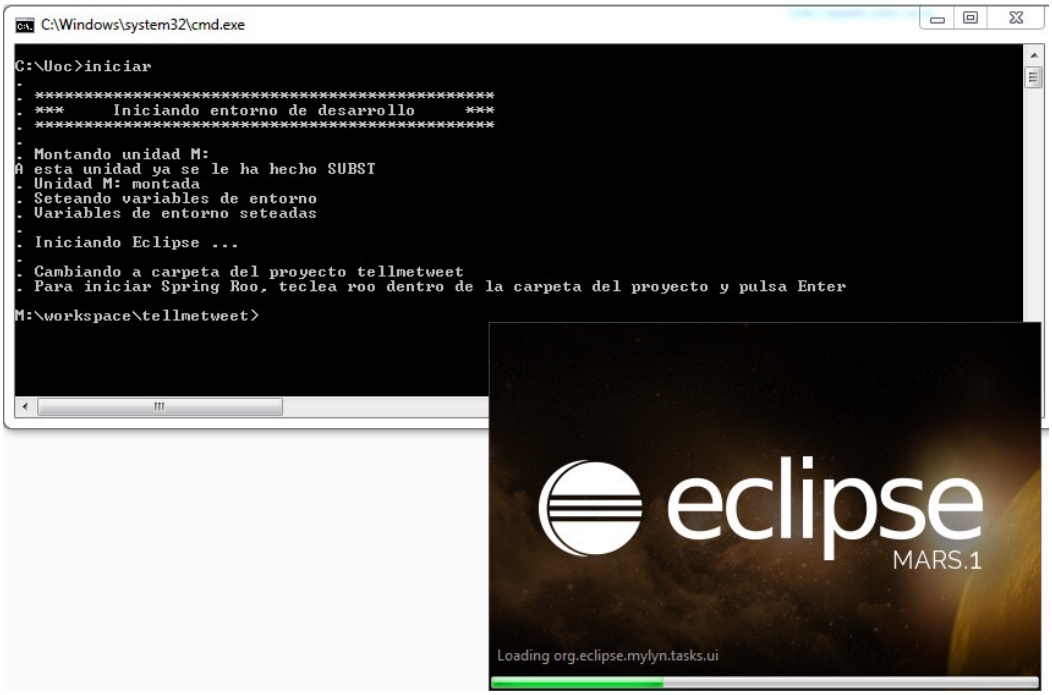


Figura 8.4

Una vez en la carpeta del proyecto, se puede teclear “roo” para iniciar Spring Roo.

Cuando se termine bastará con escribir “parar” y se desmontará la unidad “M”

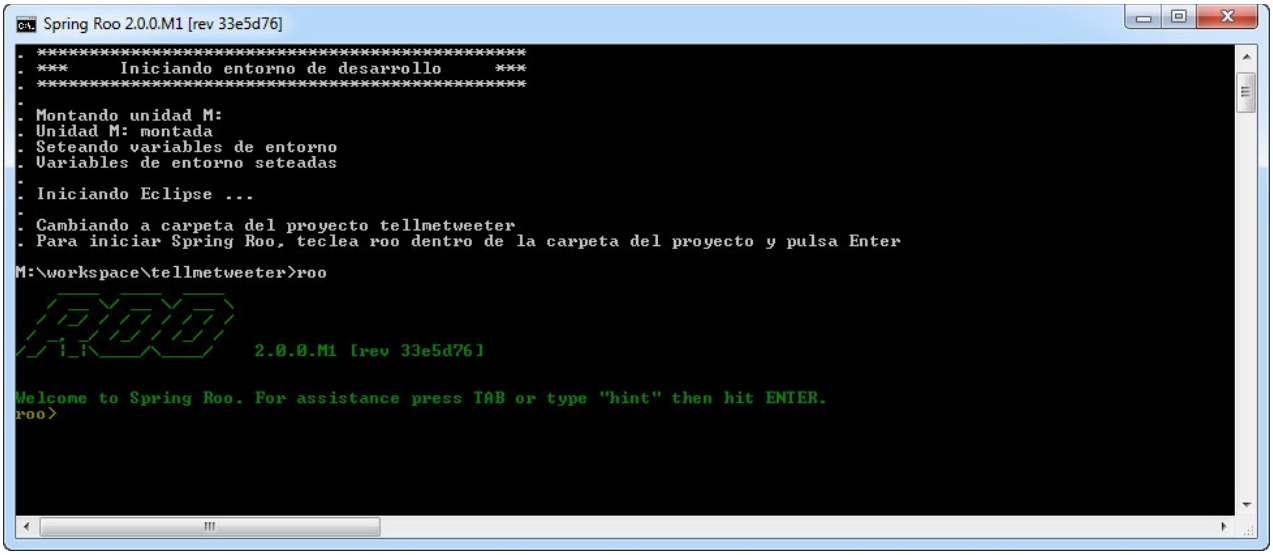


Figura 8.5

9. PLANIFICACIÓN

Nombre	Duración	Inicio	Fin
BASE DE DATOS	5 días	16/09/2015	22/09/2015
Diseño de base de datos	3 días	16/09/2015	18/09/2015
Creación de base de datos, carga inicial	2 días	21/09/2015	22/09/2015
SERVIDOR	33 días	16/09/2015	30/10/2015
Instalación en máquina virtual de pruebas	5 días	16/09/2015	22/09/2015
Carga de datos	1 día	23/09/2015	23/09/2015
Contratar servidor VPS en OVH	3 días	23/10/2015	27/10/2015
Instalar VPS	2 días	28/10/2015	29/10/2015
Desplegar aplicación en VPS	1 día	30/10/2015	30/10/2015
APLICACIÓN	33 días	23/09/2015	06/11/2015
Creación tokens seguridad Twitter	1 día	23/09/2015	23/09/2015
Puesta en marcha entorno de desarrollo	6 días	24/09/2015	01/10/2015
Creación parte Back, esqueleto inicial	3 días	02/10/2015	06/10/2015
Creación parte Front, Extjs	4 días	07/10/2015	12/10/2015
Servicio para almacenar Trends en BD	3 días	13/10/15	15/10/15
Mostrar últimos Trends en Front	3 días	16/10/2015	20/10/2015
Desplegar aplicación en servidor máquina virtual	1 día	21/10/2015	21/10/2015
Registro usuarios	10 días	30/10/2015	12/11/2015
Gestión de permisos	5 días	13/11/15	10/11/2015
Login y Logout	2 días	20/11/2015	23/11/2015
Integración Paypal	14 días	24/11/2015	11/12/2015

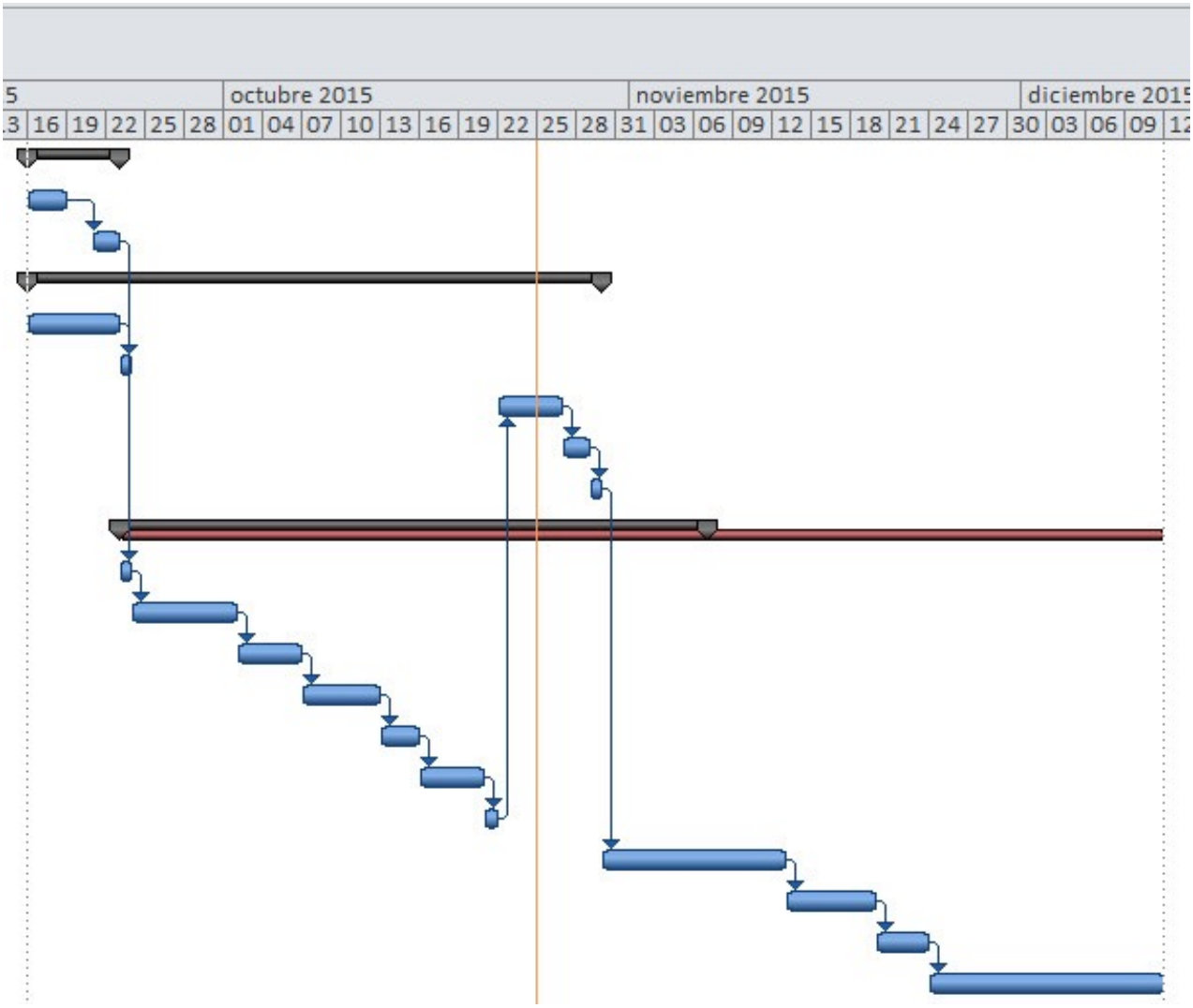

5
octubre 2015
noviembre 2015
diciembre 2015
3 16 19 22 25 28 01 04 07 10 13 16 19 22 25 28 31 03 06 09 12 15 18 21 24 27 30 03 06 09 12

Figura 9.1

10. PROCESOS DE TRABAJO / DESARROLLO

10.1 Definición del proyecto

Durante esta fase, se realiza un brainstorming de todo lo que puede y lo que no puede abarcar el proyecto, para intentar realizar una correcta definición y planificación del proyecto. Se estudian las posibles tecnologías a emplear.

10.2 Puesta en marcha del entorno de desarrollo

Una vez definido el proyecto y elegidas las tecnologías a emplear en el, se pone en marcha el entorno de desarrollo del proyecto, a su vez se realizan capturas de pantalla de las pasos de los procesos más importantes y se documentan dichos procesos.

10.3 Puesta en marcha del servidor de pruebas

En esta fase se monta un servidor Linux, distribución Centos 6 64 bits, en una máquina virtual, esta alojará la base de datos y el servidor de aplicaciones, la idea es poder desplegar en el una versión prematura de la aplicación para comprobar que todo funciona correctamente en un servidor y es viable seguir adelante con el proyecto, si se supera con éxito esta fase, se replicará en un servidor VPS, que se contratará en la empresa OVH.

10.4 Creación y puesta en marcha de la base de datos

En esta fase se realizan los scripts para crear las tablas de la base de datos y la carga de los datos iniciales, se comprueba que se puede acceder desde un cliente MySql y realizar consultas a dicha base de datos.

10.5 Puesta en marcha de la aplicación

En esta fase, se crea el esqueleto de la aplicación, con sus distintas capas, se crea por un lado el Backend haciendo uso de la herramienta “Spring Roo”, y por otro lado se crea el Frontend mediante la herramienta “Sencha Cmd”, ambos son totalmente independientes el uno del otro, se comprueba que se puede llamar a los servicios del Backend desde un navegador.

10.6 Despliegue de la aplicación

En esta fase se crean los servicios necesarios para guardar los últimos Trending Topics, de las ciudades activas en la aplicación, una vez comprobado que se puede llamar al servicio desde un navegador mediante una petición Rest, se despliega la aplicación en el servidor de pruebas y se comprueba que al realizar la llamada a través del navegador se almacenan los datos correctamente en la base de datos. Por último se crea un script bash en el servidor que llamará al servicio Rest encargado de almacenar los datos cada hora.

10.7 Primeras pantallas del frontend

Se modifican la páginas iniciales del frontend, para realizar un layout que incluye un Header, Footer, Sidebar y Centro, se muestra en el centro los últimos Trending Topics almacenados.

10.8 Replica en el VPS

Llegados a esta fase, el proyecto es viable, ha funcionado la versión inicial, por tanto se contrata el servidor VPS y se replica toda la faena realizada en el servidor montado en la máquina virtual, el principal motivo es evitar que un problema con mi máquina o con mi proveedor de internet pueda dejar el proyecto inaccesible en cualquier momento.

10.9 Entregas

En esta fase se continuará adelante con el proyecto, y con la programación y maquetación y al mismo tiempo se irá completando la documentación del proyecto para poder realizar las distintas entregas del proyecto:

- Pac 1
- Pac 2
- Pac 3
- Entrega final

11. APIS UTILIZADAS

11.1 Twitter4J

Twitter4J es una librería Java no oficial, esta librería es la que se encarga de conectar con Twitter y realizar todas las operaciones necesarias para devolver los datos o información relacionada con los procesos de extracción de datos de la red social.

Esta librería no tiene más dependencias, y hace uso del último protocolo de autenticación de Twitter OAuth, además está en el repositorio de Maven, por lo que su instalación se limita a copiar cinco líneas en el pom.xml de la aplicación.

```
<dependency>
    <groupId>org.twitter4j</groupId>
    <artifactId>twitter4j-core</artifactId>
    <version>[4.0,)</version>
</dependency>
```

La librería dispone de ejemplos y documentación en su página web, y dispone de muchas más funciones que la Api original de Twitter, estas funciones adicionales son las que me van a permitir buscar las distintas localizaciones de Twitter y extraer sus Trending Topics.

11.2 Paypal SDK

Se encarga de conectar con Paypal para realizar el cobro de la cuenta “Premium”, mediante tarjeta de crédito o débito, puede funcionar en modo SandBox y en modo normal.

El modo Sandbox es igual que el normal, con la diferencia de que en este modo no se realizan los cobros realmente, se realiza todo el proceso de conexión, autenticación y validación, pero no se descuenta dinero real en las transacciones, cambiar de este modo al normal es tan sencillo como modificar una variable.

Al igual que en Twitter4J, para instalarla basta con añadir las siguientes líneas al pom.xml de la aplicación.

```
<dependency>
     <groupId>com.paypal.sdk</groupId>
     <artifactId>rest-api-sdk</artifactId>
     <version>1.4.1</version>
</dependency>
```

12. DIAGRAMAS UML

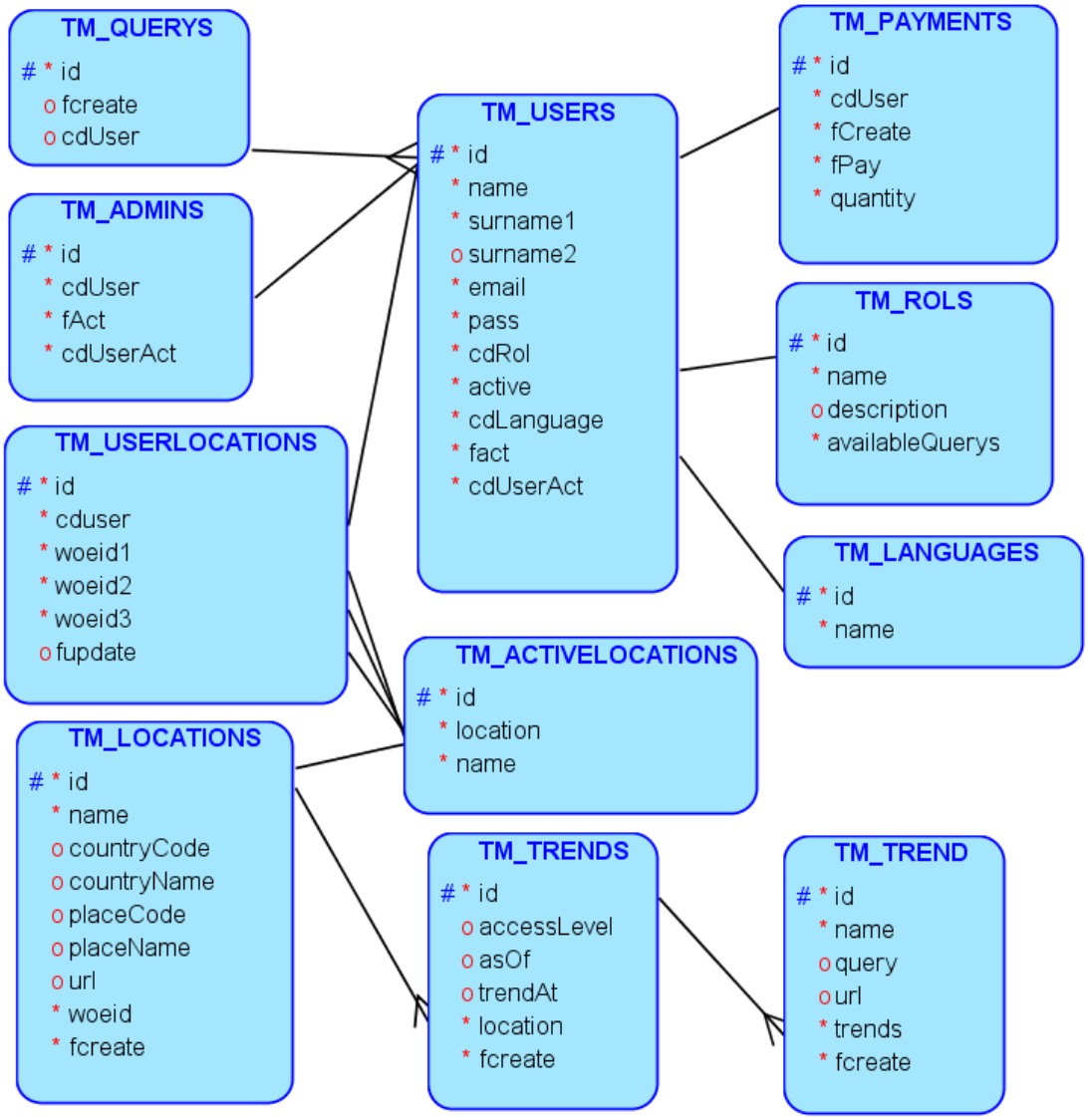


Figura 12.1

13. PROTOTIPOS

Durante la fase de prototipo inicial se han identificado cinco pantallas iniciales, las cuales se muestran a continuación.

13.1 Home

Esta pantalla, muestra, la cabecera que contiene el logo, las opciones del menú principal y los botones de Registro y Login, debajo de esta se muestran tres paneles que muestran información de algunas localizaciones. En el encabezado se muestra el nombre de la ciudad y debajo se listan los últimos Trending Topics almacenados en la base de datos. Por último en el footer se muestra la fecha de la última actualización de la base de datos.

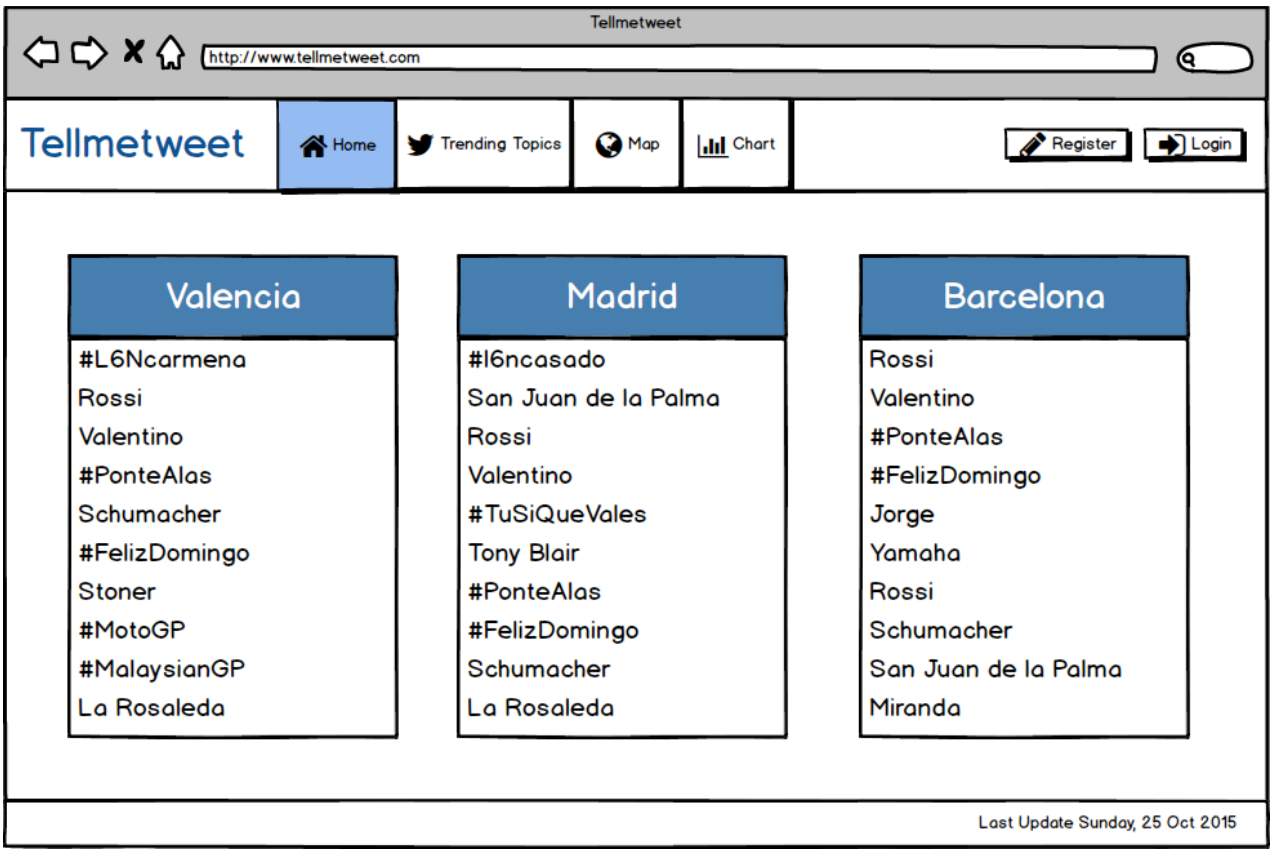


Figura 13.1

13.2 Trends

Esta pantalla muestra un Grid con los Trending topics almacenados en la base de datos, tiene un apartado de filtros que permite filtrar las búsquedas por localizaciones y fechas para afinar los resultados.

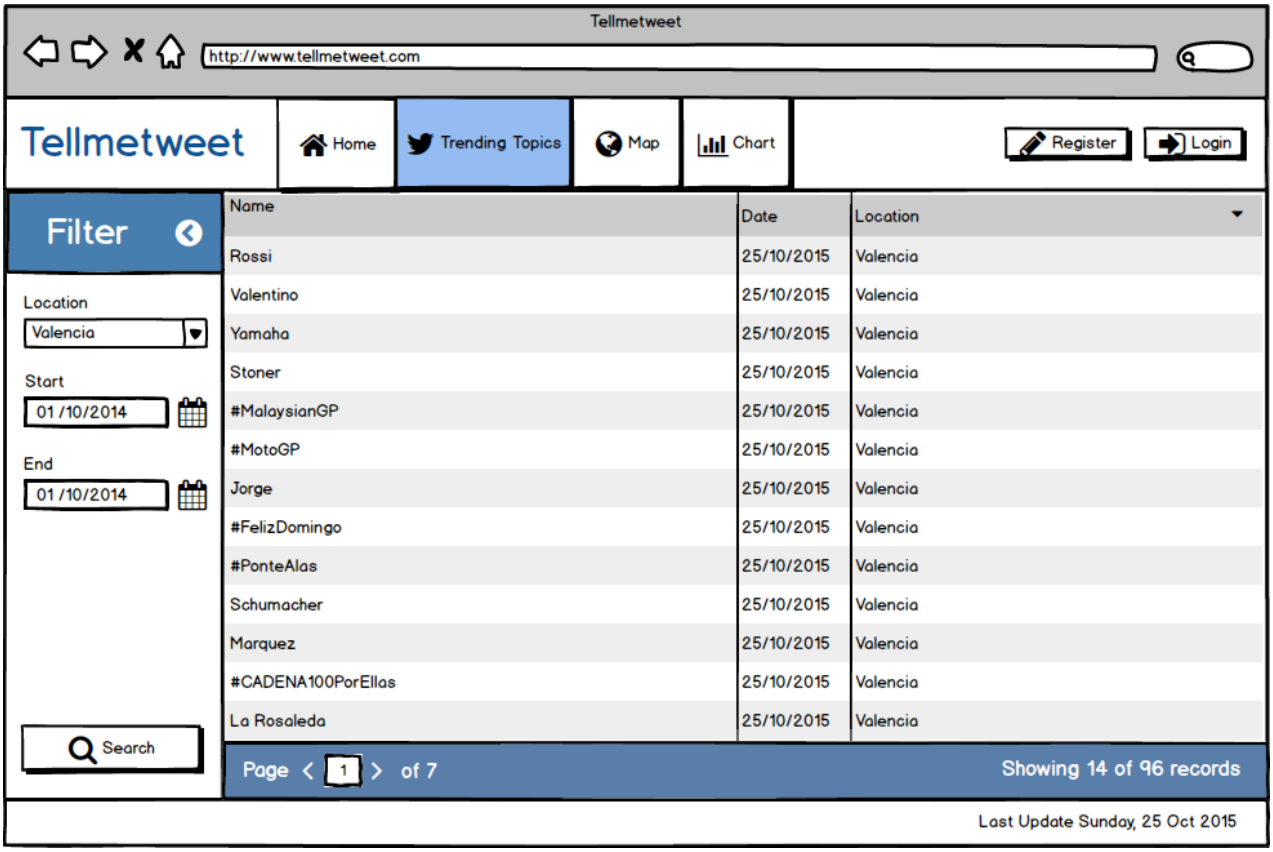


Figura 13.2

13.3 Map

Esta pantalla muestra un mapa, en el que se geolocalizan las locaciones de Twitter y sus Trending topics.

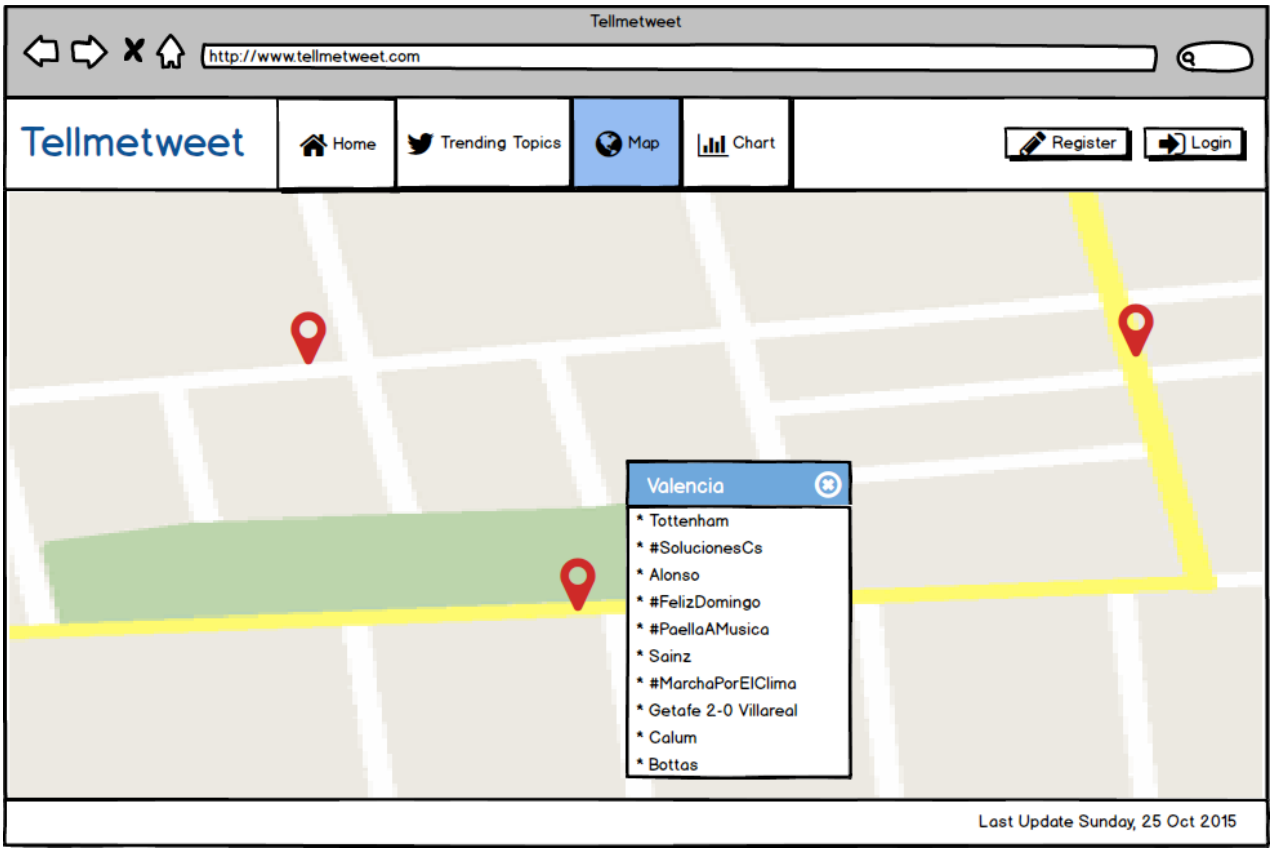


Figura 13.3

13.4 Chart

Esta pantalla muestra una gráfica de barras, en la que se puede apreciar para el día seleccionado en el filtro, cuantas horas ha sido Trending Topic un tweet, también nos muestra los horas por semana y por mes.

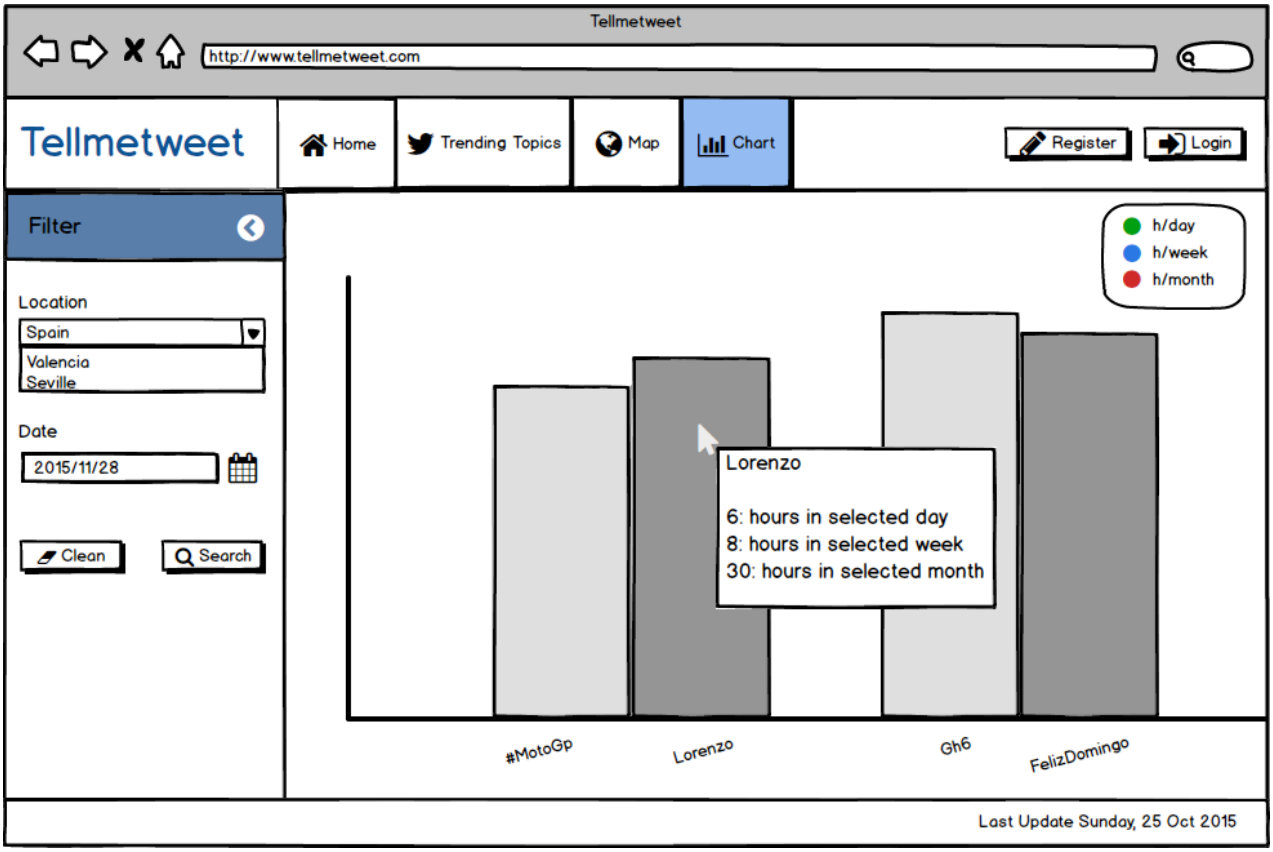


Figura 13.4

13.5 Register

Esta pantalla muestra el formulario de registro para nuevos usuarios.

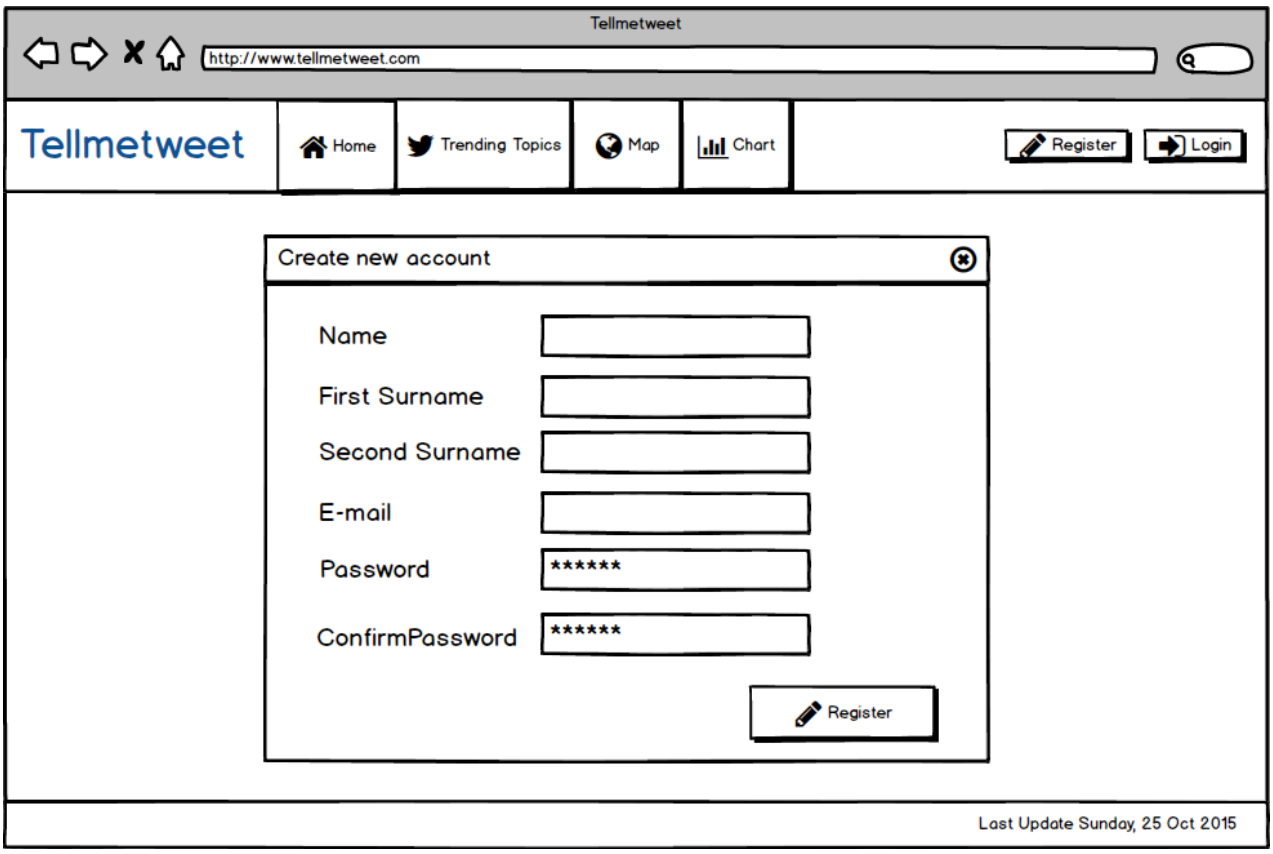


Figura 13.5

13.6 Login

Esta pantalla muestra el formulario de login, para acceder a la aplicación como usuario registrado.

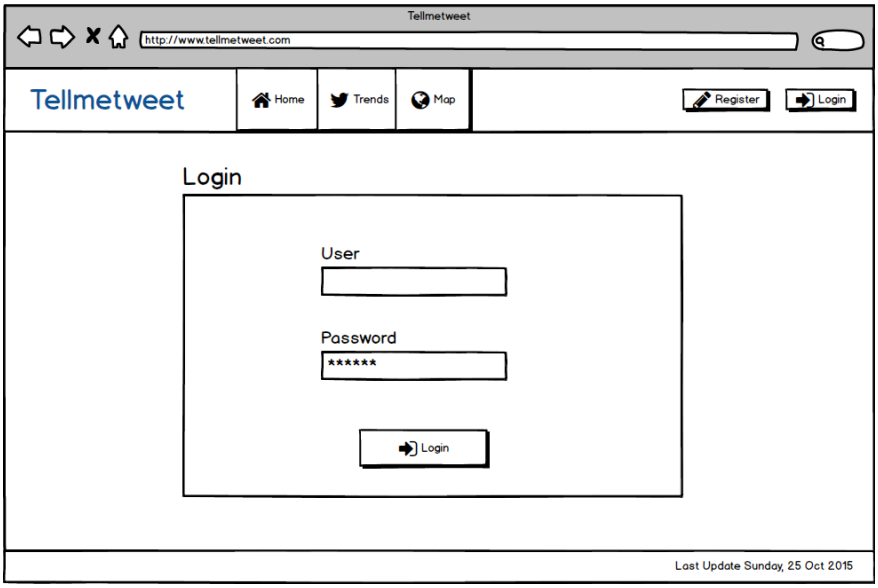


Figura 13.6

15. PERFILES DE USUARIOS

La aplicación desarrollada en el presente trabajo de fin de grado, está orientada hacia usuarios de entre 20 y 60 años, que hacen un uso habitual de internet y sienten un mínimo de interés hacia el marketing online y las redes sociales, tanto sean hombres como mujeres.

El usuario de esta aplicación puede ser un estudiante de marketing online, así como un usuario que se encuentra realizando un curso de Community Manager, también lo es el usuario que trabaja como asesor de marketing online de una empresa, o un antiguo comercial que quiere reciclarse e implantar sus estrategias a través de internet.

Los usuarios de esta herramienta, podrán comprobar como obtienen datos reales y fiables, que les ayudarán a optimizar sus campañas en Twitter Ads y a mejorar sus estrategias de marketing online, creando con ello un vinculo de confianza con la herramienta y convirtiéndola en imprescindible para su actividad diaria.

16. USABILIDAD/UX

La usabilidad de una aplicación es la facilidad con que el usuario puede realizar ciertas acciones o utilizar dicha herramienta, es importante centrarse en este concepto desde el principio de definición y desarrollo de la aplicación, pues si la aplicación no es usable, es muy probable que el usuario se canse y abandone la herramienta.

Para facilitar la usabilidad de la aplicación, esta será responsive, permitiendo adaptarse a distintas resoluciones y distintos dispositivos, los colores de la aplicación mantendrán un alto contraste, y las tipografías serán clara y grandes para que el usuario pueda leer la información con facilidad.

Los botones de la aplicación, serán lo suficientemente grandes para que el usuario pueda interactuar con ellos, tanto en la versión de ordenador como si accede a través de un dispositivo móvil, además incorporarán iconos para facilitar la interpretación de su significado, se añadirán tooltips que indicarán que ocurrirá si el usuario hace click en el botón, esto hará que el usuario sepa que es lo que va a suceder cuando haga uso del botón y no se sienta desconcertado y abandone la aplicación.

17. SEGURIDAD

Para garantizar la máxima disponibilidad de la aplicación desarrollada en el trabajo de fin de grado, esta estará desplegada en un servidor VPS contratado en la empresa OVH, desde el momento del primer despliegue hasta que finalice el trabajo de fin de grado.

Durante el tiempo que la aplicación esté desplegada en el servidor de OVH, esta estará expuesta a un gran número de hackers que no dudarán en buscar vulnerabilidades en el sistema para intentar introducirse en el.

A nivel de aplicación existen dos partes a revisar, el backend y el fronted, a nivel de servidor hay que establecer las medidas mínimas para disuadir a los hackers.

17.1 Servidor

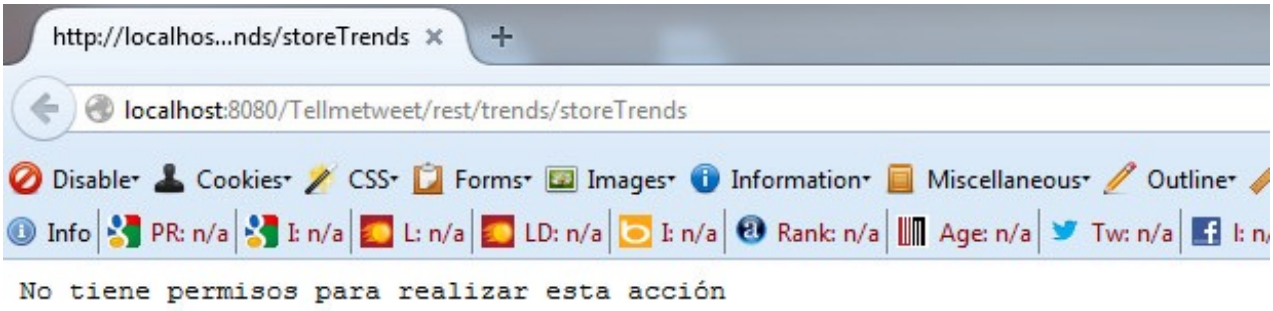
La primera medida a implementar es cambiar el puerto estándar 22(SSH), los hackers suelen rastrear la red en busca de máquinas que responden al puerto 22, cambiando este por otro, la máquina no aparecerá en sus herramientas de escaneo y se cebaran con otras máquinas que no lo han cambiado.

La segunda medida es activar el firewall, de forma que hasta día de hoy solo es posible acceder a los puertos 3306 (MySql), y 8080 (Servidor de aplicaciones) desde la propia máquina y desde mi propio ordenador, se ha abierto en el firewall únicamente el puerto 80, para poder acceder a la aplicación desde el exterior.

17.2 Backend

Se ha implementado un token de seguridad en los servicios del backend, además de la llamada rest, el servicio tiene que recibir un token de seguridad, si el token no coincide, devuelve un mensaje de error.

Figura 17.1



Además hay que comprobar que cuando se envían parámetros en las llamadas a los servicios del backend, estos parámetros no hayan sido modificados.

Las contraseñas de los usuarios se almacenan cifradas en la base de datos codificadas en MD5 128 bits almacenando en su lugar un número hexadecimal de 32 dígitos, cuando un usuario se registra, su contraseña se cifra en y se almacena en la base de datos, cuando el usuario intenta entrar en la aplicación, se envían al servidor sus credenciales, el servidor cifra la contraseña enviada y la compara con la que tiene almacenada que también está cifrada.

17.1 Fronted

En el fronted se controla la visibilidad de la aplicación en función del rol del usuario que está accediendo a la aplicación, puede que hayan acciones que un un usuario con un rol no pueda realizar, o puede que incluso algún componente se muestre no habilitado o simplemente no se muestre para un determinado rol, en cambio puede que si que lo haga para otro rol distinto, todas estas situaciones están controladas.

También se controla que el usuario no pueda hacer trampas colocando parámetros en las url de navegación, se comprueban los permisos en función del rol del usuario en todas las acciones y pantallas.

18. TESTS

18.1 Aplicación

Spring Roo al igual que ha generado la aplicación, ha generado una capa de test, que permite probar los servicios creados por el, esta parte no se ha desarrollado debido al poco tiempo disponible para el desarrollo de la aplicación.

18.2 Usuarios

Los test con usuarios, permiten comprobar el nivel de usabilidad de la aplicación, durante estos tests, se le pide al usuario que realice una serie de acciones, y se toman notas de lo que el usuario hace o dice, también puede grabarse la sesión, la persona que acompaña al usuario no debe darle indicaciones al usuario a no ser que este pregunte, se trata de comprobar si el usuario es capaz de llevar cabo por si mismo todas las acciones que se le han pedido y por tanto la aplicación es usable.

En esta entrega se han realizado varias pruebas con resultado positivo. lo que demuestra que la aplicación es fácil de manejar.

19. VERSIONES DE LA APLICACIÓN

Durante el desarrollo de la aplicación, esta transcurre por tres versiones que son las que se indican a continuación :

- Pac 2: Se entrega la versión Alpha de la aplicación, es una versión prematura en la que funciona la recogida de Trending Topics y su almacenamiento en la base de datos, para poder consultarlos posteriormente desde la propia aplicación. Esta versión no dispone del registro y login de usuarios, los estilos no se han modificado.
- Pac 3: Se entrega la versión Beta de la aplicación, es una versión completamente funcional, con la parte de registro de usuarios y login implementada, se han programado algunos estilos, los más importantes.
- Entrega final: Se entrega la versión 1.0 de la aplicación, esta versión contiene los estilos finales de la aplicación, posibles mejoras a nivel de seguridad y corrección de bugs que se detecten en la versión Beta, se ha implementado la integración con Paypal, y la configuración de localizaciones favoritas, mejorado estilos y corregidos los bugs detectados.

20. REQUISITOS DE INSTALACIÓN

Para poner en marcha la aplicación, es necesario disponer de un servidor que tenga la máquina virtual de java instalada.

A continuación se detallan los componentes empleados en la etapa de desarrollo.

Componente	Descripción
Servidor	Pentium 4, 512 Mb Ram
Sistema operativo	Centos 6
Servidor aplicaciones	Apache Tomcat 7.0.65
Base de datos	MySql 5.1.73
Máquina virtual Java	Java JDK 1.7.0_79

Los componentes empleados durante el desarrollo del presente trabajo pueden ser sustituidos por otros que realicen las mismas funciones.

No es necesario utilizar las últimas versiones de la máquina virtual de Java, MySql, o Tomcat.

Es recomendable utilizar un servidor Linux, aunque también se puede instalar en un servidor Windows.

Como servidor de aplicaciones se ha utilizado Tomcat, por su sencillez de uso, pero la aplicación se puede desplegar en cualquier otro servidor como JBoss, GlassFish, Weblogic,,,etc.

Como base de datos, se ha utilizado MySql, pero esta también puede ser sustituida fácilmente por otra base de datos como Oracle, ya que se ha usado el motor de persistencia Hibernate que es el encargado de hablar con la Base de datos, con lo que modificando la configuración de Hibernate, no sería complicado cambiar el origen de los datos de la aplicación.

21. INSTRUCCIONES DE INSTALACIÓN

21.1 Instalación del servidor

Para instalar el servidor, se accede a un terminal del servidor y se ejecutan una serie de comandos que instalan y configuran el software necesario.

21.2.1 Instalación de la base de datos

Para instalar la base de datos, hay que ejecutar los siguientes comandos:

```
yum install mysql-server mysql  -> Instala la base de datos
service mysqld start            -> Pone en marcha la BD
chkconfig mysqld on             -> La configura como servicio
                                   para que se inicie sola.
/usr/bin/mysql_secure_installation -> Para configurar la
                                      seguridad
```

21.2.2 Creación de la base de datos

El siguiente paso es crear la base de datos de la aplicación, para ello hay que acceder con un usuario que tenga permisos;

```
mysql -u root -p
```

Crear el usuario de la aplicación

```
create user 'tellme' identified by '20Tellme_151';
```

Crear la base de datos

```
create database tellme CHARACTER SET utf8 COLLATE
utf8_general_ci;
```

Conceder permisos al usuario creado en la base de datos creada

```
GRANT ALL PRIVILEGES ON tellme.* TO 'tellme'@'%' IDENTIFIED BY
'20Tellme_151' WITH GRANT OPTION;
```

Por último es necesario recargar los permisos.

```
flush privileges;
```

21.2.3 Importación de scripts

Para importar los scripts que crean las tablas de la base de datos y cargar sus datos iniciales, es necesario copiar los scripts a una carpeta del servidor, situarse en dicha carpeta y ejecutar:

```
mysql -u root -p tellme < script_creacion_bbdd.sql
mysql -u root -p tellme < script._carga_inicial_bbdd.sql
```

El primer script crea las tablas de la base de datos, el segundo carga los datos iniciales de la aplicación.

21.3.1 Instalación del servidor de aplicaciones

Para instalar el servidor de aplicaciones Apache Tomcat, hay que situarse en la carpeta donde se quiere instalar en este caso `/opt` y ejecutar los siguientes comandos:

Descargar Apache Tomcat

```
wget  http://www.us.apache.org/dist/tomcat/tomcat-7/v7.0.65/bin/
apache-tomcat-7.0.65.tar.gz
```

Descomprimir el fichero descargado

```
tar xzf apache-tomcat-7.0.65.tar.gz
cd /opt/apache-tomcat-7.0.65
```

Iniciar el servidor de aplicaciones

```
./bin/startup.sh
```

21.3.2 Configuración de Tomcat

Para acceder al manager o pantalla de administración es necesario primero disponer de un usuario que tenga privilegios para hacerlo, para ello

```
vi /opt/apache-tomcat-7.0.65/conf/tomcat-users.xml
```

Pegamos lo siguiente dentro del archivo y lo guardamos

```
<!-- user manager can access only manager section -->

<role rolename="manager-gui" />

<user username="manager" password="20Tellme_151" roles="manager-gui" />


<!-- user admin can access manager and admin section both -->

<role rolename="admin-gui" />

<user username="admin" password="20Tellme_151" roles="manager-gui,admin-gui" />
```

21.4 Desplegar la aplicación

Para desplegar la aplicación en el servidor Tomcat, primero es necesario generar el archivo WAR, esto se puede realizar de varias formas una es desde la consola de comandos ejecutando:

```
mvm package -Dmaven.test.skip=true
```

Mediante este comando, la herramienta Maven, descarga todas las librerías necesarias, compila el proyecto y genera el WAR de la aplicación.

Otra forma más sencilla es desde el propio Eclipse, realizar un Export del proyecto, esta exportación crea el fichero WAR que hay que desplegar en el servidor de aplicaciones.

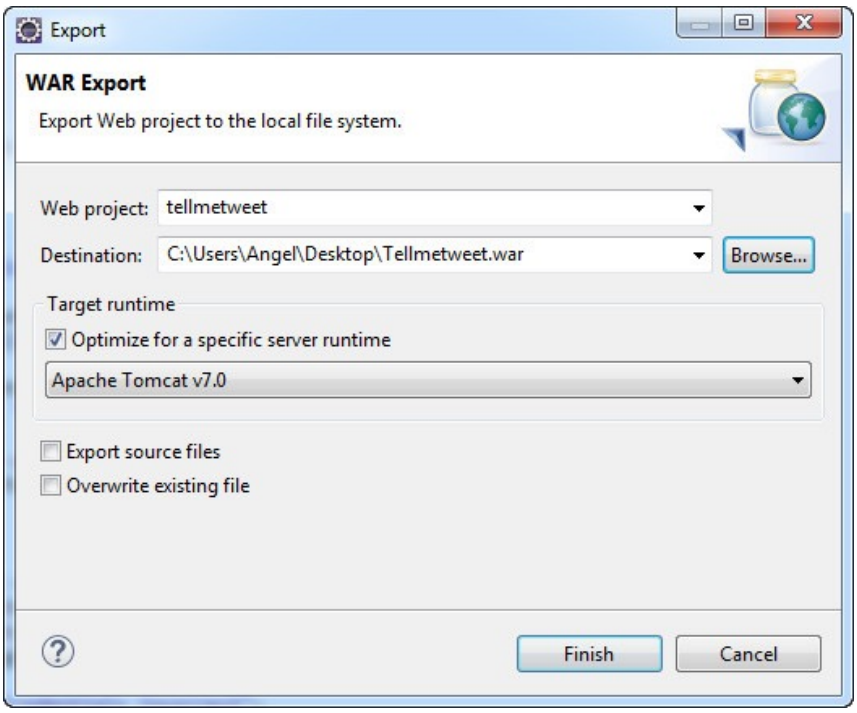


Figura 21.4

Una vez generado el archivo WAR, hay que acceder al panel de administración del servidor de aplicaciones.

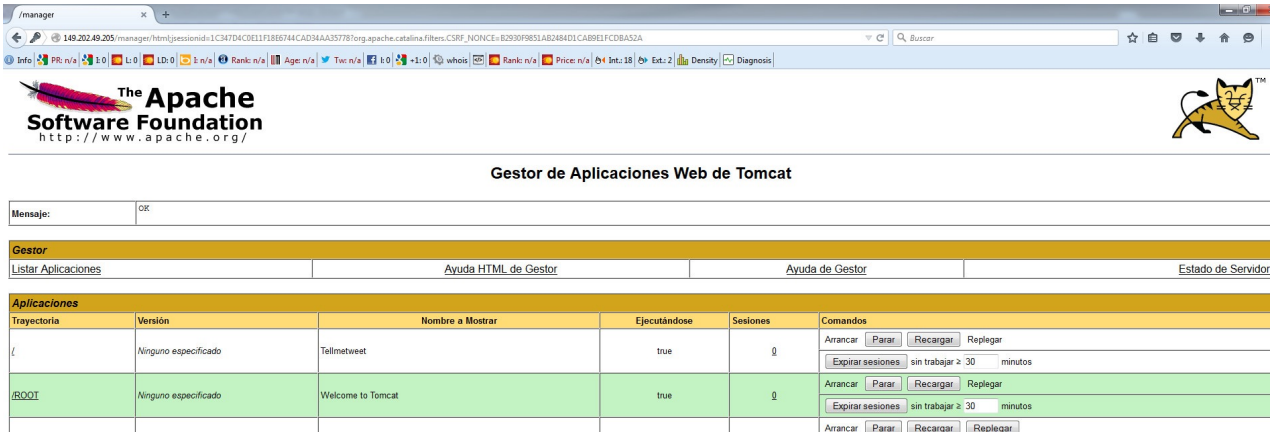


Figura 21.5

En la sección “Archivo WAR a desplegar”, con el botón “Examinar…” se localiza el archivo WAR generado,

al hacer click en el botón “Desplegar”, se despliega la aplicación en el servidor.

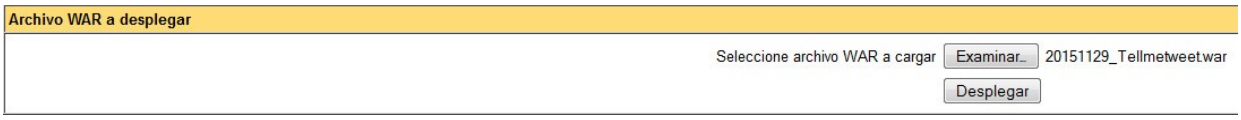


Figura 21.6

Al finalizar el proceso de despliegue, la aplicación aparece en el listado de aplicaciones.

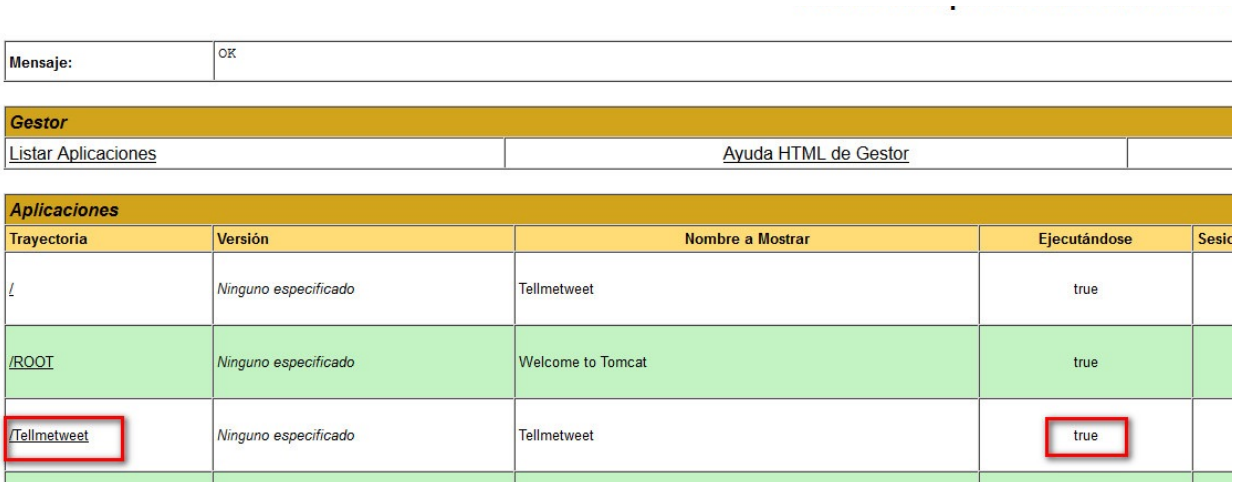


Figura 21.7

Si se hace click con el ratón, se lanza la aplicación en el navegador del usuario

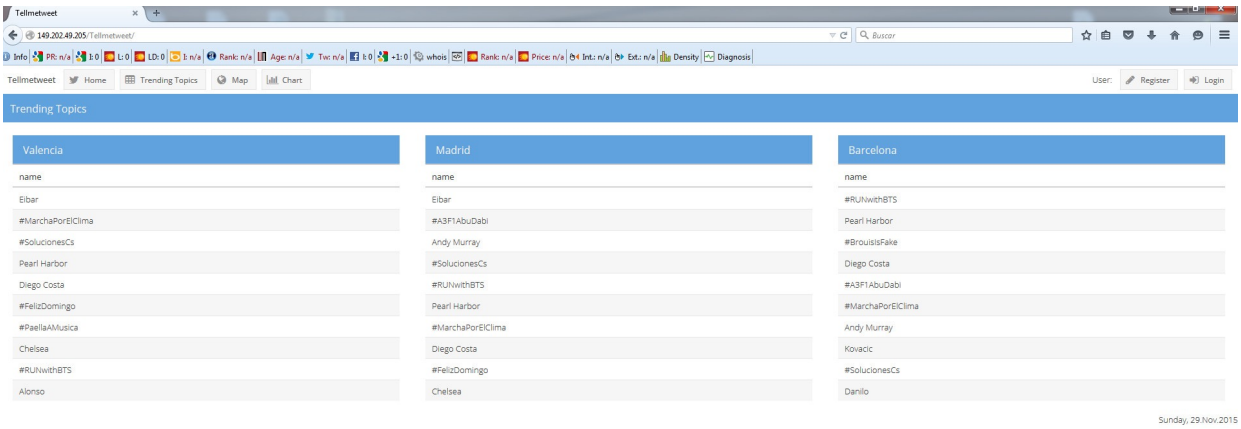


Figura 21.8

22. INSTRUCCIONES DE USO

22.1 PARTES DE LA APLICACIÓN

La aplicación consta de secciones generales a todas las pantallas, y otras que van cambiando en función la acción realizada.

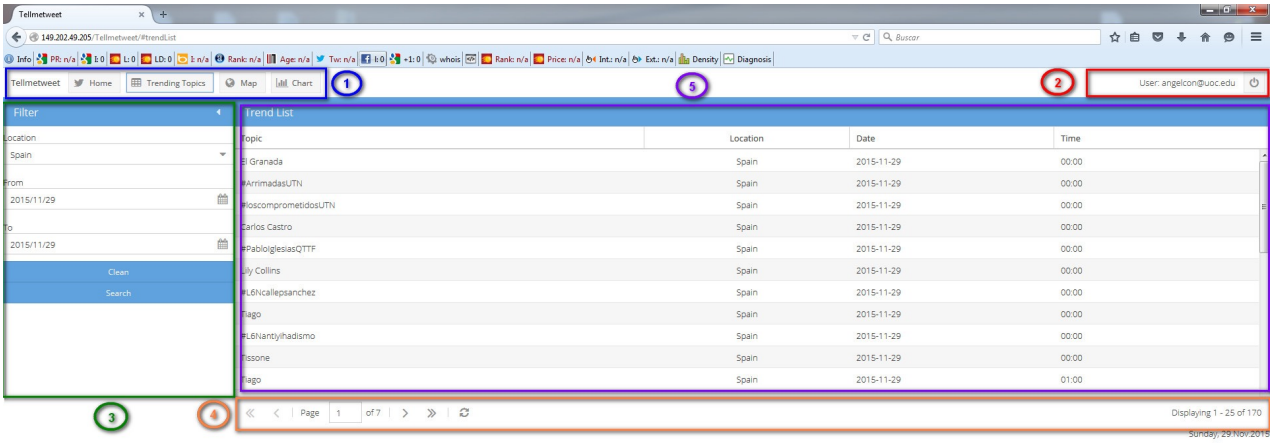


Figura 22.1

1- Menú superior: Se encuentra situado en la parte superior izquierda de la aplicación y muestra los botones que conducen al resto de pantallas de la aplicación.

2- Zona de usuario: Situada en la parte superior derecha, muestra la información del usuario logueado en la aplicación, y los botones para registrarse, hacer login, comprar la cuenta “Premium”, modificar las localizaciones favoritas, y hacer logout, estos tres últimos solo aparecen si hay un usuario identificado en la aplicación, en caso de no haber ninguno no se muestran y se muestran los botones para que el usuario pueda identificarse o registrarse.

3- Zona de filtros: Esta zona varia dependiendo de la pantalla en la que se encuentre el usuario, algunas pantallas no tienen filtro, en las pantallas que si disponen de filtro, es posible filtrar los resultados por los valores seleccionados en ellos.

4- Paginador y resultados: En esta zona se muestra el paginador y el número de resultados obtenidos, la aplicación consulta los datos la primera vez, y luego pagina en memoria, por lo que la paginación es instantánea.

5- Zona central: Esta zona situada en el centro de la pantalla, es la que va a ir cambiando en función de la acción realizada para mostrar las distintas pantallas de la aplicación.

22.2 Home

Para acceder a la aplicación hay que abrir el navegador e ir a la siguiente url:

`http://149.202.49.205/Tellmetweet/`

Aparece la pantalla principal, donde se muestran los últimos Trending Topics para las localizaciones de las ciudades de Valencia, Madrid y Barcelona.

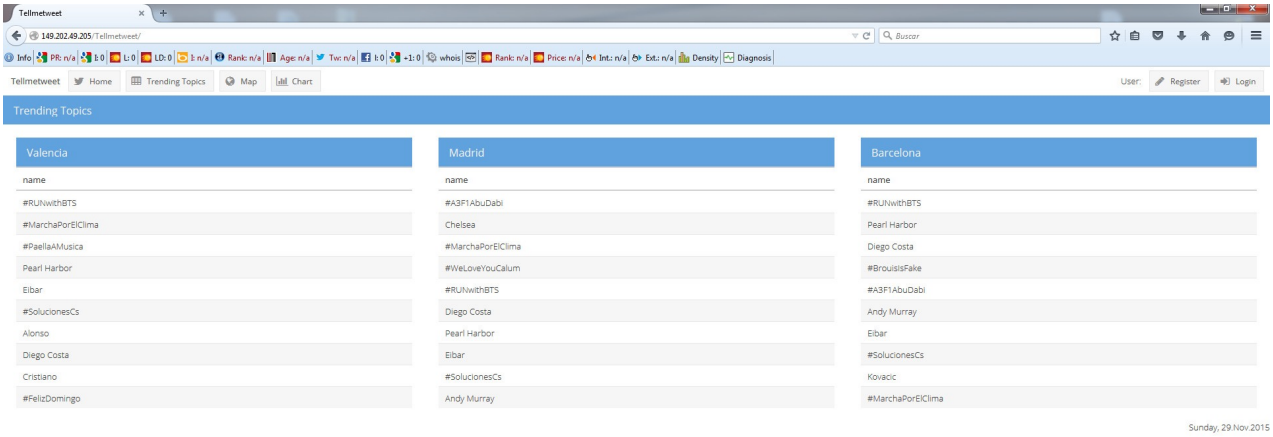


Figura 22.2.1

Si el usuario no está autenticado en la aplicación, la información del usuario aparece vacía y se muestran los botones “Register” y “Login”, si el usuario está autenticado se muestra la información del usuario y el botón “Logout”

22.3 Trending Topics

Esta opción solo está disponible para usuarios registrados, por lo que si no está autenticado en la aplicación se mostrará un mensaje de error.

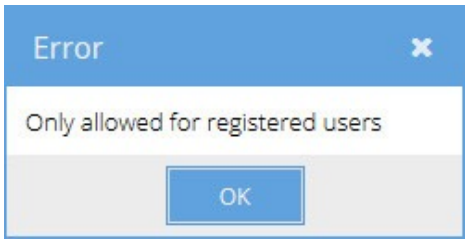


Figura 22.3.1

Accediendo con un usuario autenticado, se muestran los Trending Topics que coinciden con los parámetros seleccionados en el filtro.

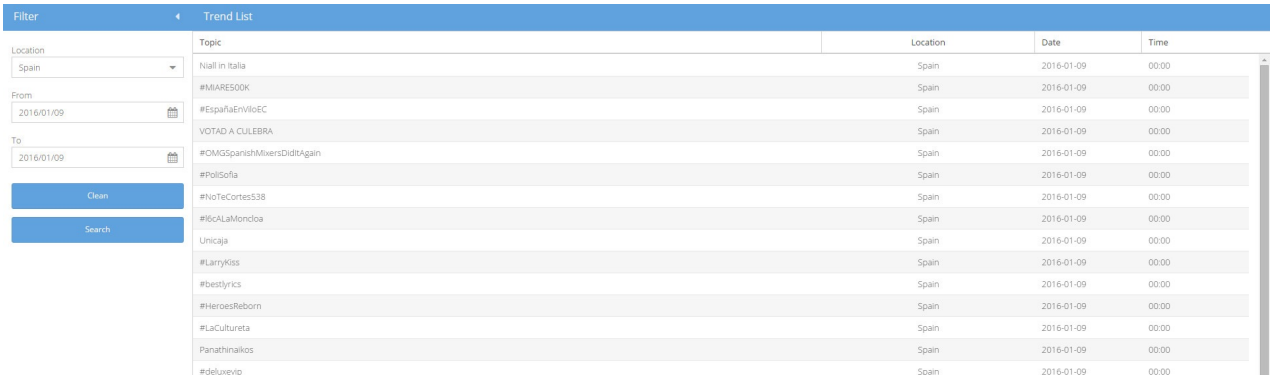


Figura 22.3.2

En la parte izquierda está el filtro, para cambiar de Location podemos seleccionar una distinta en el combo, o podemos escribir directamente en el y seleccionarla.

El rango de fecha también se puede escribir o buscar en el calendario haciendo click en el icono del calendario que mostrará el calendario para seleccionar una fecha.

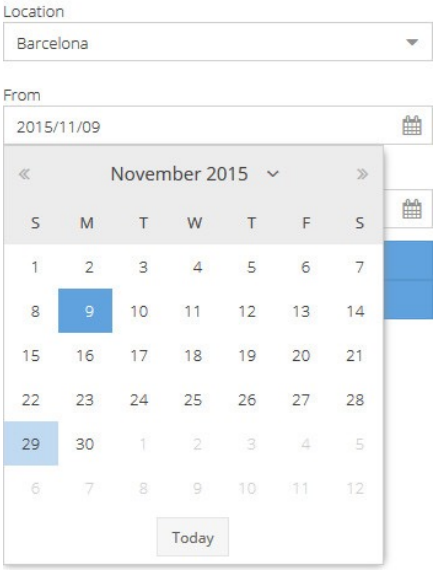


Figura 22.3.3

El botón “Clean” restablece los valores del filtro a sus valores predeterminados.

El botón “Search” realiza una nueva búsqueda aplicando los valores del filtro.

Si en los campos del filtro se introducen valores incorrectos, los botones se inhabilitan hasta que se corrijan los valores erróneos.

Topic	Location	Date	Time
Sporting	Valencia	2015-11-09	00:00
Sevilla	Valencia	2015-11-09	00:00
Sevilla	Valencia	2015-11-09	01:00
Sergio Rico	Valencia	2015-11-09	00:00
Sergio Rico	Valencia	2015-11-09	01:00
Ronaldo	Valencia	2015-11-09	00:00
Krychowiak	Valencia	2015-11-09	01:00
Hedy Lamarr	Valencia	2015-11-09	01:00
Golazo de James	Valencia	2015-11-09	00:00
Danilo	Valencia	2015-11-09	01:00
Broncos	Valencia	2015-11-09	02:00

Page 1 of 10

Figura 22.3.4

Haciendo click en la cabecera del grid, se ordena el grid, también se puede cambiar de página con el paginador, bien introduciendo la página directamente o mediante los botones.

22.4 Map

Muestra un mapa con las localizaciones activas en la aplicación, haciendo click sobre un marcador aparece un Popup, que muestra los últimos Trending topics

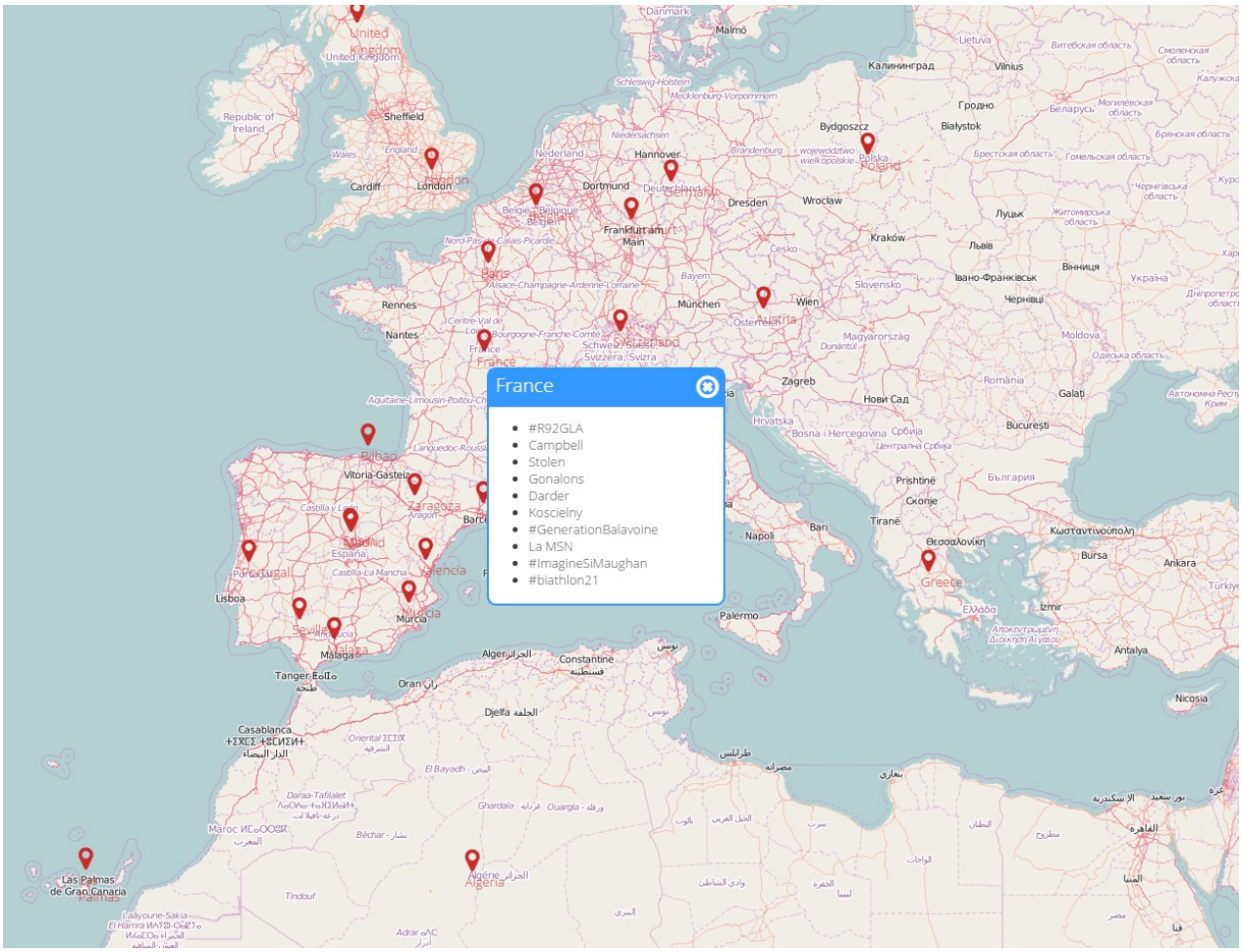


Figura 22.4.1

El Popup se cierra haciendo click en el Botón redondo con una X en su interior.

22.5 Chart

Solo disponible para usuarios autenticados. Muestra una gráfica de barras en la que aparecen todos los Trending Topics del día seleccionado en el Filtro y en la localización seleccionada en el mismo filtro.

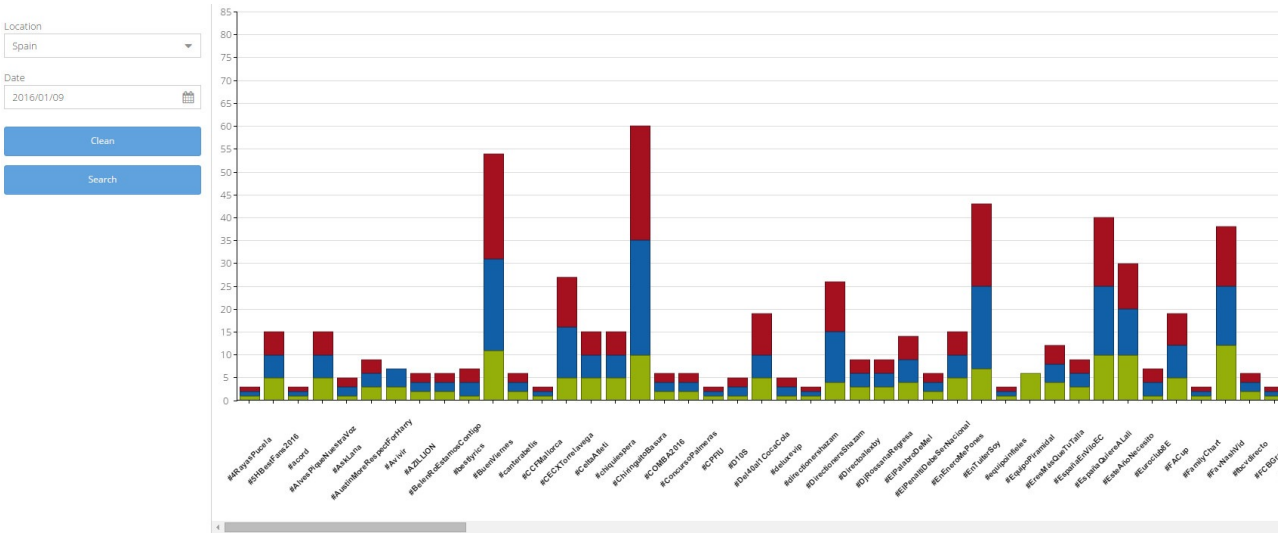


Figura 22.5.1

La gráfica permite ver durante cuantas horas ha sido Tending Topic un Tweet el día seleccionado, además muestra en distinto color cuantas horas ha sido Trending Topic en la semana en la que se encuentra el día seleccionado y lo mismo para el mes en distinto color.

En la parte derecha de la gráfica se encuentra una leyenda que permite activar o desactivar las horas por día, semana y mes.

Al pasar el ratón por encima de una barra, se muestra un Tooltip con información.

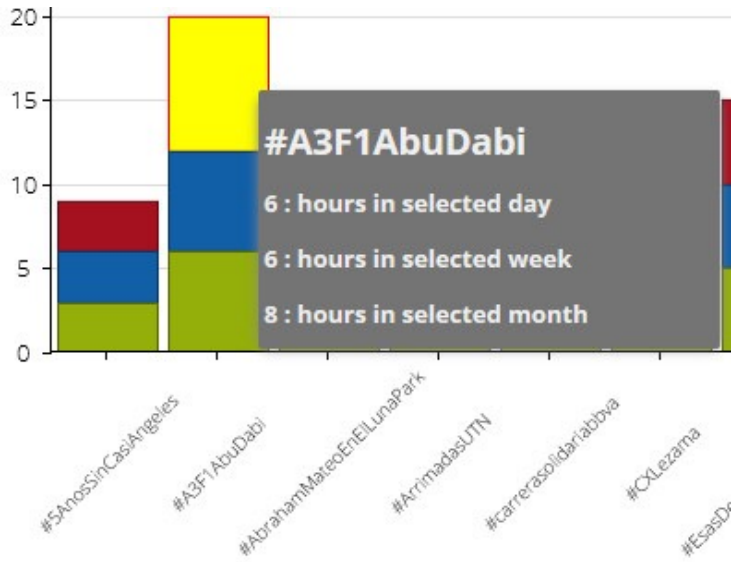
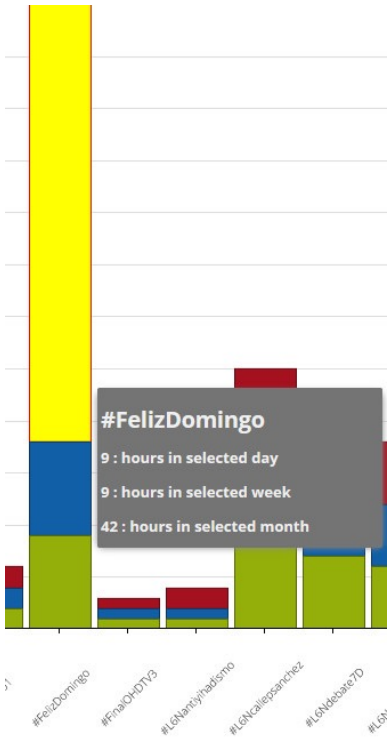


Figura 22.5.2

En el tooltip se puede apreciar como #A3F1AbuDabi ha sido Trending Topic hoy durante 6 horas, y además ha sido Trending Topic 8 horas este mes.



Aquí se puede apreciar como #FelizDomingo ha sido Trending Topic hoy durante 9 horas, pero lo más importante es que también lo ha sido 42 horas este mes, algo difícil de creer ya que según las normas de Twitter los Trending topics deben de ser originales y en este caso este no lo es, pues se ha repetido todas las semanas, ¿Que ocurre aquí?.

Figura 22.5.3

22.6 Register

Esta opción muestra un Popup, donde se introducen los datos para crear una nueva cuenta, si los datos rellenados son correctos se habilita el botón “Register”.

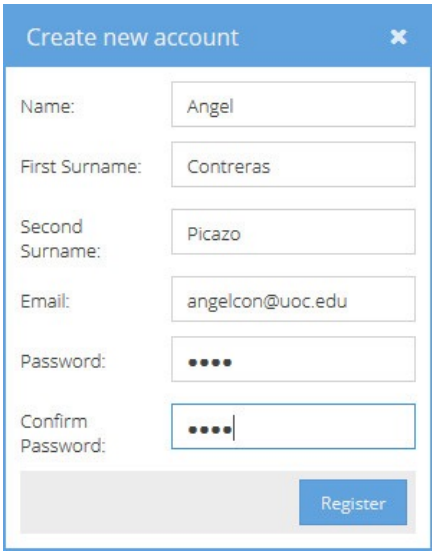


Figura 22.6.1

Al pulsar en el botón “Register” se envían los datos al servidor y el usuario es autenticado en la aplicación.

22.6 Login

Esta opción muestra el Popup de autenticación.

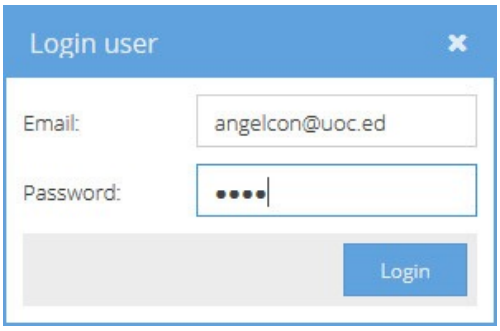


Figura 22.6.2

22.6 Logout

Esta opción desconecta al usuario del sistema

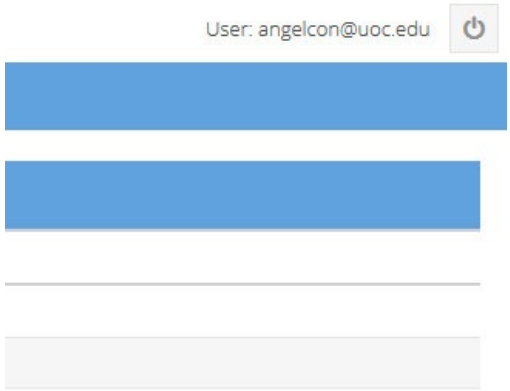


Figura 22.6

En las pantallas que requieren que el usuario esté autenticado, dependiendo del rol del usuario, este tiene un número máximo de acciones diarias que puede realizar, si llega al límite tendrá que esperar al día siguiente para poder realizar más acciones.

22.7 Configuración

Muestra un popup que permite modificar las localizaciones favoritas del usuario, estas son las que aparecen en la Home de la aplicación.

El usuario registrado solo puede realizar un cambio diario, el registro del usuario se contabiliza como un cambio, por lo que un usuario recien registrado no podrá realizar modificaciones hasta el siguiente día, el usuario con cuenta Premium puede modificarlo siempre que quiera.

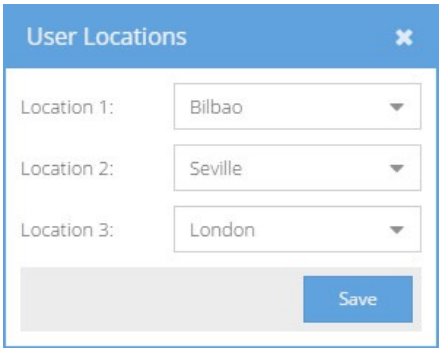


Figura 22.7

22.8 Premium

Esta opción muestra un Popup, donde se introducen los datos para comprar la cuenta "Premium", si los datos rellenados son correctos se habilita el botón "Get Premium".

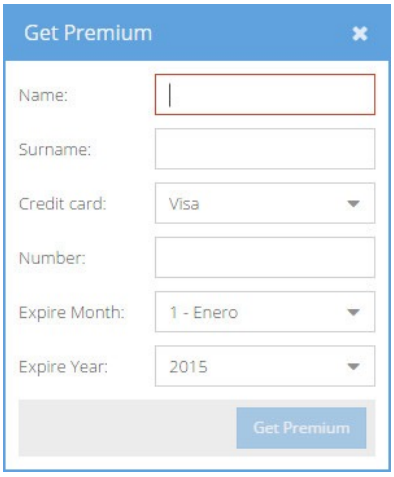


Figura 22.8

23. BUGS

23.1 API Twitter

Si bien no es un bug, es un problema que no me esperaba y me ha hecho crear nuevas funciones y dedicarle bastante tiempo.

Twitter tiene ciertas limitaciones a la hora de realizar consultas, dependiendo de lo que se consulte hay establecido un número de conexiones y si se agotan hay que esperar un tiempo establecido hasta que se restablecen dichas conexiones.

Cada vez que se consultan los Trending Topics de una localización se descuenta un intento del limite disponible, al principio hay 15 disponibles, el problema es que si se consultan los trending Topics de 15 localizaciones a la siguiente se recibe una excepción de Twitter y la aplicación genera una excepción que en caso de no tenerla controlada puede dejarla bloqueada.

Para solventar este problema se ha optado por, primero comprobar si el limite disponible es mayor que cero, en caso de serlo realizar la consulta, si el límite está en cero, se consulta el tiempo que queda para liberar las conexiones y se espera ese tiempo y unos segundos más para dar un margen de seguridad, a continuación se vuelve a consultar los intentos disponibles.

El número de intentos disponibles y tiempo, depende de lo que se consulte, así que se ha realizado una función que indicándole lo que se quiere consultar, realiza las comprobaciones correspondientes.

25. PRESUPUESTO

Nombre de tarea	Duración	Costo
Diseño de base de datos	**5 días**	**4.150,00 €**
Diseño de base de datos	3 días	1.250,00 €
Creación de base de datos y carga inicial	2 días	850,00 €
Servidor	**32 días**	**17.900,00 €**
Instalación servidor en máquina virtual de pruebas	5 días	2.050,00 €
Carga de datos	1 día	450,00 €
Contratar servidor VPS en OVH	3 días	1.250,00 €
Instalar VPS	2 días	850,00 €
Desplegar aplicación en VPS	1 día	450,00 €
Aplicación	**33 días**	**34.600,00 €**
Creación Tokens seguridad Twitter	1 día?	450,00 €
Puesta en marcha entorno de desarrollo	6 días	2.450,00 €
Creación parte Back, esqueleto inicial	3 días	1.250,00 €
Creación parte Front, Extjs	4 días	1.650,00 €
Servicio para almacenar Trends en BD	3 días	1.250,00 €
Mostrar últimos trends en Front	3 días	1.250,00 €
Desplegar aplicación en servidor máquina virtual	1 día	450,00 €
Registro usuarios	10 días	4.050,00 €
Gestión de permisos	5 días	2.050,00 €
Login y Logout	2 días	850,00 €
Integración Paypal	14 días	5.650,00 €
Total		**85.150,00 €**

El presupuesto total del proyecto es de 85.150 euros.

26. ANÁLISIS DE MERCADO

Esta aplicación va dirigida especialmente a Community managers y gestores de campañas de Marketing Online, dado que estos perfiles suelen hacer un uso intensivo de las redes sociales como Twitter, disponer de una herramienta como esta para medir sus resultados es algo imprescindible para ellos.

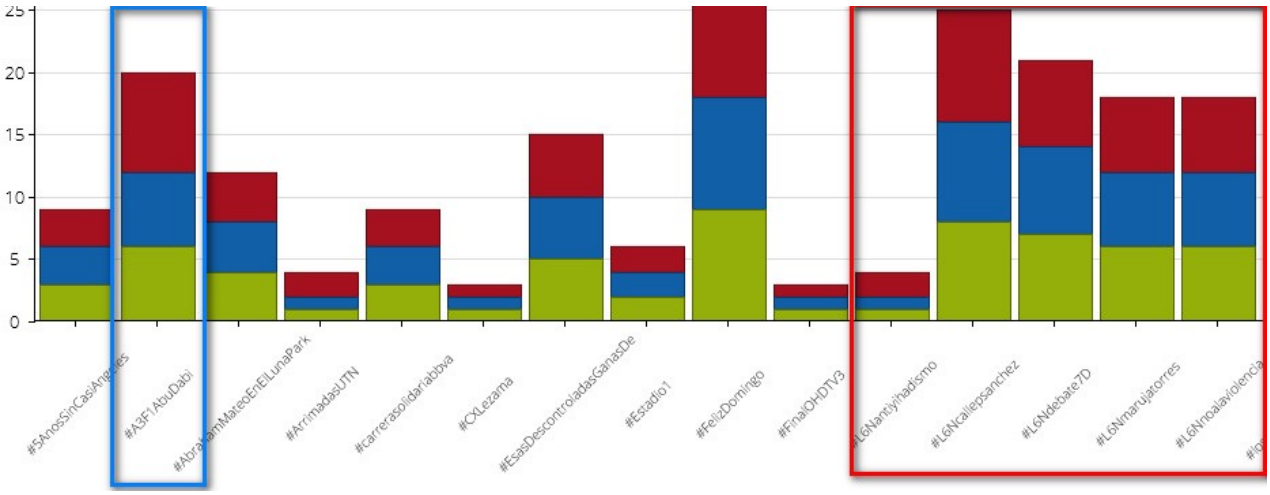


Figura 26.1

En la figura 26.1 se puede apreciar en color rojo, como el Community Manager de la Sexta ha colocado el día 29 de Noviembre cinco Trending Topics de los cuales cuatro han sido Trending Topic durante seis horas o más.

En cambio el Community Manager de Antena3 en azul, solo ha sido capaz de colocar el Trending Topic #A3F1AbuDabi durante seis horas, la diferencia es notable y está claro quien es el Community Manager que marca la diferencia y que más visitas está generando.

Otra cosa importante a destacar, es que se puede apreciar, como el resto de cadenas de la competencia no son capaces de generar Tendring Topics.

Es muy sencillo ver con esta herramienta que los Community Manager del resto de cadenas se tienen que espabilar y cambiar de estrategia.

A parte de los gestores de estrategias de Marketing online y Community managers, esta herramienta también va dirigida a cualquier persona que tenga un negocio o empresa y quiera optimizar sus campañas de publicidad en Twitter Ads.

En el apartado cuatro de esta memoria, se ha realizado un estudio sobre la competencia y aplicaciones de referencia existentes, comprobando que en la mayoría los resultados no coinciden si los comparamos con los de una cuenta de Twitter real, por lo que no pueden ser considerados datos fiables, ademas no permiten consultar los Trending Topics de por ejemplo Valencia desde el día 1 de Noviembre hasta el día 4 de Noviembre, tampoco ofrecen datos medibles como en el caso expuesto anteriormente donde se puede observar como el Community de la sexta ha colocado cinco Trending Topics durante más de seis horas, además mediante el listado de Trending Topics podemos ver en que horas han sido, el resto de herramientas no ofrecen estos datos, y *"los datos si no se pueden medir no sirven para nada"* *(Avinash Kaushik)*

Por todo ello, las aplicaciones analizadas, no son competencia, pues no representan información fiable, la única que representa información fiables es la propia aplicación de Twitter, pero esta no muestra la información que presenta esta herramienta, la propia aplicación de Twitter no nos permite conocer que el community de la sexta ha colocado cinco Trending Topics en un día.

ANNEXO 1. ENTREGABLES DEL PROYECTO

- Memoria: El presente documento.
- Pdf navegable con las imágenes del prototipo: Pdf interactivo, haciendo click con el ratón en los botones, podemos interactuar y recorrer la aplicación, los botones interactuables tienen una marca para indicarlo.
- Script para la creación de la base de datos: Este script crea las tablas de la base de datos y sus restricciones.
- Script para cargar los datos iniciales de la base de datos: Este script carga los datos iniciales necesarios para el funcionamiento de la aplicación.
- Código fuente de la aplicación: Zip que contiene el código fuente de la aplicación, los códigos personales empleados para autenticar las APIS utilizadas, han sido ligeramente modificados manteniendo la misma estructura, para no publicar información sensible.
- Presentación de la aplicación: Presentación visual de la aplicación.
- Informe de competencias trasversales: Informe con las competencias que se han desarrollado durante el transcurso del Grado de Multimedia.
- Video de la aplicación: Video que explica el desarrollo y uso de la aplicación, para la defensa ante el tribunal.
- La aplicación estará accesible en la siguiente url: http://149.202.49.205/Tellmetweet

ANNEXO 2. CÓDIGO FUENTE (EXTRACTOS)

2.1 Servidor

Script que llama al servicio de la aplicación para que almacena los Trending Topics en la base de datos.

```
file=storeTrends.log
start=$(date +"%s")
serverIp=149.202.49.205
serverPort=8080

echo . >> $file
echo ---------------------------------- >> $file
echo ---------------------------------- >> $file
echo Start storeTrends $(date '+%Y/%m/%d %H:%M:%S') >> $file

curl --request GET -H 'Content-type: application/json' -d
'{"tokenId":"2D3047EF21DD6401D506F7E04EC37C0090CB9C7240BA010DE855
B90322C4"}' http://$serverIp:$serverPort/Tellmetweet/rest/trends/
storeTrends >> $file
end=$(date +"%s")
diff=$(($end-$start))

echo End storeTrends $(date '+%Y/%m/%d %H:%M:%S') >> $file
echo Procesed in "$(($diff / 60)) minutes and $(($diff % 60))
seconds" >> $file
echo ---------------------------------- >> $file
echo ---------------------------------- >> $file
echo . >> $file
```

El script es lanzado automáticamente por Cron(servicio linux que permite la programación de eventos) cada hora, llamando al servicio de la aplicación que se encarga de obtener los trends. Al llamar al servicio se le pasa el token de seguridad, el servicio comprueba el token recibido para comprobar que tiene permisos, si el token es correcto el servicio almacena los trending topics en la base de datos y devuelve un mensaje que se almacena en el fichero storeTrends.log a modo de los para su posterior consulta.

2.2.1 Aplicación - Controller

El siguiente código muestra el servicio que se encarga de almacenar en la base de datos los trending topics de las localizaciones activas, este servicio rest es llamado automáticamente por el administrador de tareas Cron del servidor, mediante el script mostrado en el punto anterior.

```
package com.uoc.tellmetweet.web;

import java.util.ArrayList;

@Controller
@RequestMapping("/rest/trends")
public class TrendsController {

    private static Logger logger = LoggerFactory.getLogger(TrendsController.class);

    @Autowired
    TrendsBO trendsBO;

    /**
     * Almacena en la base de datos los últimos Trending Topics, para las localizaciones activas(TM_ACTIVELOCATIONS)
     * @param request
     * @return MessageDTO.toJsonArray
     */
    @RequestMapping(value="/storeTrends",headers = "Accept=application/json", method = RequestMethod.GET)
    @ResponseBody
    public ResponseEntity<String> listTrends(@RequestParam(required=false) String tokenId){

        HttpHeaders headers = new HttpHeaders();
        headers.add("Content-Type", "application/json; charset=utf-8");

        if(tokenId==null || !tokenId.equals(TellmetweetConstants.security_token)){
            return new ResponseEntity<String>(Messages.usuario_sin_permisos, headers, HttpStatus.INTERNAL_SERVER_ERROR);
        }

        List<MessageDTO> listMessageDTO = new ArrayList<MessageDTO>();
        try {
            logger.info("Iniciando proceso de recuperación de trends");
            listMessageDTO = trendsBO.storeTrends();
            logger.info("Fin proceso de recuperación de trends");
        } catch (TellmetweetBOException b) {
            logger.error(b.getMessage(), b);
            return new ResponseEntity<String>(b.getMessage(), headers, HttpStatus.INTERNAL_SERVER_ERROR);
        } catch (Exception e) {
            logger.error(e.getMessage(), e);
            return new ResponseEntity<String>(e.getMessage(), headers, HttpStatus.INTERNAL_SERVER_ERROR);
        }
        return new ResponseEntity<String>(MessageDTO.toJsonArray(listMessageDTO), headers, HttpStatus.OK);
    }
```

Las primeras anotaciones, indican que la clase hace de controlador y la ruta de acceso, lo siguiente es declarar el Logger para poder seguir las trazas de los errores.

La anotación @Autowired, hace que Spring inyecte la clase TrendsBO, de forma que no se tenga que inicializar.

Se definen las cabeceras y seguidamente se comprueba que llegue el token de seguridad y que este sea correcto, en caso de no serlo se retorna un mensaje de error terminando la ejecución del método.

Se crea una lista de MessageDTO, y se llama al método de la clase de negocio trendsBO.storeTends() asignando el resultado a la lista de mensajes, finalmente se devuelve la lista en formato Json, de forma que si en la lógica de negocio se ha añadido algún mensaje, este llegará hasta quien ha realizado la llamada al servicio.

Finalmente los bloque Try/Catch se encargan de capturar las excepciones, en caso de capturar alguna la devuelven de forma que pueda ser mostrada en el frontend de la aplicación.

2.2.2 Aplicación - Bo

El siguiente código, muestra el código de una clase de negocio que devuelve una lista de objetos DTO que contienen los últimos trending topics de la localidad perteneciente al woeid(Where On Earth IDentifier) pasado.

```
/**
 * Devuelve los trends de un woeid
 * @param woeid
 * @return List<HomeTrendsDTO>
 */
public List<HomeTrendsDTO> findHomeTrendsByWoeid(Integer woeid) throws TellmetweetBOException {

    List<HomeTrendsDTO> listHomeTrendsDTO = new ArrayList<HomeTrendsDTO>();
    try {

        /* 1- Obtener la activeLocation del woeid pasado */
        TmActivelocations activeLocation = activeLocationDAO.findActiveLocationsByWoeid(woeid);
        /* 2- Recoger los últimos trend de la activeLocation */
        Set<TmTrends> listTmTrends = activeLocation.getLocation().getTmTrendss();
        for (TmTrends tmTrends : listTmTrends) {

            Set<TmTrend> listTmTrend = tmTrends.getTmTrends();
            for (TmTrend tmTrend : listTmTrend) {
                HomeTrendsDTO homeTrendsDTO = new HomeTrendsDTO();
                homeTrendsDTO.setName(tmTrend.getName());
                homeTrendsDTO.setLocationName(tmTrends.getLocation().getName());
                homeTrendsDTO.setWoeid(tmTrends.getLocation().getWoeid());
                listHomeTrendsDTO.add(homeTrendsDTO);
            }
        }

    } catch (Exception e) {
        logger.error(e.getMessage(), e);
        throw new TellmetweetBOException(e.getMessage(), e);
    }
    return listHomeTrendsDTO;
}
```

```
throws TellmetweetBOException
```

Si ocurre una excepción, esto hará que se pase a quien ha llamado al método o sea cualquier excepción será devuelta al controlador.

Se crea una lista de objetos DTO, se busca la localización perteneciente al woeid pasado, es decir si llega el woeid “776688” se obtendrá la localización de “Valencia”, seguidamente se obtiene una colección con sus Trending Topics, se recorre la colección y se añaden todos a la lista de objetos DTO creada inicialmente que es lo que devuelve el método si no ocurre ninguna excepción, en caso de ocurrir alguna el bloque try/catch la captura y la relanza al controlador mediante el “throws” incluido en la declaración del método.

2.2.3 Aplicación - Interfaz de un Dao

El siguiente método muestra el código de la interfaz de un DAO. A un DAO se accede a través de su interfaz, esto permite desacoplar el código, aquí podemos ver una interfaz con dos métodos.

```
package com.uoc.tellmetweet.dao;

import java.util.List;

public interface ActiveLocationDAO {

    /**
     * Devuelve un listado de tmActivelocations
     * @return tmActivelocations
     */
    public List<TmActivelocations> findAllActiveLocations() throws TellmetweetDAOException;

    /**
     * Devuelve un activeLocation pasando un woeid
     * @param woeid
     * @return TmActivelocations
     */
    public TmActivelocations findActiveLocationsByWoeid(Integer woeid) throws TellmetweetDAOException;

}
```

El uso de interfaces permite que el cambio de código en la implementación del DAO sea totalmente transparente, incluso se puede cambiar de implementación y el método de la lógica de negocio que llama a la interfaz no tiene porque saberlo ni conocer que hay en dicha implementación del DAO.

2.2.4 Aplicación - Implementación de un Dao

El siguiente método muestra el código de la implementación de un DAO, las implementaciones son llamadas desde las interfaces.

```
package com.uoc.tellmetweet.dao;


import org.slf4j.Logger;

@Transactional
@Repository
public class AdminsDAOImpl extends GenericDAOImpl implements AdminsDAO {

    private static Logger logger = LoggerFactory.getLogger(GenericDAOImpl.class);

    /**
     * Guarda un objeto TmAdmins en la base de datos
     */
    public void saveAdmin(TmAdmins admins) throws TellmetweetDAOException {

        try {
            admins.persist();
        } catch (Exception e) {
            logger.error("Error: " + e.getMessage(), e);
            throw new TellmetweetDAOException(e.getMessage(), e);
        }
    }

}
```

La anotación @Transactional comprueba que se ejecuten todas las transacciones, si alguna falla realiza un “rollback” en la base de datos y si se realizan todas, hace un “commit” en la base de datos.

Los daos se pueden declarar en Spring como beans o bien en un archivo xml o bien mediante la anotación @Repository en el propio dao como es el caso, esta anotación también se encarga de que si un DAO produce una excepción de JPA, esta sea traducida a la jerarquía de excepciones de Spring.

```
admin.persist();
```

Persiste el objeto admin en la base de datos, o sea inserta un nuevo admin.

ANNEXO 3. LIBRERÍAS/CÓDIGO EXTERNO

3.1 Twitter4J

La librería Twitter4J permite realizar consultas y obtener información a través de una cuenta de Twitter, para poder conectar mediante Twitter4J a Twitter, es necesario ir a:

https://apps.twitter.com/

Y hacer click en “Create New APP”

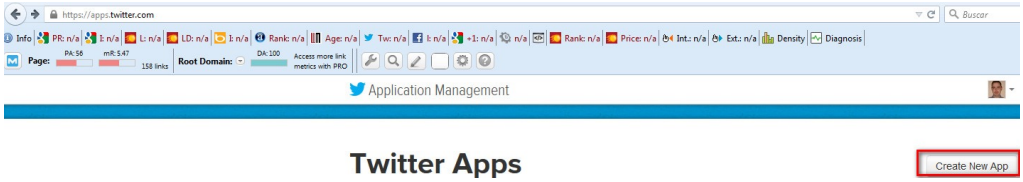


Figura A3.1.1

Seguidamente se rellenan los datos y se acepta la licencia, y se crea la aplicación con los token y códigos de acceso que son necesarios pasarle a Twitter4J para que pueda autenticarse en Twitter y poder realizar consultas.

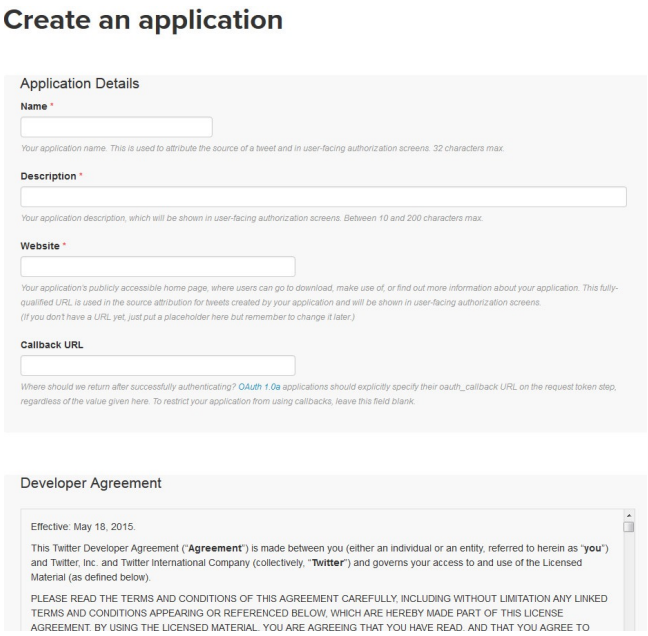


Figura A3.1.2

Mediante Twitter4J se obtiene la conexión con Twitter, se valida que queden conexiones disponibles para realizar consultas y se obtienen los Trending Topics de las localizaciones disponibles en la aplicación.

3.2 Paypal SDK

La librería de Paypal, permite realizar cobros utilizando los formularios propios creados en la aplicación, para ello es necesario ir a:

https://developer.paypal.com/developer/accounts/

El proceso es similar al de la librería Twitter4J, creamos una cuenta, si se dispone de una cuenta Paypal es posible foguearse con ella, seguidamente se obtienen los códigos de seguridad necesarios para que el API pueda autenticarse en Paypal.

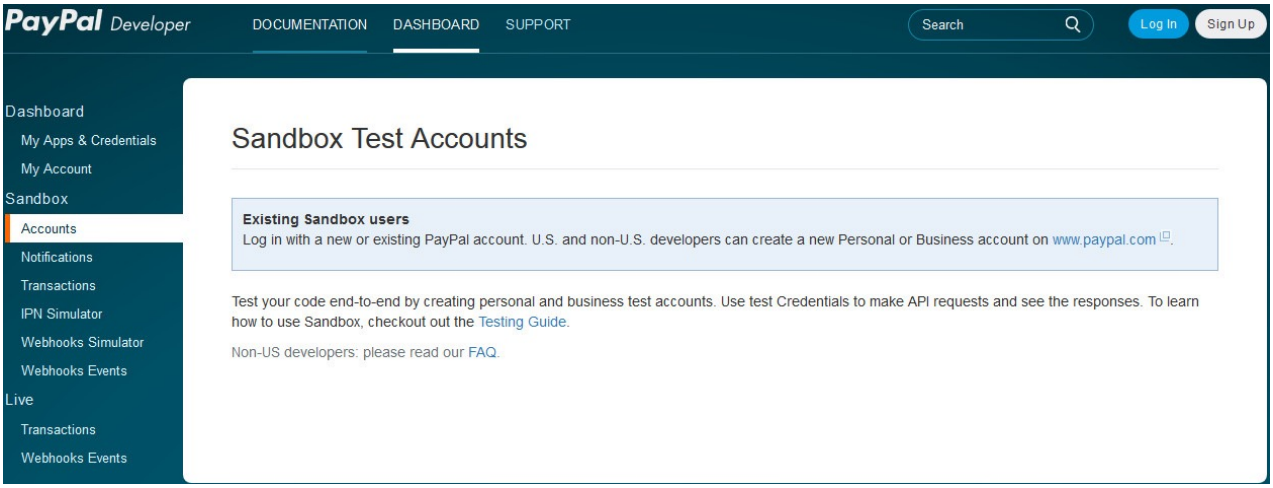


Figura A3.2

Sandbox es una herramienta que permite simular transacciones, para realizar pruebas y una vez se comprueba que todo es correcto se cambia al modo normal para que realmente se realicen las transacciones.

Se ha utilizado esta librería en modo Sandbox para simular el cobro de la cuenta Premium, el API realiza las mismas validaciones que el modo normal y el funcionamiento es el mismo salvo que las transacciones realmente no se realizan.

ANNEXO 4. CAPTURAS DE PANTALLA

Anexo 4.1 Puesta en marcha de la aplicación

Iniciamos Roo y escribimos:

```
project setup --topLevelPackage com.uoc.tellmetweet --projectName
Tellmetweet --packaging WAR
```

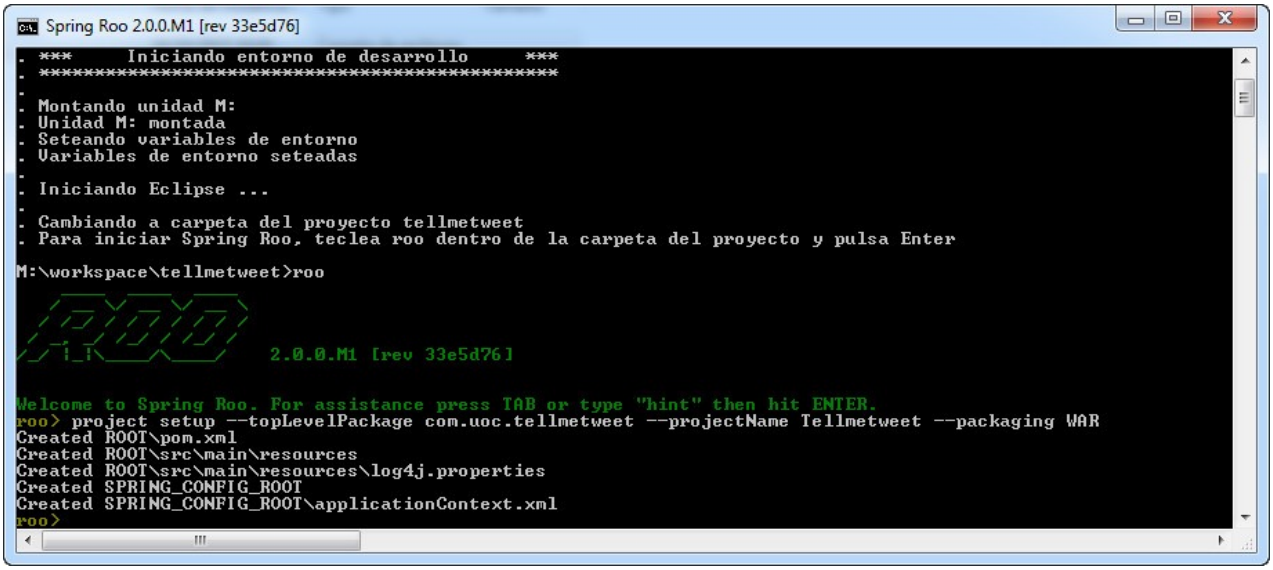


Figura A4.1

Este comando crea la aplicación, como podemos ver en la figura A4.1, crea el fichero de Maven “pom.xml”, el fichero de configuración del logger “log4j.properties” y el fichero de configuración de Spring “applicationContext.xml”

A continuación se configura la conexión con la base de datos:

```
jpa setup --provider HIBERNATE --database MYSQL --databaseName
tellme?autoReconnect\=true&useUnicode\=true&characterEncoding
\=UTF-8 --userName tellme --password 20Tellme_151 --hostName
149.202.49.205
```

```
Spring Roo 2.0.0.M1 [rev 33e5d76]
. Variables de entorno seteadas
.
. Iniciando Eclipse ...
.
. Cambiando a carpeta del proyecto tellmetweet
. Para iniciar Spring Roo, teclea roo dentro de la carpeta del proyecto y pulsa Enter

M:\workspace\tellmetweet>roo

    2.0.0.M1 [rev 33e5d76]

Welcome to Spring Roo. For assistance press TAB or type "hint" then hit ENTER.
roo> project setup --topLevelPackage com.uoc.tellmetweet --projectName Tellmetweet --packaging WAR
Created ROOT\pom.xml
Created ROOT\src\main\resources
Created ROOT\src\main\resources\log4j.properties
Created SPRING_CONFIG_ROOT
Created SPRING_CONFIG_ROOT\applicationContext.xml
roo> jpa setup --provider HIBERNATE --database MYSQL --databaseName tellme?autoReconnect\=true&useUnicode\=true&characterEncoding\=UTF-8 --userName tellme --password 20Tellme_15 --hostName 149.202.49.205
Created SPRING_CONFIG_ROOT\database.properties
Please update your database details in src/main/resources/META-INF/spring/database.properties.
Updated SPRING_CONFIG_ROOT\applicationContext.xml
Created ROOT\src\main\resources\META-INF\persistence.xml
Updated ROOT\pom.xml [added dependencies mysql:mysql-connector-java:5.1.18, org.hibernate:hibernate-core:4.3.6.Final, org.hibernate:hibernate-entitymanager:4.3.6.Final, org.hibernate.javax.persistence:hibern
rg.springframework:spring-jdbc:${spring.version}, org.springframework:spring-orm:${spring.version}, commons-pool:commons-pool:1.5.6, commons-dbcp:commons-dbcp:1.4]
roo>
```

Figura A4.2

Una vez está configurada la conexión a la base de datos, se aplica ingeniería inversa, para que lea las tablas de la base de datos y genere el dominio de la aplicación.

```
database reverse engineer --schema tellme --package ~.domain –
testAutomatically
```

```
Spring Roo 2.0.0.M1 [rev 33e5d76]
    2.0.0.M1 [rev 33e5d76]

Welcome to Spring Roo. For assistance press TAB or type "hint" then hit ENTER.
roo> project setup --topLevelPackage com.uoc.tellmetweet --projectName Tellmetweet --packaging WAR
Created ROOT\pom.xml
Created ROOT\src\main\resources
Created ROOT\src\main\resources\log4j.properties
Created SPRING_CONFIG_ROOT
Created SPRING_CONFIG_ROOT\applicationContext.xml
roo> jpa setup --provider HIBERNATE --database MYSQL --databaseName tellme?autoReconnect\=true&useUnicode\=true&characterEncoding\=UTF-8 --userName tellme --password 20Tellme_15 --hostName 149.202.49.205
Created SPRING_CONFIG_ROOT\database.properties
Please update your database details in src/main/resources/META-INF/spring/database.properties.
Updated SPRING_CONFIG_ROOT\applicationContext.xml
Created ROOT\src\main\resources\META-INF\persistence.xml
Updated ROOT\pom.xml [added dependencies mysql:mysql-connector-java:5.1.18, org.hibernate:hibernate-core:4.3.6.Final, org.hibernate:hibernate-entitymanager:4.3.6.Final, org.hibernate.javax.persistence:hibern
rg.springframework:spring-jdbc:${spring.version}, org.springframework:spring-orm:${spring.version}, commons-pool:commons-pool:1.5.6, commons-dbcp:commons-dbcp:1.4]
roo> database reverse engineer --schema tellme --package ~.domain --testAutomatically
Searching 'com.mysql.jdbc.Driver' on installed repositories
1 matches found with 'com.mysql.jdbc.Driver' on installed repositories
ID   BUNDLE SYMBOLIC NAME                                   DESCRIPTION
--------------------------------------------------------------------------------
01   org.springframework.roo.wrapping.mysql-connector-java   Spring Roo - Wrapping - mysql-connector-java

[HINT] use 'addon info bundle --bundleSymbolicName' to see details about a search result
[HINT] use 'addon install bundle --bundleSymbolicName' to install a specific add-on version
Located add-on that may offer this JDBC driver
JDBC driver not available for 'com.mysql.jdbc.Driver'
roo>
```

Figura A4.3

Si da un error diciendo que el driver JDBC no está disponible, se instala con:

```
addon install bundle --bundleSymbolicName
org.springframework.roo.wrapping.mysql-connector-java;
```

```
roo> addon install bundle --bundleSymbolicName org.springframework.roo.wrapping.mysql-connector-java;
Target resource(s):
--------------------
   Spring Roo - Wrapping - mysql-connector-java (5.1.18.0002)

Deploying...
done.

Starting org.springframework.roo.wrapping.mysql-connector-java; id: 86 ...
Started!

Successfully installed add-on: Spring Roo - Wrapping - mysql-connector-java [version: 5.1.18.0002]
roo>
```

Figura A4.4

Ahora sí, se realiza la ingeniería inversa correctamente:

```
roo> database reverse engineer --schema tellme --package ~.domain --testAutomatically
Created ROOT\src\main\resources\dbre.xml
Updated ROOT\pom.xml
Updated ROOT\src\main\resources\META-INF\persistence.xml
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmActivelocations.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmAdmins.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLanguages.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLocations.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmPayments.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmRols.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmTrend.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmTrends.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmUsers.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmActivelocations.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmActivelocationsDataOnDemand.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmActivelocationsIntegrationTest.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmAdmins.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmAdminsDataOnDemand.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmAdminsIntegrationTest.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLanguages.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmLanguagesDataOnDemand.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmLanguagesIntegrationTest.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLocations.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmLocationsDataOnDemand.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmLocationsIntegrationTest.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmPaymentsDataOnDemand.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmPaymentsIntegrationTest.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmRols.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmRolsDataOnDemand.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmRolsIntegrationTest.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmTrend.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmTrendDataOnDemand.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmTrendIntegrationTest.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmTrends.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmTrendsDataOnDemand.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmTrendsIntegrationTest.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmUsers.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmUsersDataOnDemand.java
Created SRC_TEST_JAVA\com\uoc\tellmetweet\domain\TmUsersIntegrationTest.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmActivelocations_Roo_Configurable.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmActivelocations_Roo_Jpa_Entity.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmActivelocations_Roo_ToString.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmActivelocations_Roo_DbManaged.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmActivelocations_Roo_Jpa_ActiveRecord.a
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmAdmins_Roo_Configurable.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmAdmins_Roo_Jpa_Entity.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmAdmins_Roo_ToString.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmAdmins_Roo_DbManaged.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmAdmins_Roo_Jpa_ActiveRecord.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLanguages_Roo_Configurable.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLanguages_Roo_Jpa_Entity.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLanguages_Roo_ToString.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLanguages_Roo_DbManaged.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLanguages_Roo_Jpa_ActiveRecord.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLocations_Roo_Configurable.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLocations_Roo_Jpa_Entity.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLocations_Roo_ToString.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLocations_Roo_DbManaged.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLocations_Roo_Jpa_ActiveRecord.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmPayments_Roo_Configurable.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmPayments_Roo_Jpa_Entity.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmPayments_Roo_ToString.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmPayments_Roo_DbManaged.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmPayments_Roo_Jpa_ActiveRecord.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmRols_Roo_Configurable.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmRols_Roo_Jpa_Entity.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmRols_Roo_ToString.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmRols_Roo_DbManaged.aj
```

Figura A4.5

Y se crea todo el dominio.

```
json all
```

```
roo> json all
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLocations.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmPayments.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmTrend.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmAdmins.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLanguages.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmTrends.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmRols.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmUsers.java
Updated SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmActivelocations.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmTrends_Roo_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLocations_Roo_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmActivelocations_Roo_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmTrend_Roo_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmPayments_Roo_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmRols_Roo_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmUsers_Roo_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmLanguages_Roo_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\domain\TmAdmins_Roo_Json.aj
roo>
```

Figura A4.6

Añade los métodos y clases necesarias para trabajar con objetos Json.

Para crear las capas de la arquitectura que proporcionan los servicios Rest Json, hay que escribir:

```
web mvc json setup
```

```
roo> web mvc json setup
Created SRC_MAIN_WEBAPP\WEB-INF\spring
Created SRC_MAIN_WEBAPP\WEB-INF\spring\webmvc-config.xml
Created SRC_MAIN_WEBAPP\WEB-INF\web.xml
Updated SRC_MAIN_WEBAPP\WEB-INF\web.xml
Updated ROOT\pom.xml [added dependency org.springframework:spring-webmvc:${spring.version}]
roo>
```

Figura A4.7

Por último solo queda generar la capa controladora de la aplicación usada para la interacción entre las entidades y la capa de presentación.

```
web mvc json all --package ~.web
```

```
roo> web mvc json all --package ~.web
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmActivelocationsController.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmAdminsController.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmLanguagesController.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmLocationsController.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmPaymentsController.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmRolsController.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmTrendController.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmTrendsController.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmUsersController.java
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmPaymentsController_Roo_Controller_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmUsersController_Roo_Controller_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmTrendController_Roo_Controller_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmTrendsController_Roo_Controller_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmRolsController_Roo_Controller_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmAdminsController_Roo_Controller_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmLanguagesController_Roo_Controller_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmActivelocationsController_Roo_Controller_Json.aj
Created SRC_MAIN_JAVA\com\uoc\tellmetweet\web\TmLocationsController_Roo_Controller_Json.aj
roo>
```

Figura A4.8

Con esto ya está el backend creado y si se inicia la aplicación ya se puede llamar a los servicios y obtener datos de la base de datos.

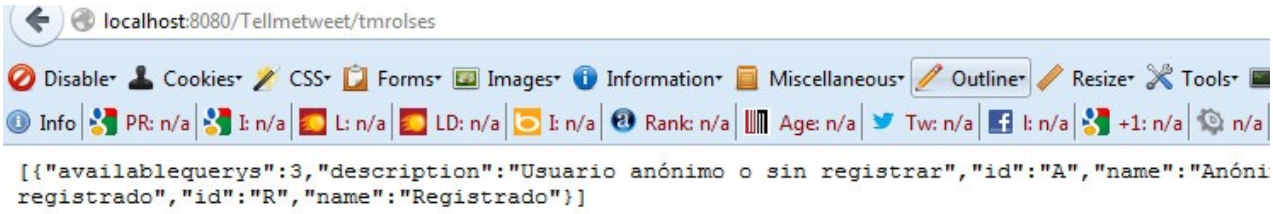


Figura A4.9

En la figura A4.9 puede verse como se llama a un servicio Rest para obtener los roles de los usuarios y este los devuelve en un Json que contiene los roles.

Spring Roo permite poner en marcha el backend de la aplicación ejecutando siete comandos.

Anexo 4.2 Puesta en marcha del frontend

Para poner en marcha la parte frontend es necesario ejecutar dos comandos mediante la utilidad Sencha CMD, hay que situarse en la carpeta del proyecto.

```
cd workspace\tellmetweet\src\tellmetweet
```

```
M:\workspace\tellmetweet>sencha generate app -ext -classic
Tellmetweet src\tellmetweet
```

```
M:\workspace\tellmetweet>sencha generate app -ext -classic tellmetweet src\tellmetweet
Sencha Cmd v6.0.1.76
[INF] Copying framework to M:\workspace\tellmetweet\src\tellmetweet\ext
[INF] Processing Build Descriptor : default
[INF] Starting server on port : 1841
[INF] Mapping http://localhost:1841/~cmd to M:\software\Sencha\Cmd\6.0.1.76...
[INF] Mapping http://localhost:1841/ to M:\workspace\tellmetweet\src\tellmetweet...
[INF] Application available at http://localhost:1841
[INF] Loading app json manifest...
[INF] Appending content to M:\workspace\tellmetweet\src\tellmetweet\bootstrap.js
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\bootstrap.json
[INF] merging 248 input resources into M:\workspace\tellmetweet\src\tellmetweet\build\deve
[INF] merged 248 resources into M:\workspace\tellmetweet\src\tellmetweet\build\development
[INF] merging 12 input resources into M:\workspace\tellmetweet\src\tellmetweet\build\devel
[INF] merged 11 resources into M:\workspace\tellmetweet\src\tellmetweet\build\development\
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\sass\example\bootstrap.j
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\sass\example\bootstrap.j
```

Figura A4.10

El comando sencha generate app, lo que hace es descargar ExtJs 6, y copiarlo en src \tellmwtweet\ext, al terminar de copiar los archivos se coloca en la ruta M:\workspace \tellmetweet\src\tellmetweet, ahora se ejecuta:

```
sencha app build --destination ..\main\webapp testing
```

```
M:\workspace\tellmetweet\src\tellmetweet>sencha app build --destination ..\main\webapp testing
Sencha Cmd v6.0.1.76
[INF] Processing Build Descriptor : default
[INF] Starting server on port : 1841
[INF] Mapping http://localhost:1841/~cmd to M:\software\Sencha\Cmd\6.0.1.76...
[INF] Mapping http://localhost:1841/ to M:\workspace\tellmetweet\src\tellmetweet...
[INF] Application available at http://localhost:1841
[INF] Loading app json manifest...
[INF] Appending content to M:\workspace\tellmetweet\src\tellmetweet\bootstrap.js
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\bootstrap.json
[INF] Writing concatenated output to file M:\workspace\tellmetweet\src\main\webapp\app.js
[INF] merging 248 input resources into M:\workspace\tellmetweet\src\main\webapp\resources
[INF] merged 248 resources into M:\workspace\tellmetweet\src\main\webapp\resources
[INF] merging 12 input resources into M:\workspace\tellmetweet\src\main\webapp
[INF] merged 11 resources into M:\workspace\tellmetweet\src\main\webapp
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\sass\example\bootstrap.json
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\sass\example\bootstrap.js
[INF] writing sass content to M:\workspace\tellmetweet\src\tellmetweet\build\temp\testing\tellmetweet\slicer-temp\tellmetweet-example.scss.tmp
[INF] writing sass content to M:\workspace\tellmetweet\src\tellmetweet\build\temp\testing\tellmetweet\slicer-temp\config.rb
[INF] Building ..\..\build\temp\testing\tellmetweet\slicer-temp\tellmetweet-example.scss
[INF] Slicing images...
[INF] Slicing complete - generated 202 images
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\sass\example\bootstrap.json
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\sass\example\bootstrap.js
[INF] writing sass content to M:\workspace\tellmetweet\src\tellmetweet\build\temp\testing\tellmetweet\sass\tellmetweet-all.scss.tmp
[INF] appending sass content to M:\workspace\tellmetweet\src\tellmetweet\build\temp\testing\tellmetweet\sass\tellmetweet-all.scss.tmp
[INF] appending sass content to M:\workspace\tellmetweet\src\tellmetweet\build\temp\testing\tellmetweet\sass\tellmetweet-all.scss.tmp
[INF] writing sass content to M:\workspace\tellmetweet\src\tellmetweet\build\temp\testing\tellmetweet\sass\config.rb
[INF] Using base bref : M:\workspace\tellmetweet\src\tellmetweet/build/temp/testing/tellmetweet/slicer-temp/tellmetweet-example.css/
[INF] Build complete for ..\..\build\temp\testing\tellmetweet\slicer-temp\tellmetweet-example.scss
[INF] Building ..\..\build\temp\testing\tellmetweet\sass\tellmetweet-all.scss
[INF] Application available at http://localhost:1841
[INF] Appending content to M:\workspace\tellmetweet\src\tellmetweet\bootstrap.js
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\bootstrap.json
[INF] Copying page resources to M:\workspace\tellmetweet\src\main\webapp
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\..\main\webapp\app.json
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\..\main\webapp\microloader.js
[INF] Appending content to M:\workspace\tellmetweet\src\tellmetweet\..\main\webapp\microloader.js
[INF] Building output markup to M:\workspace\tellmetweet\src\tellmetweet\..\main\webapp/index.html
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\..\main\webapp\index.html

M:\workspace\tellmetweet\src\tellmetweet>
```

Figura A4.11

El comando sencha app build, construye la aplicación, y con esto ya está el front preparado, se puede ver ejecutando:

sencha app watch

```
M:\workspace\tellmetweet\src\tellmetweet>sencha app watch
Sencha Cmd v6.0.1.76
[INF] Processing Build Descriptor : default
[INF] Starting server on port : 1841
[INF] Mapping http://localhost:1841/~cmd to M:\software\Sencha\Cmd\6.0.1.76...
[INF] Mapping http://localhost:1841/ to M:\workspace\tellmetweet\src\tellmetweet...
[INF] Application available at http://localhost:1841
[INF] Application available at http://localhost:1841
[INF] Loading app json manifest...
[INF] Appending content to M:\workspace\tellmetweet\src\tellmetweet\bootstrap.js
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\bootstrap.json
[INF] merging 248 input resources into M:\workspace\tellmetweet\src\tellmetweet\build\development\tellmetweet\resources
[INF] merged 1 resources into M:\workspace\tellmetweet\src\tellmetweet\build\development\tellmetweet\resources
[INF] merging 12 input resources into M:\workspace\tellmetweet\src\tellmetweet\build\development\tellmetweet
[INF] merged 1 resources into M:\workspace\tellmetweet\src\tellmetweet\build\development\tellmetweet
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\sass\example\bootstrap.json
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\sass\example\bootstrap.js
[INF] writing sass content to M:\workspace\tellmetweet\src\tellmetweet\build\temp\development\tellmetweet\sass\tellmetweet-all.scss.tmp
[INF] appending sass content to M:\workspace\tellmetweet\src\tellmetweet\build\temp\development\tellmetweet\sass\tellmetweet-all.scss.tmp
[INF] appending sass content to M:\workspace\tellmetweet\src\tellmetweet\build\temp\development\tellmetweet\sass\tellmetweet-all.scss.tmp
[INF] Building ..\..\build\temp\development\tellmetweet\sass\tellmetweet-all.scss
[INF] Application available at http://localhost:1841
[INF] Appending content to M:\workspace\tellmetweet\src\tellmetweet\bootstrap.js
[INF] Writing content to M:\workspace\tellmetweet\src\tellmetweet\bootstrap.json
[INF] Application available at http://localhost:1841
[INF] Waiting for changes...
```

Figura A4.11

El comando sencha app watch, lanza un servidor que se queda escuchando en el puerto 1841, como se puede ver en la figura A4.11 al final pone “Waiting for changes…” esto quiere decir que cualquier cambio que se realice en el front, será procesado automáticamente y solo se tendrá que refrescar la página para verlo. Puede verse el front de ExtJs funcionando accediendo al puerto 1841 de la máquina donde se ha lanzado el comando.

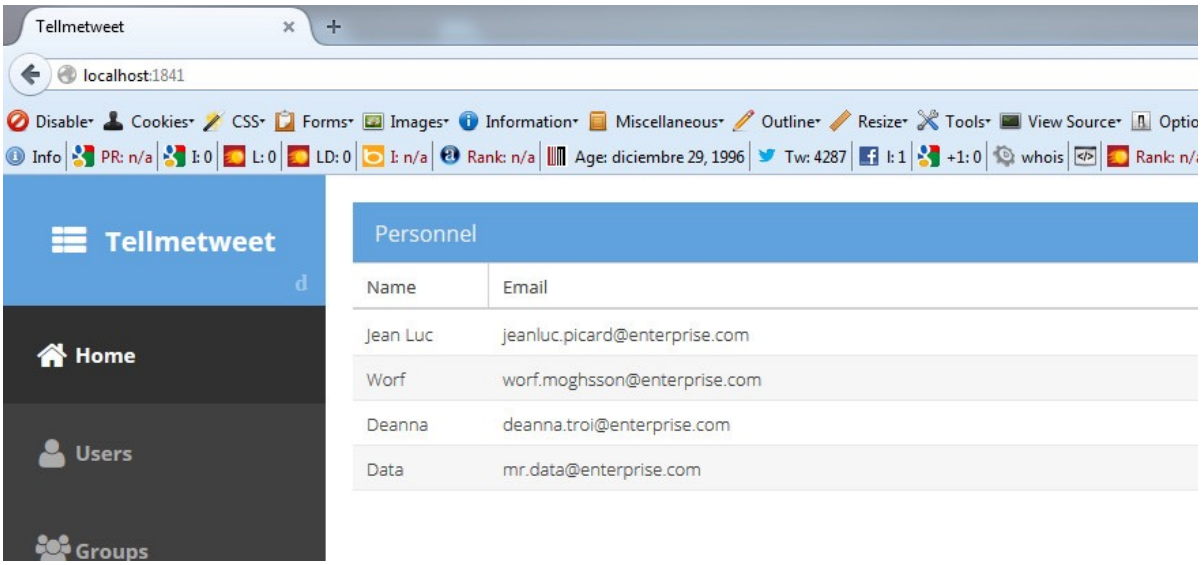


Figura A4.12

Sencha crea un layout inicial de ejemplo, con varias pestañas y datos de prueba ficticios de usuarios.

La ventaja de usar watch como servidor frente a un servidor de aplicaciones como Tomcat, es que una vez que watch se encuentra escuchando, es muy rápido trabajar en el front de la aplicación, watch compila automáticamente todos los cambios que detecta, por lo que para ver los cambios realizados solo es necesario refrescar la pantalla ahorrando tener que hacer un Refresh y Clean & Build como en Tomcat.

ANNEXO 5. GUIA DE USUARIO

Anexo 5.1 Perfiles de usuario

Existen 3 perfiles de usuario:

- Anónimo, o sin registrar: puede acceder a algunas pantallas de la aplicaciónn, no tiene localizaciones favoritas, en su defecto se cargarán Valencia, Madrid y Barcelona.
- Registrado: Puede acceder a todas las pantallas de la aplicación, puede modificar las localizaciones favoritas una vez por día, el momento del registro cuenta como una modificación, puede realizar cinco consultas diarias en las pantallas “Trending Topics” y “Chart”.
- Premium: Puede acceder a todas las pantallas de la aplicación, puede modificar las localizaciones favoritas ilimitadamente, puede realizar veinte consultas diarias en las pantallas “Trending Topics” y “Chart”, se muestra el logotipo “Premium” en letras blancas con fondo azul al lado del usuario.

Figura A5.1

Anexo 5.2 Validación de campos

Los campos de los formularios validan que se introduzcan datos correctos, las validaciones son realizadas conforme se van rellenando los campos, no permitiendo introducir números ni caracteres raros cuando se requiere texto, en caso de no ser valido el campo se remarca en color rojo y si se pasa por encima el puntero del ratón se muestra un tooltip con información sobre el error, además el botón correspondiente para enviar el formulario o grabar la información no se activará hasta que los datos introducidos sean validos.

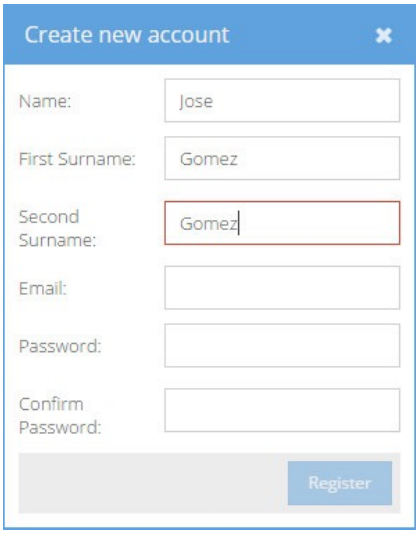


Figura A5.2

Anexo 5.3 Número de consultas diarias

Si se alcanza el número de consultas diarias, el usuario no podrá realizar más consultas hasta el siguiente día que será restablecido el numero de consultas disponibles.

Anexo 5.4 Trend List

Se pueden ordenar los resultados pinchando en la cabecera de las columnas.

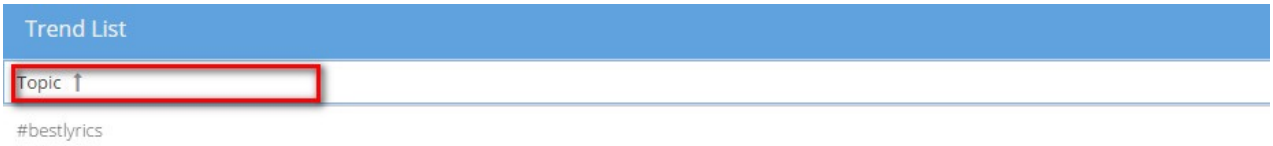


Figura A5.4.1

El paginador muestra por orden:

- Pagína actual: esta se puede modificar, introduciendo la nueva página en el campo de texto o mediante los botones con los símbolos <> situados en su lateral.
- Número total de páginas recuperadas.
- Número de resultados mostrados.
- Número total de resultados recuperados.

Figura A5.4.2

Anexo 5.5 Mapa

Se puede ampliar y reducir el mapa desplazando la rueda del ratón, o mediante los botones situados en la parte superior izquierda.

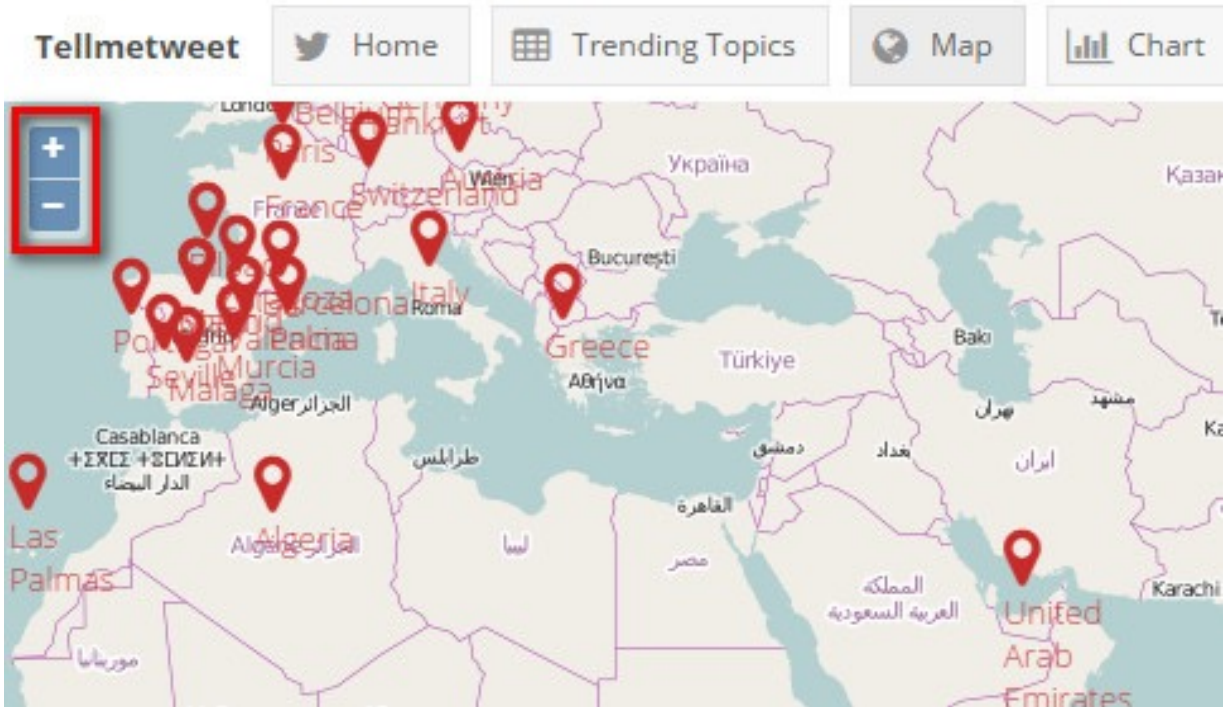


Figura A5.5.1

El mapa se desplaza moviendo el ratón a la vez que se mantiene presionado cualquier botón o la rueda central.

Al pasar el puntero del ratón por encima de un marcador, el icono a la forma de una mano, que indica que podemos hacer click para que aparezca el popup que muestra los últimos trending topics de la localización seleccionada, el popup se cierra haciendo click en el botón circular con una x blanca en su interior.

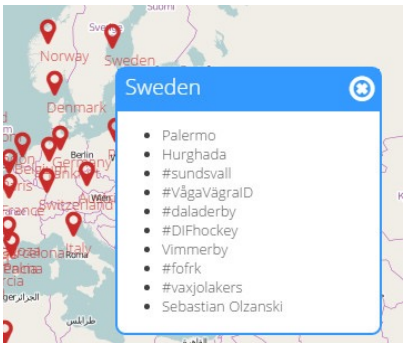


Figura A5.5.2

Anexo 5.6 Chart

El eje izquierdo indica el número de horas que ha sido Trending Topic el Tweet.

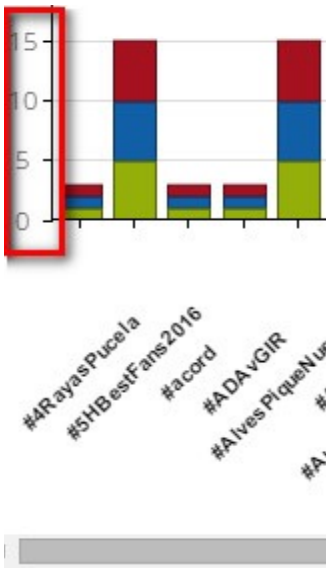


Figura A5.6.1

El eje inferior muestra el nombre de los Trending Topics.

El scroll situado en la parte inferior permite desplazarse por la gráfica, mediante la leyenda situada en la parte derecha se pueden activar y desactivar las categorías de la gráfica.

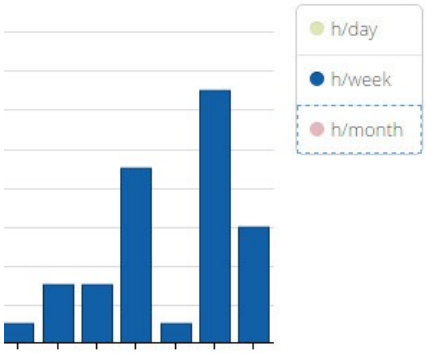


Figura A5.6.2

Anexo 5.7 Localizaciones favoritas

Mediante el botón de configuración.

Figura A5.7

Es posible modificar las localizaciones que aparecen en la pantalla principal de la aplicación, una vez al día para los usuarios registrados e ilimitadamente para los usuarios con cuenta Premium.

ANNEXO 6 LIBRO DE ESTILO

Icono y logotipo

Figura A6.1

Paleta de colores

#5FA2DD

#D900FF

#CF4C35

#343434

#404040

#919191

#F6F6F6

#FFFFFF

Paleta tipográfica

Open Sans, Helvetica 16px

Open Sans, Helvetica 16px

Open Sans, Helvetica 13px

Open Sans, Helvetica 12px

Margenes

10px

Iconos

Generados con la fuente Awesome

· Unicode: f099 · Unicode: f0ce · Unicode: f0ac

· Unicode: f080 · Unicode: f040 · Unicode: f085

$ · Unicode: f155 · Unicode: f011 · Unicode: f090

ANNEXO 8. GLOSARIO / ÍNDICE ANALÍTICO

Backend: Contiene los servicios, y demás lógica de la aplicación necesaria para realizar operaciones con la base de datos, los resultados obtenidos son entregados al frontend para que los represente en pantalla.

Bo(Business Object): Capa que contiene la lógica de negocio, contiene las validaciones o reglas de negocio.

Cron: Servicio Linux que sirve para programar y ejecutar tareas automáticamente.

Dao(Direct Access Object): Es la capa de persistencia que tiene acceso a la base de datos.

Dto(Data Transfer Object): Objeto usado para la transferencia de datos entre capas.

Frontend: Es la parte visual de la aplicación, es la que muestra lo que el usuario ve y con la que interactua.

Jpa(Java Persistence Api): El Api de persistencia de Java se encarga de manejar los datos relacionales de la base de datos.

Location: Termino empleado por Twitter para referirse a una localización geográfica.

ANNEXO 9. BIBLIOGRAFÍA

Guias ExtJs - http://docs.sencha.com/extjs/6.0

Api ExtJs 6.01 - http://docs.sencha.com/extjs/6.0/6.0.1-classic

Spring Roo - http://projects.spring.io/spring-roo

Maven - https://maven.apache.org

Sencha Cmd - https://www.sencha.com/products/sencha-cmd

Documentación Eclipse - http://help.eclipse.org/mars/index.jsp?nav=%2F0

Twitter4J - http://twitter4j.org/en/index.html

localStorage - https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage

Building a Login System - http://docs.sencha.com/extjs/6.0/getting_started/login_app.html

GeoExt 3 - https://geoext.github.io/geoext3

Paypal developer - https://developer.paypal.com/developer/accounts/

Local storage - https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage

Building a Login System - http://docs.sencha.com/extjs/6.0/getting_started/login_app.html