



**UNIVERSIDADE DA CORUÑA**

## **Informe de Resultados y Cierre de Proyecto**

*Versión:4.0*  
*Fecha: 17/12/2014*



**Tuiter 11**  
**Informe de resultados y cierre de proyecto.**

HOJA DE CONTROL

<b>Organismo</b>			
<b>Proyecto</b>	Tuiter 11		
<b>Entregable</b>	Informe de resultados y cierre de proyecto		
<b>Autor</b>	René Balay Lorenzo Samuel Paredes Sánchez Xoán Iago Suárez Canosa		
<b>Versión/Edición</b>	4.0	<b>Fecha Versión</b>	DD/MM/AA AA
<b>Aprobado por</b>	René Balay Lorenzo	<b>Fecha Aprobación</b>	DD/MM/AA AA
		<b>Nº Total de PáginaS</b>	15



[1. INTRODUCCIÓN](#)

[1.1 Objetivo](#)

[1.2 Alcance](#)

[2. DESCRIPCIÓN DEL PROYECTO](#)

[2.1 Equipo de trabajo](#)

[2.2 Entregables](#)

[2.3 Funcionalidades](#)

[2.4 Metodología](#)

[2.5 Herramientas utilizadas](#)

[3 CARACTERÍSTICAS DEL PROYECTO](#)

[3.1 Version 1.0](#)

[3.2 Version 2.0](#)

[3.3 Version 3.0](#)

[3.4 Version 4.0](#)

[4 COSTE DEL PROYECTO](#)

[5 MANUAL DE USUARIO](#)

[6 ANEXOS](#)



# 1. INTRODUCCIÓN

## 1.1 Objetivo

El objetivo de este proyecto ha sido el correcto empleo de las distintas herramientas existentes utilizadas en la asignatura de *Ferramentas de Desenvolvimento* para facilitar el desarrollo de aplicaciones. En este caso, y como se explicará en el apartado “2. Descripción del Proyecto”, dichas herramientas han servido para el desarrollo de una aplicación web basada en Twitter.

## 1.2 Alcance

La aplicación ha sido modelada con una funcionalidad básica que permite publicar tuits, retuitear, y guardar tuits en favoritos, así como también tener conversaciones con varios usuarios y varias funcionalidades de búsqueda (búsqueda de usuarios y palabras clave). Además también proporciona registro y login de usuarios y funcionalidades propias de un administrador de una app web. Dado los recursos de los que se ha dispuesto (3 personas y entre 1 y 2 semanas para cada release), el proyecto se considera pequeño.

# 2. DESCRIPCIÓN DEL PROYECTO

Tuiter11 es una aplicación web basada en la famosa red social *Twitter*. La aplicación permite enviar mensajes de texto plano, que se muestran en la página principal del usuario. Los usuarios pueden seguir a otros usuarios, mostrándose así en su timeline las publicaciones de éstos. Los tuits son públicos, a excepción de los usuarios con cuentas privadas, en los que sólo se verán si hay un seguimiento mutuo entre dos usuarios. Los usuarios también pueden bloquear a otros usuarios para que no puedan ver sus publicaciones. También podemos guardar tuits como favoritos para poder visualizarlos fácilmente en el momento que se desee. Por último, la aplicación permite hacer retuit de una publicación, apareciendo de esta forma como tuit del usuario que lo ha retuiteado, pero siempre haciendo referencia al usuario de la publicación original.

A mayores, y como se ha comentado anteriormente, se le permite a los usuarios buscar a otros usuarios mediante palabras clave y ver sus perfiles, así como editar la información de los suyos propios, y además buscar tuits por palabras clave.



## 2.1 Equipo de trabajo

El equipo de trabajo está compuesto por 3 miembros:

- Rene Balay Lorenzo
- Samuel Paredes Sánchez
- Xoan Iago Suárez Canosa

## 2.2 Entregables

El proyecto realizado se ha dividido en 4 versiones :

Número de versión	Fecha Inicio	Fecha Fin
1.0	25/09/2014	10/10/2014
2.0	16/10/2014	31/10/2014
3.0	06/11/2014	28/11/2014
4.0	04/12/2014	19/12/2014

## 2.3 Funcionalidades

Las funcionalidades implementadas se han incorporado al sistema de manera iterativa e incremental, quedando de la siguiente forma (ver [apartado 3](#) para explicación de dichas funcionalidades):

**V1.0:** Gestión de usuarios, añadir nick al usuario, publicar tuits, ver tuits propios, almacenar tuits en MongoDB.

**V2.0:** Seguir/Dejar de seguir usuario, mostrar timeline, cuentas privadas, bloquear usuario, crear maven site del proyecto.

**V3.0:** Supervisión del administrador en altas de usuario y en tuits, alerta de palabras ofensivas, retuits, favoritos, conversaciones.

**V4.0:** Desarrollo de un servicio REST de búsqueda de tuits con el cajetín de búsqueda correspondiente en la app web y realización de pruebas de carga sobre dicho servicio.



## 2.4 Metodología

Se ha utilizado una metodología “mixta”, ya que se han aplicado prácticas de SCRUM y eXtreme Programming. Esta decisión viene tomada debido al corto período de entrega de releases, el “Product Backlog” definido previamente, las reuniones diarias del equipo de trabajo, el trabajo mediante “pair programming”, la aplicación de TDD para algunas funcionalidades, etc.

## 2.5 Herramientas utilizadas

Para el diseño e implementación de la aplicación hemos usado la plataforma JavaEE, así como distintos frameworks de dicha plataforma como lo son Spring y Spring MVC, Hibernate, tecnologías para la capa web como Thymeleaf, y bases de datos en memoria como HSQLDB para guardar la información relativa a los usuarios y NoSQL como MongoDB para la persistencia de tuits.

Como entorno de desarrollo integrado (IDE) para el desarrollo de la aplicación, se ha escogido Eclipse Luna con diferentes plugins que nos faciliten las tareas llevadas a cabo. Dichos plugins son algunos como EGit (con su respectivo conector m2e) para la integración con el repositorio Git proporcionado por terceros, mylyn-redmine-connector para la integración con Redmine, MonjaDB para la visualización de los datos de MongoDB, etc.

Para el control de versiones del proyecto hemos usado Git. Nos proporciona mantenimiento de las versiones de nuestra aplicación con rapidez y eficiencia.

Para la gestión del proyecto hemos usado la herramienta Redmine, que incluye, entre otras funcionalidades, un sistema de seguimiento, calendario de actividades, diagramas de Gantt para la representación visual de la línea del tiempo del proyecto, repositorio de control de versiones, control del flujo de trabajo basado en roles, ...

Ha sido de gran utilidad para la gestión y organización del trabajo en base a peticiones que luego pueden ser sincronizadas con el IDE en base al plugin anteriormente indicado y así poder “cronometrar” el tiempo dedicado a cada tarea.

Otra herramienta útil en el desarrollo del proyecto ha sido Jenkins, un software open-source que se ha utilizado para dotar de integración continua a éste mediante su conexión con el repositorio Git de la aplicación.



Por último, para la gestión de calidad del código hemos usado la herramienta SonarQube, que nos permite gestionar la calidad del código. Entre otras cosas, nos permite visualizar parámetros de vital importancia como pueden ser:

- Duplicidad de código fuente.
- Cobertura de código fuente por pruebas unitarias.
- Complejidad ciclomática.
- Violaciones críticas y de importancia mayor en el código fuente

## 3 CARACTERÍSTICAS DEL PROYECTO

Como hemos mencionado en el apartado anterior hemos dividido el proyecto en 4 versiones, en las cuáles cada una tiene una serie de tareas asignadas. En cada versión cada miembro del equipo de trabajo ha tenido un rol específico (Desarrollador o Jefe de Proyecto).

### 3.1 Version 1.0

En la primera versión del proyecto, hemos usado, además de las herramientas de desarrollo y de control de versiones, la herramienta Redmine, que como hemos indicado anteriormente nos ayuda a hacer un seguimiento de cerca sobre las tareas realizadas.

- Roles:

Rene Balay Lorenzo	Jefe de Proyecto
Samuel Paredes Sánchez	Desarrollador
Xoan Iago Suárez Canosa	Desarrollador

- Peticiones realizadas:

En base a las tareas descritas en el apartado anterior, hemos realizado una serie de peticiones, repartidas de manera equitativa por el jefe de proyecto entre los miembros del equipo de trabajo:



## Tuiter 11

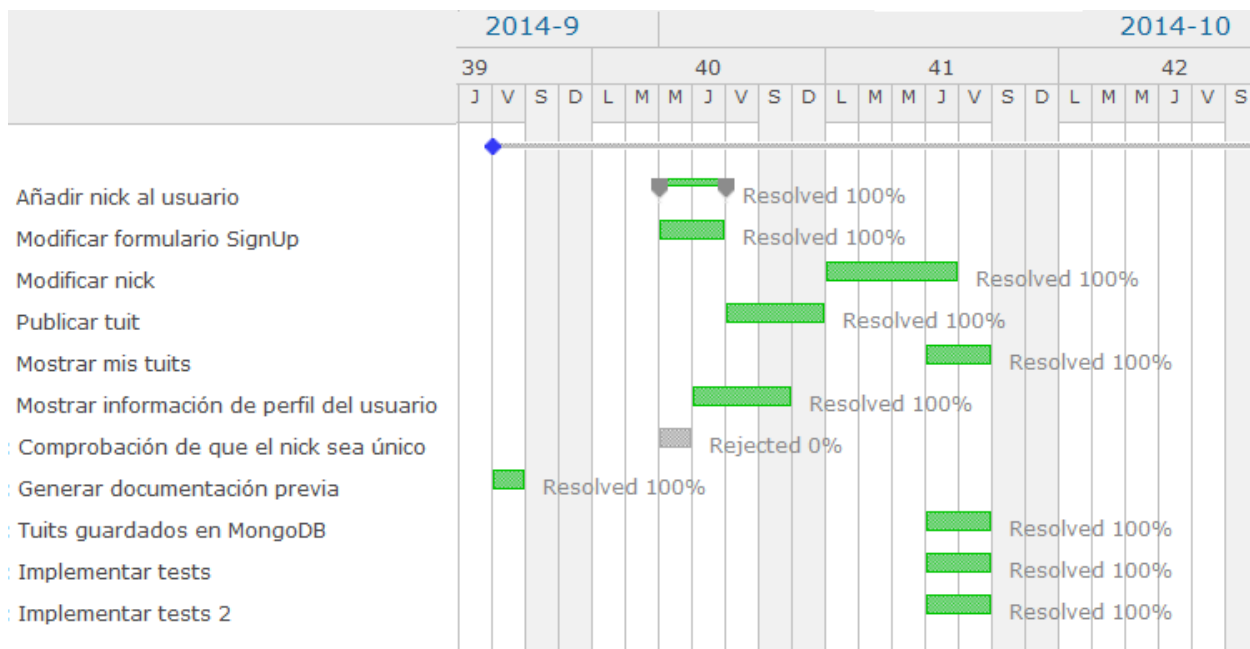
### Informe de resultados y cierre de proyecto.

#	Tipo	Estado	Prioridad	Asunto	Asignado a
898	Support	Resolved	High	Implementar tests 2	René Balay Lorenzo
897	Support	Resolved	High	Implementar tests	Samuel Paredes Sánchez
878	Support	Resolved	Normal	Tuits guardados en MongoDB	Xoan Iago Suárez Canosa
801	Bug	Resolved	Normal	Modificar formulario SignUp	René Balay Lorenzo
779	Support	Resolved	Urgent	Generar documentación previa	Xoan Iago Suárez Canosa
778	Feature	Rejected	High	Comprobación de que el nick sea único	
777	Feature	Resolved	Normal	Mostrar información de perfil del usuario	Samuel Paredes Sánchez
776	Feature	Resolved	High	Mostrar mis tuits	Xoan Iago Suárez Canosa
775	Feature	Resolved	High	Publicar tuit	René Balay Lorenzo
774	Feature	Resolved	High	Modificar nick	Samuel Paredes Sánchez
773	Feature	Resolved	Normal	Añadir nick al usuario	René Balay Lorenzo

Todas las peticiones han sido resueltas satisfactoriamente, a excepción de la petición 778, debido a que ya se había resuelto con la terminación de la tarea 773.

Cabe indicar también que se ha hecho una mala gestión de las tareas 897 y 898, puesto que deberían estar fusionadas en una única tarea, o en su defecto, haber asignado un nombre más preciso a dichas tareas.

- Diagrama de Gantt: En la siguiente imagen mostramos el diagrama de gantt que permite visualizar la línea base de ésta versión:



Las dependencias entre tareas no están asignadas debido a que en esta versión el jefe de proyecto lo ha omitido.

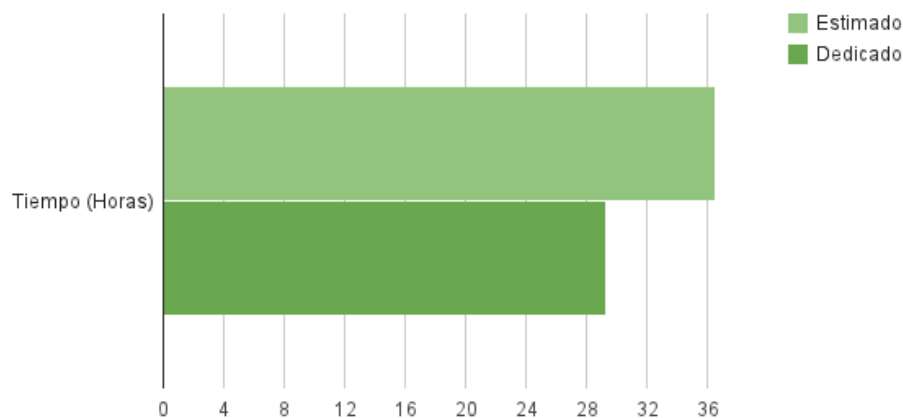




## Tuiter 11

### Informe de resultados y cierre de proyecto.

- Desviaciones:



El tiempo total estimado para esta versión ha sido 36.5 horas, y le hemos dedicado 29.3, por tanto nos ha llevado menos tiempo de lo previsto. Esto es debido a una falta de imputación de horas en una petición (en concreto la petición “Mostrar información de perfil de usuario”).

Hemos usado este gráfico y no el de Redmine debido a que en esta versión el jefe de proyecto ha pasado por alto el hecho de crear una nueva versión y asignarle las tareas correspondientes.

- Factores de calidad:

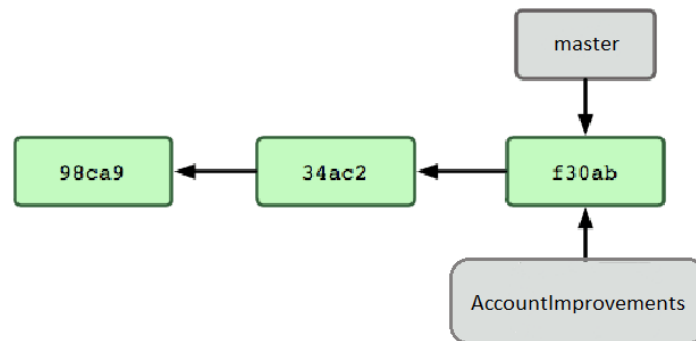
No hemos tenido en cuenta los factores de calidad debido a que aún no teníamos puesto en marcha la herramienta Sonar.



## 3.2 Version 2.0

En esta versión hemos integrado la herramienta Jenkins, que nos permite hacer integración continua de nuestro proyecto.

Además, hemos creado una nueva rama a partir de la rama master para implementar las funcionalidades relativas a las cuentas de usuario.



- Roles:

Rene Balay Lorenzo	Desarrollador
Samuel Paredes Sánchez	Desarrollador
Xoan Iago Suárez Canosa	Jefe de Proyecto

- Peticiones realizadas:  
En base a las tareas descritas en el apartado 2.3, hemos realizado una serie de peticiones, repartidas de manera equitativa por el jefe de proyecto entre los miembros del equipo de trabajo:



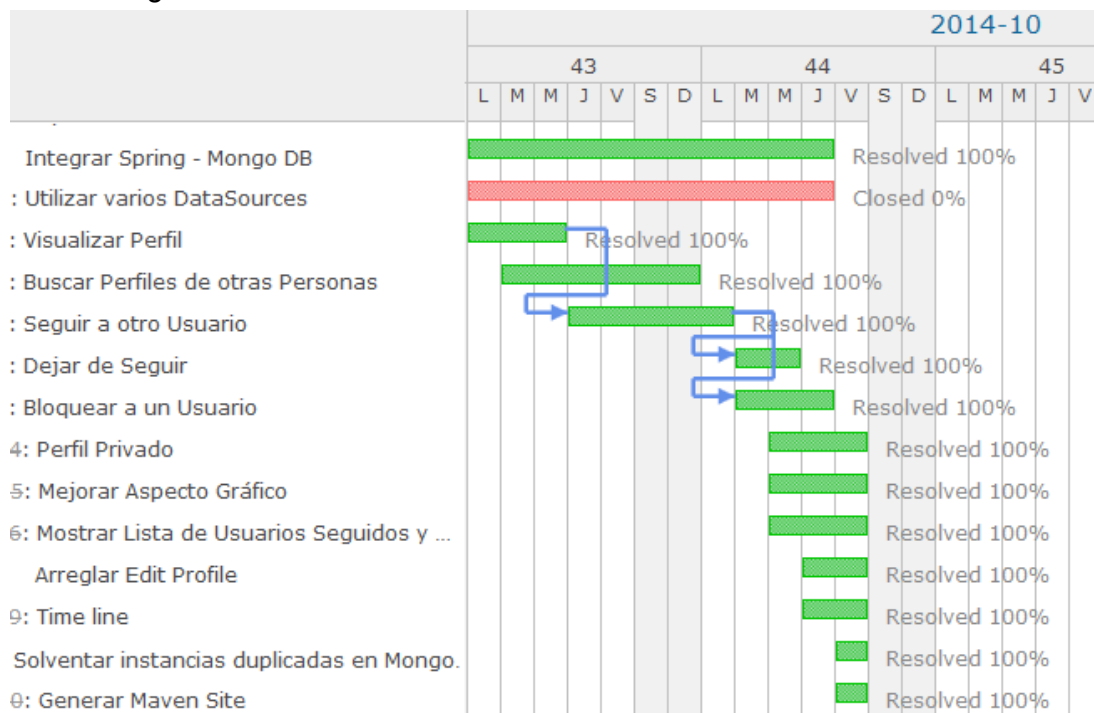
## Tuiter 11

### Informe de resultados y cierre de proyecto.

#	Tipo	Estado	Prioridad	Asunto	Asignado a
1070	Feature	Resolved	Normal	Generar Maven Site	
1068	Bug	Resolved	Normal	Solventar instancias duplicadas en MongoDB - HSQLDb	Xoan Iago Suárez Canosa
1059	Feature	Resolved	Urgent	Time line	Xoan Iago Suárez Canosa
1058	Bug	Resolved	High	Arreglar Edit Profile	René Balay Lorenzo
1046	Feature	Resolved	Normal	Mostrar Lista de Usuarios Seguidos y de Seguidores	Xoan Iago Suárez Canosa
1045	Support	Resolved	Normal	Mejorar Aspecto Gráfico	René Balay Lorenzo
1044	Feature	Resolved	High	Perfil Privado	René Balay Lorenzo
939	Feature	Resolved	High	Bloquear a un Usuario	Xoan Iago Suárez Canosa
938	Feature	Resolved	Normal	Dejar de Seguir	Samuel Paredes Sánchez
937	Feature	Resolved	High	Seguir a otro Usuario	Samuel Paredes Sánchez
936	Feature	Resolved	High	Buscar Perfiles de otras Personas	René Balay Lorenzo
935	Feature	Resolved	High	Visualizar Perfil	Xoan Iago Suárez Canosa
934	Support	Closed	Low	Utilizar varios DataSources	René Balay Lorenzo
933	Bug	Resolved	Normal	Integrar Spring - Mongo DB	Samuel Paredes Sánchez

La tarea 934 “Utilizar varios DataSources” se ha marcado como cerrada debido a una malinterpretación de los requisitos del usuario, puesto que ya se utilizaban varios DataSources en la aplicación.

- Diagrama de Gantt:

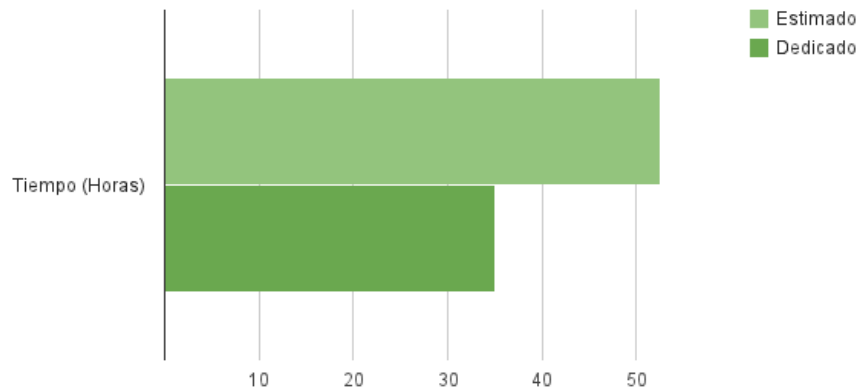




## Tuiter 11

### Informe de resultados y cierre de proyecto.

- Desviaciones:



Como podemos observar en la gráfica, vuelve a ocurrir lo mismo que en la anterior versión, falta una imputación de horas en una de las tareas (935 - Visualizar Perfil), además de que se ha estimado un poco por encima de lo que realmente ha llevado realizar dichas tareas. Al igual que en la versión anterior, hemos usado este gráfico y no el de Redmine por el mismo motivo.

- Factores de calidad:  
No hemos tenido en cuenta los factores de calidad debido a que aún no teníamos puesto en marcha la herramienta sonar.
- Integración continua:

## Resultados de los tests

1 fallidos (+1)

31 tests (+3)  
Tardó 7.5 Seg.  
[añadir descripción](#)

### Todos los test con fallos

Nombre del test	Duración	Antigüedad
<a href="#">es.udc.fi.dc.fd.tuit.TuitTest.createNewTuitAndFindIt</a>	72 Ms	1

### Todos los tests

Paquete	Duración	Fallidos	(diferencias)	Omitir	(diferencias)	Pass	(diferencias)	Total	(diferencias)
<a href="#">es.udc.fi.dc.fd.account</a>	6.4 Seg	0		0		11	+1	11	+1
<a href="#">es.udc.fi.dc.fd.block</a>	0.14 Seg	0		0		5		5	
<a href="#">es.udc.fi.dc.fd.follow</a>	0.15 Seg	0		0		5		5	
<a href="#">es.udc.fi.dc.fd.signin</a>	63 Ms	0		0		2		2	
<a href="#">es.udc.fi.dc.fd.signup</a>	0.53 Seg	0		0		3		3	
<a href="#">es.udc.fi.dc.fd.tuit</a>	0.2 Seg	1	+1	0		4	+1	5	+2



## Tuiter 11

### Informe de resultados y cierre de proyecto.

Debido a un pequeño error de configuración en el Jenkins, esta versión nos ha dado un test fallido, el cual hemos resuelto inmediatamente una vez encontrado, como se observa en la siguiente figura:

#### Resultados de los tests

0 fallidos (-1)

31 tests (±0)  
Tardó 7 Seg.  
[añadir descripción](#)

#### Todos los tests

Paquete	Duración	Fallidos	(diferencias)	Omitir	(diferencias)	Pass	(diferencias)	Total	(diferencias)
<a href="#">es.udc.fi.dc.fd.account</a>	6 Seg	0		0		11		11	
<a href="#">es.udc.fi.dc.fd.block</a>	0.14 Seg	0		0		5		5	
<a href="#">es.udc.fi.dc.fd.follow</a>	0.13 Seg	0		0		5		5	
<a href="#">es.udc.fi.dc.fd.signin</a>	38 Ms	0		0		2		2	
<a href="#">es.udc.fi.dc.fd.signup</a>	0.41 Seg	0		0		3		3	
<a href="#">es.udc.fi.dc.fd.tuit</a>	0.2 Seg	0	-1	0		5	+1	5	

### 3.3 Version 3.0

En esta versión se ha empezado a trabajar con la herramienta de gestión de calidad Sonar. Cabe destacar que en esta iteración se ha decidido desarrollar las funcionalidades correspondientes a la administración de la aplicación en una rama aparte denominada “adminfeatures”.

- Roles:

Rene Balay Lorenzo	Desarrollador
Samuel Paredes Sánchez	Jefe de Proyecto
Xoan Iago Suárez Canosa	Desarrollador



## Tuiter 11

### Informe de resultados y cierre de proyecto.

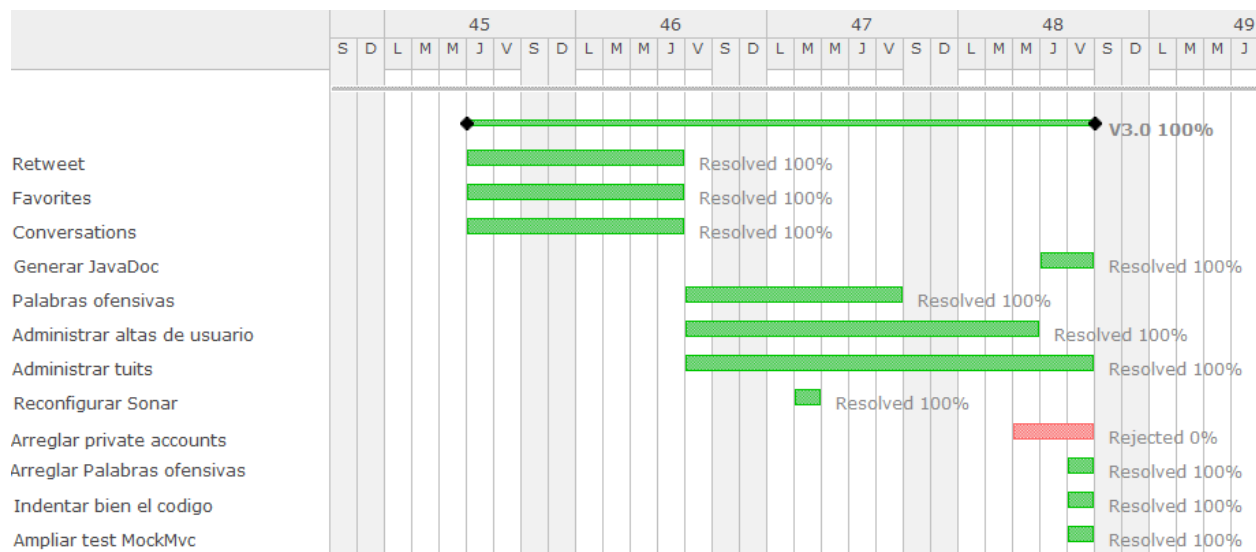
- Peticiones realizadas:

#	Tipo	Estado	Prioridad	Asunto	Asignado a
1286	Support	Resolved	High	Ampliar test MockMvc	René Balay Lorenzo
1283	Support	Resolved	Normal	Indentar bien el codigo	Xoan Iago Suárez Canosa
1282	Bug	Resolved	Urgent	Arreglar Palabras ofensivas	René Balay Lorenzo
1236	Bug	Rejected	Normal	Arreglar private accounts	
1200	Support	Resolved	Immediate	Reconfigurar Sonar	Xoan Iago Suárez Canosa
1190	Feature	Resolved	Normal	Administrar tuits	Samuel Paredes Sánchez
1189	Feature	Resolved	Normal	Administrar altas de usuario	Xoan Iago Suárez Canosa
1188	Feature	Resolved	Normal	Palabras ofensivas	René Balay Lorenzo
1121	Support	Resolved	Low	Generar JavaDoc	Samuel Paredes Sánchez
1099	Feature	Resolved	Normal	Conversations	René Balay Lorenzo
1098	Feature	Resolved	Normal	Favorites	Samuel Paredes Sánchez
1097	Feature	Resolved	Normal	Retweet	Xoan Iago Suárez Canosa

La petición 1236 - “Arreglar private accounts” ha sido rechazada debido a que se dió de alta por parte del jefe de proyecto porque se creía que su funcionamiento era incorrecto, aunque realmente estaba funcionando correctamente. Ha sido una error a causa de hacer las pruebas con un binario que no era la última versión.

Con respecto a la tarea 1200 - “Reconfigurar Sonar” hay que comentar que se definió como inmediata puesto a que había un fallo en la ejecución de los tests unitarios por parte de Sonar y se necesitó su adecuada reconfiguración a través del plugin JaCoCo.

- Diagrama de Gantt:



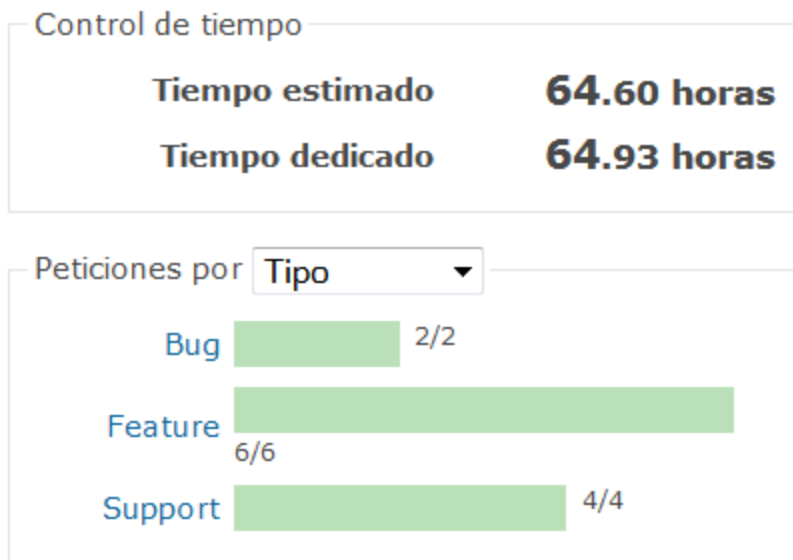


## Tuiter 11

### Informe de resultados y cierre de proyecto.

Cabe observar que no hemos marcado las dependencias entre tareas porque el jefe de proyecto ha tenido una idea ligeramente equivocada del término dependencia, y por lo tanto decidió no establecer ninguna.

- Desviaciones:



Como podemos observar en la imagen superior extraída de Redmine, esta vez hemos gestionado mejor la planificación de las tareas y nos ha llevado aproximadamente lo estimado, aunque podrían haber sido menos horas de las estimadas puesto que la tarea 1099 - “Conversations” ha consumido bastante tiempo de la planificación por problemas de funcionamiento y de comprensión de los requisitos.

- Integración continua:

### Resultados de los tests

0 fallidos (-3)

64 tests (±0)  
Tardó 13 Seg.  
[añadir descripción](#)

### Todos los tests

Paquete	Duración	Fallidos	(diferencias)	Omitir	(diferencias)	Pass	(diferencias)	Total	(diferencias)
<a href="#">es.udc.fi.dc.fd.account</a>	10 Seg	0	-1	0		19	+1	19	
<a href="#">es.udc.fi.dc.fd.block</a>	0.21 Seg	0		0		5		5	
<a href="#">es.udc.fi.dc.fd.conversations</a>	1.4 Seg	0		0		8		8	
<a href="#">es.udc.fi.dc.fd.follow</a>	0.14 Seg	0		0		5		5	
<a href="#">es.udc.fi.dc.fd.search</a>	92 Ms	0		0		3		3	
<a href="#">es.udc.fi.dc.fd.signin</a>	51 Ms	0		0		2		2	
<a href="#">es.udc.fi.dc.fd.signup</a>	0.23 Seg	0		0		5		5	
<a href="#">es.udc.fi.dc.fd.tuit</a>	0.81 Seg	0	-2	0		17	+2	17	

Con respecto a la versión anterior hemos implementado 31 test más, sin fallas.



## Tuiter 11

### Informe de resultados y cierre de proyecto.

- Factores de calidad:

Grupo11 -  
Ferramentas de  
Desenvolvimento  
2.0  
09/Dic/2014

Líneas de código	3,251
Complejidad	619
Comentarios (%)	31.2%
Líneas duplicadas (%)	20.5%
Evidencias	192
Cobertura	54.4%
Architecture	64.0%
Ciclos entre paquetes	12
Dependencias entre paquetes a cortar	7
Bloques duplicados	64
Quality Index	8.7
Code Quality	83.5%
API pública documentada (%)	100.0%
Índice de cumplimiento de reglas	81.6%
Cobertura de líneas	61.0%
Cobertura de rama	37.0%
DRYness	79.5%

Debido a que ha habido algún despiste a la hora de subir versiones del proyecto al repositorio, no se ha etiquetado bien el versionado de cada feature, lo cual nos ha complicado bastante la observación de los parámetros de calidad en el Sonar. Por eso, y dado que la versión 2.0 de Sonar (la versión 3.0 del proyecto) data a 09/Diciembre/2014 y la fecha de salida de la release ha sido el 28/Noviembre/2014 se ha optado por visualizar dichos parámetros, puesto que el sistema no había cambiado a esas alturas, de hecho, se incluyeron dos tareas en la versión 4.0 (1222 y 1237) que realmente





## Tuiter 11

### Informe de resultados y cierre de proyecto.

pertenecían a la versión 3.0 pero que se quedaron pendientes por problemas de funcionamiento.

Con respecto a la cobertura, es importante destacar que, aunque para la capa modelo se han realizado una batería importante de tests unitarios, hemos experimentado problemas en las 3 primeras versiones con los tests de los controladores mediante MockMVC y no ha sido hasta una etapa tardía del desarrollo cuando se ha dominado dicha tecnología y se comenzaron a incluir dichos tests.

Además, una gran parte del código duplicado se considera un “falso positivo” puesto que la gran mayoría de dicho código son métodos “equals”, “hashCode” y “getters-setters” autogenerados. Este factor también influye negativamente en la cobertura del código.

## 3.4 Version 4.0

En esta versión hemos separado el proyecto en módulos, y hemos implementado un servicio REST de búsqueda simple para buscar tuits a partir de una palabra clave. También se ha realizado un estudio de profiling para comprobar la eficiencia de la aplicación, dicho estudio se adjunta en el mismo directorio que el presente informe.

- Roles:

Rene Balay Lorenzo	Jefe de Proyecto
Samuel Paredes Sánchez	Desarrollador
Xoan Iago suárez Canosa	Desarrollador

- Peticiones realizadas:

# ▼	Tipo	Estado	Prioridad	Asunto	Asignado a
1454	Support	In Progress	High	Elaboración de informe de test de carga	René Balay Lorenzo
1422	Feature	Resolved	High	Poblar las bases de datos	René Balay Lorenzo
1421	Support	Resolved	Normal	Gestionar calidad del código	Xoan Iago Suárez Canosa
1357	Support	In Progress	High	Elaboración del informe	Samuel Paredes Sánchez
1353	Support	Resolved	Normal	Recopilación de información	René Balay Lorenzo
1322	Feature	Resolved	High	Creación del servicio REST	Samuel Paredes Sánchez
1321	Feature	Resolved	High	Separación en módulos de la aplicación	Xoan Iago Suárez Canosa
1320	Support	Resolved	High	Test de MockMVC para HomeSignedInController	René Balay Lorenzo
1237	Bug	Resolved	Normal	Crear IntanceNotFoundException	Xoan Iago Suárez Canosa
1222	Support	Resolved	High	Refactorización de código	Samuel Paredes Sánchez

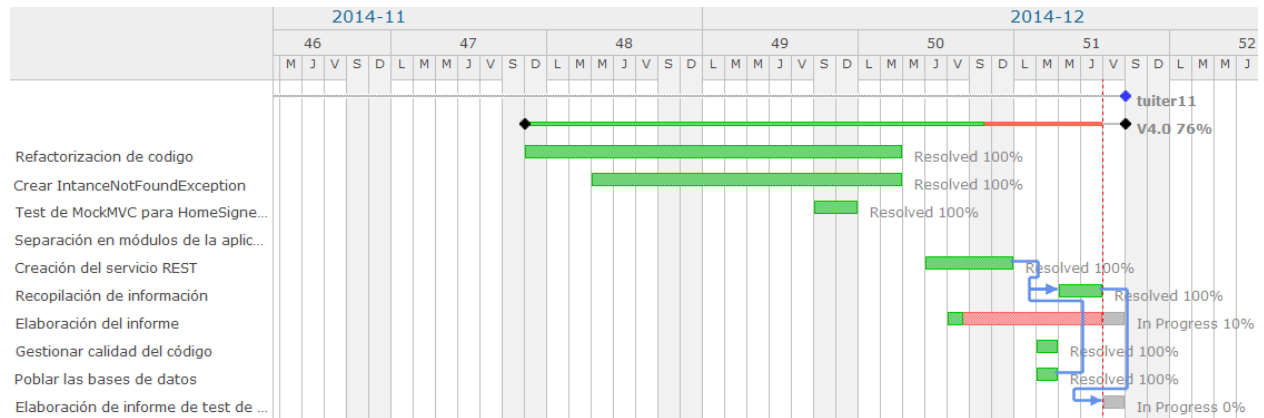
Algunas de las peticiones de esta versión está en progreso porque los informes a redactar son tareas a realizar en la misma.



## Tuiter 11

### Informe de resultados y cierre de proyecto.

- Diagrama de Gantt:



- Desviaciones:



- Integración Continua

## Resultado del test

0 fallidos (±0)

70 tests (±0)			
Módulo	Fallos	(diferencias) Total	(diferencias)
<a href="#">es.udc.fi.dc.fd:tuiter11</a>	0	0	
<a href="#">es.udc.fi.dc.fd:tuiter11-model</a>	0	46	
<a href="#">es.udc.fi.dc.fd:tuiter11-restservice</a>	0	0	
<a href="#">es.udc.fi.dc.fd:tuiter11-webapp</a>	0	24	

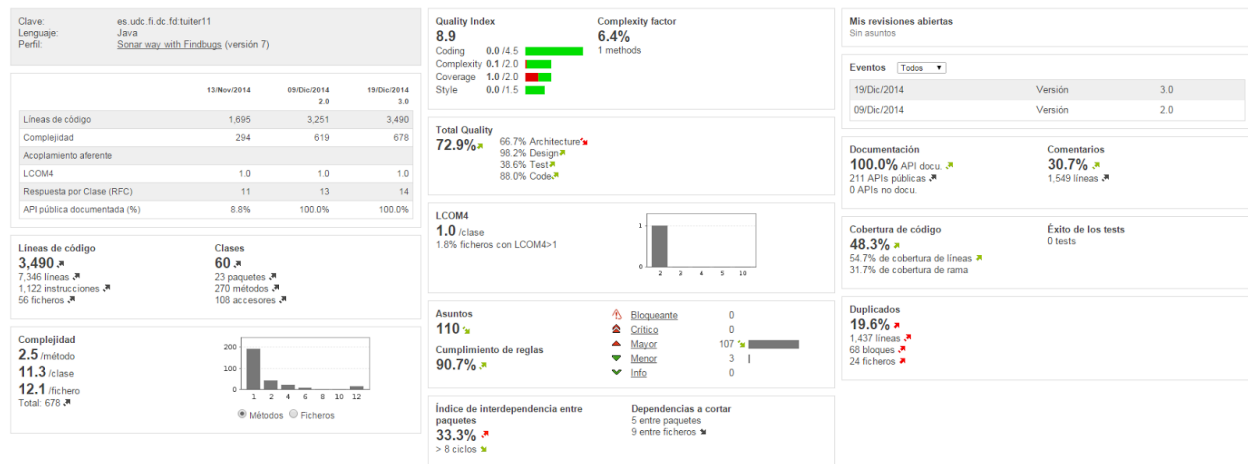
Con respecto a la versión 3.0 hemos añadidos 6 test, sin fallas.

- Factores de calidad



## Tuiter 11

### Informe de resultados y cierre de proyecto.



Como podemos observar en la imagen anterior nuestro proyecto ha mejorado en cobertura de código, en diseño, test y calidad de código, aunque al separarlo en módulos la calidad de nuestra arquitectura y cobertura ha disminuido levemente.

La calidad final de nuestro proyecto es de 72,9%, un porcentaje bastante aceptable.

## 4 COSTE DEL PROYECTO

**V1.0:** De las 35 horas planificadas, se han trabajado 29.3 horas reales a 12€/hora, suma un total de 351,6€, frente a los 438€ planificados.

**V2.0:** De las 52.5 horas planificadas, se han trabajado 35 horas reales a 12€/hora, suma un total de 420€, frente a los 638€ planificados.

**V3.0:** De las 64.60 horas planificadas, se han trabajado 64.93 horas a 12€/hora, suma un total de 779,16€, frente a los 775,2€ planificados.

**V4.0:** De las 47.5 horas planificadas, se han trabajado 56.58 horas hasta el momento. Vamos a aproximar el valor de las horas restantes a 1 hora a mayores, por lo tanto se trabajaría unas 57.58 horas a 12€/hora, que suman un total de 690.96€.

El Valor planificado es de  $438€ + 638€ + 775.2€ + 570€ = 2421.2€$

El coste total por mano de obra es de  $351.6€ + 420€ + 779.16€ + 690.96€ = 2241.18€$

Hemos gastado menos de lo planificado por tanto, con respecto al proyecto, tenemos un balance positivo económico de  $2421.2€ - 2241.18€ = 180.02€$ . Sin embargo, una de las causas de que tengamos este balance positivo es la imputación de horas que no se anotaron por parte de los desarrolladores en algunas de las versiones del proyecto.



## 6 APÉNDICE

### 6.1 MANUAL DE INSTALACIÓN

Como requisitos para poder lanzar la aplicación tenemos que tener instaladas las herramientas que hemos descrito en el apartado 2.5.

Para instalar Eclipse entramos en la url

<http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/lunasr1>

y nos descargamos la versión que se adapte a la arquitectura de nuestro PC.

Para instalar el controlador de versiones git y maven, entramos en la consola y tecleamos:

```
sudo apt-get install git maven
```

El repositorio remoto nos lo proporciona la organización (UDC).

Además, en una terminal debemos teclear lo siguiente (siempre que tengamos la correcta clave ssh configurada) suponiendo un sistema operativo tipo UNIX:

```
git clone git@git.fic.udc.es:tuitier11.git
```

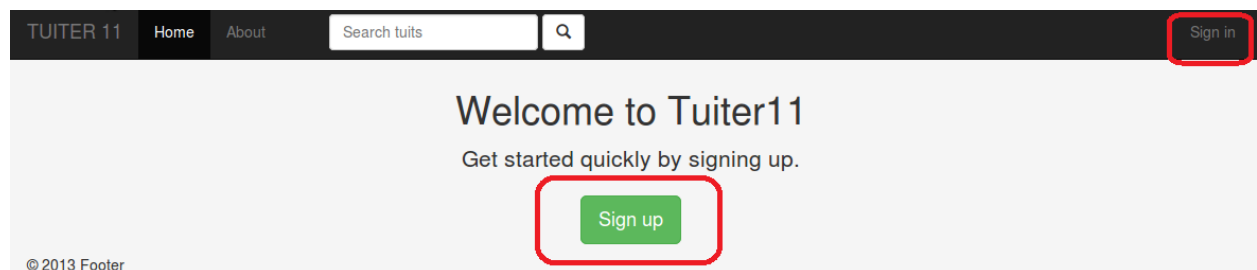
```
cd tuitier11
```

```
mvn install tomcat7:run
```

Este comando compilará la aplicación, la empaquetará y la desplegará en el servidor Tomcat

### 6.2 MANUAL DE USUARIO

Una vez arrancada la aplicación, entramos en la url <http://localhost:8080/tuitier11-webapp/>





## Tuitr 11

### Informe de resultados y cierre de proyecto.

En esta pantalla tenemos una cabecera que contiene un enlace a home, otro a about, un buscador de tuits por palabra clave y a la derecha enmarcado en rojo el enlace a la pagina de login. En el cuerpo de la página esta el botón de Darse de alta.

Si pulsamos en *Sign up* nos lleva a la siguiente ventana:

### Please Sign Up

Email

Nick

Password



Already have an account? [Sign In](#)

Introducimos los datos de usuario, y si son correctos, es decir, si el email y el nick no existen nuestra petición de Alta quedará pendiente hasta que el administrador la confirme, como se observa a continuación:

**TUITR 11** [Home](#) [About](#)   [Logout](#)

## Welcome to tuitr11

The user has not yet been confirmed, please wait for confirmation of the administrator to use our great application.

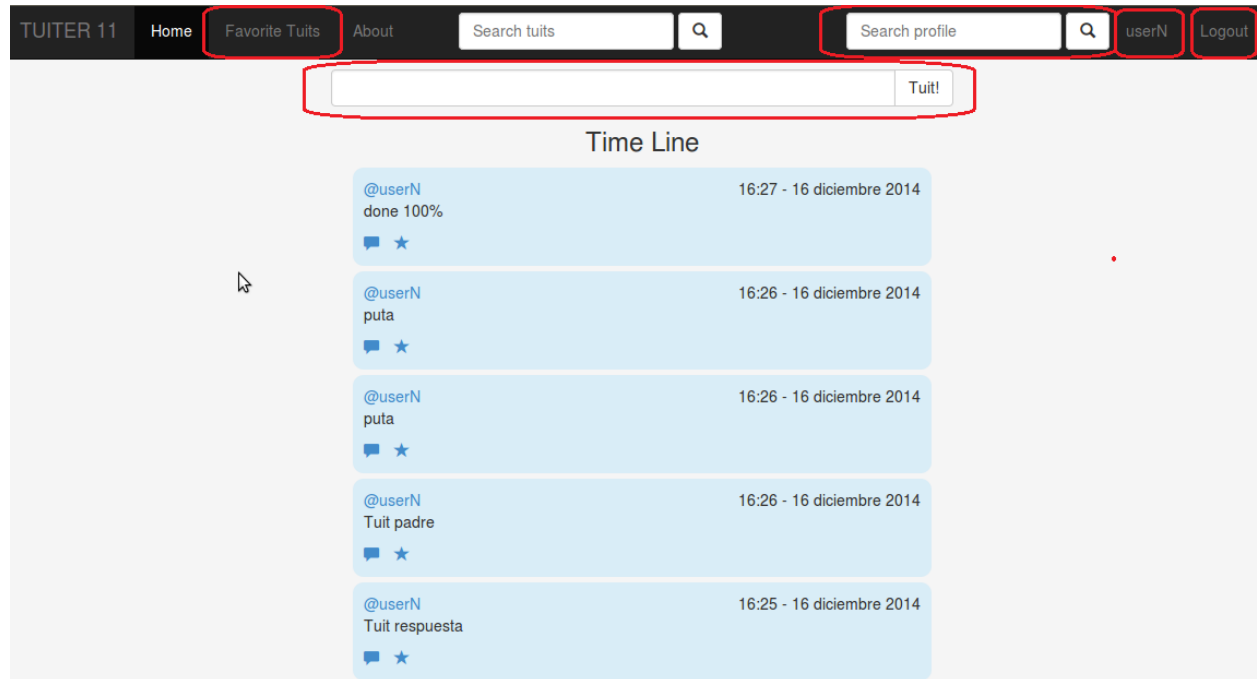
© 2013 Footer

Una vez confirmada el alta, nuestra pagina principal tendrá un aspecto similar al siguiente:



## Tuiter 11

### Informe de resultados y cierre de proyecto.



La cabecera de la página se ha actualizado, tenemos un Campo de texto para poder crear nuestras publicaciones y un timeline donde se verán todos nuestros tuits y retuits, además de los de los usuarios a los que sigamos(explicaremos más adelante como seguir a un usuario). Cada tuit contiene el usuario que lo ha publicado, la fecha de la publicación, el contenido y 3 iconos, que sirven para contestar a un tuit, hacer favorito a un tuit, o retuitear un tuit(esté último no aparece en la imagen) respectivamente. No podremos retuitear nuestros propios tuits pero si podemos contestar a nosotros mismo y marcar nuestros propios tuits como favoritos.

Para las conversaciones, si hacemos click en el botón, nos mostrará una nueva página con todo el hilo de conversación del tuit, y a partir de ahí poder responder al tuit:





## Tuiter 11

### Informe de resultados y cierre de proyecto.



@userN20:05 - 19 diciembre 2014

Hola holaaaaaa



@userN21:17 - 19 diciembre 2014

hola que tal





Response to @userN

Response

@userN21:18 - 19 diciembre 2014

bien y tu?





En la cabecera, que ya hemos citado en el párrafo anterior se ha añadido el enlace a los tuits favoritos del usuario, donde podrá visualizar en una nueva página dichos tuits.

TUITER 11HomeFavorite TuitsAboutSearch tuitsSearch profileuserNLogout

Favorite List



@userN16:27 - 16 diciembre 2014

done 100%



@userN16:26 - 16 diciembre 2014

Tuit padre



En la parte superior derecha tenemos el campo de búsqueda de usuarios donde podremos buscar a uno o varios usuarios en base a su nick La búsqueda nos dará una nueva página como la que se muestra a continuación:



## Tuiter 11

### Informe de resultados y cierre de proyecto.

TUITER 11 Home Favorite Tuits About Search tuits Search profile userN Logout

Search results:

- @userN  
Panel content
- @userN2  
Panel content
- @userNick  
Panel content
- @userN3  
Panel content

Si hacemos click en uno de los nombres de la lista de resultados iremos al perfil de dicho usuario(más adelante mostraremos la página resultante).

Continuando con la cabecera, tenemos otro enlace que nos lleva a la visualización de nuestro perfil de usuario, y un último enlace para cerrar la sesión.  
la pagina de perfil tiene la siguiente apariencia:

TUITER 11 Home Favorite Tuits About Search tuits Search profile userN Logout

@userN

E-mail user

Nick userN

Edit Profile

Followed List

Followers List

- @userN done 100% 16:27 - 16 diciembre 2014
- @userN puta 16:26 - 16 diciembre 2014
- @userN puta 16:26 - 16 diciembre 2014
- @userN Tuit padre 16:26 - 16 diciembre 2014
- @userN Tuit respuesta 16:25 - 16 diciembre 2014





## Tuiter 11

### Informe de resultados y cierre de proyecto.

Como podemos observar en la página de perfil se visualizan los tuits y retuits del usuario concreto, además de las listas de seguidores y seguidos por el propio usuario. Además, mostramos la información de perfil y un botón para poder editar la cuenta.

The screenshot shows the 'Edit Profile' form in the TUITER 11 application. The form is centered on a light gray background. It has a dark gray header bar with the text 'TUITER 11' and navigation links: 'Home', 'Favorite Tuits', and 'About'. There are two search bars: 'Search tuits' and 'Search profile', both with a magnifying glass icon. The user's name 'userN' and a 'Logout' link are visible on the right. The 'Edit Profile' form itself contains the following fields and options:

- Edit Profile** (title)
- New Email**: Input field with 'user'.
- New Nick**: Input field with 'userN'.
- New Password**: Input field with '....'.
- Old Password**: Input field with 'Old Password'.
- Private account**: A checkbox that is currently unchecked.
- Save Changes**: A button at the bottom.

Podremos cambiar el nick, el email y/o la contraseña, además de poder hacer nuestra cuenta privada.

Una vez explicados todos los enlaces y botones de la página principal y las páginas a las que llevan dichos enlaces/botones pasamos a explicar como seguir/Dejar de seguir a un usuario, o bloquearlo/desbloquearlo.

Como se ha comentado anteriormente que clickando en el nombre de un usuario en los resultados de una búsqueda, nos devolverá una página similar a la de nuestro perfil, con una pequeña modificación: tendremos los botones de seguir y bloquear:

The screenshot shows the user profile page for '@userN2' in the TUITER 11 application. The page has a dark gray header bar with the text 'TUITER 11' and navigation links: 'Home', 'Favorite Tuits', and 'About'. There are two search bars: 'Search tuits' and 'Search profile', both with a magnifying glass icon. The user's name 'userN' and a 'Logout' link are visible on the right. The profile page itself contains the following elements:

- @userN2** (username)
- E-mail**: Input field with 'user2'.
- Nick**: Input field with 'userN2'.
- Follow**: A green button.
- Block**: A red button.
- Followed List**: A button at the bottom left.
- Followers List**: A button at the bottom right.
- Tweets**: Two tweets are displayed, both from '@userN2' on '19:53 - 19 diciembre 2014'. The first tweet says 'estoy muy contento con tuiter 11,! es la mejor app en mucho tiempo!' and the second says 'hola amigos ya soy miembro de tuiter 11'. Each tweet has icons for reply, retweet, and favorite.



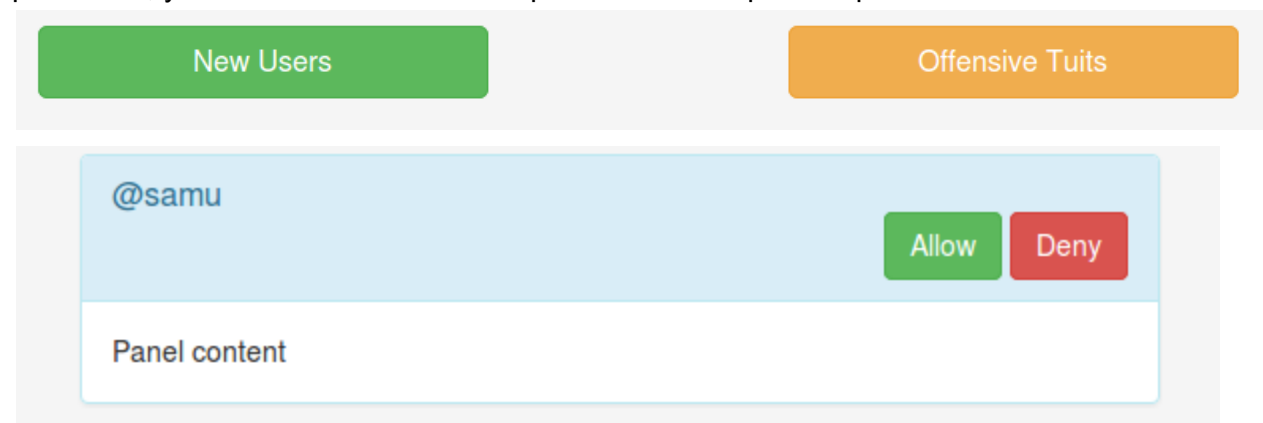
## Tuitter 11

### Informe de resultados y cierre de proyecto.

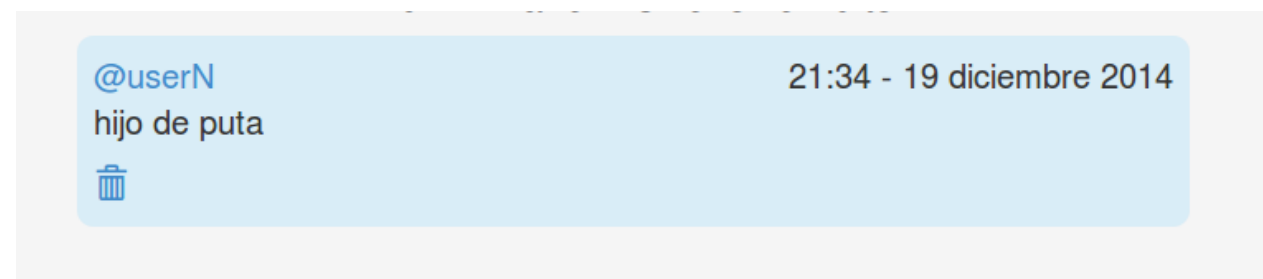
Una vez seguimos a un usuario nos aparecerá en nuestra lista de seguidos, y el usuario seguido le aparece el usuario seguidor en su lista de seguidores.

una vez hecho click en el boton de *follow*, se actualizará automáticamente para poder dejar de seguir a dicho usuario en el momento que nos plazca. Lo mismo pasa con el botón de bloquear usuario. Si bloqueamos a un usuario dicho usuario no podrá seguirnos y en casa de que ya nos estuviera siguiendo, dejará de hacerlo. una vez bloqueado un usuario el botón se actualizará automáticamente para poder desbloquearlo en el momento que al usuario le plazca.

Esto es todo con respecto a un usuario de la aplicación. Sin embargo, si somos administradores de la misma, tendremos a mayores un panel de administración en el cuál tendremos los los usuarios pendientes de confirmación y podrá aceptar o denegar dichas peticiones, y la lista de tuits ofensivos que los usuarios puedan publicar.



A raíz de esto, el administrador podrá eliminar cualquier tuit a su antojo, teniendo en cada tuit un icono de papelera para poder eliminarlos.





**Tuiter 11**  
**Informe de resultados y cierre de proyecto.**