



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Desarrollo de una aplicación para el análisis de los sentimientos de los alumnos durante una clase

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Jose Manuel Martínez Serrano

**Tutores:** Miguel Rebollo Pedruelo

Jose Luis Poza Luján

Ángeles Calduch Losa

Curso 2016/2017



# Resumen

---

Hoy en día, con la actual influencia de las redes sociales en nuestra vida cotidiana, surge la necesidad para los docentes de conocer el estado emocional y satisfacción en sus clases. Es por ello que nace este proyecto, permitiendo a los docentes conocer en tiempo real la satisfacción del alumnado a través de la red social Twitter en cada una de sus clases mejorando la calidad educativa. El objetivo principal de este proyecto es que el alumnado pueda evaluar tanto su satisfacción personal como su estado anímico a través de Twitter donde, además, podrán añadir una opinión personal de como se está desarrollando la sesión. Para llevar a cabo este proyecto se ha desarrollado una aplicación móvil, así como el análisis y la visualización mediante gráficas de los datos almacenados.

**Palabras clave:** Twitter, aplicación móvil, análisis de datos, mejora educativa, visualización gráfica, alumnado, satisfacción, estado emocional, tiempo real, redes sociales.

# Abstract

---

Nowadays, with the current influence of social networks in our quotidian life, teachers have the necessity to know the emotional state and the satisfaction in their classes. For this reason, this project is born like an instrument which will allow to all teachers improve their education quality, knowing student's satisfaction in real time through the social network Twitter. The main objective is that students can evaluate their emotional state and their personal satisfaction about the session as well as the possibility of adding comments describing how the session is being developed. This objective has been achieved through a mobile application, the analysis and displaying graphics of data stored.

**Keywords:** Twitter, mobile application, data analyze, educational improvement, display graphic, students, satisfaction, emotional state, real time, social network.





# Tabla de contenidos

---

<b>1. Introducción .....</b>	<b>10</b>
<b>1.1 Motivación.....</b>	<b>10</b>
<b>1.2 Objetivos.....</b>	<b>10</b>
<b>1.3 Estructura de la memoria .....</b>	<b>11</b>
<b>2. Entorno de desarrollo .....</b>	<b>13</b>
<b>2.1 Introducción.....</b>	<b>13</b>
<b>2.2 Herramientas similares .....</b>	<b>13</b>
2.2.1 Twitter Analytics .....	13
2.2.2 Sprout Social .....	16
2.2.3 Hashtracking .....	18
<b>2.3 Análisis.....</b>	<b>19</b>
2.3.1 Características cualitativas.....	19
2.3.2 Características cuantitativas .....	20
<b>2.4 Síntesis .....</b>	<b>21</b>
<b>2.5 Tecnología utilizada.....</b>	<b>21</b>
<b>2.6 Conclusión.....</b>	<b>22</b>
<b>3. Especificación de requisitos .....</b>	<b>23</b>
<b>3.1 Introducción.....</b>	<b>23</b>
3.1.1 Propósito .....	23
3.1.2 Ámbito de sistema .....	24
3.1.3 Definiciones .....	25
<b>3.2 Descripción general .....</b>	<b>26</b>
3.2.1 Perspectiva del producto .....	26
3.2.2 Funciones del producto .....	27
3.2.3 Características de los usuarios.....	28
3.2.4 Restricciones .....	28
3.2.5 Suposiciones y dependencias.....	29
3.2.6 Requisitos futuros .....	29
<b>3.3 Requisitos específicos .....</b>	<b>29</b>
3.3.1 Requisitos de diseño.....	30
3.3.2 Requisitos de funcionalidad .....	31
3.3.3 Otros requisitos.....	33
<b>3.4 Conclusiones .....</b>	<b>33</b>
<b>4. Diseño .....</b>	<b>34</b>
<b>4.1 Introducción.....</b>	<b>34</b>
<b>4.2 Inspiración.....</b>	<b>34</b>
<b>4.3 Especificación conceptual .....</b>	<b>35</b>
<b>4.4 Especificación formal.....</b>	<b>38</b>
4.4.1 Capa de presentación. ....	38
4.4.2 Capa lógica .....	42



4.4.3 Capa persistencia .....	47
<b>5. Implementación .....</b>	<b>53</b>
5.1 Introducción.....	53
<b>5.2 Conceptos generales previos .....</b>	<b>53</b>
5.2.2 Creación de una App con Twitter Developers .....	53
5.2.3 Conexión con FireBase de Google. ....	54
5.2.4 Bibliotecas utilizadas en la aplicación móvil .....	54
<b>5.3 Implementación .....</b>	<b>56</b>
5.3.1 Implementación inicio de sesión con Twitter.....	56
5.3.2 Implementación selección de grupo y clase. ....	58
5.3.3 Implementación publicación de un tweet. ....	59
5.3.4 Implementación consultar y mostrar estadísticas. ....	62
<b>5.5 Implantación .....</b>	<b>66</b>
<b>5.6 Testing .....</b>	<b>66</b>
<b>5.7 Conclusiones .....</b>	<b>68</b>
<b>6. Conclusiones .....</b>	<b>70</b>
6.1 Dificultades y soluciones .....	71
6.2 Aportaciones .....	72
6.3 Trabajo futuro.....	73



# Índice de ilustraciones

---

ILUSTRACIÓN 1 : ACTIVIDAD DEL TWEET CON TWITTER ANALYTICS.....	14
ILUSTRACIÓN 2: AUDIENCIA DEL USUARIO CON TWITTER ANALYTICS.....	15
ILUSTRACIÓN 3: AUDIENCIA CON TWITTER ANALYTICS .....	15
ILUSTRACIÓN 4: EJEMPLO DE EVENTO CON TWITTER ANALYTICS .....	16
ILUSTRACIÓN 5: AUDIENCIA DEMOGRÁFICA CON SPROUT SOCIAL .....	17
ILUSTRACIÓN 6: ESTADÍSTICA SEGUIDORES CON SPROUT SOCIAL .....	17
ILUSTRACIÓN 7: ANÁLISIS #FELIZLUNES CON SPROUT SOCIAL .....	18
ILUSTRACIÓN 8: DIAGRAMA DE CASOS DE USO .....	27
ILUSTRACIÓN 9: PANTALLA APLICACIÓN MÓVIL DEL PROYECTO FELIZÓMETRO2016 .....	35
ILUSTRACIÓN 10: DIAGRAMA ESQUEMA CONCEPTUAL .....	35
ILUSTRACIÓN 11: DISEÑO DIAGRAMA DE CLASES GENERAL.....	37
ILUSTRACIÓN 12: DIAGRAMA DE CLASES CAPA PRESENTACIÓN .....	39
ILUSTRACIÓN 13: MOCKUPS INICIO DE SESIÓN.....	40
ILUSTRACIÓN 14: MOCKUPS DISEÑO MENÚ PRINCIPAL.....	40
ILUSTRACIÓN 15: MOCKUPS DISEÑO PUBLICACIÓN DE UN TWEET .....	41
ILUSTRACIÓN 16: MOCKUPS DISEÑO SELECCIÓN DE SESIÓN.....	41
ILUSTRACIÓN 17: MOCKUPS DISEÑO CONSULTA DE ESTADÍSTICAS.....	42
ILUSTRACIÓN 18: DIAGRAMA DE CLASES CAPA LÓGICA.....	43
ILUSTRACIÓN 19: DIAGRAMA DE SECUENCIA CU01.....	44
ILUSTRACIÓN 20: DIAGRAMA DE SECUENCIA CU02 Y CU03 .....	45
ILUSTRACIÓN 21: DIAGRAMA DE SECUENCIA CU04 .....	46
ILUSTRACIÓN 22: DIAGRAMA DE CLASES CAPA PERSISTENCIA .....	48
ILUSTRACIÓN 23: DISEÑO MODELO ENTIDAD RELACIÓN BASE DE DATOS .....	48
ILUSTRACIÓN 24: ESQUEMA JSON NO RELACIONAL.....	50
ILUSTRACIÓN 25: DISEÑO ESQUEMA CON FIREBASE .....	51
ILUSTRACIÓN 26: PANTALLAS DE INICIO DE SESIÓN CODIFICADAS.....	57
ILUSTRACIÓN 27: PANTALLAS SELECCIÓN DE GRUPO Y CLASE CODIFICADAS.....	59
ILUSTRACIÓN 28: PANTALLA DE PUBLICACIÓN DE UN TWEET CODIFICADA.....	61
ILUSTRACIÓN 29: PANTALLAS DE CONSULTA DE ESTADÍSTICAS CODIFICADAS .....	65
ILUSTRACIÓN 30: RESULTADO TEST UNITARIO PUBLICAR TWEET .....	67



# Índice de tablas

---

TABLA 1: CARACTERÍSTICAS CUALITATIVAS .....	20
TABLA 2: CARACTERÍSTICAS CUANTITATIVAS.....	20
TABLA 3: DESCRIPCIÓN DE LAS CARACTERÍSTICAS A CUMPLIMENTAR. ....	21
TABLA 4: MIEMBRO MIGUEL REBOLLO PEDRUELO .....	23
TABLA 5: MIEMBRO ÁNGELES CALDUCH LOSA.....	24
TABLA 6: MIEMBRO JOSÉ LUIS POZA LUJÁN.....	24
TABLA 7: MIEMBRO JOSÉ MANUEL MARTÍNEZ SERRANO .....	24
TABLA 8: DEFINICIONES TÉCNICAS. ....	25
TABLA 9: CASOS DE USO .....	27
TABLA 10: ROL USUARIOS .....	28
TABLA 11: REQUISITOS DE INTEROPERABILIDAD DE LA APLICACIÓN. ....	30
TABLA 12: REQUISITOS DE FORMULARIOS DE LA APLICACIÓN.....	30
TABLA 13: REQUISITOS DE DISEÑO DE LA BASE DE DATOS .....	31
TABLA 14: REQUISITOS INICIO DE SESIÓN CON TWITTER.....	31
TABLA 15: REQUISITOS PUBLICACIÓN DE TWEETS.....	32
TABLA 16: REQUISITOS CONSULTAR ESTADÍSTICAS. ....	32
TABLA 17: REQUISITOS DE ADECUACIÓN CON LOS ESTÁNDARES Y LA NORMATIVA .....	33
TABLA 18: BIBLIOTECA FABRIC.....	55
TABLA 19: BIBLIOTECA PICASSO. ....	55
TABLA 20: BIBLIOTECA FIREBASE. ....	55
TABLA 21: BIBLIOTECA GRAPHVIEW.....	55
TABLA 22: BIBLIOTECA RADARCHART .....	55
TABLA 23: BATERÍA DE PRUEBAS CORRESPONDIENTE AL INICIO DE SESIÓN .....	67
TABLA 24: BATERÍA DE PRUEBAS CORRESPONDIENTE A LA SELECCIÓN DE GRUPO Y PUBLICACIÓN DE UN TWEET .....	68
TABLA 25: BATERÍA DE PRUEBAS CORRESPONDIENTE A LA CONSULTA DE ESTADÍSTICAS .....	68



# Índice de códigos

---

CÓDIGO 1: INICIAR LLAMADA PARA EL INICIO DE SESIÓN CON TWITTER.....	56
CÓDIGO 2: USUARIOS VALIDADOS.....	56
CÓDIGO 3: USUARIOS NO VALIDADOS.....	57
CÓDIGO 4: INSERCIÓN EN EL SPINNER.....	58
CÓDIGO 5: PASAR DATOS ENTRE DIFERENTES ACTIVITYS.....	58
CÓDIGO 6: OBTENER LA SATISFACCIÓN Y EL ESTADO EMOCIONAL .....	59
CÓDIGO 7: GENERACIÓN DE UN OBJETO DE TIPO STATUSSERVICE.....	60
CÓDIGO 8: CUERPO DEL MENSAJE DEL TWEET. ....	60
CÓDIGO 9: PUBLICAR UN TWEET.....	61
CÓDIGO 10: CREAR REFERENCIA EN LA BASE DE DATOS .....	62
CÓDIGO 11: CONSULTA PARA OBTENER LAS SESIONES .....	62
CÓDIGO 12: GENERACIÓN DE LA GRÁFICA EVOLUCIÓN.....	63
CÓDIGO 13: CONSULTA GRÁFICO RADAR .....	63
CÓDIGO 14: OBTENER LA SESIÓN ACTIVA. ....	64
CÓDIGO 15: CONSULTA DE MEDIAS GENERALES Y PERSONALES. ....	64
CÓDIGO 16: TEST UNITARIO PUBLICACIÓN DE UN TWEET .....	67



# 1. Introducción

---

## 1.1 Motivación

En la actualidad, un alto porcentaje de los jóvenes de la sociedad en la que vivimos disponen de teléfonos móviles, pero si nos centramos en los jóvenes universitarios observamos que los usos de estos aparatos se han hecho imprescindibles en las aulas por diversos motivos. Los jóvenes necesitan expresar sus emociones en clase a través de las redes sociales o simplemente buscan la forma de distraerse.

El sector de los jóvenes es el que sostiene que las redes sociales tengan tanta popularidad. Diversas redes sociales como Facebook, Twitter o Instagram permiten compartir hechos con el resto de la comunidad de usuarios, las personas sentimos la necesidad de compartir y es por esto que en las aulas la presencia de los teléfonos móviles está viva y presente. De ahí surge nuestro proyecto.

Existe un gran debate<sup>1</sup> sobre si el uso de los teléfonos móviles debería estar o no permitido en las aulas, puesto que en la mayoría de los casos es un método de distracción del alumno lo que dificulta la comunicación alumno/profesor. También puede provocar o influir negativamente en los profesores puesto que la falta de concentración al encontrarse una clase desconectada puede reducir la calidad de las explicaciones, pero, ¿y si en vez de utilizar los teléfonos móviles con un motivo de ocio los utilizáramos con un motivo educativo?

## 1.2 Objetivos

Como introducimos en la sección 1.1, esta idea surge para mejorar el entendimiento entre el profesor y los alumnos, mejorando la calidad de la enseñanza, así como la relación entre ambos.

El proyecto constará de una aplicación móvil en la que los alumnos pueden valorar al profesor en todo momento en tiempo real indicando su nivel de satisfacción con respecto a la clase en ese momento determinado, así como su estado emocional a través de la red social Twitter. El alumno podrá indicarle al profesor si su clase o la temática de la clase en ese momento tiene una

---

<sup>1</sup> <http://blog.tiching.com/el-movil-en-el-aula-problema-o-herramienta/>



satisfacción alta, o por el contrario no es de su agrado. Para añadir más información, el alumno también indicará su estado emocional o estado anímico, dando posibilidad a extraer más conclusiones que determinen el nivel de satisfacción del alumno. De esta forma, el profesor será capaz de saber en tiempo real si la temática que está dando es apropiada para ese momento o por el contrario, si debería cambiar el enfoque o planteamiento de la misma.

Este proyecto solo contempla el desarrollo de la aplicación móvil, que concierne la parte de publicación de *tweets* y el análisis de estadísticas interesantes para los alumnos. La extracción de datos y su análisis por parte del profesor se realizará en otro proyecto diferente pero que trabaja conjuntamente con éste.

### 1.3 Estructura de la memoria

La memoria está estructurada en seis capítulos diferentes que explican paso a paso las tareas y los materiales necesarios para desarrollar el proyecto. La estructura de esta memoria es la siguiente:

- Capítulo 2: Entorno de desarrollo.

En este capítulo se describen diferentes herramientas existentes en el mercado que se orientan en el análisis de datos enfocándose en la red social Twitter, pero sin discriminar al resto de redes sociales. Se han elaborado una serie de tablas, especificando las características cuantitativas y cualitativas para su análisis. Se definirán las características que cumplirá nuestra aplicación y se valorarán de menos a más importante. Por último, se ha detallado la tecnología empleada para llevar a cabo este proyecto.

- Capítulo 3: Especificación de requisitos.

En este capítulo se detallan los requisitos necesarios para el desarrollo de nuestro proyecto. En dicho capítulo, se seguirá el modelo de estándar IEE 830 para la especificación de requisitos software dejando claros los puntos clave que consistirán en una introducción, descripción general, requisitos específicos y una conclusión que nos permitirán conocer con más claridad las bases y las necesidades del proyecto.

- Capítulo 4: Diseño.

En este capítulo se detalla el diseño de la aplicación móvil. Hemos elaborado una introducción en la que hemos hecho una puesta en situación, seguidamente detallaremos de que idea surgió la inspiración del diseño del presente proyecto, hemos elaborado un diagrama UML en el que hemos reflejado todas las clases que intervienen en el proyecto, también hemos especificado de forma



conceptual un esquema de cómo funciona el sistema. Se ha elaborado una especificación formal en la que entraremos a diseñar nuestra aplicación a nivel de capas software y, por último, se ha elaborado una conclusión.

- Capítulo 5: Implementación.

En este capítulo del tema se han abarcado principalmente los pasos necesarios para la implementación de código. Para una correcta implementación hemos elaborado primero una introducción con los aspectos iniciales a tener en cuenta antes de la codificación. Seguidamente se puede encontrar especificado cómo se implementarán cada uno de los casos de uso en el que nos hemos apoyado, con ilustraciones de código y de diseño real. Finalmente, hay una fase de testeo en el que desarrollaremos una prueba automática y diferentes pruebas manuales. Hemos detallado una sección de implantación que permitirá a los alumnos poder utilizar esta aplicación de forma sencilla en sus teléfonos móviles y garantizándonos una total seguridad en su funcionamiento. Por último, se ha elaborado una conclusión.

- Capítulo 6: Conclusiones.

En este capítulo final, hemos establecido unas conclusiones generales a todo el proyecto en el que hemos hablado de las dificultades ocasionadas y la forma de resolver estas dificultades. Además, hemos añadido un apartado de aportaciones tanto tecnológicas como académicas que puede conllevar el desarrollo de este proyecto. También hemos dedicado un apartado a especificar el trabajo futuro que en proyectos posteriores se puede llevar a cabo a partir de este. Por último, se ha elaborado un apartado de referencias bibliográficas que nos han servido de apoyo y documentación para llevar a cabo el presente proyecto.



## 2. Entorno de desarrollo

---

### 2.1 Introducción

Como ya sabemos, Twitter hoy en día es una de las redes sociales más populares que existe, ocupando actualmente el cuarto lugar en el ranking mundial de redes sociales más utilizadas por usuarios, con más de 320 mil usuarios en la red<sup>2</sup>. Como ya sabemos, muchos de estos usuarios consultan día a día sus estadísticas, es decir, si su estado nuevo tiene muchos “me gusta” o simplemente si es muy compartido por el resto de usuarios.

Para ello surgen una serie de herramientas para ayudar a los usuarios de Twitter a conocer más cosas al respecto de sus estadísticas, o bien para realizar distintos tipos de análisis. Es por ello que en este apartado estudiaremos y analizaremos distintas herramientas que trabajan con Twitter para el diagnóstico de los datos. Cabe destacar que cada herramienta se centra en un aspecto distinto, por lo que posteriormente a medida que vayamos avanzando en este capítulo encontraremos una tabla a modo de resumen sobre lo que analiza cada una de éstas. Una vez detalladas a modo esquema las distintas herramientas con las que compararemos nuestro proyecto, escogeremos las tecnologías empleadas para llevar a cabo nuestro sistema hasta su respectivo funcionamiento.

### 2.2 Herramientas similares

#### 2.2.1 Twitter Analytics

Desde hace poco más de tres años, Twitter está empezando a dar estadísticas bastante interesantes sobre los perfiles para conocer el funcionamiento de cada cuenta haciendo que el uso de aplicaciones externas sea menor, ya que desde un solo sitio podemos tener muchos de los datos necesarios para sacar conclusiones.

---

<sup>2</sup> <https://about.twitter.com/es/company>



De este modo surge *Twitter Analytics*<sup>3</sup>, una aplicación que nos ofrece esta famosa red social para conocer estadísticas sobre nuestros *tweets*, seguidores y los *posts* o páginas de nuestra Web.

Para configurarlo únicamente tenemos que editar un par de aspectos en la configuración desde la interfaz web de Twitter. Para ello, vamos al panel de configuración y seleccionamos la pestaña que dice “Anuncios Twitter”. Una vez allí, entramos dentro de la herramienta.

Como hemos mencionado ya anteriormente, Twitter Analytics permite medir tres tipos de datos:

### Datos sobre nuestra actividad del *tweet*:

Como se puede observar en la ilustración 1, existe un gráfico en el que se pueden medir las impresiones, que son las veces que la gente vio ese *tweet* en Twitter y, las interacciones totales, que son las veces que lo usuarios interactuaron con este *tweet*.

En esta última medición podemos desglosarla en las siguientes mediciones para obtener unas estadísticas con más nivel de detalle: *clicks* en el perfil, respuestas, me gusta, abrir el detalle.

*Ilustración 1 : Actividad del tweet con Twitter Analytics*



### Datos sobre la audiencia

En este apartado, la herramienta permite calcular la audiencia de los seguidores de un usuario o la audiencia de todo usuario de Twitter.

<sup>3</sup> <https://analytics.twitter.com/about>



Para la primera opción, la herramienta muestra una estadística del número de seguidores de un usuario en los últimos 30 días a través de un diagrama de barras como el que veremos en la ilustración 2.

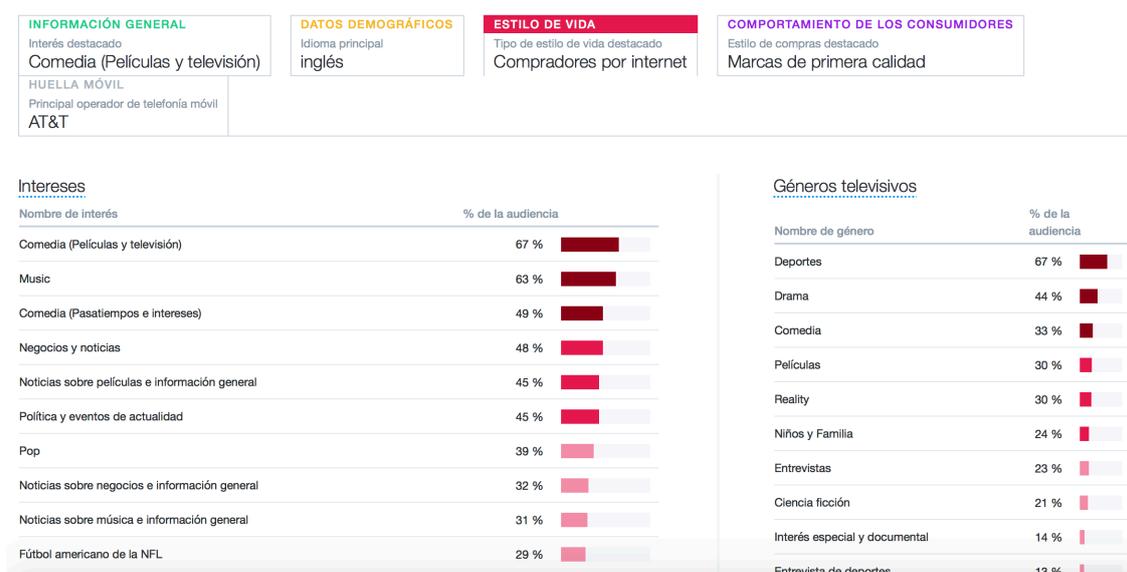
Para la segunda opción, la audiencia viene dada según todos los usuarios de Twitter, es decir, a través de ciertos campos, la herramienta es capaz de facilitar un porcentaje de seguidores en según qué ámbito. Unos de los ámbitos destacados y que también vemos en la ilustración 2 puede ser según sus intereses, según sus categorías de ingresos familiares, datos demográficos o estilo de vida entre otros muchos. De esta forma es más factible poder obtener conclusiones concretas.

*Ilustración 2: audiencia del usuario con Twitter Analytics*



La ilustración 3 muestra la audiencia de todos los usuarios de Twitter en torno al estilo de vida como campo general.

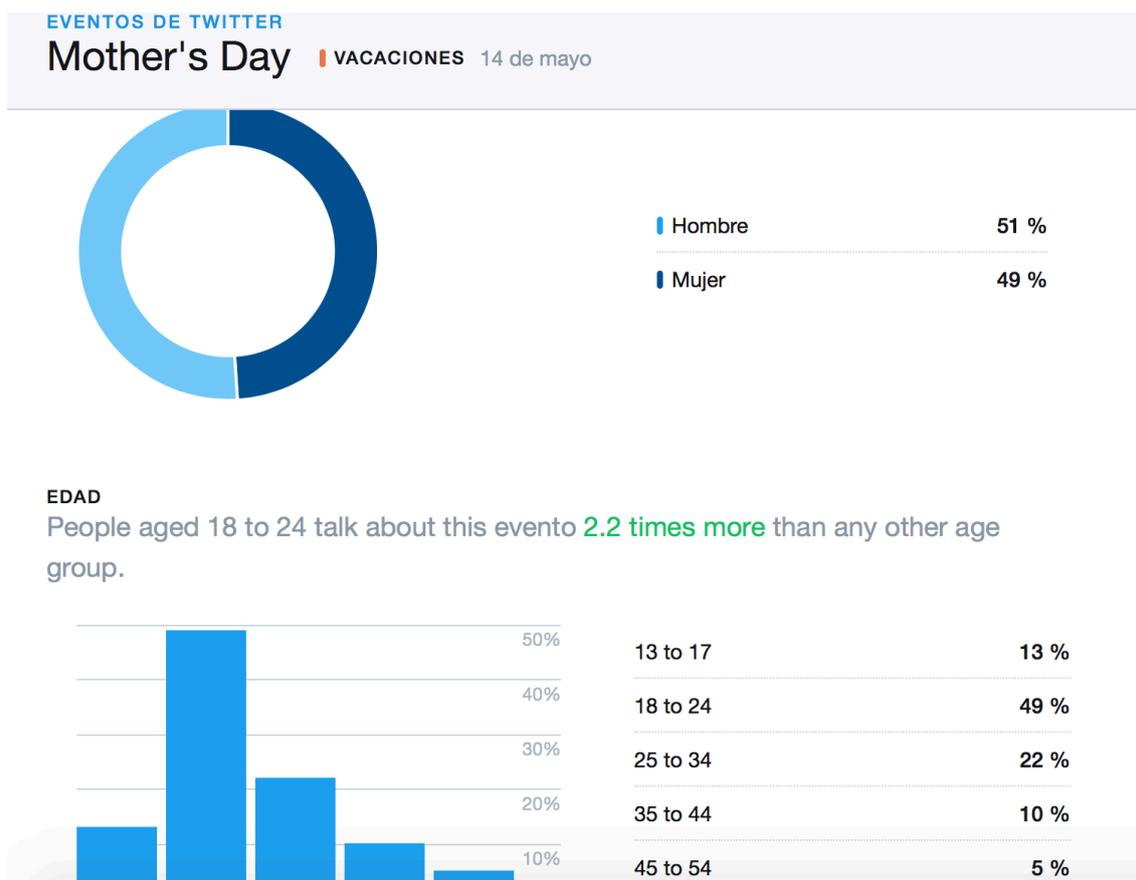
*Ilustración 3: Audiencia con Twitter Analytics*



## Datos sobre eventos

En este tercer apartado, la herramienta Twitter Analytics muestra estadísticas sobre los eventos. En el ejemplo que se puede ver en la ilustración 4, refleja datos sobre el evento del día de la madre, como la cantidad de hombres o mujeres que asistieron, en qué lugares del mundo o simplemente el porcentaje de rango de edades de los participantes.

Ilustración 4: Ejemplo de evento con Twitter Analytics



### 2.2.2 Sprout Social

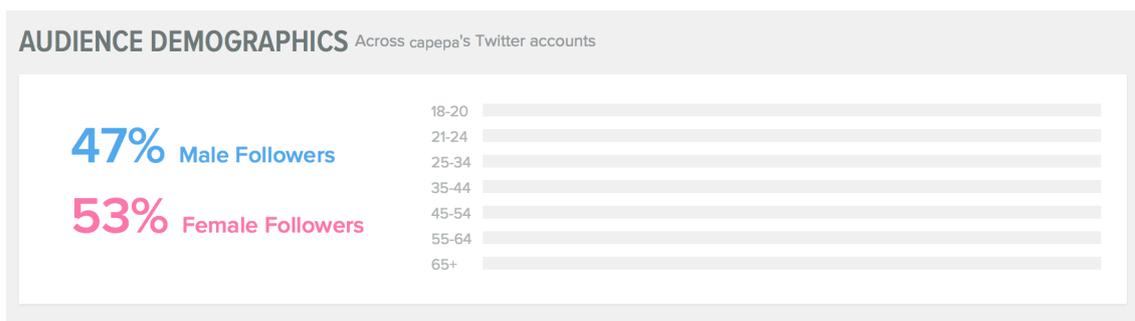
Sprout Social<sup>4</sup> es una herramienta de pago que también permite extraer estadísticas sobre la red social Twitter. Gracias a su versión de treinta días de prueba hemos observado cómo funciona realmente. Resulta ser bastante completa e interesante para los usuarios de Twitter, permitiendo calcular datos como en qué momento del día es mejor publicar un *tweet* para obtener más audiencia.

<sup>4</sup> <https://sproutsocial.com>

Para ello cuenta con una bandeja de administración de entrada social para mantener múltiples cuentas de Twitter, administrar mensajes, seguidores o simplemente agruparlos por orden de importancia, pero la mayor utilidad de esta herramienta es porque presenta una poderosa variedad de informes de Twitter entre los cuales destaca la identificación de tu mejor contenido, es decir, analiza la participación de tendencias con informes interactivos de Twitter. Esta herramienta también es capaz de informar a través de varios perfiles cuyo objetivo es comparar el crecimiento de los seguidores, el número de vinculaciones entre otras cosas, en períodos de tiempo personalizados. Por último, realiza un seguimiento del uso de palabras clave y *hashtags* en las que descubre tendencias en el uso de palabras clave y *hashtags* históricos en cualquier rango de datos.

A través de Sprout Social se han podido hacer unas mediciones y estadísticas en las que, por ejemplo, destacamos datos interesantes como el porcentaje de número de seguidores que son mujeres y hombres (ver la ilustración 5).

*Ilustración 5: Audiencia demográfica con Sprout Social*



Otra de las estadísticas que hemos comprobado a través de la herramienta es el número de impresiones que tienen mis *tweets*, el ratio de publicaciones por día o simplemente los nuevos usuarios que sigues. (Ver ilustración 6).

*Ilustración 6: Estadística seguidores con Sprout Social*



### 2.2.3 Hashtracking

Esta potente herramienta se basa fundamentalmente en el análisis a través de los *hashtags*, hecho que según el criterio que se plantea para nuestro proyecto es una herramienta bastante similar a lo que nosotros queremos hacer, ya que en nuestro sistema tenemos que filtrar por etiquetas.

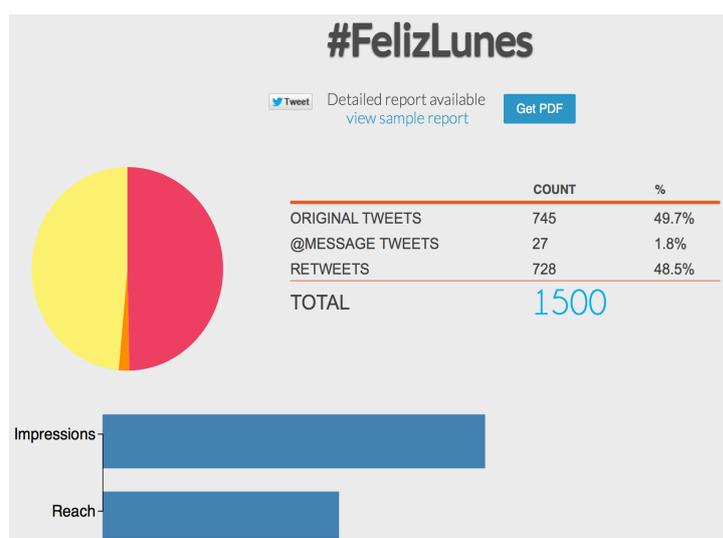
Hashtracking<sup>5</sup>, en un mundo donde los usuarios de la comunidad de Twitter acostumbran a utilizar etiquetas para categorizar y encontrar contenidos, resulta bastante interesante. A través de esta herramienta podremos analizar un *hashtag* de cara a incluirlo en nuestros *tweets* y obtener los mejores resultados. Con solo poner un hashtag y hacer la búsqueda, el usuario puede conocer los siguientes elementos: cuántas veces ha sido utilizado un determinado *hashtag* en *tweets* originales, cuántos para mensajes y cuántos ha sido retuiteados.

En el análisis también tienen un papel destacado los usuarios, pues nos mostrará quiénes son más influyentes para el *hashtag* y cuáles son los que más han publicado contenido relacionado.

En definitiva, lo mejor de esta herramienta es que nos ofrece la posibilidad de comparar *hashtags* y ver cuál de ellos tiene mayor alcance y, por tanto, posibilidad de ayudar a nuestros contenidos. Si utilizamos las etiquetas adecuadas, a partir de ellas muchos usuarios llegarán a nuestras publicaciones, por lo que es necesario elegir las bien y analizar nuestro sector en este sentido.

Para poder entender mejor en qué consiste esta herramienta veremos la ilustración 7 en la que se puede ver cómo se analiza un *hashtag*.

Ilustración 7: Análisis #felizLunes con Sprout Social



<sup>5</sup> <https://www.hashtracking.com>

## 2.3 Análisis

Después de exponer diferentes tipos de herramientas existentes en el mercado que trabajan con Twitter en el análisis de datos y estadísticas, vamos a proceder a analizarlas para extraer más información entre estas tres herramientas y poder compararlas con mayor precisión.

En este apartado del capítulo nos centraremos en dos tipos de análisis, según sus características cuantitativas y sus características cualitativas.

Para las características cualitativas se valorará la calidad de dichos programas de forma subjetiva mediante una valoración. Para las características cuantitativas se emplearán técnicas objetivas; como la observación controlada y estructurada, de la que se extraen inferencias externas más allá de los datos. Las características cuantitativas y cualitativas se verán reflejadas en las tablas 1 y 2.

### 2.3.1 Características cualitativas

Este análisis se ha realizado con las siguientes características que permiten elaborar una tabla objetiva valorando sus puntos fuertes y débiles en el ámbito cualitativo, estos datos se recogerán en la tabla 1:

- **Diseño:** Aspecto y estética del sistema.
- **Facilidad de aprendizaje:** Capacidad para un usuario que desconozca la herramienta para aprender a utilizar toda la información.
- **Facilidad de uso:** Capacidad del usuario para manejar la herramienta. Esfuerzo que requiere la herramienta para su utilización.
- **Contenido:** Medición del contenido de la herramienta al sistema.

Para valorar estas características se ha utilizado una escala de Likert. La escala es la siguiente:

*1 total desacuerdo.*

*2 en desacuerdo.*

*3 ni de acuerdo ni en desacuerdo.*

*4 de acuerdo.*

*5 totalmente de acuerdo.*



Tabla 1: Características cualitativas

Sistema	Diseño	Facilidad de aprendizaje	Facilidad de uso	Contenido
<b>Twitter Analytics</b>	4	3	3	4
<b>Sprout Social</b>	3	4	3	3
<b>HashTracking</b>	4	3	4	3

### 2.3.2 Características cuantitativas

Para elaborar un análisis cuantitativo de las tres herramientas propuestas existentes en el mercado, hemos clasificado una serie de características detalladas a continuación y que mostraremos en la tabla 2:

**Precio:** Determina si la herramienta es de pago indicando su coste o gratuita.

**Autenticación:** Determina si es necesario autenticarse con una cuenta Twitter o no.

**Cuentas:** Determina si gestiona varias redes sociales o solo una.

**Información:** Determina si muestra información del propio usuario o de más usuarios.

**Descarga:** Determina si es posible o no descargar los datos analizados en un fichero externo.

Tabla 2: Características cuantitativas.

Sistema	Precio	Autenticación	Cuentas	Información	Descarga
<b>Twitter Analytics</b>	Gratuito	Sí	Twitter	Propio usuario	Sí
<b>Sprout Social</b>	25€/mes	Sí	Varias	Todos	Sí
<b>HashTracking</b>	Desde 50€/mes	No	Varias	Todos	No

Como conclusión de los sistemas analizados anteriormente, podemos extraer como información útil que las herramientas analizadas presentan una mayor dificultad de aprendizaje en los sistemas más completos, es decir, aquellos sistemas que incorporan más funcionalidades. También podemos apreciar que los sistemas de pago trabajan con un número de redes sociales más amplio, permitiendo trabajar a los usuarios en distintos ámbitos. En general, son herramientas más elaboradas que están en continuo desarrollo y perfeccionamiento. Debido a todos estos razonamientos, destacamos la herramienta de pago Hashtracking como la mejor, puesto que es la más



completa de las tres. Además, esta herramienta es utilizada por múltiples empresas por la cantidad de información que podemos obtener sobre cuáles son los *hashtags* más populares y de mayor influencia en la red Twitter. Como única desventaja sobre las demás es su precio, pues es la herramienta más cara que hemos analizado.

Con nuestro sistema obtendremos análisis sobre datos de interés únicamente centrándonos en el sector de la educación, logrando que la etapa docente mejore tanto para el profesor como para el alumnado. Además, nuestra herramienta será completamente gratuita.

## 2.4 Síntesis

Una vez hemos analizado las características de relevancia en las distintas herramientas seleccionadas, pasaremos a proceder a detallar las características que más nos podrán servir de utilidad para proporcionar la funcionalidad de nuestro sistema.

Estas características se detallarán en la tabla 3.

*Tabla 3: Descripción de las características a cumplimentar.*

Nombre	Descripción
<b>Co1</b>	La red social que se utilizará será Twitter a través de su API.
<b>Co2</b>	El sistema será gratuito en su totalidad.
<b>Co3</b>	El sistema constará de una aplicación Android en la que los usuarios interactuarán con la red social Twitter enviando sus estados emocionales y su nivel de satisfacción en clase.
<b>Co4</b>	Los datos serán almacenados y podrán consultarse en una base de datos externa.
<b>Co5</b>	El sistema requerirá autenticación previa con una cuenta de Twitter.
<b>Co6</b>	El sistema en todo momento será conocedor del grupo de clase y de la asignatura que esté cursando en ese momento el alumno.

## 2.5 Tecnología utilizada

La tecnología utilizada para desarrollar nuestro proyecto consistirá en una base de datos, así como un entorno de desarrollo y nuestra plataforma de Twitter. Para poder utilizar todas estas funcionalidades hemos empleado las siguientes herramientas y tecnologías disponibles en el mercado y de uso gratuito:



- **Android Studio** <sup>6</sup>: Entorno de desarrollo para plataformas de desarrollo Android.
- **Emulador 5x Nexus** <sup>7</sup>: Simulador de móvil para ejecutar la aplicación Android.
- **Fabric sdk.Android:Twitter** <sup>8</sup>: API de Twitter integrada en el entorno de desarrollo Android Studio.
- **Picasso** <sup>9</sup>: Módulo para la visualización de los contenidos gráficos en Android.
- **FireBase** <sup>10</sup>: Base de datos no relacional de tipo *JSON* utilizada para el almacenamiento y la consulta de datos.

## 2.6 Conclusión

Hoy en día muchas herramientas trabajan con las redes sociales por su gran número de usuarios y por su gran impacto en la sociedad. Concretamente, en la red social Twitter existen numerosas herramientas que permiten la extracción de datos para sacar conclusiones, pero no existe ninguna que se centre en el estudio de datos en la docencia. Nuestra aplicación está orientada en ese sentido. Por medio de una aplicación móvil permitirá a los alumnos que estudian en la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València de evaluar las clases según sus sensaciones. De esta forma, nuestro sistema garantiza un aumento de la calidad en la enseñanza en las aulas permitiendo tanto al alumnado como a los docentes emplear técnicas y adquirir flexibilidad a la hora de impartir las clases. En este capítulo se han analizado diferentes herramientas, en las cuales se han elaborado dos tablas, permitiendo obtener sus características cuantitativas y cualitativas. Posteriormente, se han elaborado una serie de funcionalidades que tendrá nuestro sistema que no tienen el resto de las herramientas analizadas. Dichas funcionalidades se han agrupado por características en el apartado de síntesis y por último se ha mencionado la tecnología que ha sido empleada en nuestro sistema.

---

<sup>6</sup> <https://developer.android.com/studio/index.html?hl=es-419>

<sup>7</sup> <https://www.visualstudio.com/es/vs/msft-android-emulator>

<sup>8</sup> <https://docs.fabric.io/android/fabric/overview.html>

<sup>9</sup> <http://square.github.io/picasso/>

<sup>10</sup> <https://firebase.google.com>



# 3. Especificación de requisitos

---

Para la especificación de requisitos de nuestro proyecto seguiremos el estándar IEEE 830 que está dividido en las siguientes fases:

## 3.1 Introducción

### 3.1.1 Propósito

Este sistema desarrollado forma parte de un proyecto colectivo, es decir, es dependiente de otros proyectos y que comparten las mismas especificaciones. En el capítulo de especificación de requisitos se pretende determinar la función completa del sistema.

Las tablas 4 a 7 describen los roles de los actores involucrados en el proyecto. La nomenclatura que hemos utilizado es la siguiente:

**Nombre:** Nombre de la persona o entidad involucrada en el proyecto.

**Rol:** Función en la está involucrado.

**Cargo profesional:** Cargo que ocupa

**Contacto:** Email de contacto de la persona o entidad.

*Tabla 4: Miembro Miguel Rebollo Pedruelo*

<b>Nombre:</b>	<b>Miguel Rebollo Pedruelo</b>
<b>Rol:</b>	<ul style="list-style-type: none"><li>• Supervisar y ayudar en el desarrollo del proyecto.</li><li>• Resolver aclaraciones y dudas expuestas por el desarrollador.</li></ul>
<b>Cargo profesional</b>	Profesor Titular de la Universidad.
<b>Contacto</b>	mrebollo@upv.es

Tabla 5: Miembro Ángeles Calduch Losa.

<b>Nombre:</b>	<b>Ángeles Calduch Losa</b>
<b>Rol:</b>	<ul style="list-style-type: none"><li>• Supervisar y ayudar en el desarrollo del proyecto.</li><li>• Resolver aclaraciones y dudas expuestas por el desarrollador.</li></ul>
<b>Cargo profesional</b>	Profesora Colaboradora.
<b>Contacto</b>	mcalduch@eio.upv.es

Tabla 6: Miembro José Luis Poza Luján.

<b>Nombre:</b>	<b>José Luis Poza Luján</b>
<b>Rol:</b>	<ul style="list-style-type: none"><li>• Supervisar y ayudar en el desarrollo del proyecto.</li><li>• Resolver aclaraciones y dudas expuestas por el desarrollador.</li></ul>
<b>Cargo profesional</b>	Profesor Contratado Doctor.
<b>Contacto</b>	jopolu@disca.upv.es

Tabla 7: Miembro José Manuel Martínez Serrano

<b>Nombre:</b>	<b>José Manuel Martínez Serrano</b>
<b>Rol:</b>	<ul style="list-style-type: none"><li>• Desarrollar la aplicación Android.</li><li>• Testear el funcionamiento del proyecto.</li><li>• Coordinación con el personal de proyectos ajenos directamente involucrados con este.</li></ul>
<b>Cargo profesional</b>	Estudiante universitario.
<b>Contacto</b>	jomarse7@inf.upv.es

### 3.1.2 Ámbito de sistema

Este sistema que hemos desarrollado recibirá el nombre de ‘El felizometro’. Esta aplicación recibe este nombre puesto que está enfocada en medir el estado emocional y la satisfacción de los alumnos en el ámbito académico. A continuación, se detallarán las funcionalidades que implementa la aplicación y las que no implementa.

### El sistema permitirá:

- Iniciar sesión con una cuenta de Twitter. En caso de no recordar la contraseña de Twitter, el usuario tendrá la posibilidad de cambiar la contraseña mediante el correo electrónico.
- La aplicación podrá publicar *tweets* que reflejan el estado emocional de los alumnos.
- La aplicación permitirá al alumno escoger un grupo de clase de la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València. La aplicación permitirá al alumno escoger una asignatura ofertada en la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València.
- Publicar varios *tweets* con diferentes estados en una misma sesión.

### El sistema no permitirá:

- Acceder a la sesión si no es con una cuenta de *Twitter* activa.
- Eliminar *tweets* después de haber sido publicados anteriormente.

Con este proyecto se espera tener un mayor conocimiento sobre la calidad de las clases de los profesores, ayudándoles a mejorar y a favorecer un ámbito de aprendizaje más ameno. La aplicación permite detectar retroalimentación del sentimiento de los alumnos ante las actividades que se plantean en el aula, de forma que el profesor pueda corregir rápidamente situaciones negativas como aburrimiento o falta de comprensión y refuerce las positivas.

### 3.1.3 Definiciones

A continuación, hemos detallado en la tabla 8 las definiciones que hemos utilizado durante todo el proyecto y que nos permitirán entender con mayor exactitud su significado.

Tabla 8: Definiciones técnicas.

Nombre	Descripción
<b>Android Studio</b>	Entorno de desarrollo integrado para la plataforma Android.
<b>API</b>	Es un conjunto de subrutinas, funciones y procedimientos que ofrece la biblioteca Fabric de Twitter para ser utilizado en nuestra aplicación móvil.
<b>Emulador</b>	Es un software que simula un terminal Android pensado para ejecutar nuestro programa



<b>Favorito</b>	Son los <i>tweets</i> que se presentan con un icono en forma de estrella, los usuarios tienen derecho de seleccionarlos y guardarlos para su posible actualización en el futuro.
<b>Followers</b>	Es un usuario que sigue a otro en Twitter. Es decir, que está recibiendo las actualizaciones que está generando el usuario cual sigue.
<b>Hashtag</b>	Son palabras precedidas del símbolo “#” y son una forma de etiquetar o clasificar los mensajes de Twitter, de tal forma que se puedan agrupar alrededor de un tema en común.
<b>Impresiones</b>	Son una medida de dimensión. Es una forma de analizar los <i>tweets</i> según la repercusión social que puedan tener.
<b>Mención</b>	Mensaje que pone en comunicación a diferentes usuarios. Para poder utilizar una mención es necesario utilizar el carácter “@” seguido del usuario con el que se desea comunicar.
<b>Mensajes</b>	Publicación de 140 caracteres que puede hacer un usuario de Twitter.
<b>Muro</b>	Espacio donde se muestran todos los <i>tweets</i> y los <i>retweets</i> de los usuarios seguidos.
<b>Notificaciones</b>	Son las actualizaciones e información que envía Twitter al correo electrónico del usuario.
<b>Perfil de usuario</b>	Información que incluye datos básicos e información adicional que varía según el tipo y la funcionalidad de la red.
<b>Retweet</b>	Tipo de mensaje que permite reenviar un mensaje recibido.
<b>Sistema</b>	<ul style="list-style-type: none"> <li>• Aplicación móvil alumnado.</li> <li>• Aplicación móvil docente.</li> <li>• Base de datos.</li> </ul>
<b>Tweet</b>	Es el mensaje que podemos enviar a través de Twitter. (Ver mensaje).
<b>Usuario</b>	Es una persona o una identidad registrada en la red social Twitter.

## 3.2 Descripción general

### 3.2.1 Perspectiva del producto

El sistema a desarrollar forma parte de un proyecto colectivo. Este proyecto es un subproyecto de un estudio amplio que trabaja con las redes sociales tanto de *LinkedIn*, *Twitter* o *Facebook*. Este proyecto se apoya con otro TFG que se está llevando a cabo de título Desarrollo de una aplicación para el análisis de los sentimientos de los alumnos durante una clase (Vázquez, 2017), que desarrollará un sistema capaz de recoger y analizar todos los datos almacenados en Twitter por nuestra aplicación y analizar. Concretamente, desarrollará el entorno que utilizará el profesor al que le interesa conocer el estado de la situación anímica y emocional en tiempo real de sus alumnos. El proyecto estará desarrollado por los actores mencionados en la sección 3.1.1.

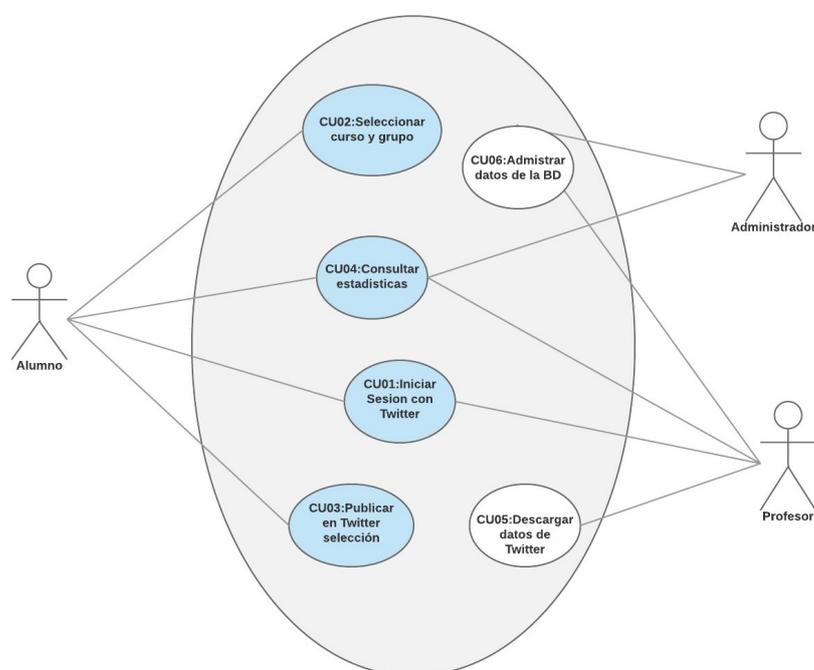


### 3.2.2 Funciones del producto

Las funciones principales del sistema completo las podemos recoger en los diferentes casos de uso que se detallarán en la ilustración 8. En esta gráfica se indicarán los casos de uso del sistema completo.

En este proyecto únicamente hemos desarrollado los casos de uso que aparecen en azul puesto que es la funcionalidad utilizada por el alumno.

Ilustración 8: Diagrama de casos de uso



A continuación, en la tabla 9, hemos relacionado los casos de uso anteriormente descritos en el gráfico con las características detalladas en la sección 2.4.

Tabla 9: Casos de uso

Caso de uso	Descripción	Característica
<b>CU01</b>	Iniciar sesión con Twitter.	C05
<b>CU02</b>	Seleccionar curso y grupo.	C06
<b>CU03</b>	Publicar <i>tweet</i> .	C03
<b>CU04</b>	Consultar estadísticas.	C04

### 3.2.3 Características de los usuarios

En el sistema existen tres tipos de usuarios: los usuarios alumnos, los usuarios profesores y los usuarios administradores. En este proyecto únicamente nos centraremos en los usuarios alumnos puesto que ellos son los encargados de publicar su nivel de satisfacción y su estado emocional.

Tabla 10: Rol usuarios

<b>Usuario</b>	<b>Rol/Función</b>
<i>Alumno</i>	<ul style="list-style-type: none"><li>• Publicar estados emocionales.</li><li>• Publicar nivel de satisfacción.</li><li>• Consultar estadísticas.</li></ul>

### 3.2.4 Restricciones

En el proyecto desarrollado se identifican dos tipos de restricciones: las restricciones de los desarrolladores y las restricciones del usuario.

Restricciones del desarrollador:

- El desarrollador del proyecto está limitado a trabajar con la red social Twitter.
- El desarrollador del proyecto debe trabajar con una base de datos conjunta para fomentar la cohesión con el sistema.
- El desarrollador tendrá restringido todo aquello que no está sujeto al contrato inicial donde se determinaron las condiciones de este proyecto.

Restricciones del usuario:

- El usuario necesitará obligatoriamente una cuenta de Twitter.
- El usuario necesitará utilizar un dispositivo con sistema operativo Android.
- El usuario necesitará que ese dispositivo Android sea un teléfono móvil, no permitiéndose el uso de tabletas electrónicas.
- El usuario necesitará en todo momento acceso a Internet.



### 3.2.5 Suposiciones y dependencias

Destacamos que en este proyecto, para el correcto funcionamiento de la aplicación, necesitamos un número notable de alumnos que valoren una clase para poder tener estadísticas más precisas.

Por otra parte, necesitamos a lo largo del proceso dependencia de Internet, puesto que no utilizamos un servidor privado, si no que utilizamos el servidor de Twitter y una base de datos en línea para intercambiar información sobre estas distintas aplicaciones.

### 3.2.6 Requisitos futuros

Para mejorar el sistema en futuras versiones podríamos incluir los siguientes requisitos:

- Desarrollar el sistema en plataformas iOS o web.
- Poder configurar las mediciones según el interés del profesor. Es decir, este sistema mide la satisfacción del alumno en clase y su estado emocional, pero bajo demanda del profesor podría configurar una serie de mediciones de su interés.
- Ampliar el servicio para docentes de diferentes escuelas o universidades.

## 3.3 Requisitos específicos

En este punto se detallan concretamente todos los requisitos a cumplir para que el desarrollo de nuestro sistema sea óptimo.

Para ello se ha establecido unas etiquetas que se han dividido en tres apartados que contienen los requisitos específicos de diseño, los requisitos específicos de funcionalidad y otros requisitos específicos con el contenido que se detalla a continuación:

**Nombre:** Establece el nombre del requisito específico a cumplimentar.

**Descripción:** Establece de manera detallada en qué consiste ese requerimiento específico.

**Tipo de requisito:** Establece qué tipo es (diseño, funcionalidad u otro requisito).

**Prioridad:** Prioridad de implementación por los desarrolladores.



**Caso de uso:** Establece con que caso de uso lo podremos relacionar de entre los mencionados anteriormente en el capítulo tres, apartado funciones del producto.

### 3.3.1 Requisitos de diseño

*Tabla 11: Requisitos de interoperabilidad de la aplicación.*

<i>Nombre</i>	RDo1: Interoperabilidad de la aplicación.
<i>Descripción</i>	<ul style="list-style-type: none"> <li>El sistema permitirá adaptarse a cualquier dispositivo móvil que funcione con la plataforma Android.</li> </ul>
<i>Tipo de requisito</i>	Requisito específico de diseño.
<i>Prioridad</i>	Media/Baja.
<i>Caso de uso</i>	CU01.

*Tabla 12: Requisitos de formularios de la aplicación*

<i>Nombre</i>	RDo2: Formularios de la aplicación.
<i>Descripción</i>	<ul style="list-style-type: none"> <li>El diseño de los formularios será para una aplicación Android para teléfonos móviles.</li> <li>El usuario escogerá entre una asignatura y un curso.</li> </ul>
<i>Tipo de requisito</i>	Requisito específico de diseño.
<i>Prioridad</i>	Alta.
<i>Caso de uso</i>	CU02.

Tabla 13: Requisitos de diseño de la base de datos

<i>Nombre</i>	RDO3: Diseño en la base de datos.
<i>Descripción</i>	Se consultarán datos sobre las estadísticas calculadas por el otro proyecto del sistema.
<i>Tipo de requisito</i>	Requisito específico de diseño.
<i>Prioridad</i>	Alta.
<i>Caso de uso</i>	CU04.

### 3.3.2 Requisitos de funcionalidad

Tabla 14: Requisitos inicio de sesión con Twitter.

<i>Nombre</i>	RF01: Inicio de sesión con Twitter.
<i>Descripción</i>	<ul style="list-style-type: none"> <li>• Será obligatorio tener una cuenta de Twitter para poder acceder a la aplicación móvil.</li> <li>• A través de un API de Twitter integrada en el entorno de desarrollo Android Studio podremos desarrollar el inicio de sesión con Twitter.</li> </ul>
<i>Tipo de requisito</i>	Requisito específico de funcionalidad.
<i>Prioridad</i>	Alta.
<i>Caso de uso</i>	CU01.

Tabla 15: Requisitos publicación de tweets

<i>Nombre</i>	RF02: Publicación de <i>tweets</i> .
<i>Descripción</i>	<ul style="list-style-type: none"> <li>• El usuario, al final del formulario publicará un <i>tweet</i> con dos etiquetas. La primera etiqueta contendrá el nombre de la asignatura, mientras que la segunda etiqueta contendrá el grupo al que pertenece dicho usuario.</li> <li>• Además de publicar las etiquetas, también se incluirán las valoraciones elegidas por el usuario en cuanto al nivel de satisfacción y su estado emocional.</li> <li>• Todos estos datos se asociarán a un <i>tweet</i>.</li> </ul>
<i>Tipo de requisito</i>	Requisito específico de funcionalidad.
<i>Prioridad</i>	Alta.
<i>Caso de uso</i>	CU03.

Tabla 16: Requisitos consultar estadísticas.

<i>Nombre</i>	RF03: Consultar estadísticas.
<i>Descripción</i>	<ul style="list-style-type: none"> <li>• El alumno también tendrá la posibilidad de consultar sus estadísticas antes de valorar una sesión académica.</li> <li>• Estas estadísticas se mostrarán en forma de gráficos.</li> <li>• Se incluirán varios típicos de gráficos indicando el nivel de satisfacción medio de la clase o el nivel de motivación que los alumnos presentan en ese momento.</li> </ul>
<i>Tipo de requisito</i>	Requisito específico de funcionalidad.
<i>Prioridad</i>	Media.
<i>Caso de uso</i>	CU04.

### 3.3.3 Otros requisitos.

Tabla 17: Requisitos de adecuación con los estándares y la normativa

<i>Nombre</i>	OR01: Adecuación con los estándares y la normativa.
<i>Descripción</i>	<ul style="list-style-type: none"><li>• El proyecto, en todo momento cumple con los diseños y estándares de la normativa IEEE 830.</li><li>• El proyecto cumple la normativa de servicios de uso de Twitter, y se adecúa a sus restricciones.</li><li>• El proyecto integra patrones de diseño que garantizan el cumplimiento de las recomendaciones de las reglas de diseño software.</li><li>• El proyecto garantiza y asegura mantenibilidad, facilidad de aprendizaje y facilidad de uso.</li></ul>
<i>Tipo de requisito</i>	Otros requisitos.
<i>Prioridad</i>	Media.
<i>Caso de uso</i>	-

## 3.4 Conclusiones

En este capítulo se han especificado los requisitos que se han regido por el estándar IEEE 830 para la especificación de requisitos software. También se han establecido los diferentes casos de uso que definen la funcionalidad del sistema. Por otra parte, hemos detallado los requisitos específicos que se han adaptado a los casos de uso mencionados en las funciones del producto y se ha añadido su nivel de prioridad de implementación para capítulos posteriores.

# 4. Diseño

---

## 4.1 Introducción

En primer lugar, detallaremos de dónde surge la idea de inspiración para el diseño de nuestra aplicación móvil correspondiente al alumno. Posteriormente se ha elaborado un esquema conceptual en el que se mostrará de forma genérica cómo funcionará nuestro sistema. Una vez detallados todos estos aspectos se redactarán de forma más precisa todos los componentes de nuestro sistema. Para ello se han elaborado una serie de diagramas que permitirán explicar con más claridad dichos componentes. Finalmente se ha elaborado una conclusión para resumir aspectos y matices importantes.

## 4.2 Inspiración

Este sistema está inspirado en otro TFG de título Extracción de información de Twitter para la creación de redes de interacción entre usuarios (Torró, 2016), el cual hizo un sistema para determinar el sentimiento de los alumnos de la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica. En este proyecto hemos decidido ampliar los conceptos y satisfacer nuevas necesidades, tanto como para los alumnos como para los profesores.

En la ilustración 9 podemos ver un ejemplo del diseño de la aplicación elaborada por Torró y que nos ha servido de inspiración para la elaboración del presente proyecto.

En el sistema que se ha elaborado este año está compuesto por dos proyectos en el que en este tiene como objetivo elaborar una aplicación móvil para el alumnado y en otro proyecto se implementará la aplicación móvil que usará el profesorado para poder ser valorados y ser conocedores de sus estadísticas dentro de las aulas.

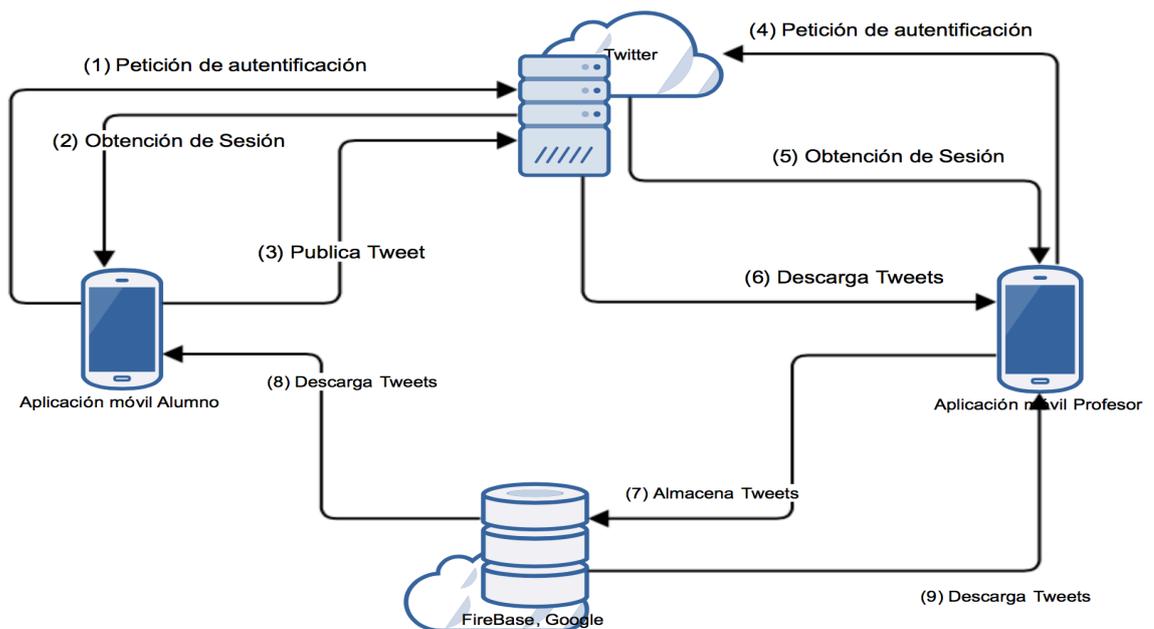
Ilustración 9: Pantalla aplicación móvil del proyecto Felizómetro2016



### 4.3 Especificación conceptual

El esquema de la ilustración 10 muestra el ciclo de funcionamiento de la aplicación y los distintos componentes involucrados.

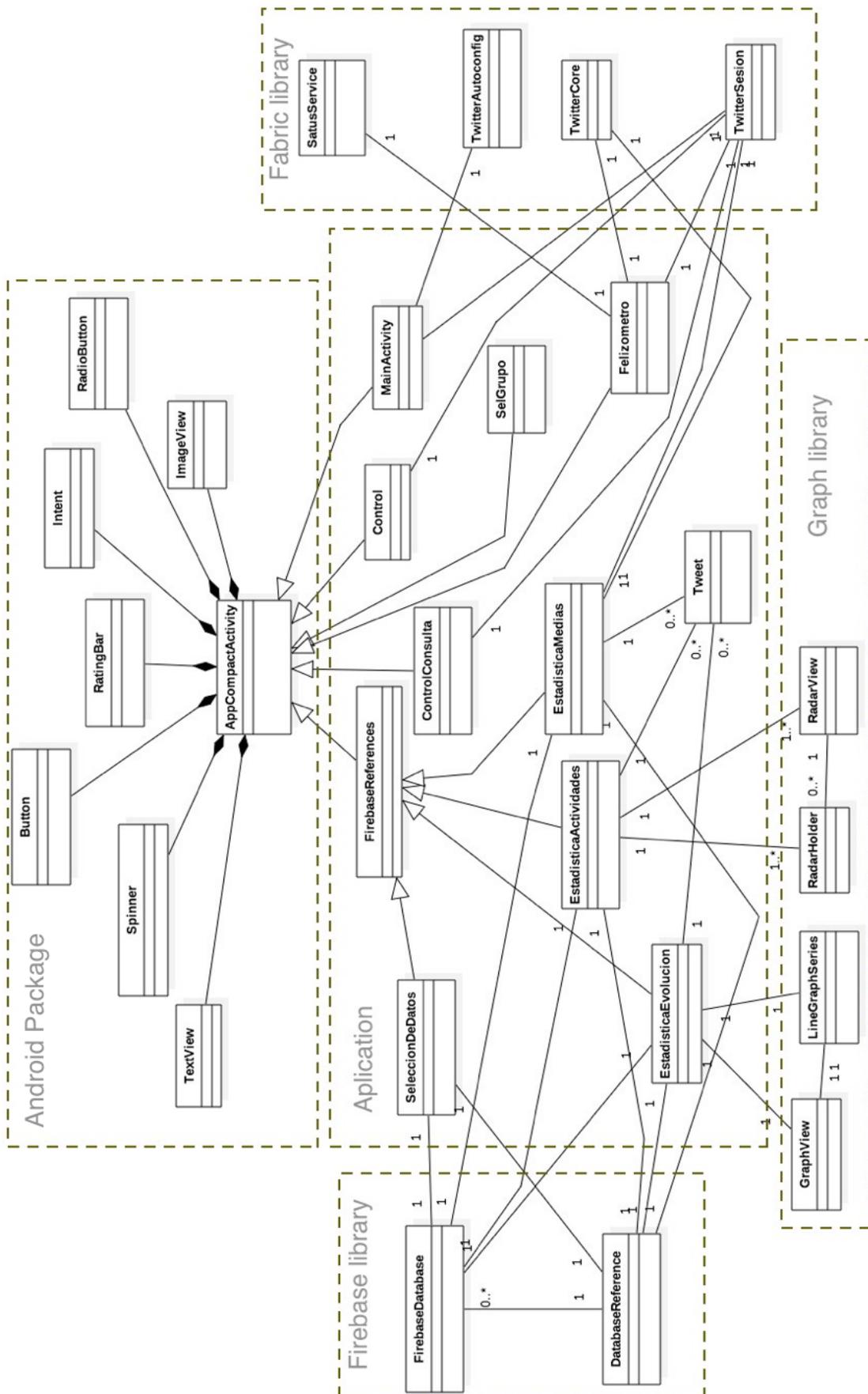
Ilustración 10: Diagrama esquema conceptual



Como se detalla en este esquema de la ilustración 10, el usuario de tipo alumno se identifica con su cuenta de Twitter (1) para obtener su sesión (2). Una vez con su sesión activa en la aplicación móvil, el usuario de tipo alumno puede publicar *tweets* a lo largo de la sesión docente (3). A su vez, la aplicación que utiliza el profesor también se identifica con una cuenta Twitter (4) y obtiene su sesión activa (5). La aplicación profesor descarga la lista de *tweets* publicados en Twitter (6) y los almacena en la base de datos (7). Finalmente, tanto la aplicación móvil del alumno como del profesor descargan la lista de *tweets* (8) (9) de la base de datos para mostrar los resultados.

La ilustración 11 contiene un diagrama de clases de UML en el que se detallan todas las clases que contiene nuestra aplicación, así como las clases de las distintas bibliotecas que utilizamos y las clases de las herramientas que nos proporciona Android. En esta ilustración únicamente se mostrará el nombre de las clases y como están relacionadas entre sí. También veremos reflejado unos recuadros que delimitan tanto las bibliotecas utilizadas en nuestro proyecto (Fabric de Twitter, paquete Android, GraphView para dibujar las gráficas y Firebase para la base de datos) como nuestra aplicación (recuadro Application). En el apartado siguiente de especificación formal, se mostrarán fragmentos de este diagrama con más detalle.

Ilustración 11: Diseño diagrama de clases general



## 4.4 Especificación formal

En este apartado se especificarán formalmente los diferentes componentes que interactúan en el sistema de forma más detallada. Para poder elaborar bien la fase de diseño de especificación formal, hemos decidido establecer el modelo por capas en el que se han distinguido tres capas, que son la capa de presentación, la capa lógica o de negocio y la capa de persistencia.

### 4.4.1 Capa de presentación.

En la capa presentación se han especificado a través de *mockUps* el diseño de la interfaz externa que verá el alumno cuando utilice la aplicación móvil. También se han detallado las diferentes interacciones para poder navegar por la aplicación y de esta forma poder entender mejor el funcionamiento del sistema a través de esta capa externa y que es una de las más significativas para los usuarios.

El diseño de la aplicación móvil será lo más sencillo e intuitivo posible para mejorar la facilidad de uso de los usuarios. Se pretende hacer un diseño que sea sencillo de entender por todos y que sea lo más rápido posible.

En esta capa, únicamente se ha detallado la especificación formal de la aplicación móvil del alumno. Para su correcto diseño antes de ver los diferentes *mockUps* veremos un diagrama UML con las clases implicadas directamente en la capa de presentación. Este diagrama se ha extraído del diagrama de clases (Ilustración 11) en el que se muestran las clases de la aplicación junto con la clase del *Android Package* de la cual hereda nuestro proyecto, así como las clases de la biblioteca de gráficos para poder mostrarlo. Cada clase de la aplicación tiene asociado un *layout* que mostrará el contenido gráfico de cada ventana. Por otra parte, la clase *AppCompatActivity* será la clase padre de la cual hereden todas las clases de la aplicación, esta clase entre otras cosas permite la utilización de eventos Android. Por último, a través de las clases *RadarView* y *GraphView* podremos utilizar las funciones de los gráficos.

Una vez mostrado el diagrama, se procederá con las explicaciones de los diseños gráficos con sus diferentes interacciones para tener una mayor claridad sobre la capa de presentación.



Ilustración 12: Diagrama de clases capa presentación

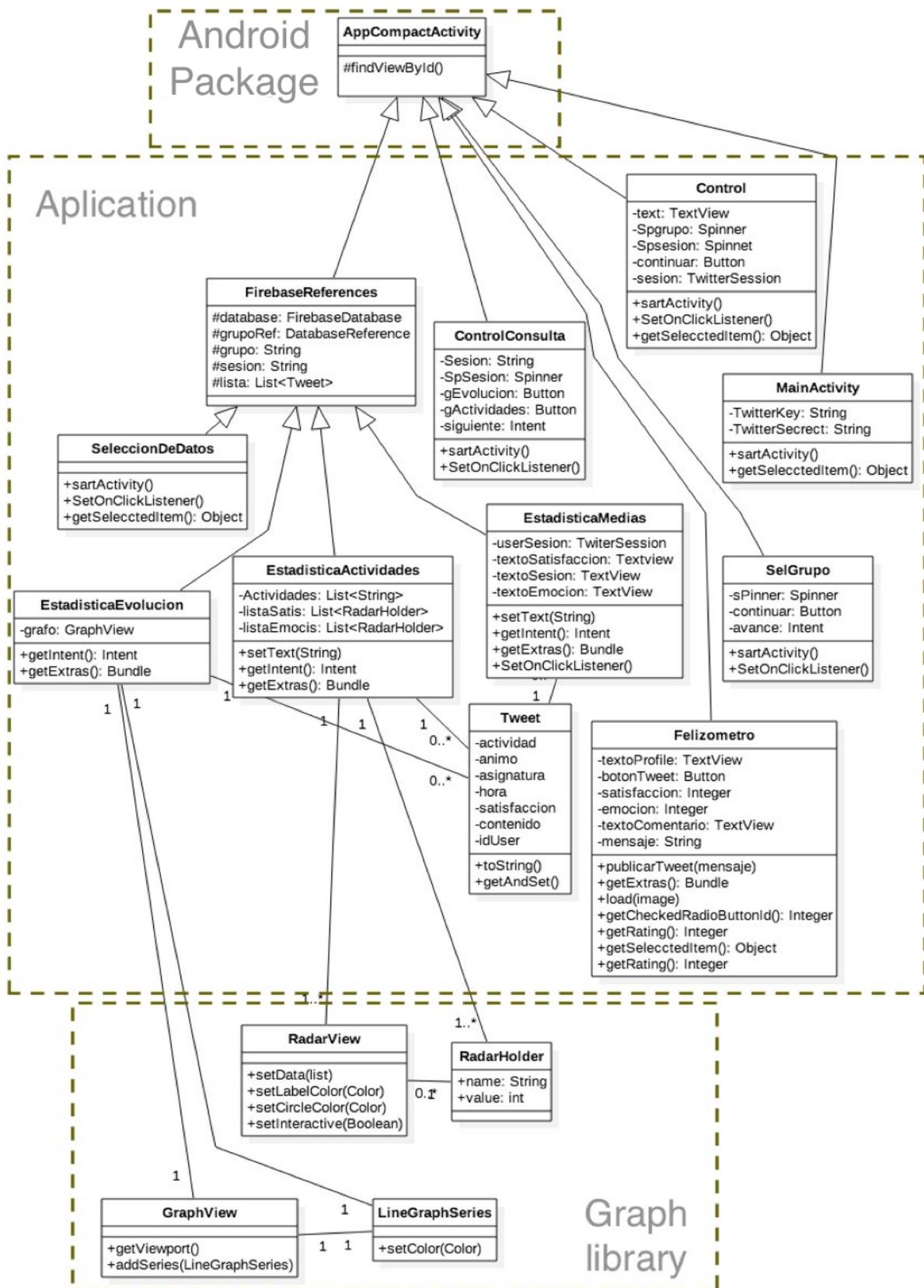
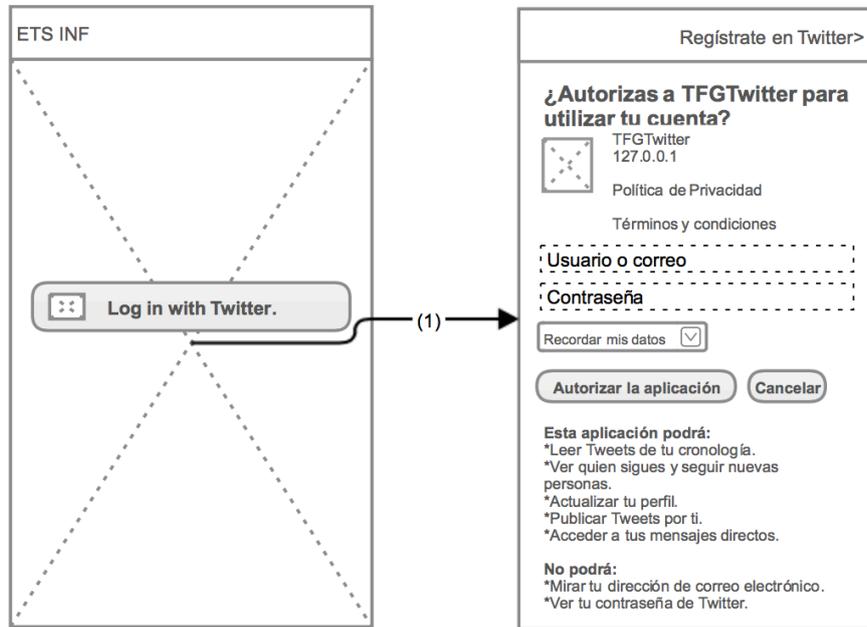
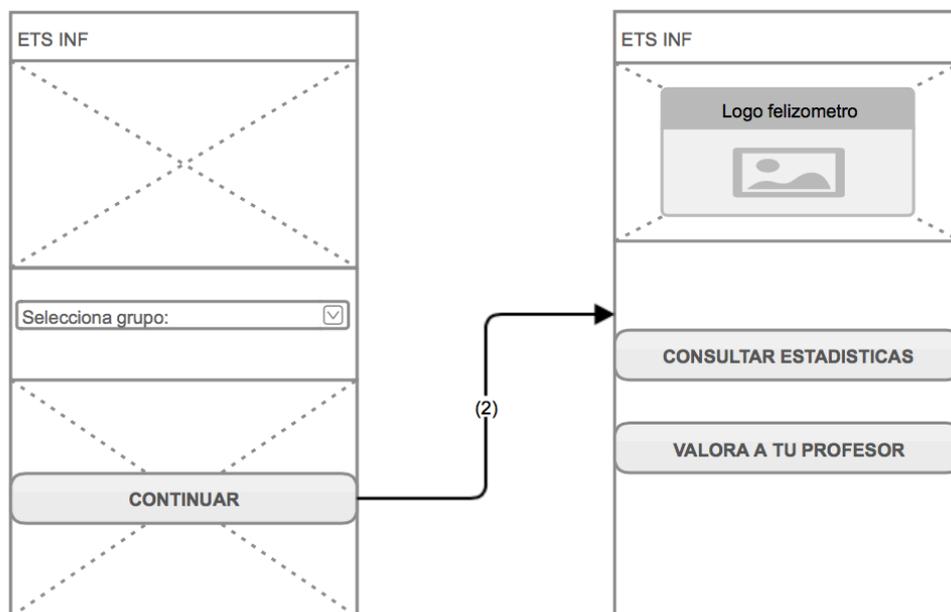


Ilustración 13: MockUps inicio de sesión



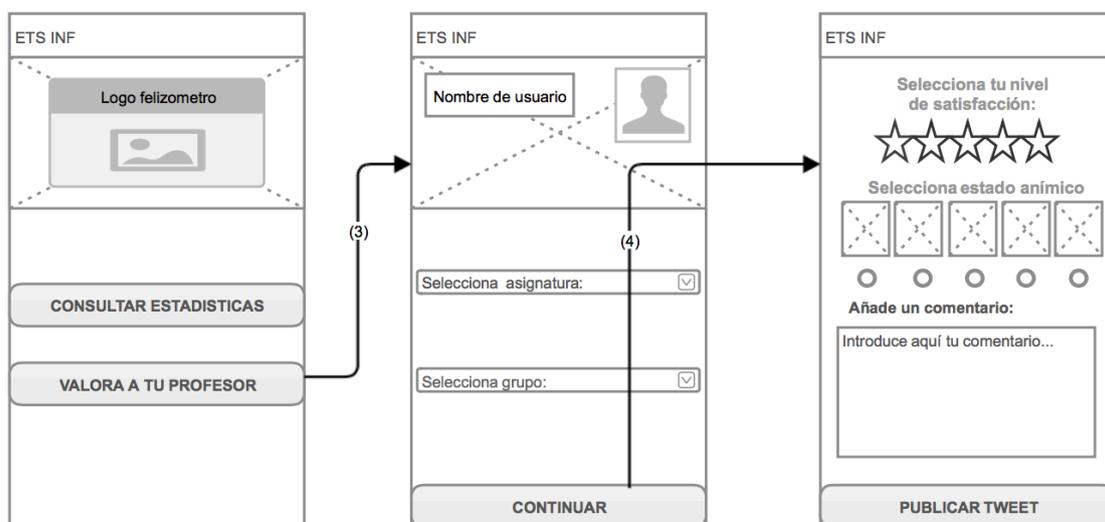
Como podemos observar en la ilustración 13 a través de el botón Log in with Twitter (1) saltamos a la pantalla de inicio de sesión, donde tendremos que indicar nuestro usuario y contraseña en Twitter.

Ilustración 14: MockUps diseño menú principal



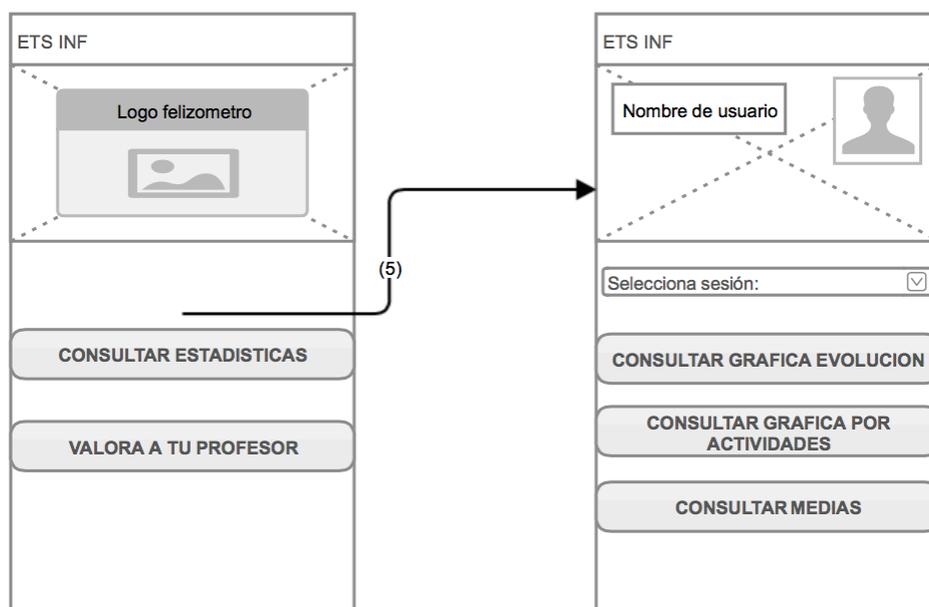
En el menú principal seleccionamos el grupo y al pulsar el botón “Continuar” nos llevará (2) al menú principal de nuestra aplicación móvil.

Ilustración 15: MockUps diseño publicación de un tweet



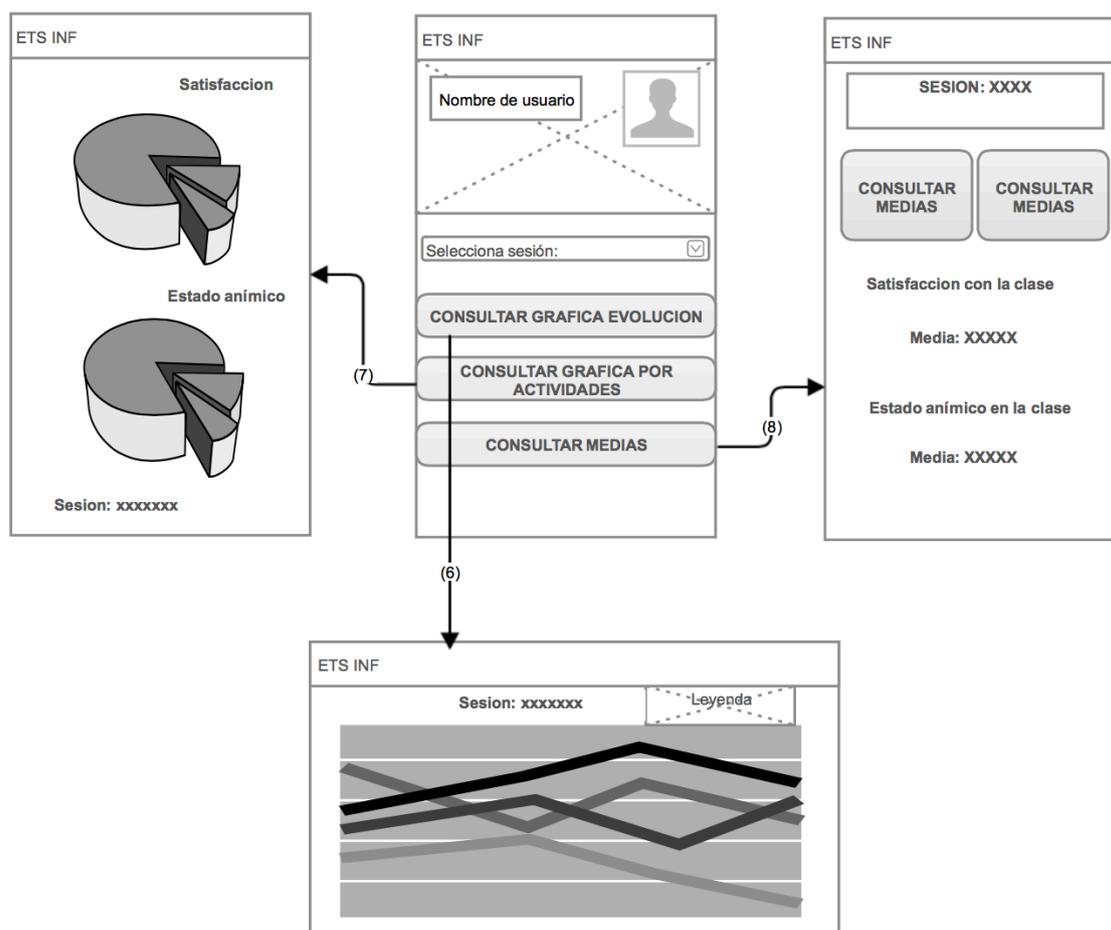
En el menú principal eligiendo la opción “Valora a tu profesor” nos llevará (3) a una pantalla en la que nos pedirá de nuevo la selección del grupo y de la asignatura que queremos valorar, continuando con la única opción disponible nos llevará (4) a una nueva pantalla en la que mediante estrellas podremos elegir el nivel de satisfacción y mediante unos iconos el estado anímico, así como la posibilidad de añadir un comentario opcional. Con el botón “publicar tweet”, si tenemos todos los campos bien rellenos, la aplicación publicará el *tweet* en Twitter.

Ilustración 16: MockUps diseño selección de sesión.



Si seleccionamos la opción (5) “Consultar estadísticas”, llegaremos a un menú donde tendremos que seleccionar una sesión disponible y tendremos varias opciones que detallaremos en la ilustración 17.

Ilustración 17: MockUps diseño consulta de estadísticas



Como podemos observar en la ilustración 17, tendremos que seleccionar una sesión disponible para poder visualizar las estadísticas. Si pulsamos el botón “Consultar gráfica evolución” nos llevará (6) a una gráfica de líneas en la que una línea representará nuestra satisfacción a lo largo de toda la sesión y la otra representará el estado emocional. Si seleccionamos la segunda opción “Consultar gráfica por actividades” nos llevará (7) a dos graficas en las que en la primera se representa la media general de satisfacción de la clase según qué actividad se esté impartiendo y en la segunda lo mismo, pero con los datos del estado emocional. Por último, si elegimos la opción “Consultar medias” nos llevará (6) a una pantalla en la que podremos elegir entre dos tipos de medias, las personales o las generales del conjunto de la clase. Según lo que escojamos, la aplicación mostrará las medias tanto de satisfacción como de estado emocional.

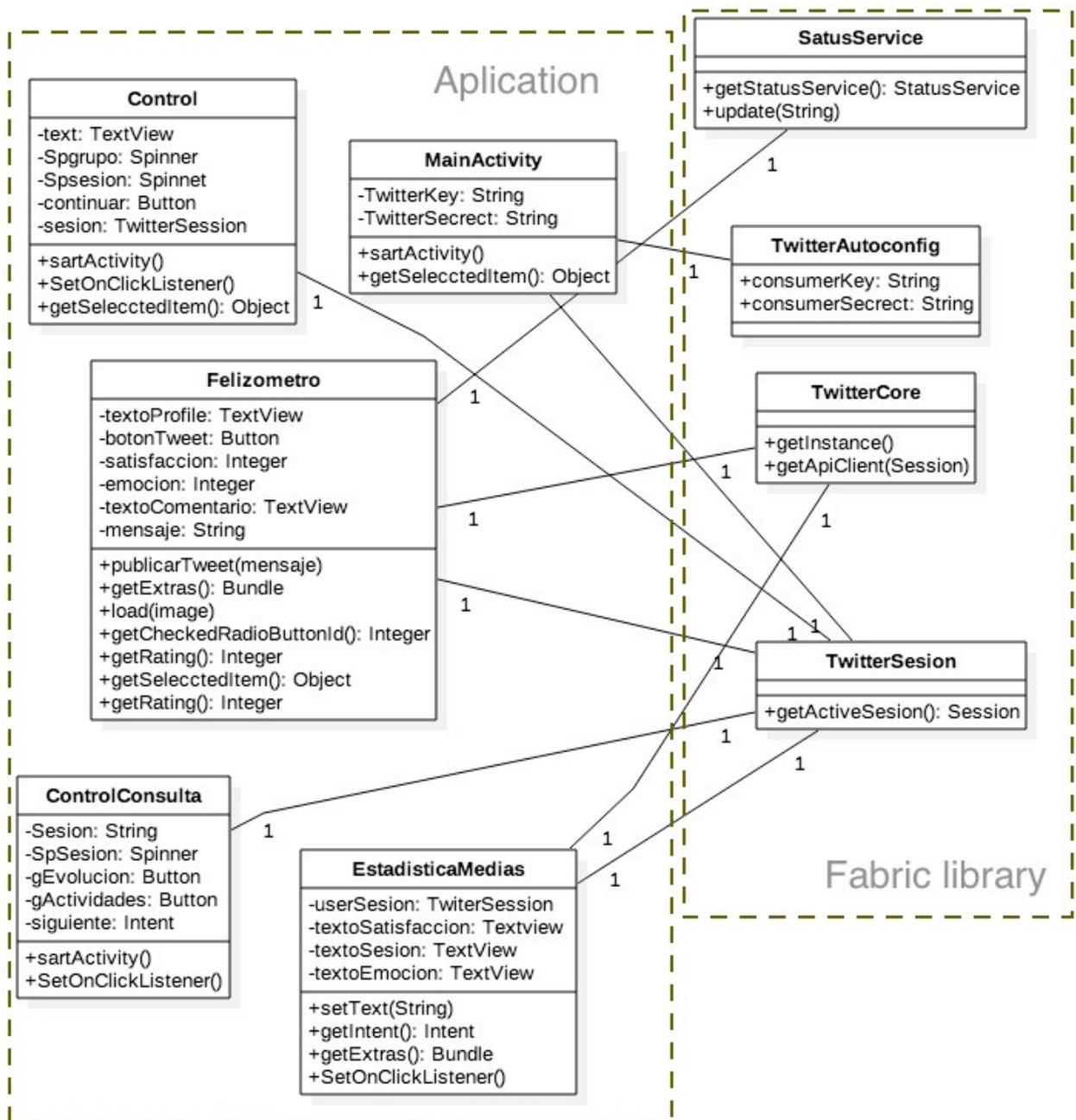
#### 4.4.2 Capa lógica

En este apartado del diseño, hemos detallado con diagramas de secuencia cómo son los diferentes casos de uso que expusimos en el capítulo 3 (Ilustración 8) de



especificación de requisitos, de forma que quede detallado gráficamente su diseño. Para una mayor aclaración sobre esta capa, también mostraremos un diagrama UML con las clases de nuestra aplicación, así como las clases de la biblioteca de Twitter implicadas en la lógica o negocio. Este diagrama nos ayudará a entender mejor el funcionamiento de las mismas, y está en la Ilustración 18.

Ilustración 18: Diagrama de clases capa lógica



En la ilustración 18 podemos apreciar que las clases Control, MainActivity, Felizómetro, ControlConsulta y EstadísticaMedias (clases pertenecientes a la aplicación) interactúan con las cuatro clases TwitterAutoconfig, SatusService,

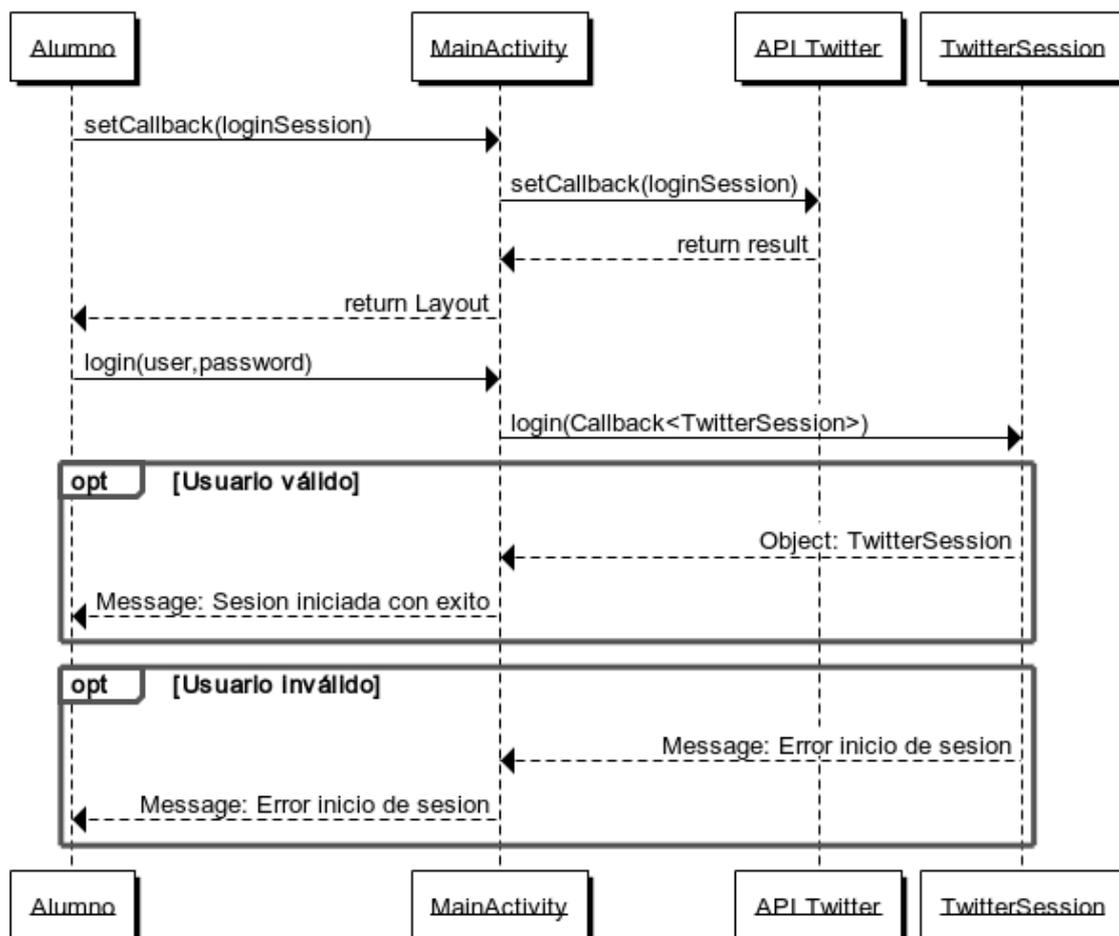


TwitterCore y TwitterSesion de Fabric. Estas cuatro últimas clases mencionadas son muy importantes en nuestro proyecto, puesto que permiten establecer comunicación con Twitter. En la clase TwitterCore tenemos los métodos principales de esta biblioteca, TwitterAutoconfig es imprescindible para establecen la comunicación con el servidor de Twitter y, TwitterSesion y StatusService son clases importantes para recuperar la sesión activa de un usuario y para publicar *tweets*.

Como mencionábamos en el capítulo 3, hemos elaborado un diagrama de secuencia para cada caso de uso (Ilustración 8). En la ilustración 19 se mostrará cómo iniciar sesión con Twitter (Cu01). En la ilustración 20 se mostrará como seleccionar curso y grupo, y publicar en Twitter la selección (Cu02 y Cu03). En la ilustración 21 se mostrará un diagrama en el que se iniciará una secuencia para consultar estadísticas (Cu04).

Ilustración 19: Diagrama de secuencia Cu01

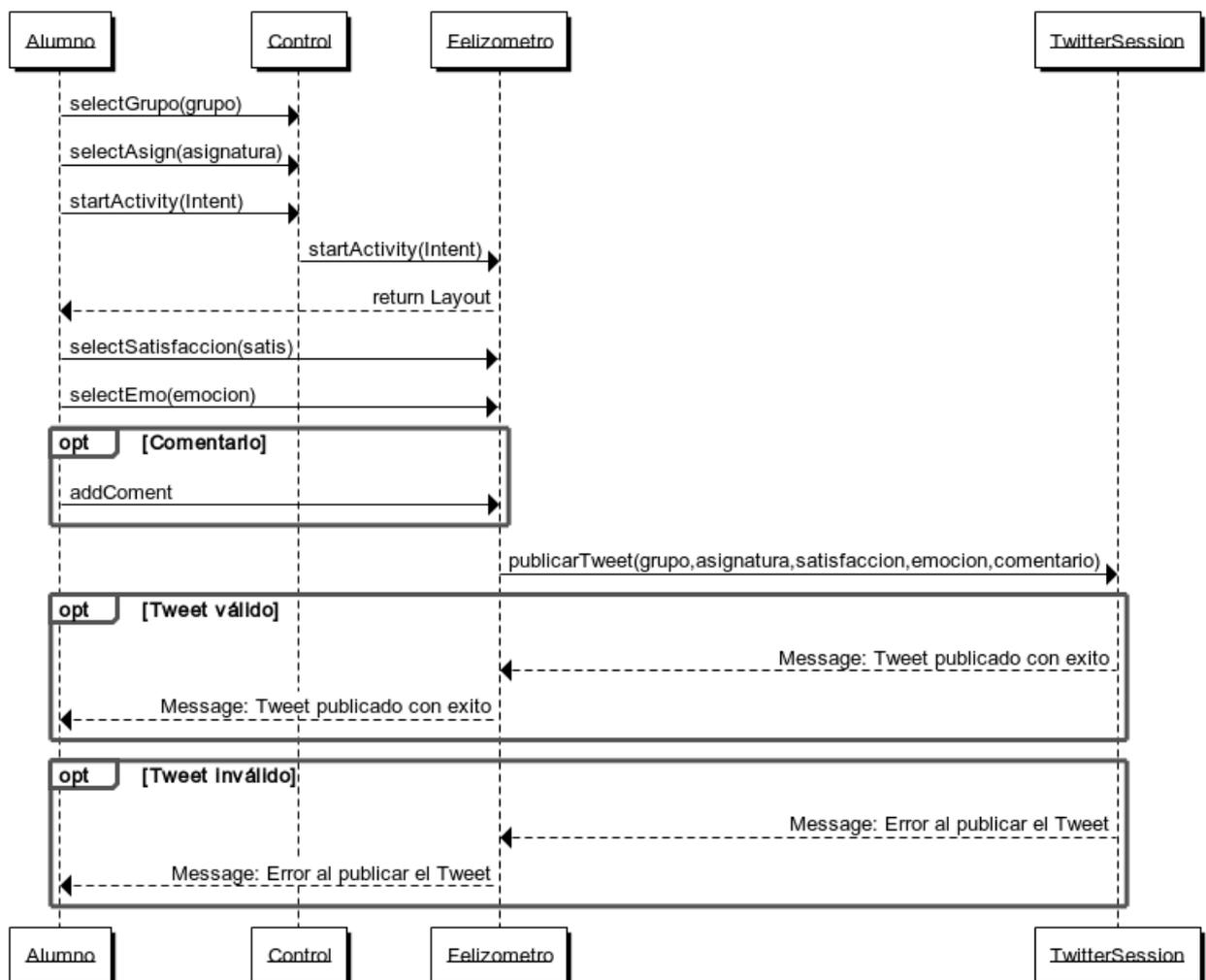
### CU02 y CU03 Selección de grupo y asignatura y publicación de un Tweet



Como se puede apreciar en la ilustración 19, el alumno a través de el botón de inicio de Sesión (Interacción 1 de Ilustración 13) podrá solicitar una petición de inicio de sesión. El API de Twitter contestará una pantalla de autenticación en la que el usuario deberá rellenar con su usuario y contraseña. Si el usuario o la contraseña son válidos, automáticamente devolverá los datos de la sesión activa y mostrará un mensaje de éxito. En caso contrario, se enviará una notificación de fallo de inicio de sesión.

Ilustración 20: Diagrama de secuencia CU02 y CU03

### CU02 y CU03 Selección de grupo y asignatura y publicación de un Tweet

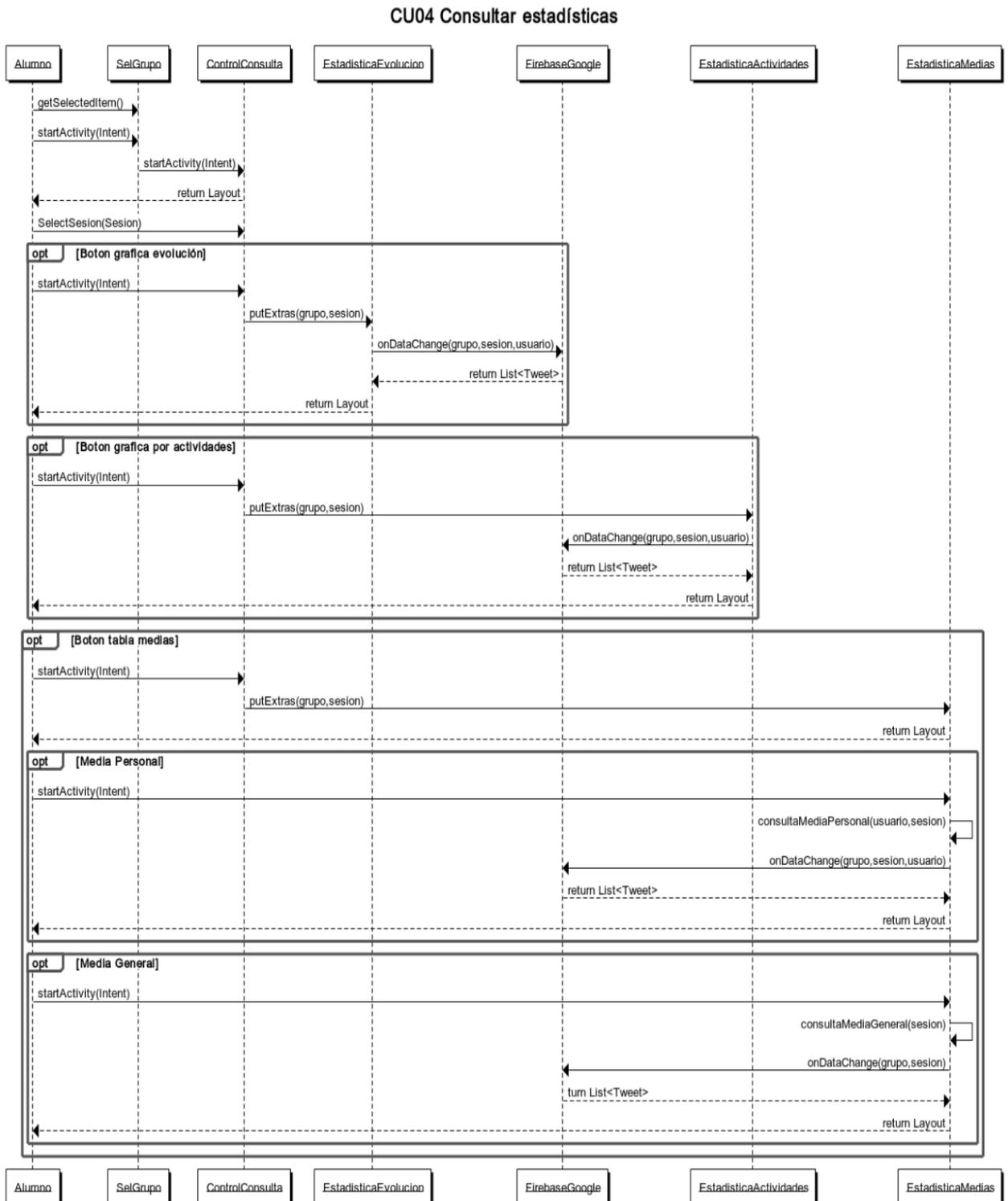


Como podemos observar en la ilustración 20, para la selección de grupo y asignatura interactuaremos con la clase Control situada en la capa lógica que nos permitirá seleccionar un grupo y una asignatura para posteriormente, tras pulsar el botón de avance, saltar a otra clase controladora llamada Felizómetro encargada de activar otra vista (Interacción 3, ilustración 15). Ahora el alumno tendrá que elegir su nivel de satisfacción como su estado emocional añadiéndose la posibilidad de escribir un comentario. Una vez todo este proceso



de selección por parte del alumno haya finalizado, la clase Felizometro publicará un *tweet* con todos los datos anteriormente escogidos por el alumno. Si todos los parámetros del *tweet* son válidos, la clase TwitterSession emitirá un mensaje de éxito. En caso contrario, esta misma clase emitirá un mensaje de error informando al alumno de ello.

Ilustración 21: Diagrama de secuencia Cuo4



Como se detalla en el diagrama de la ilustración 21 correspondiente al caso de uso CU04, el alumno primero seleccionará el grupo mediante la clase SelGrupo. Una vez avanzado a la siguiente pantalla perteneciente a la clase de negocio ControlConsulta, tendrá que seleccionar una sesión activa con la opción de escoger entre estas tres consultas que le permitirán visualizar cada una de las estadísticas planteadas en la aplicación móvil. Estas clases de estadísticas se corresponden con la capa de presentación (Interacciones 6, 7 y 8 de la Ilustración 17). Cabe destacar que cada una de estas estadísticas calculadas son opcionales y en ningún caso el alumno tendrá la obligación de acceder a ellas.

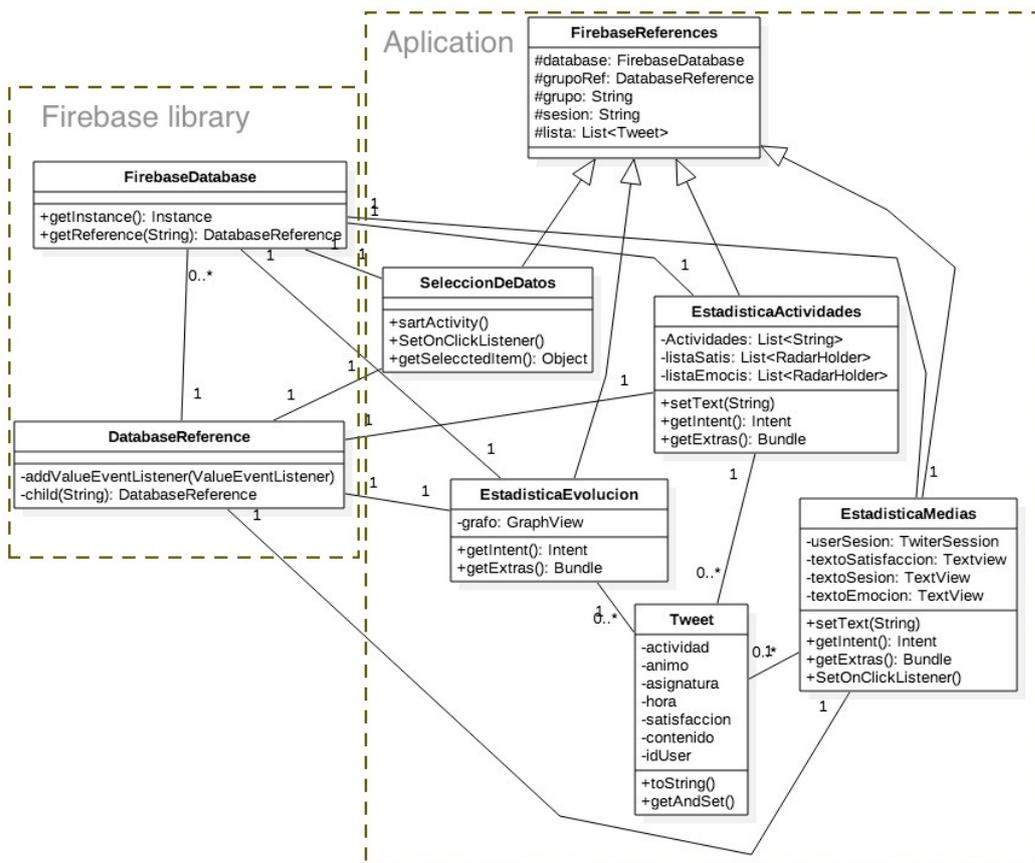
#### 4.4.3 Capa persistencia

En esta capa trataremos aspectos generales del diseño relacionados con la base de datos. Como se mencionó en la sección 2.5, la base de datos que hemos utilizado es la de Google *Firebase*. Esta base de datos permite una estructura no-relacional con almacenamiento de datos como objetos de tipo *JSON* diferente a las bases de datos relacionales tradicionalmente utilizadas.

En la ilustración 22 se muestra un diagrama en el que aparecen las clases de nuestra aplicación, así como las clases de la biblioteca de la base de datos implicadas en esta capa de persistencia. Las clases *FirebaseDatabase* y *DatabaseReference* son importantes en nuestra aplicación, ya que la primera se encarga de establecer conexión con la base de datos de Google y la segunda establece referencias de búsqueda para las consultas que realizaremos durante el proyecto.

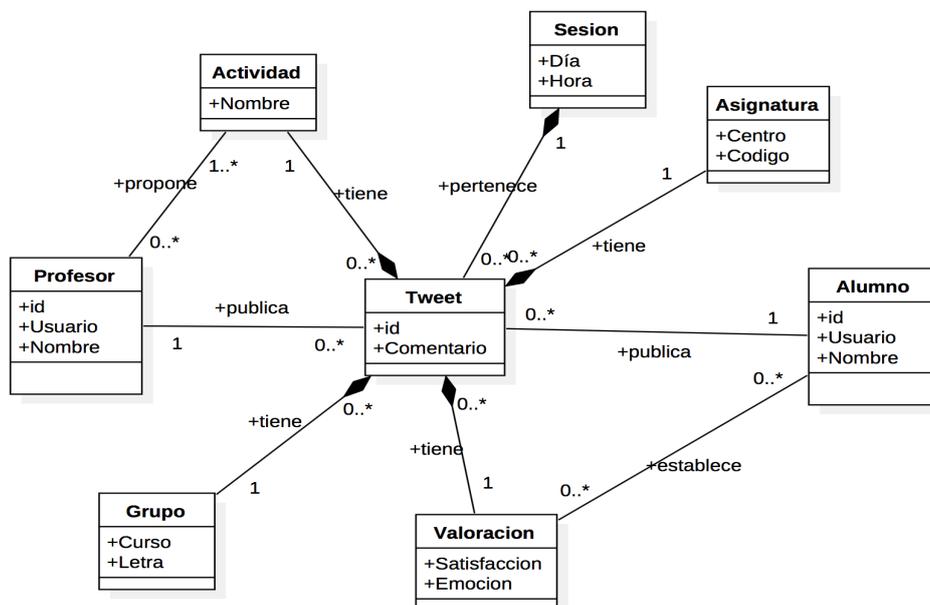


Ilustración 22: Diagrama de clases capa persistencia



En la ilustración 23 hemos procedido a hacer un esquema conceptual inicial. Posteriormente se detallarán los cambios realizados sobre el modelo de diseño no-relacional del que nos hemos servido utilizando *Firebase*.

Ilustración 23: Diseño modelo entidad relación base de datos



La información que recoge la ilustración 23 del esquema entidad relación se explica a continuación:

1. **Tabla Alumno:** Tabla de todos los usuarios que pertenecen a la modalidad alumno.
  - Id: Código identificador del alumno.
  - Usuario: Usuario que utiliza el alumno para identificarse.
  - Nombre: Nombre del alumno.
2. **Tabla Profesor:** Tabla de todos los usuarios que pertenecen a la modalidad profesor.
  - Id: Código identificador del profesor.
  - Usuario: Usuario que utiliza el profesor para identificarse.
  - Nombre: Nombre del profesor.
3. **Tabla Tweet:** Tabla que contiene toda la información que necesita un *tweet*.
  - Id: Código identificador del *tweet*.
  - Comentario: Texto que tiene asociado un *tweet*.
4. **Tabla Valoración:** Contenido de valoración que puntúa el alumno.
  - Satisfacción: Número entero que evalúa la satisfacción de un alumno
  - Emoción: Número entero que evalúa el estado emocional de un alumno en clase.
5. **Tabla Grupo:** Grupo al que pertenece un *tweet*.
  - Curso: Número entero que representa el curso.
  - Letra: Letra que representa la clase de un curso.
6. **Tabla Asignatura:** Asignatura a la que pertenece un *tweet*.
  - Centro: Lugar donde se imparte la asignatura.
  - Código: Abreviatura que identifica a una asignatura.
7. **Tabla Sesión:** Tabla que representa el comienzo de una sesión docente.
  - Día: Fecha del día que se imparte la sesión.
  - Hora: Hora en la que comienza la sesión.
8. **Tabla Actividades:** Tabla que representa el contenido de la actividad que forma parte un *tweet* y que propone un profesor.
  - Nombre: Nombre de la actividad propuesta por un profesor perteneciente a un *tweet*.

Al tratarse de una base de datos orientada a documentos, podemos observar en la ilustración 24 que el diseño varía. Los datos se almacenan como documentos en un esquema no relacional.



Ilustración 24: Esquema JSON no relacional.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "definitions": {},
  "id": "http://example.com/FirebaseDatabase.json",
  "properties": {
    "Sesion": {
      "id": "/properties/Sesion",
      "properties": {
        "Dia": {
          "id": "/properties/Sesion/properties/Dia",
          "type": "string"
        },
        "Hora": {
          "id": "/properties/Sesion/properties/Hora",
          "type": "string"
        },
        "TweetID": {
          "id": "/properties/Sesion/properties/TweetID",
          "properties": {
            "Actividad": {
              "id": "/properties/Sesion/properties/TweetID/properties/Actividad",
              "type": "string"
            },
            "Asignatura": {
              "id":
"/properties/Sesion/properties/TweetID/properties/Asignatura",
              "type": "string"
            },
            "Comentario": {
              "id":
"/properties/Sesion/properties/TweetID/properties/Comentario",
              "type": "string"
            },
            "Emocion": {
              "id": "/properties/Sesion/properties/TweetID/properties/Emocion",
              "type": "integer"
            },
            "Grupo": {
              "id": "/properties/Sesion/properties/TweetID/properties/Grupo",
              "type": "string"
            },
            "Hora": {
              "id": "/properties/Sesion/properties/TweetID/properties/Hora",
              "type": "string"
            },
            "Satisfaccion": {
              "id":
"/properties/Sesion/properties/TweetID/properties/Satisfaccion",
              "type": "integer"
            },
            "Usuario": {
              "id": "/properties/Sesion/properties/TweetID/properties/Usuario",
              "type": "string"
            }
          }
        },
        "type": "object"
      }
    },
    "type": "object"
  }
}
```

En el ejemplo de la ilustración 25 podemos observar cómo la información se agrupa por sesiones con un identificador día y hora, y que a su vez contiene la lista de *tweets* por cada sesión. En este ejemplo mostrado existe una sesión activa del día 03/05/2017 que empieza a las 10:30 en la que hay un *tweet* publicado por el alumno “Jose TFG” con todas sus características.

*Ilustración 25: Diseño esquema con Firebase*



## 4.5 Conclusiones

En este capítulo se ha elaborado una introducción donde se detalla qué es lo que se abarca en la fase de diseño de nuestro proyecto y una sección donde nos centramos en las ideas iniciales de inspiración. Sucesivamente se ha presentado el diagrama de clases UML (Ilustración 11) en el que recogemos todas las clases de nuestra aplicación Android, las clases que utilizaremos del Android Package, las clases encargadas de establecer conexión y comunicación con el servidor de Twitter, las clases involucradas en la comunicación con la base de datos y, por último, las clases que permiten representar gráficas. Hemos mostrado un esquema conceptual del funcionamiento del sistema (Ilustración 12) y para concluir este capítulo hemos elaborado una fase de especificación formal en la que hemos dividido por capas el diseño del presente proyecto. En esta fase intervienen: la capa de presentación en la que se ha mostrado un diagrama de clases UML (Ilustración 12) más detallado en el que se muestran las clases involucradas con esta capa, así como los diseños MockUps de las pantallas (Ilustraciones 13-17). La capa lógica o de negocio en la que se ha mostrado un

diagrama (Ilustración 18) con las clases implicadas con el servidor Twitter y hemos establecido para cada caso de uso un diagrama de secuencia (Ilustraciones 19-21). Por último, la capa persistencia en la que hemos mostrado un diagrama (Ilustración 22) con las clases implicadas con la interacción con la base de datos, un esquema conceptual entidad-relación (Ilustración 23), un esquema final de tipo no relacional (Ilustración 24) y un ejemplo de inserción de datos en Firebase (Ilustración 25).



# 5. Implementación

---

## 5.1 Introducción

Para desarrollar una aplicación móvil, hemos visto que existe una gran variedad de herramientas, tecnologías, lenguajes de programación o bibliotecas que den soporte a nuestras necesidades, por lo que es muy importante hacer una buena elección de la herramienta que vamos a utilizar.

En este apartado abarcaremos todo lo relacionado a la codificación e implementación de la aplicación que hemos desarrollado. También se menciona la tecnología empleada mencionada en la sección 2.5, así como el entorno de desarrollo elegido para codificar nuestra aplicación.

Este capítulo se estructura primero con un apartado de configuraciones generales previas a la codificación e implementación de la aplicación. Después procederemos a visualizar la codificación de los diferentes casos de uso por separado expuestos en el Capítulo 3 (Ilustración 8). Para una mejor asimilación del resultado se incluirán imágenes reales de nuestra aplicación que se podrán comparar con las expuestas en la sección 4.4.1, así como unos fragmentos de código relevante en nuestro sistema. Por último, se concluye con una fase de *testing* en la que se comprueba que el sistema proporciona garantías de funcionamiento.

## 5.2 Conceptos generales previos

### 5.2.2 Creación de una App con Twitter Developers

Antes de empezar a codificar la aplicación tendremos que tener una cuenta de Twitter, la necesitamos para crear una nueva aplicación en *Twitter Developers*, en la siguiente URL: <http://dev.twitter.com> . Ahí rellenaremos una serie de formularios. Una vez realizado todo este proceso tendremos que anotar nuestro código de *Consumer Key API* y *Consumer Secret* que nos harán falta para poder conectar con el API de Twitter. También es necesario establecer unos permisos para permitir a nuestra aplicación móvil poder iniciar sesión sin ningún tipo de restricción, para ello, en el apartado de *Settings* tendremos que activar la opción “*Enable to Callback Login*”



En los siguientes apartados de implementación se especificará qué hacer con estas claves para poder enlazar la aplicación creada anteriormente con el API de Twitter y así implementar la primera fase de codificación referente al inicio de sesión con Twitter a través de nuestra herramienta.

### 5.2.3 Conexión con FireBase de Google.

Antes de empezar a utilizar la base de datos de Google tendremos que crear una cuenta con *Firebase*, previamente necesitamos tener una cuenta de *Google*. Para ello accederemos a la siguiente página: <https://firebase.google.com> y elegiremos “crear un nuevo proyecto”. Después de haber rellenado todos los pasos de creación se tendrá que elegir la opción “añadir FireBase a tu aplicación de Android” y posteriormente introducir el archivo JSON en el directorio de tu proyecto.

Todos estos pasos explicados de forma resumida se detallan con mayor precisión en la página web mencionada anteriormente en la pestaña de documentación.

### 5.2.4 Bibliotecas utilizadas en la aplicación móvil

Para poder realizar el presente proyecto, antes de empezar a codificar necesitamos una serie de bibliotecas que nos darán soporte para poder utilizar las herramientas o las tecnologías que vamos a emplear para su desarrollo.

La tecnología que vamos a utilizar como entorno de desarrollo es la herramienta facilitada por Google llamada Android Studio. En ella generaremos un nuevo proyecto y posteriormente cargaremos las bibliotecas necesarias en el *Gradle* de nuestra aplicación. Para ello tendremos que añadir líneas de “Código dependencia” que aparecen en las tablas 18-22 en el apartado de dependencias que automáticamente importarán las bibliotecas a nuestro proyecto, estas se especificarán en tres campos que son los siguientes:

**Nombre:** Nombre de la biblioteca que vamos a utilizar.

**Código dependencia:** Código que hay que añadir en el Gradle del proyecto para descargarla.

**Descripción:** Breve resumen de lo que hace esta biblioteca y para que la vamos a utilizar.



Tabla 18: Biblioteca Fabric.

Nombre	Fabric
<b>Código dependencia</b>	classpath <code>'io.fabric.tools:gradle:1.+'</code>
<b>Descripción</b>	API oficial de Twitter que nos servirá en el proyecto para conectarnos a una sesión Twitter y para poder publicar <i>tweets</i> .

Tabla 19: Biblioteca Picasso.

Nombre	Picasso
<b>Código dependencia</b>	compile <code>'com.squareup.picasso:picasso:2.5.2'</code>
<b>Descripción</b>	Biblioteca que permite la visualización de imágenes por medio de una URL.

Tabla 20: Biblioteca FireBase.

Nombre	FireBase
<b>Código dependencia</b>	compile <code>'com.google.firebase:firebase-core:9.0.0'</code> compile <code>'com.google.firebase:firebase-database:9.0.0'</code>
<b>Descripción</b>	Biblioteca que permite descargar el soporte de la base de datos para realizar consultas o inserciones.

Tabla 21: Biblioteca GraphView.

Nombre	GraphView
<b>Código dependencia</b>	compile <code>'com.jjoe64:graphview:4.2.1'</code>
<b>Descripción</b>	Biblioteca que permite dibujar gráficas. En nuestro proyecto la utilizaremos para crear la gráfica lineal de evolución del alumno.

Tabla 22: Biblioteca RadarChart.

Nombre	RadarChart
<b>Código dependencia</b>	compile <code>'com.ultramegasoft.radarchart:radarchart:0.1.2'</code>
<b>Descripción</b>	Biblioteca que permite dibujar gráficas tipo Radar. En nuestro proyecto la usaremos para crear un grafo radar para separarlo por actividades.



## 5.3 Implementación

Para mostrar el proceso de codificación de la aplicación móvil, hemos optado por separar los casos de uso expuestos en el capítulo 3 (Ilustración 8), por lo que habrá una explicación de codificación para el inicio de sesión con Twitter, otra para la selección de grupo y asignatura, otra para la publicación de un *tweet* y por último para la consulta de estadísticas.

### 5.3.1 Implementación inicio de sesión con Twitter.

Previamente, para poder implementar este caso de uso necesitamos tener correctamente instalada la biblioteca Fabric (Tabla 18). En lo que respecta a la codificación de este caso de uso, necesitaremos hacer una llamada para el intento de inicio de sesión. Si no dispone de conexión a Internet, el programa lanzará una excepción. El código 1 muestra cómo hacer la llamada de inicio de sesión:

*Código 1: Iniciar llamada para el inicio de sesión con Twitter*

```
loginButton = (TwitterLoginButton)
findViewById(R.id.twitter_login_button);
description = (TextView) findViewById(R.id.text);

loginButton.setOnClickListener(new Callback<TwitterSession>() {
```

Posteriormente, Twitter cargará automáticamente la pantalla de inicio de sesión y nos pedirá que nos identifiquemos con nuestro usuario o correo y contraseña. Esta imagen se podrá ver en la ilustración 26 de la implementación del inicio de sesión con Twitter, donde se facilitará una imagen del diseño real de la aplicación.

Si el usuario es válido tendremos que tratar los datos con un método para el evento de la llamada anterior (Código 2):

*Código 2: Usuarios validados*

```
@Override
public void success(Result<TwitterSession> result) {
    TwitterSession session = result.data;
    String msg = "Sesion @" + session.getUserName() + " iniciada con
    exito";
    Toast.makeText(getApplicationContext(), msg,
    Toast.LENGTH_LONG).show();
    Intent nextStep = new Intent(MainActivity.this, selGrupo.class);
    startActivity(nextStep);
}
```



Como se puede ver en el código 2, guardamos la sesión, mostramos un mensaje de éxito y pasamos a la siguiente actividad.

En caso de que el usuario no sea válido tendremos que implementar un código de fallo que nos permita tratar el error (Código 3).

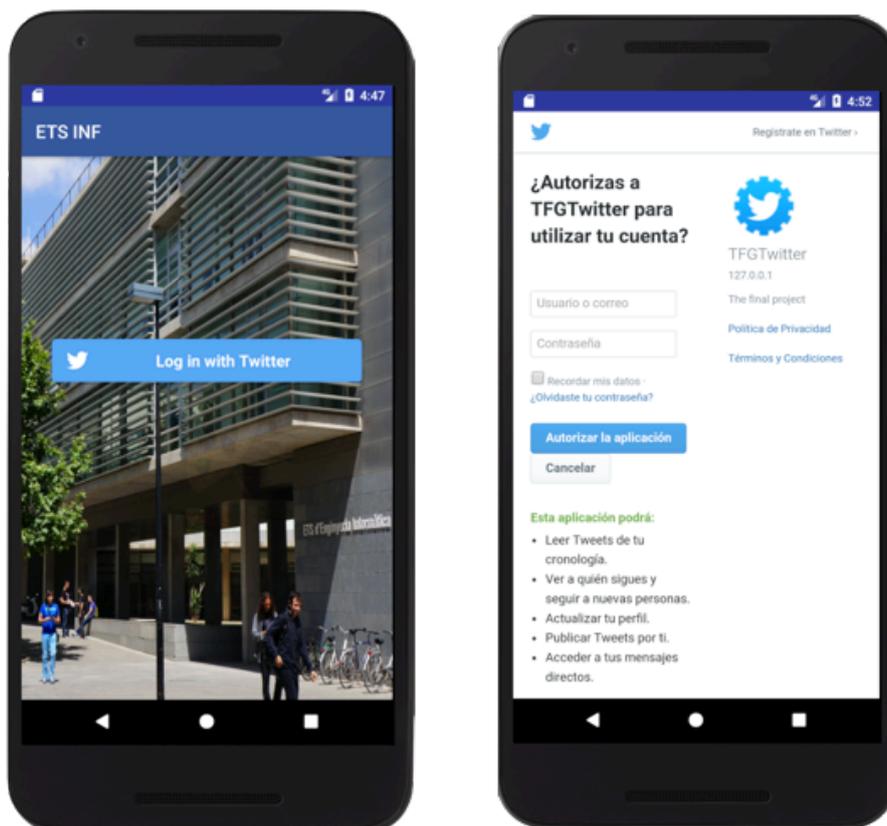
Únicamente se mostrará un mensaje en el que indica el fallo y la excepción.

*Código 3: Usuarios no validados*

```
@Override
public void failure(TwitterException exception) {
    Log.d("TwitterKit", "Login with Twitter failure", exception);
}
```

De esta forma tendremos implementado nuestro inicio de sesión con Twitter (Ilustración 26):

*Ilustración 26: Pantallas de inicio de sesión codificadas.*



### 5.3.2 Implementación selección de grupo y clase.

En primer lugar, para la selección de grupo y clase tendremos dos listas de *arrays* internas que nos permitirán a través de un *Spinner* mostrar una lista de selección. Una vez seleccionado, la clase y el grupo, tendremos que pasar los datos a la posterior actividad para que el siguiente caso de uso pueda tratar con estos dos datos. Cabe destacar que este diagrama de secuencia (Ilustración 20) es el más fácil de implementar, puesto que únicamente se hace una selección de datos por el alumno y se pasa a la siguiente pantalla. Mediante el código 4 conseguiremos introducir estos valores en el *Spinner* para ser seleccionado:

*Código 4: Inserción en el Spinner.*

```
ArrayAdapter<CharSequence> adapter2 = ArrayAdapter.createFromResource  
    (this, R.array.options2, android.R.layout.simple_spinner_item);  
opcionesGrupo.setAdapter(adapter);  
opcionesAsignatura.setAdapter(adapter2);
```

Una vez seleccionados todos los datos tendremos que guardarlos en un objeto de tipo *Intent* para pasarlos a la siguiente actividad (Código 5):

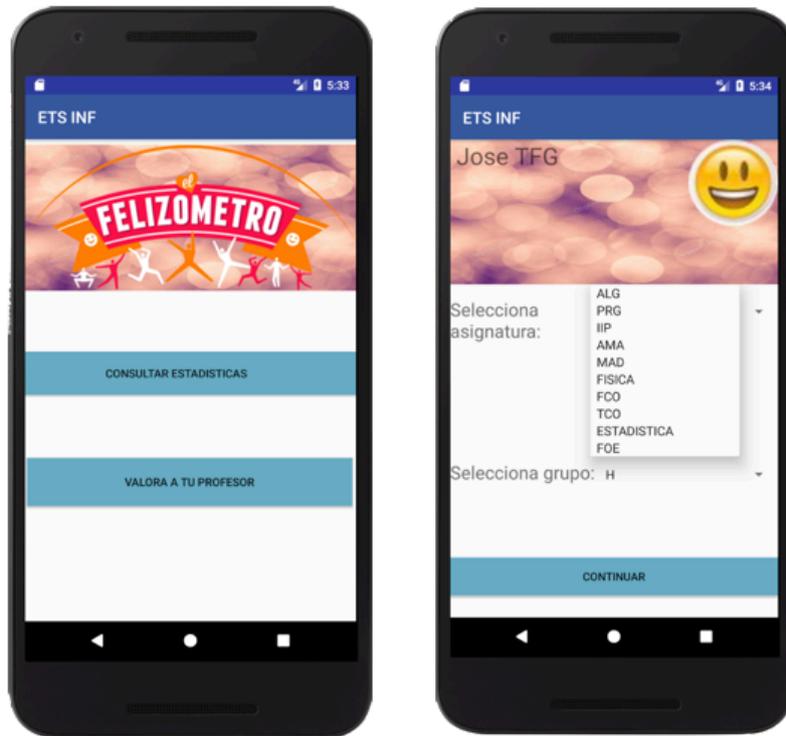
*Código 5: Pasar datos entre diferentes Activitys.*

```
String grupo = opcionesGrupo.getSelectedItem().toString();  
String asignatura = opcionesAsignatura.getSelectedItem().toString();  
  
siguiente.putExtra("grupo", grupo);  
siguiente.putExtra("asignatura", asignatura);  
startActivity(siguiente);
```

Con todo lo mencionado anteriormente, y basándonos en los diseños de *mockUps* de la sección 4.4.1 (Ilustración 15), el diseño en la aplicación real quedará como muestra la ilustración 27:



Ilustración 27: Pantallas selección de grupo y clase codificadas.



### 5.3.3 Implementación publicación de un tweet.

Procederemos a crear el cuestionario referente a la satisfacción y al estado emocional. Para la implementación de estos dos componentes utilizaremos un *RatingBar* para establecer la satisfacción y un *RadioButton* para elegir un icono del estado anímico del alumno. Para ello usaremos el código 6, que recogerá los datos de ambos siempre y cuando los dos hayan sido seleccionados y no nos hayamos dejado ningún campo en blanco; en caso contrario mostrará una alerta de aviso de falta de información:

*Código 6: Obtener la satisfacción y el estado emocional*

```
if(grupo.getCheckedRadioButtonId()<0 || satisfacion.getRating()<1){  
    Toast.makeText(getApplicationContext(),  
        "Error al marcar una de las opciones",  
        Toast.LENGTH_LONG).show();  
}  
else{  
    int seleccionado = grupo.getCheckedRadioButtonId()%10;  
    int numStars = (int) satisfacion.getRating();  
}
```

Posteriormente tendremos la opción de escribir un comentario opcional que acompañe al *tweet*, puesto que al publicarse en una red social como Twitter con la cuenta personal del alumno es visible para muchas personas que no usan nuestra aplicación.

Después de tener todos estos pasos codificados pasamos a la implementación del *tweet*. Haremos la llamada a la clase *TwitterSession* para extraer la sesión activa y a la clase *StatusService* que nos permitirá publicar un *tweet* sin tener que salir de la aplicación móvil, lo cual, la hace eficiente a clases como por ejemplo *TwitterComposer* que para publicar nos llevará a Twitter directamente.

Con el código 7 recuperaremos la sesión activa:

*Código 7: Generación de un objeto de tipo StatusService.*

```
final TwitterSession session1 =  
    Twitter.getSessionManager().getActiveSession();  
final StatusesService statusesService =  
TwitterCore.getInstance().getApiClient(session1).getStatusesService();
```

A continuación, crearemos el cuerpo del mensaje con el grupo y la asignatura recolectados del caso de uso anterior más la selección de satisfacción y estado emocional y el comentario emocional y lo guardaremos (Código 8):

*Código 8: Cuerpo del mensaje del tweet.*

```
//Parte en la que publicamos un Tweet con los datos obtenidos.  
String mensaje= "#"+comprimido.getString("grupo")+"ETSINF"+  
    "#"+comprimido.getString("asignatura")+"ETSINF "+  
    "satisfacion:"+numStars+ " animico:"+seleccionado+  
    "+" "+comentario;
```

Y, por último, invocamos al método que nos permite enviar el mensaje sin tener que salir de la aplicación. Si todo sale bien, el mensaje se publicará automáticamente y ejecutará el código del *Callback*, mientras que si hay algún error indicará una alerta y el código de error que se ha producido uso (Ilustración 20). De esta forma será más fácil poder identificar cuál ha sido la excepción que ha saltado. (Código 9):

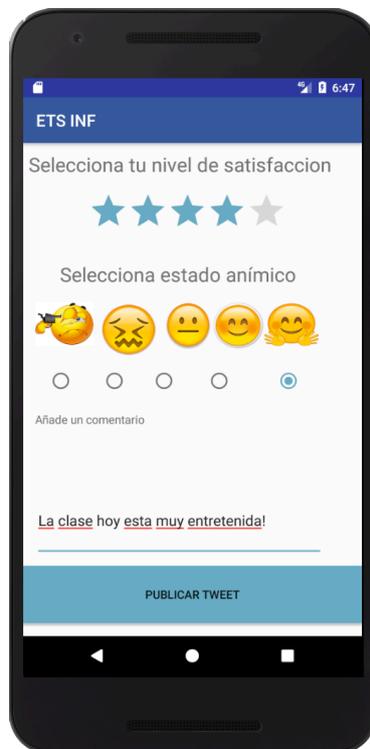


Código 9: Publicar un tweet.

```
statusesService.update(mensaje, null, false,
    null, null, null, true, false, null)
    .enqueue(new Callback<Tweet>() {
        @Override
        public void success(Result<Tweet> tweetResult) {
            Toast.makeText(getApplicationContext(),
                "Publicado con éxito"
                , Toast.LENGTH_LONG).show();
        }
        public void failure(TwitterException p) {
            TwitterApiException e = (TwitterApiException) p;
            System.out.println("Error code: "
                +e.getErrorCode());
            System.out.println("Error mensaje: "
                +e.getErrorMessage());
        }
    });
```

De esta forma obtendremos la pantalla de la ilustración 28, de la cual nos basamos en el diseño del *mockUp* del capítulo 4 de diseño de la capa de presentación (Ilustración 15):

*Ilustración 28: Pantalla de publicación de un tweet codificada*



### 5.3.4 Implementación consultar y mostrar estadísticas.

Para implementar este diagrama de secuencia es necesario previamente estar conectado a la base de datos de Firebase y tener conexión a Internet para poder acceder a ella.

Siguiendo los pasos explicados en la sección 5.2.3, conectaremos la base de datos, pero ahora mediante codificación crearemos una referencia que será el grupo seleccionado en la pantalla ilustración 27. (Código 10):

*Código 10: Crear referencia en la base de datos*

```
String group = "G";
comprimido = getIntent().getExtras();
if (comprimido != null) {
    group = comprimido.get("grupo").toString();
}
final FirebaseDatabase dataBase = FirebaseDatabase.getInstance();
DatabaseReference grupoRef = dataBase.getReference(group);
```

Posteriormente se implementará la consulta a la base de datos para obtener las sesiones disponibles que hay para un grupo determinado (Código 11).

*Código 11: Consulta para obtener las sesiones*

```
grupoRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        int i=0;
        for (DataSnapshot tweetSnapshot: dataSnapshot.getChildren())
        {
            String t = tweetSnapshot.getKey();
            sesiones.add(tweetSnapshot.getKey());
            i++;
        }
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
        Log.e("ERROR", databaseError.getMessage());
    }
});
```

A continuación, aparecerá una pantalla con las sesiones disponibles y tres opciones de estadísticas para visualizar. Si pulsamos la primera, obtendremos una gráfica lineal de evolución, para ello tendremos que descargar todos los



*tweets* de una sesión con la siguiente consulta e ir acumulando los puntos para después mostrarlos en una gráfica.

En el código 12 hay un extracto de código donde vemos cómo se van almacenando los puntos mientras vamos recorriendo todos los *tweets* descargados en la consulta:

*Código 12: Generación de la gráfica evolución.*

```
DataPoint[] YSatis = new DataPoint[listaTweets.size()];
DataPoint[] YEmocis = new DataPoint[listaTweets.size()];
String [] tempo = new String[listaTweets.size()*5];

for (Tweet tweet: listaTweets) {

    Log.d("RESULTADO", tweet.toString());
    Log.d("RESULTADOPUNTOS", "(x:"+tweet.getAnimo()+
        " ,y:"+tweet.getSatisfacción()+")");
    YSatis[i] = new DataPoint(i,tweet.getSatisfacción());
    YEmocis[i] = new DataPoint(i,tweet.getAnimo());
    tempo[i]="t"+i;
    i++;
}
```

En la segunda gráfica se mostrará un gráfico de radar en el que se dibujará una media de satisfacción y estado emocional según las actividades que existen. Para esto haremos una consulta y por cada *tweet* iremos viendo de qué tipo de actividad es, para ir sumando sus características. De esta forma, después podremos calcular la media y dibujarla en el gráfico de radar. En el código 13 se muestra un ejemplo de la consulta de cómo separamos según las actividades:

*Código 13: Consulta gráfico radar.*

```
for (Tweet tweet: listaTweets) {
    if(tweet.getActividad().equals("Ejercicios")){
        mSatisEjercicios = mSatisEjercicios +
            tweet.getSatisfacción();
        mEmoEjercicios = mEmoEjercicios +
            tweet.getAnimo();
        numEj++;
    }
}
```

En el fragmento del código 14 únicamente enseñamos cómo extraemos los datos de una actividad, pero hay que pensar que con el resto de actividades sería lo mismo.



Código 14: Obtener la sesión activa.

```
user.enqueue(new Callback<User>(){  
  
    @Override  
    public void success(Result<User> userResult){  
        User user = userResult.data;  
  
        userName=user.name;  
    }  
});
```

Para finalizar, en la última pantalla de nuestra aplicación vamos a mostrar una tabla que nos indica la media general o la media personal de cada usuario. Esta pantalla se implementa de forma similar a la anterior, solo que esta vez no filtramos por actividades, si no que tendremos que filtrar por usuarios. Para ello primero tenemos ayudarnos del API de Twitter para sacar el nombre de usuario de la sesión activa:

Una vez obtenida la sesión activa, en la consulta hay que compararla con el usuario propietario de ese *tweet* y deben coincidir. De esta forma, descargaremos de nuestros propios *tweets* para poder sacar nuestra media personal. La media general la obtendremos extrayendo los datos de todos los *tweets*. En el código 15 mostramos un extracto de la consulta donde comprueba los usuarios de cada *tweet*:

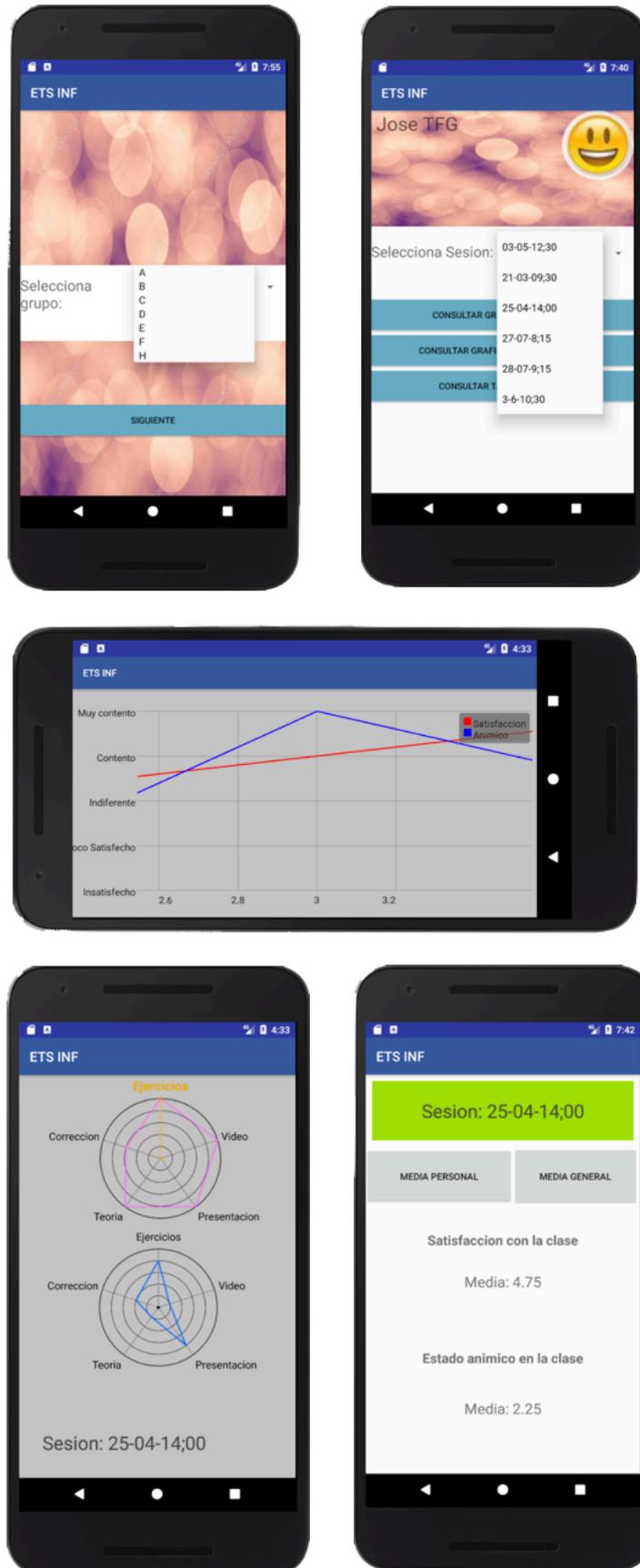
Código 15: Consulta de medias generales y personales.

```
for (Tweet tweet : listaTweets) {  
    if (tweet.getAlumno().equals(userName)) {  
        UserSatisfaccion = UserSatisfaccion +  
            tweet.getSatisfacción();  
        UserEmocion = UserEmocion +  
            tweet.getAnimo();  
        numUser++;  
    }  
    claseSatisfaccion = claseSatisfaccion  
        + tweet.getSatisfacción();  
    claseEmocion = claseEmocion +  
        tweet.getAnimo();  
    numClase++;  
}
```

Así podemos dar por finalizada esta funcionalidad (Ilustración 21) en la que la mayor complicación es trabajar con la base de datos, y procederemos a visualizar la interfaz gráfica real. Como en las anteriores pantallas, este diseño de la interfaz también se corresponde con los *mockUps* (Ilustración 17) presentados en la sección 4.4.1.



Ilustración 29: Pantallas de consulta de estadísticas codificadas.



## 5.5 Implantación

En este apartado explicaremos lo necesario para la puesta en marcha de nuestra aplicación móvil para los alumnos en clase.

Para poder utilizar la aplicación móvil, los alumnos necesitarán un teléfono con sistema operativo Android e instalar la aplicación móvil. Para ello necesitará obtener el *APK* de Android. (*Application Package File*). Posteriormente tendremos que activar en el menú de configuración del teléfono móvil la opción: “Instalar aplicaciones de origen desconocido” y seguir los pasos de instalación.

Una vez efectuadas todas estas operaciones, la aplicación quedará instalada en el dispositivo móvil y se podrá empezar a utilizar de inmediato.

## 5.6 Testing

### 5.6.1 Pruebas unitarias

En esta parte del test se ha decidido automatizar el testeo de un caso de uso de los mencionados en el capítulo 3 (Ilustración 8). Para ello hemos decidido testear el *Cuo3* que consiste en la publicación de un *tweet*.

Como podemos comprobar en el código 16, hemos testeado el método *publicarTweet* de la clase *Felizómetro*. Para ello generaremos el cuerpo de un mensaje y se lo pasaremos como parámetro a dicho método. Si todo el proceso funciona bien, este método devolverá el valor verdadero. Por último, pasamos a la parte final de nuestro test, en el que comprobamos que este método siempre dará como valor verdadero, de esta forma nos garantizamos que nuestra aplicación móvil siempre publicará el *tweet*.



Código 16: Test unitario publicación de un tweet.

```
public class FelizometroTest {
    @Test
    public void testTweet(){
        Felizometro objeto = new Felizometro();
        String grupo="E";
        String asignatura="AMA";
        int satisfaccion = 3;
        int emocion=5;
        String comentario="Esto es una prueba automatica";

        boolean resultado = objeto.publicarTweet
            (grupo,asignatura,satisfaccion
            ,emocion,comentario);

        //Comprobamos de forma automatica que el
        // tweet se publica de forma correcta
        assertTrue(resultado);
    }
}
```

En la ilustración 30 podemos comprobar la ejecución de este test en el que el resultado es de éxito.

Ilustración 30: Resultado test unitario publicar tweet.



### 5.6.2 Pruebas externas

Para verificar el correcto funcionamiento del sistema se ha realizado una batería de pruebas de forma manual y testeadas por el desarrollador de la aplicación. Las pruebas han validado los casos de uso mencionados en la sección 3.2.2 (Ilustración 8). Toda la información se puede recoger almacenada en las tablas 23-25:

Tabla 23: Batería de pruebas correspondiente al Inicio de sesión

Prueba	Caso de uso cubierto	Resultado
Testo1	Cu01	El usuario inicia sesión de manera correcta en la aplicación.
Testo2	Cu01	La sesión siempre se mantiene activa a pesar de salir de la aplicación y volver a entrar.
Testo3	Cu01	La aplicación siempre muestra un mensaje con el nombre de usuario si este usuario es válido.
Testo4	Cu01	La aplicación siempre muestra un mensaje de error y permitirá volver a insertar un usuario y contraseña después de que el usuario no haya sido validado por Twitter.



Tabla 24: Batería de pruebas correspondiente a la selección de grupo y publicación de un tweet.

Prueba	Caso de uso cubierto	Resultado
Testo5	Cu02	El usuario tiene la opción de escoger entre todos los grupos y asignaturas de primer curso de forma correcta.
Testo6	Cu03	El alumno no tiene la posibilidad de publicar un <i>tweet</i> si no ha seleccionado un estado anímico y su satisfacción con la clase, la aplicación avisa de forma correcta de los campos no rellenados.
Testo7	Cu03	Los <i>tweets</i> se envían correctamente a Twitter con todos sus parámetros correspondientes.
Testo8	Cu03	La aplicación genera las etiquetas de forma correcta.

Tabla 25: Batería de pruebas correspondiente a la consulta de estadísticas.

Prueba	Caso de uso cubierto	Resultado
Testo9	Cu04	La base de datos permanece siempre conectada a la aplicación de forma satisfactoria.
Testo10	Cu04	La consulta a la base de datos para extraer la información acerca de los <i>tweets</i> se hace de forma correcta.
Testo11	Cu04	Las gráficas se generan de forma correcta y representan los datos de forma precisa almacenados en la base de datos.
Testo12	Cu04	La aplicación muestra de forma correcta las sesiones únicamente disponibles de un grupo.

## 5.7 Conclusiones

En este capítulo se detalla la fase de implementación del presente proyecto. Se ha elaborado una sección de configuraciones previas a tener en cuenta antes de la fase de implementación, en ella, se ha especificado cómo configurar una app de Twitter desde la página de desarrolladores de Twitter, cómo configurar la base de datos con Firebase para poder conectarla a nuestro proyecto y se ha detallado cómo descargar e instalar las bibliotecas que vamos a emplear (Tablas 18-22). Posteriormente hemos detallado cómo codificar la aplicación por los casos de uso. Para cada caso de uso se han mostrado fragmentos de código relevante (Códigos 1-15) y la pantalla final después de su codificación (Ilustraciones 26-29). Además, se ha realizado una sección de testeado en la que hemos realizado un test automático en el método publicarTweet del presente proyecto (Código 16) y una fase de batería de pruebas manuales en las que se comprueba el correcto funcionamiento de la aplicación (Tablas 23-25). Para



concluir, se ha especificado una fase de implantación en la que se detalla cómo poner en funcionamiento la aplicación una vez codificada.

Con todo realizado se procederá al capítulo final de las conclusiones generales del presente proyecto.

## 6. Conclusiones

---

En este proyecto se ha desarrollado una aplicación móvil para medir el estado emocional de los alumnos en una sesión docente a través de Twitter. Para llevar a cabo el trabajo se ha elaborado un estudio previo en el que se han analizado las herramientas Twitter Analytics, Sprout Social y Hashtracking para compararlas con nuestras necesidades y obtener las características principales que tiene la aplicación.

En la especificación de requisitos se han establecido las funcionalidades que requiere la aplicación:

- Tener una cuenta de Twitter para poder iniciar sesión.
- Identificar el curso y el grupo al que pertenecen la sesión docente.
- Por cada publicación los alumnos reflejan el estado emocional y la satisfacción en distintos momentos de la sesión docente.
- La aplicación permite publicar más de un estado en una misma sesión docente.
- La aplicación recupera los datos de una base de datos y los visualiza mediante gráficas.

Estas funcionalidades se vieron reflejadas en los casos de uso (Ilustración 8). En el apartado de diseño hemos detallado un esquema conceptual general del funcionamiento del sistema (Ilustración 12). Se ha elaborado un diagrama de clases UML en el que intervienen todas las clases de nuestro proyecto (Ilustración 11). En él se han incluido las clases de la propia aplicación y además se muestran las clases que utilizamos para:

- la utilización de gráficas con GraphView,
- la conexión con la base de datos con Google Firebase,
- la conexión con el servidor de Twitter con Fabric,
- la biblioteca del Android Package.

Se ha detallado el diseño específico de la capa de presentación mediante la realización de *mockUps* (ilustraciones 13-17), la funcionalidad de la capa lógica o de negocio mediante diagramas de secuencia (ilustraciones 19-21) y, el de la capa de persistencia mediante un esquema conceptual y otro detallado de la base de datos (ilustraciones 23-24). Se realizó un diagrama de clases UML detallado para cada capa (ilustraciones 12, 18 y 22) con las clases que intervienen en ellas.

Por último, hemos abordado la implementación del sistema. En ella se han especificado las configuraciones previas para:

- Conectar Firebase a la aplicación Android.
- Crear una aplicación Twitter por medio de la página oficial.
- Descargar e instalar las bibliotecas (tablas 18-22).

Después ha implementado el del presente proyecto en el que mostramos fragmentos de código relevante para codificar:

- El inicio de sesión con Twitter (códigos 1-3) en el que hemos explicado cómo hacer llamadas al servicio de Twitter y obtener la sesión activa.
- La implementación de selección de grupo y clase mediante *Spinners* y el paso de datos entre *Activitys*.
- La publicación de un *tweet* (códigos 6-9) en el que obtenemos la satisfacción y el estado emocional del alumno, se genera un mensaje con todos los datos recolectados y se llama al método `update` de la clase `StatusService` para la publicación de este mensaje.
- Consultar y mostrar estadísticas (códigos 10-15) en el que hemos visto cómo hacer consultas con el método `onDataChange` de la clase `FirestoreReferences` y después según el tipo de gráfica seleccionada filtrar por actividades o por nombre de usuario. Para obtener la lista de los *tweets* del usuario activo se ha realizado una consulta a la clase `StatusService` de Twitter para obtener el nombre de usuario de la sesión activa y compararlo con el usuario propietario del *tweet* publicado (códigos 14 y 15).

Finalmente se ha explicado cómo implantar la aplicación en un teléfono móvil y se elabora una fase de testing en el que se comprueba de forma automática que la aplicación garantiza la publicación de un *tweet* (código16) y una fase de batería de pruebas manuales en las que se comprueba el correcto funcionamiento de la aplicación (tablas 23-25).

## 6.1 Dificultades y soluciones

A continuación, se detallarán las dificultades surgidas a lo largo de todo el proyecto software elaborado. Estas dificultades han aparecido tanto en las fases iniciales de estudio de las diferentes tecnologías, pasando por la fase de especificación de requisitos hasta llegar a la implementación.

A la hora de realizar la especificación de requisitos se tuvo dificultades para seleccionar cuáles serían los casos de uso que mejor se adaptarían a la funcionalidad del sistema y al objetivo planteado inicialmente. Este problema se resolvió elaborando este diagrama con precisión, una vez realizada la fase de diseño.



En la fase de diseño es donde han surgido más dificultades, puesto que el diseño de una aplicación software puede variar constantemente si no es preciso. Los diferentes diagramas expuestos en este capítulo, así como los *mockUps* o los diagramas de clases se elaboraron y se modificaron progresivamente en el tiempo hasta dar con el diseño idóneo que facilitara la implementación de la aplicación. Muchos de los errores en el diseño han sido muy perjudiciales en la fase de implementación. Un mal diseño de este puede conllevar innumerables cambios.

En la fase de implementación han surgido diferentes dificultades en la codificación. Surgieron problemas a la hora de codificar el inicio de sesión con Twitter y la publicación de *tweets* que se resolvieron finalmente con la documentación de la biblioteca Fabric en Internet. También surgieron problemas a la hora de realizar consultas y extraer toda la lista de *tweets* de la base de datos, pero de igual forma que los anteriores casos, se resolvieron con la documentación técnica de Firebase.

## 6.2 Aportaciones

### 6.2.1 Aportaciones tecnológicas

La aportación principal de este proyecto es la aplicación móvil desarrollada para el alumno en clase, la base de datos donde se almacenan los datos para el análisis y este documento en el que se detalla cómo llevar a cabo este proyecto especificando un estudio previo en el capítulo dos de herramientas similares, así como una especificación de requisitos y fase de diseño que podrían utilizarse en proyectos similares a éste.

### 6.2.2 Aportaciones académicas

Por medio de este proyecto se espera como objetivo aportar confianza y evolución en el sector académico haciendo que aumente la calidad de la enseñanza, así como la calidad de aprendizaje por parte de los alumnos. Además, este proyecto facilita aportaciones para terceros sistemas que integren el estudio de estos datos almacenados y someterlos a un estudio más exhaustivo que permita solucionar los problemas en las aulas y llevar a cabo varias conclusiones estadísticas.



## 6.3 Trabajo futuro

Este proyecto se puede ampliar en diferentes ámbitos, lo que mejoraría la calidad de la aplicación. Actualmente, las aplicaciones móviles tienen que ir renovándose y añadiendo funcionalidades para que los usuarios cubran sus necesidades en todo momento. Es por esto que en un momento dado, los trabajos futuros a partir de éste podrán implantar un sistema en el que los diseños gráficos de la aplicación mejoren con respecto a los aquí presentes. También se podría implantar este sistema en diferentes plataformas y diferentes tipos de dispositivos para una mayor participación de los alumnos. No obstante, los datos también pueden sufrir un gran cambio en trabajos futuros, puesto que se pueden verse ampliados considerablemente aumentando el tipo de docencia y distintos lugares de docencia. Por último, esta aplicación podría incluir mejoras en la funcionalidad, por ejemplo se podrían configurar las mediciones en función del gusto del profesor que imparte una sesión docente.

## 6.4 Referencias

*Android Developers.* <https://developer.android.com/index.html>

Ceballos sierra, J (2010). *Java2, curso de programación.* Madrid: RA-MA editorial.

Estándar IEEE 830.

<https://www.fdi.ucm.es/profesor/gmendez/docs/iso809/ieee830.pdf>

*Fabric.* <https://get.fabric.io>

Gómez, C., Mayol, E., Olivé, A. y Teniente, E. (2014). *Diseño de un sistema Software UML.* Valencia: Universitat Politècnica de València.

*GraphView.* <http://www.android-graphview.org/simple-graph/>

*Lenguaje Java.*

<http://www.oracle.com/technetwork/java/javase/documentation>

*Twitter developers.* <https://www.dev.twitter.com/>

Zechner, M (2011). *Desarrollo de juegos Android.* Madrid: Anaya multimedia.

Torró Segura, Carles (2017). *Extracción de información de Twitter para la creación de redes de interacción entre usuarios.* Valencia: Universitat Politècnica de València.

