

Programación de aplicaciones

Tarea 4

Curso 2011

1 Información Administrativa.

La tarea comienza el **lunes 14 de noviembre** y finaliza el **domingo 4 de diciembre**. La tarea constará de múltiples entregas parciales a realizarse durante las clases de monitoreo cuando el docente lo requiera, y la entrega final de código a realizarse hasta el **domingo 4 de diciembre** a las **23:59hs** (entrega vía mail).

2 Objetivos.

La presente tarea pretende la puesta en práctica y profundización por parte del estudiante de los siguientes puntos:

- **Experimentar con una Metodología Iterativa e Incremental.**
- **Arquitectura Distribuida**
- Integración con servicios externos

3 Planificación General.

Basados en los lineamientos definidos en la Descripción del Proyecto [1], en esta cuarta etapa, se continuará con el desarrollo de los componentes hasta ahora desarrollados de forma de completar su desarrollo.

4 Visión.

Para esta cuarta iteración del proceso de desarrollo se ha generado el siguiente anexo de la descripción de la realidad.

Se desea que la aplicación pueda sugerir amigos a los integrantes de la red social. Para ello se cuenta con el siguiente método de sugerencia de amigos: dado un usuario registrado en la red social, se toman todos los amigos de sus amigos. De dicho conjunto de usuarios, se considerarán aquellos que aún no son amigos, ordenando los mismos de acuerdo a la aprobación que hayan tenido en las publicaciones de sus muros (de mejor a peor promedio).

5 Requerimientos Funcionales.

5.1 Portal Web

Para esta iteración del proceso de desarrollo se modificó el siguiente caso de uso implementado en iteraciones anteriores (los cambios se muestran **en negrita**):

Nombre	Ver Perfil de Usuario
Actores	Usuario
Sinopsis	Este caso de uso permite ver los datos del usuario así como su modificación y su muro. Las entradas del muro se deben ordenar por fecha de publicación, colocando las más recientes primero, mostrando la fecha de publicación, el contenido de cada entrada. Además deberán mostrarse aquellas entradas (de muros de sus amigos o grupos a los que pertenece) en las que el usuario está etiquetado. Para cada entrada deberá mostrarse sus comentarios (omitiendo aquellos cuyo puntaje está por encima del umbral mínimo) de forma anidada,

	mostrando para cada comentario su fecha, publicador, puntaje y su contenido. En caso de tratarse de una entrada de tipo imagen, se deberán mostrar los usuarios etiquetados (aplica también para muro de grupo). Finalmente deberá mostrarse la información completa del calendario de eventos del usuario mostrando para cada evento su nombre, tipo, si es repetible o no y cuando ocurre.
--	--

Para esta iteración del proceso de desarrollo se agregaron los siguientes casos de uso para este componente:

Nombre	Solicitar amistad
Actores	Usuario
Sinopsis	El objetivo del caso de uso es permitir al usuario solicitar una amistad con otro miembro de la red. Para ello, el sistema lista los usuarios con los que no tenga una solicitud pendiente ni sean amigos y el usuario elige uno. La amistad va a quedar pendiente hasta que el otro usuario la confirme a través del caso de uso Confirmar/Rechazar solicitud de amistad . En caso de una respuesta afirmativa el sistema generará una entrada en los muros de ambos usuarios de tipo texto indicando el inicio de la amistad entre los dos usuarios. La fecha de publicación de ambas entradas corresponderá al inicio del vínculo de amistad entre ellos.

Nombre	Confirmar/Rechazar solicitud de amistad
Actores	Usuario
Sinopsis	El objetivo del caso de uso es responder a una solicitud de amistad realizada por otro usuario de la red. Para ello, el sistema lista todas las solicitudes pendientes, y el usuario puede confirmar o rechazar aquellas que desee. Si el usuario confirma, pasan a ser amigos en la red, y en cualquier caso la solicitud deja de estar pendiente.

Nombre	Eliminar amigo
Actores	Usuario
Sinopsis	Este caso de uso permite eliminar un amigo del usuario. El caso de uso comienza listando sus amigos, y permite elegir a uno o más que serán eliminados.

Nombre	Agregar Contenido Remoto
Actores	Usuario
Sinopsis	Este caso de uso permite subir contenidos almacenados en forma remota. El caso de uso comienza cuando el usuario ingresa el link a un contenido remoto. El sistema debe inferir el tipo del contenido a partir de la extensión de la URL. El contenido proveniente de algunos orígenes se debe manejar en forma especial. (Ver requerimiento especial 6.1)

Nombre	Sugerir Amigo
Actores	Usuario
Sinopsis	El objetivo de este caso de uso es permitir obtener una sugerencia de posibles amigos basada en las relaciones existentes entre los usuarios de la red. El usuario solicita una sugerencia, y el sistema lista hasta tres usuarios con los que tenga un amigo en común (o sea, los pertenecientes al conjunto de los amigos de sus amigos). El sistema debe mostrar primero aquellos que tengan mayor promedio en los puntajes de sus publicaciones. Además, debe existir la posibilidad de solicitar amistad con cualquiera de ellos.

6 Requerimientos no funcionales.

6.1 Reproducción embebida de contenidos

En el portal web se podrán ingresar URLs de contenido remoto para videos de *Youtube* y *DailyMotion* que serán considerados como videos remotos. Se espera que un usuario del sitio web pueda copiar y pegar la URL original del contenido y pegarlo en un cuadro de texto en la página web de ACFC. Ejemplos de URLs que podrá utilizar el usuario para indicar el contenido remoto son:

- http://www.youtube.com/watch?v=CZeRkzl_bDw&feature=related
- http://www.dailymotion.com/video/x342hv_olivia-wilde-talks-house_shortfilms

Para videos de *Dailymotion* el usuario deberá ingresar la información del video manualmente. En el caso de videos de *Youtube* se deberá inferir la información a partir de la URL del video. Esto se podrá realizar utilizando la API pública de Google [11]. Los videos remotos de estos sitios deberán ser mostrados en forma embebida en la página web del Servidor Central utilizando los reproductores originales provistos por los sitios web [12].

6.2 Obtener información para contenidos de tipo audio

Se desea que cuando el usuario va a agregar un contenido de tipo audio, se le permita complementar la información del mismo mediante un servicio externo. Para ello, se debe incluir la posibilidad de ingresar el título y artista del contenido, y obtener su duración, el álbum al que pertenece y su portada, así como información adicional del mismo. El usuario puede confirmar que la información obtenida es correcta y la misma va a quedar almacenada para mostrarla junto al contenido, en caso contrario el usuario puede completar estos campos en forma manual.

Para resolver este requerimiento se propone utilizar la API de *Last.fm* [2]. Para facilitar su consumo, se recomienda el uso de *last.fm API bindings* para Java [3].

6.3 Encriptación de las contraseñas.

Con el fin de mejorar la seguridad en el sitio web y se deberá asegurar las contraseñas de los usuarios con la finalidad de prevenir el robo de contraseñas. Por este motivo se deberá cifrar las comunicaciones entre los navegadores web y el Servidor Web usando el protocolo HTTPS en aquellas páginas en las cuales se intercambia la contraseña del usuario. Para implementar el protocolo será necesario generar certificados autofirmados. Véase [10] para más información de cómo configurar SSL en Tomcat. En segundo lugar es necesario poder persistir las contraseñas encriptadas en la persistencia de modo de no poder recuperar las contraseñas originales de los usuarios en caso de que un atacante pueda acceder a la base de datos del Servidor Central. Cuando un usuario se autentique su contraseña será validada contra la contraseña encriptada que se guardó previamente. Para implementar el almacenamiento y verificación de contraseñas es posible utilizar librerías como *Jasypt* [5].

6.4 Persistencia del dispositivo móvil usando JPA.

El dispositivo móvil deberá persistir a disco aquellos cambios que no hayan sido enviados al Servidor de Comunicaciones al momento de ser apagado utilizando una base de datos (se recomienda el uso de HSQLDB [6] o SQLite [13]), la cual deberá ser accedida a través de JPA [8]. La base de datos deberá tener toda la información de todos los usuarios que utilizaron el dispositivo móvil incluyendo información como el historial local de mensajes, los mensajes en la bandeja de salida o las imágenes de los usuarios que hayan sido *cachedas* en el dispositivo.

7 Se Pide

- Desarrollo completo en Java de los casos de uso y requerimientos no funcionales definidos anteriormente.

8 Referencias

[1]	Descripción del Proyecto http://www.fing.edu.uy/inco/cursos/tprog/field.php?n=Lab.Letras
[2]	Last.fm API http://www.lastfm.es/api
[3]	Last.fm API bindings for Java http://www.u-mass.de/lastfm
[4]	Cryptography with Java http://www.informit.com/articles/article.aspx?p=170967&seqNum=5
[5]	Jasypt: Java simplified encryption http://www.jasypt.org/
[6]	HSQLDB – HyperSQL DataBase http://hsqldb.org/
[7]	JDBC – Java Data Base Connectivity http://java.sun.com/javase/technologies/database/
[8]	JPA – Java Persistence API http://java.sun.com/developer/technicalArticles/J2EE/jpa/
[9]	Introduction to SSL http://docs.sun.com/source/816-6156-10/contents.htm
[10]	Apache Tomcat 6.0 - SSL Configuration HOW-TO http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html
[11]	Guía para desarrolladores: Protocolo del API de datos – Recuperación de datos de un único vídeo - Herramientas y API de YouTube - Google Code http://code.google.com/intl/es/apis/youtube/2.0/developers_guide_protocol_video_entries.html
[12]	Embedding videos : Getting Started http://www.google.com/support/youtube/bin/answer.py?hl=en&answer=57788
[13]	SQLiteJDBC http://www.zentus.com/sqlitejdbc/