



UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Urb. Montalbán – La Vega- Apartado 29086

Teléfono: +58212 - 4074407 Fax: +58212 - 4074447

Caracas, 1021 –Venezuela

Facultad de Ingeniería

Escuela de Ingeniería Informática



Contexto Inicial

El objetivo principal de este proyecto, es desarrollar una red social que permita gestionar álbumes multimedia a partir de imágenes, sonidos y videos registrados en la redes sociales **Instagram**, **SoundCloud** y **YouTube**. El producto creado podrá ser clasificado como un **Mashup** o aplicación híbrida que se nutrirá de la información y capacidades provistas por terceros así como, por la información gestionada internamente por ella.

Como segundo objetivo el presente proyecto, pretende introducir al estudiante al uso de herramientas orientadas al **Cloud Computing** para ello se espera que las aplicaciones resultantes sean desplegadas en un ambiente de producción real, provisto por alguno de los grandes competidores en el área como los son: **Amazon Web Services** y **Google App Engine**.

Especificaciones Técnicas

A continuación se describen las principales características funcionales y no funcionales de la aplicación que debe ser construida:

Debe permitirse la gestión del perfil del usuario, esto implica operaciones básicas como registro, actualización de datos, consulta de información del usuario. Para el proceso de registro, cualquier persona con una cuenta de usuario válida en **Google** o **Facebook** podrá formar parte de esta nueva red social. Se recomienda obtener mayor información sobre los procesos de autenticación en ambas plataformas en los siguientes enlaces [Federated Login Google Account Users](#) y [Facebook Login](#) Como sucede en todas las redes sociales los nuevos usuarios, deben estar en capacidad de gestionar un “grafo social” que les permita relacionarse con el resto. Se sugiere implementar un buscador similar al utilizado en **Facebook**, el cual a partir del nombre o apellido de alguna persona despliegue cierta información relacionada a su perfil en la red social.

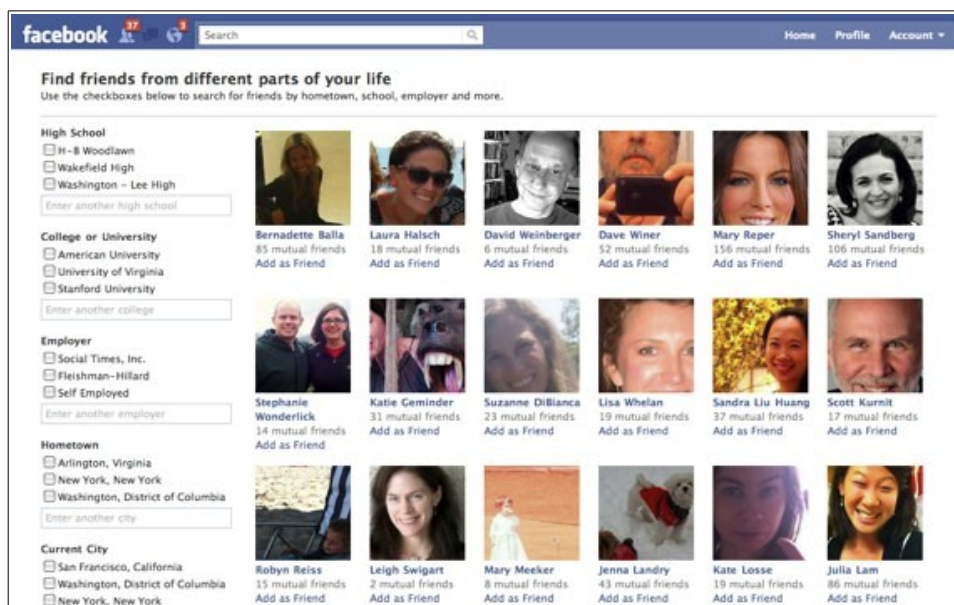


Figura 1.- Sugerencia del buscador de personas dentro de la red social.

Debe existir como parte de la aplicación una sección que permita administrar los datos de su cuenta. Se espera que todos los usuarios provean a nuestro sistema de los siguientes datos: Nombres y Apellidos, Nombre de Usuario, Correo Electrónico, Fecha de Nacimiento, Ubicación Actual, Enlaces a otras redes sociales (Google +, Twitter, Facebook). Es importante ofrecer la posibilidad de definir niveles de privacidad para el contenido publicado por cada usuario, existiendo dos esquemas **público** y **restringido a sus amigos**. Para el primero de ellos cualquier usuario ajeno a esta red social podrá visualizar el contenido de todos los álbumes de los usuarios con tan solo efectuar una búsqueda en la página principal de la aplicación. En el segundo caso, la información de los usuarios sólo podrá ser visualizada por aquellos que se encuentren el “grafo social” del usuario. Desde la pagina de cada usuario la aplicación debe mostrar algún tipo de opción para conocer sus conexiones (amigos) a partir de esta lista de conexiones debe existir un enlace que permita llegar al perfil de los amigos de un usuario, tomando como referencia un esquema propuesto por **Facebook** en su momento, la figura 3 muestra esta propuesta.

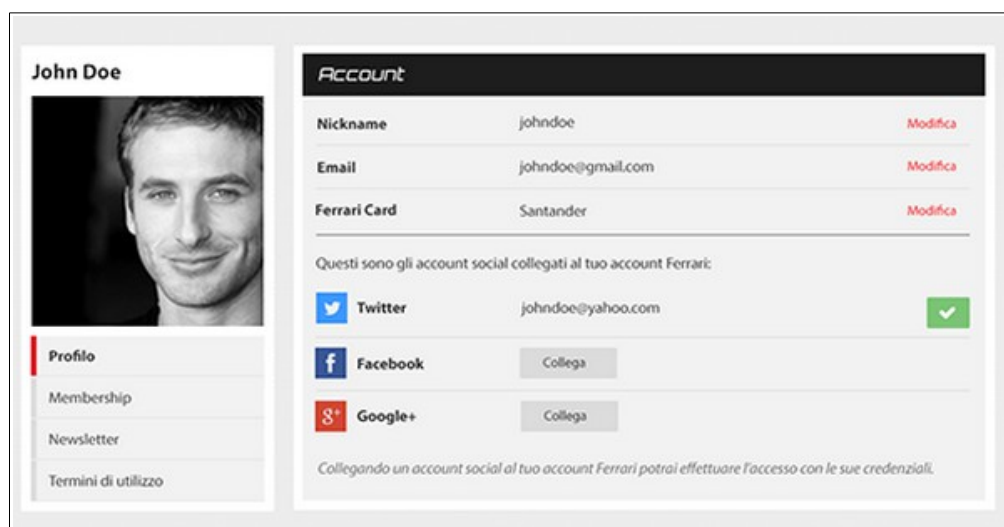


Figura 2.- Boceto de la sección que permitirá la gestión de datos del perfil de usuario.

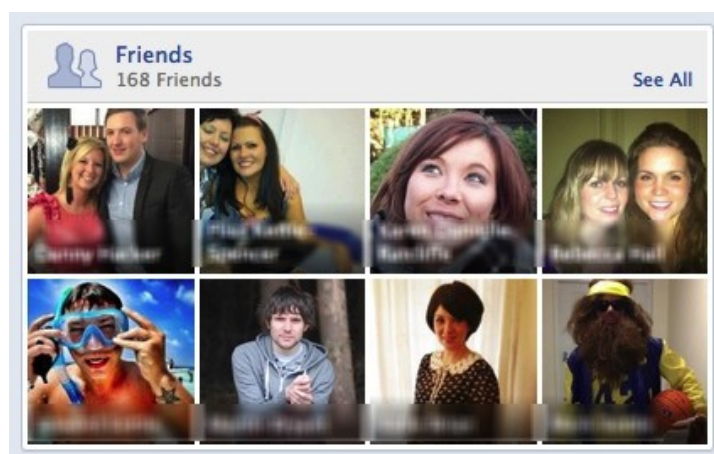


Figura 3.- Propuesta para visualizar los amigos de un usuario.

Una de las funcionalidades esenciales de esta red social es la posibilidad de gestionar una biblioteca multimedia a través de un álbum en el cual, los usuarios asociarán contenido proveniente de múltiples fuentes como **Instagram**, **SoundCloud** y **YouTube**. Para cualquiera de estos casos se debe presentar una barra de búsqueda de contenido, que tendrá como entrada una palabra clave como criterio para interactuar con el API de las plataformas mencionadas. Una vez obtenidos los resultados provenientes de las fuentes, el sistema debe presentar en forma intuitiva el compendio de ítems permitiendo seleccionar aquellos que formaran parte de su álbum. Todo nuevo álbum debe contar con un nombre unívoco y una breve descripción (opcional) que permita tener información relevante de su contenido. De igual manera, se debe proveer de una imagen en miniatura para identificar el nuevo álbum, si el usuario no asocia esta imagen nuestro sistema debe asociar una por defecto.

Por último el dueño de cada álbum, debe indicar si desea hacer público o no la nueva biblioteca de contenido. La información multimedia de esta nueva publicación debe visualizarse en un esquema atractivo para el usuario, para ello la figura 4 muestra un carrusel con cierto contenido, como sugerencia para cumplir con este requerimiento.

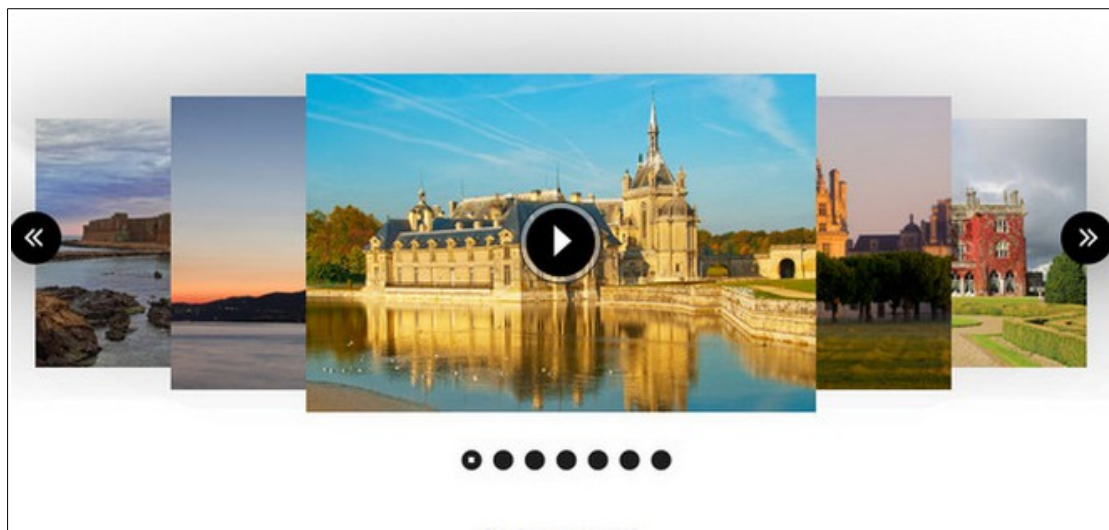


Figura 4.- Carrusel con contenido multimedia sugerido para mostrar los ítems que forman parte del álbum.

Efectuada la publicación del nuevo contenido, el resto de usuarios que forman parte del “grafo social” del usuario, serán notificados por medio de un mensaje que será entregado en una bandeja de notificaciones destinada a tal fin.

El contenido publicado en cada álbum podrá ser alterado, añadiendo o eliminando contenido provenientes de las fuentes de información mencionadas con anterioridad. Cada modificación del álbum implica una notificación a los miembros del grafo social.

Los miembros del grafo social de cada usuario podrán agregar comentarios como parte de un hilo de conversación asociado a dicho álbum. En este hilo de conversación, los comentarios tendrán una o mas respuestas y ofrecerán la posibilidad de aprobar o desaprobar el comentario (*Me Gusta, No Me Gusta*) tal como ilustra la figura 5.

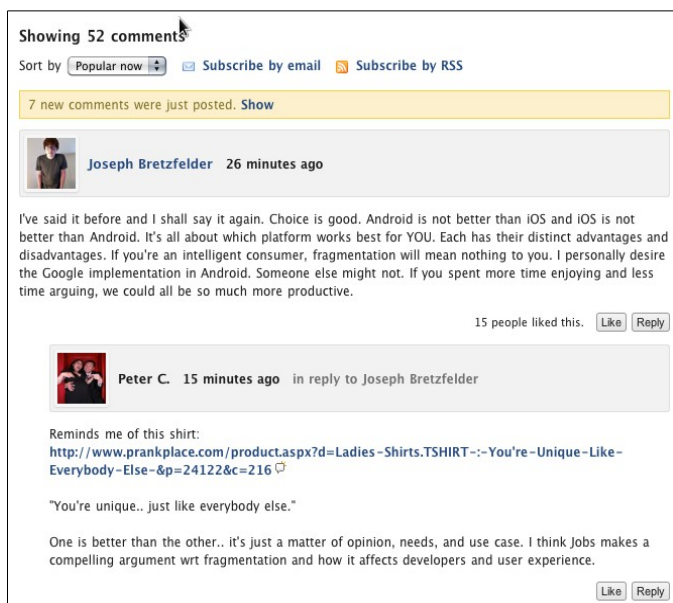


Figura 5.- Escenario propuesto para el despliegue de los comentarios, respuestas, cantidad de puntaje positivos/negativos.

La respuesta a un comentario de un usuario, debe ser notificada a través un mensaje que será enviado a la bandeja del usuario cuyo comentario o álbum recibió dicha respuesta. El hilo de una conversación solamente podrá ser modificado por el dueño del álbum, esto quiere decir que la potestad de eliminar comentarios dependerá única y exclusivamente de él.

Actualmente, hay una tendencia en el desarrollo de aplicaciones web denominada *Flat Design*. Se espera que el producto final cuente con algunos de estos elementos.

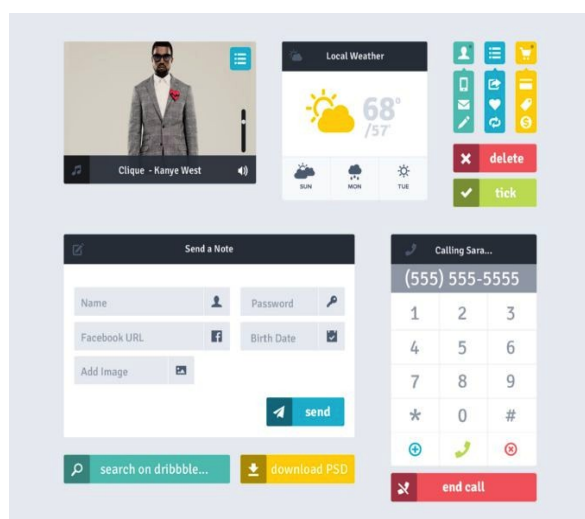
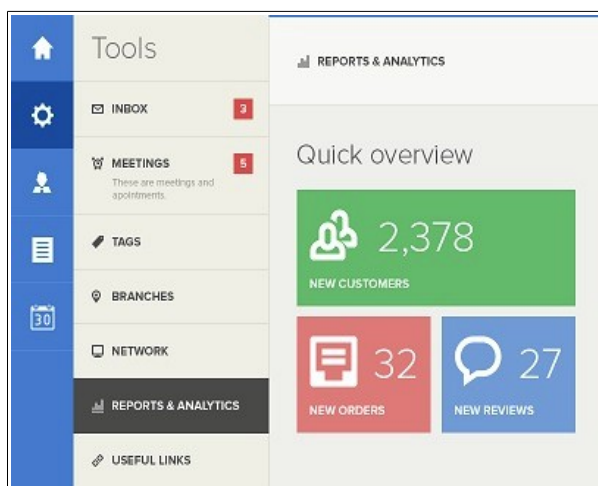


Figura 6.- Elementos usados en el interfaces con diseños *Flat*

Evaluación y Criterios de Corrección

Entrega	Items a ser evaluados
1	<p>a).- Plataforma Social y Perfiles de Usuario.</p> <p>Gestión de Usuarios, <i>Single Sign On</i> (Google o Facebook). Gestión de Perfil.</p> <p>Gestión del “grafo social”.</p> <p>Gestión de álbumes (sin contenido multimedia). Aplicación de reglas de privacidad.</p>
2	<p>a).- Integración con Instagram</p> <p>Gestión del contenido multimedia del álbum, búsqueda y adición de contenido proveniente de Instagram.</p> <p>Interacción social de usuarios del “grafo social” con los álbumes de otros usuarios.</p>
3	<p>a).- Integración con YouTube y SoundCloud.</p> <p>Gestión del contenido multimedia del álbum, búsqueda y adición de contenido proveniente de SoundCloud y YouTube.</p> <p>Notificaciones de eventos en la gestión de álbumes.</p> <p><i>Suite</i> de Pruebas de toda la plataforma. Pruebas Unitarias y Pruebas Funcionales (Selenium).</p>

El desarrollo de la aplicación estará condicionado al ambiente al proveedor de la nube utilizado para desplegar la solución en producción. A continuación se muestran las opciones aceptadas para el desarrollo de la solución:

Cloud Provider	Lenguaje y Frameworks aceptados
Google App Engine	Java <ul style="list-style-type: none"> • <i>Java Server Pages</i> • <i>Java Server Faces 2</i> Python <ul style="list-style-type: none"> • <i>Django</i>
Amazon Web Services	Ruby <ul style="list-style-type: none"> • <i>Rails</i> Play framework <ul style="list-style-type: none"> • <i>Scala o Java</i> PHP <ul style="list-style-type: none"> • <i>Cake</i> Java <ul style="list-style-type: none"> • <i>Spring MVC</i>

1. Como almacén de datos puede usarse cualquier solución: **MySql, SqlServer, Oracle, PostgreSQL, MongoDB, Cassandra**. La herramienta escogida, nuevamente, dependerá del proveedor usado para colocar en producción el proyecto.
2. Es de carácter obligatorio desplegar en el ambiente de producción: **Google App Engine** o **Amazon Web Services**. No será aceptada otra solución. Para ello se requiere que cada equipo realice una investigación exhaustiva y delicada sobre ambas plataformas. En los siguientes enlaces [Amazon Web Services](#) y [Google App Engine](#) . Ambas plataformas tienen acuerdos y límites de uso que deben ser leídos y entendidos previos al despliegue de la solución.
3. La **corrección 1** del proyecto pautada para el Sábado 26 de Octubre del 2013, se llevará a cabo haciendo uso de la herramienta **Google Hangouts**. Los horarios para la corrección de cada equipo se informarán oportunamente.
4. **El uso de herramientas para el manejo de log's de actividad en la aplicación es de uso obligatorio**, está será la constancia de los eventos que se están produciendo en el servidor de ejecución, y serán el único mecanismo al que podrán recurrir los integrantes del equipo para diagnosticar errores o eventos de interés que se generen en la aplicación.
5. **Es de carácter obligatorio hacer uso del servicio de control de versiones GIT**. Para cada entrega debe ser creada la etiqueta (tag) correspondiente, se establece como fecha tope para la generación de este tag, el día previo a la corrección **hora tope 6:00pm**. En este *tag* serán evaluados elementos de carácter teórico y de aplicación de correctas prácticas de diseño, documentación entre otros.
6. No está permitido el uso de otras herramientas o lenguajes diferentes a los mencionados en esta sección.