DIRECTORIO INTERACTIVO DE LA ESPOL

Stalin Alberto Arroyabe[1], Gonzalo Antonio Ordóñez[2], Ing. Fabricio Echeverria[3]
Facultad de Ingenieria en Electricidad y Computación (FIEC)
Escuela Superior Politécnica del Litoral
Campus "Gustavo Galindo Velasco",- Km. 30,5 Vía Perimetral
Apartado 09-01-5863, Guayaquil - Ecuador
sarroyab@fiec.espol.edu.ec[1], gordonez@fiec.espol.edu.ec[2], pechever@espol.edu.ec[3]

Resumen

En este proyecto se ha elaborado una Aplicación web llamada "Directorio Interactivo de la ESPOL", que permite a los estudiantes, publicar información personal en su perfil de usuario.
La información es extraída desde las redes sociales más populares en nuestro medio (Facebook, Hi5, Youtube), y desde la bases de datos universitaria, para luego compartirla con los demás usuarios del sistema, formando de esta manera una red social politécnica.
Para elaborar la funcionalidad detallada anteriormente utilizamos:

- Gdata Library de Google para la comunicación con los servidores de Youtube.
- Servicios Web basados en REST, para la comunicación con los servidores de Hi5.
- Servicios de Web basados en SOAP, para la comunicación con los servidores de la ESPOL
- JAVASCRIPT y FQL para la comunicación con los servidores de Facebook.

Palabras Claves: Web 2.0, Redes Sociales, REST, SOAP, WEB SERVICES, JAVASCRIPT Y FQL

Abstract

This project has developed a web application called "Directory ESPOL Interactive", which allows students to publish personal information in your user profile.
The information is extracted from the most popular social networks in our area (Facebook, Hi5, Youtube), and from the university database, then share with other users of the system, thus forming a polytechnic network.
To develop the functionality listed above use:

- Google Gdata Library for communicating with servers Youtube.
- REST based Web Services for communication with servers Hi5.
- Web Services based on SOAP, to communicate with servers ESPOL
- JAVASCRIPT and FQL for communication with the Facebook servers.

Keywords: Web 2.0, Social Networking, REST, SOAP, Web Services, JavaScript and FQL

1. INTRODUCCION

La ESPOL cuenta con un sin números de tecnologías que ayudan al funcionamiento estratégico y operacional de la institución, lo que ha permitido cumplir con efectividad y eficiencia las responsabilidades adquiridas con la sociedad. Los estudiantes de la Comunidad Politécnica, hacemos uso diario de la tecnología para cumplir con nuestras labores, por eso tenemos un alto grado de usabilidad en el manejo de tecnologías.
Los estudiantes dedicamos un tiempo muy significativo a la interacción con las siguientes redes sociales: Facebook, Hi5 y Youtube. En estas redes sociales se encuentra información personal muy valiosa de acceso público, siempre y cuando se cumpla con las políticas de privacidad declaradas en cada sitio.

La universidad no cuenta con una plataforma tecnológica, que permita el fortalecimiento de la relaciones entre los estudiantes. Es por eso que surge la idea del "Directorio Interactivo de la ESPOL"; él cual permitirá la búsqueda y acceso (no restringido) a la información de los estudiantes. La información es extraída de las redes sociales anteriormente mencionadas.

2. Comunicación con los Servicios web de la ESPOL

Antes de hablar de la comunicación entre los servicios web de la ESPOL y nuestra aplicación debemos conocer el concepto sobre Servicios web y Protocolos de Comunicación.

2.1. Servicios Web

Son un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Las aplicaciones pueden estar desarrolladas en lenguajes de programación diferentes, y ejecutarse sobre cualquier plataforma. Por lo general, el intercambio de datos se lo realiza sobre Internet, pero se lo puede realizar sobre cualquier red.
La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Los dos tipos de comunicación más importantes son REST y SOAP [1].

2.2. Protocolo SOAP

Es un protocolo estándar de intercambio de mensajes en un sólo sentido, sin estado, pero las aplicaciones pueden crear patrones de interacción más complejos (por ejemplo, petición/respuesta, petición/respuestas múltiples, etc.).
SOAP no interfiere en la semántica de los datos que transfiere, ni tampoco en el enrutamiento de los mensajes.
Proporciona un marco de trabajo para que las aplicaciones puedan comunicarse de forma extensible. También proporciona una descripción completa de las acciones que debe realizar un nodo SOAP al recibir un mensaje SOAP [2].
Los mensajes constan de 3 secciones:

- Envelope (envoltura): Es el elemento raíz del mensaje para describir su contenido y la forma de procesarlo.
- Header (encabezado): Es la información de identificación del contenido. Un grupo de reglas de codificación para expresar los tipos de datos definidos por la aplicación.
- Body (cuerpo): Es el contenido del mensaje. Una convención para representar las llamadas y las respuestas a procedimientos remotos.

2.3. Protocolo REST

La Transferencia de Estado Representacional (Representation State Transfer - REST) describe un estilo arquitectónico de sistemas en red. El término fue utilizado por primera vez en el año 2000 durante una disertación doctoral por Roy Fielding, uno de los principales autores de la especificación HTTP.
REST contiene limitaciones y principios arquitectónicos.
La solicitud de interacción entre el cliente y el servidor no maneja estado. Por eso debe contener toda la información necesaria para ser comprendida. El servidor posee recursos que representan el estado y la funcionalidad de la aplicación. Un recurso es un elemento que puede ser accedido por el cliente, algunos ejemplos de recursos son: objetos de aplicaciones, registros de bases de datos, algoritmos, etc. Cada recurso tiene acceso único a través de una URI (Universal Resource Identifier – identificador de recursos universal). Se usan métodos estándares de HTTP como GET, PUT, POST y DELETE. Los recursos se interconectan mediante hipervínculos. Las limitaciones arquitectónicas REST permiten que la aplicación logre un alto grado de escalabilidad. También se reduce la latencia (suma de retardos temporales dentro de una red) en la interacción entre clientes y servidores.

2.4. Comunicación con la ESPOL mediante el protocolo SOAP

Para mostrar la manera de comunicación de nuestra aplicación con los Servicios WEB de la ESPOL vamos a realizar el siguiente requerimiento, verificar que los datos de inicio de sesión (usuario y contraseña) estén asociados algún estudiante politécnico.

Figura 1.- Mecanismo de autenticación de un usuario

Solución:
Para la validación de un usuario del sistema se utilizó la operación de "autenticación" que tiene el Servicio web "directorioEspol" [3]
Para acceder a la operación "autenticación" se construyó un mensaje de petición en versión SOAP1

```
POST /services/directorioEspol.asmx HTTP/1.1
Host: www.academico.espol.edu.ec
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://academico.espol.edu.ec/webservices/autenticacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
        <autenticacion xmlns="http://academico.espol.edu.ec/webservices/">
                <varUser>string</varUser>
                <varContrasenia>string</varContrasenia>
        </autenticacion>
</soap:Body>
</soap:Envelope>
```

Figura 2.- Mensaje de petición en la autenticación de un usuario

El servidor envía el siguiente mensaje de respuesta:

```
POST /services/directorioEspol.asmx HTTP/1.1
Host: www.academico.espol.edu.ec
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://academico.espol.edu.ec/webservices/autenticacion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
        <autenticacion xmlns="http://academico.espol.edu.ec/webservices/">
                <varUser>string</varUser>
                <varContrasenia>string</varContrasenia>
        </autenticacion>
</soap:Body>
</soap:Envelope>
```

Figura 3.- Mensaje de respuesta en la autenticación a un usuario

Para la comunicación con los Servicios web y el procesamiento del mensaje de respuesta se utilizaron las siguientes librerías:

Tabla 1.: Librerías Utilizadas para la comunicación con los servicios web de la ESPOL

Librería	Versión
Axis	1.0
Saaj	1.2
Axis-ant	1.4
commons-discovery	0.2
Jaxrpc	1.1
log4j-1.2.8	1.2.8
Mail	1.3.3
wsdl4j	1.5.1

3. Comunicación con los Servicios Web de Facebook

Para comunicarnos con los Servicios Web de Facebook utilizamos FQL.

3.1. Facebook Query Language (FQL) [4]

FQL es una forma de consulta de datos en Facebook. Hay otras formas de acceso mediante métodos del API. Muchas de las llamadas normales de la API son contenedores simples para las consultas de FQL. Una consulta típica se ve algo como esto:
SELECT name FROM user WHERE uid=1630691750;
Nos da como resultado Gonzalo Antonio Ordóñez Rodríguez. Con este simple ejemplo nos damos cuenta que FQL es muy fácil de utilizar. Las principales ventajas de utilizar FQL son las siguientes:

- XML liviano: reduce los costos de ancho de banda y análisis. En lugar de obtener toda la información disponible acerca de un gran conjunto de elementos, se puede obtener sólo los campos que se desee de acuerdo a la condición específica. Usted puede solicitar el conjunto específico de información mediante la adición de las restricciones a la cláusula WHERE, y sólo lista algunos campos en la cláusula SELECT.
- Mayor cantidad de requerimientos pueden reducir el número de solicitudes necesarias. A menudo, los datos que usted está tratando de obtener depende de los resultados de una llamada al método anterior. Por ejemplo, con la API tradicional, para obtener los nombres de los amigos de una persona, primero llame friends.get y luego pasar el resultado directamente de nuevo a users.getInfo. Ahora sólo se puede ejecutar una consulta FQL que utiliza una subconsulta para obtener el conjunto de amigos - lo que reduce un viaje extra de ida y vuelta, y la latencia asociada a ella. Si se llama fql.multiquery, puede

realizarse consultas aún más poderosas porque utiliza los resultados de una consulta en la convocatoria de otra consulta dentro de la misma llamada.

- Proporciona una única interfaz coherente y unificada, para todos los datos. En lugar de tener que aprender muchos métodos diferentes, puede hacer todas sus peticiones con una función que tiene un tipo de cambio constante. Además, si es necesario llamar a cualquiera de los métodos tradicionales, el retorno de XML es muy similar, por lo que el coste de transferencia es insignificante.

3.2. Java Script Client Library

Nos permite hacer llamadas a la API REST de Facebook mediante el método fql_query(String query, Object onRequestCompleted) que permite utilizar una interfaz al estilo SQL con mayor facilidad de consulta de datos sobre la Red Social. El cliente puede hacer sus propias consultas a las Tablas FQL

3.3. Tablas FQL[5]

Usted puede utilizar las siguientes tablas de referencia para la construcción de las consultas de FQL.

- album
- application
- comment
- cookies
- connection
- daily_metrics
- developer
- event
- event_member
- family
- friend
- friend_request
- friendlist
- friendlist_member
- group
- group_member
- link
- link_stat
- listing
- mailbox_folder
- message
- metrics
- note
- notification
- page
- page_admin
- page_fan
- permissions
- photo
- photo_tag
- profile
- standard_friend_info
- standard_user_info
- status
- stream
- stream_filter
- thread
- translation
- user
- video
- video_tag

3.4. Requerimientos que se implementaron

Tabla 2.: Requerimientos Facebook.

Requisito	Descripción
1. Autenticación con la red social	El usuario ingresa su correo y contraseña para validar si se encuentra registrado en esta red social.
2. Cargar álbumes	Una vez terminado el requisito anterior hacemos una petición al servidor de Facebook vía FQL para que consulte todos los álbumes del usuario que inicio sesión y luego poder mostrarlos.
3. Guardar álbumes	Permite que los usuarios seleccionen los álbumes que quiera que se muestren en el Directorio Interactivo de la ESPOL, solo se guarda ciertos valores tales como, identificador del usuario, identificador del álbum, identificador de la foto, entre otros.
4. Consulta de toda la información concerniente a un álbum	Las fotos y los comentarios de un álbum se consultan en tiempo real al servidor de "Servicios Web" de ésta red.

3.5 Diagrama de la Solución

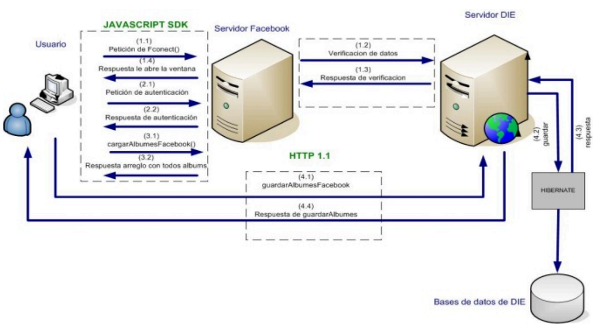

Figura 4. Diagrama de Funcionamiento de Facebook con nuestra aplicación

La figura 4 nos muestra la manera como se comunica el Directorio Interactivo de la ESPOL con los servidores de Facebook, toda ésta se la hace del lado del cliente por medio de javascript, el servidor de Facebook va una sola vez a nuestro servidor para verificar la solicitud de identificación (ID de la aplicación que previamente la obtenemos en Facebook Developer, registrando algunos datos de nuestra aplicación), paso seguido el usuario guardar los álbumes que desea mostrar en el Directorio Interactivo, para esto nosotros solamente guardamos el Id del álbum en nuestro servidor.

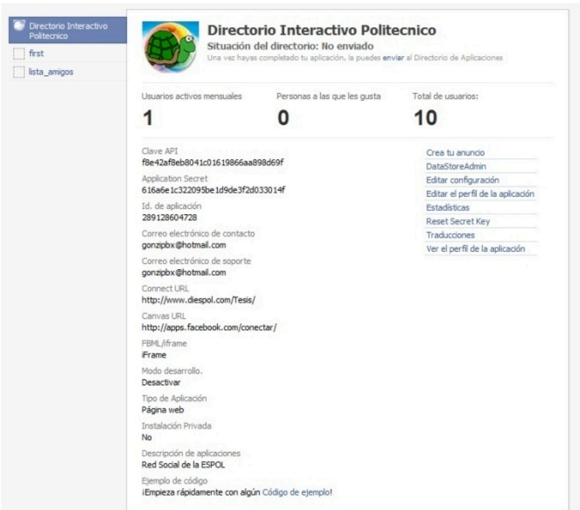

Figura 5: Registro de nuestra aplicación en Facebook Developer.

4. Comunicación con los Servicios web de Hi5

Hi5 tiene 2 formas de acceder a sus servidores de servicio:

- SOAP: Se usa a través de un conjunto de WSI Basic Profile compatible con SOAP v1.1, además es compatible con XML-binary Optimized Packaging (XOP) y con el Mecanismo de Optimización de Transmisión de Mensajes para SOAP (MTOM), para la transmisión de datos binarios.
- REST: permite acceder a un conjunto de recursos a través de un conjunto fijo de operaciones.

Para acceder a la información de un usuario solo se utiliza su correo, pero su perfil debe ser público.

4.1. Requerimientos que se implementaron

Tabla 3: Requerimientos Hi5

Requisito	Descripción
Autenticación con la red social	El usuario ingresa solamente su correo y el web services de ésta red social valida su existencia siempre
Consulta de todos los álbumes que posee un usuario	Solo se podrán acceder a los álbumes de un usuario que tenga su perfil como público.
Almacenamiento de solo aquellos álbumes que un usuario desee que tengan acceso público	El almacenamiento no es físico, solo se guarda ciertos valores, tales como: El identificador del usuario, identificador del álbum, descripción del álbum, entre otros.
Consulta de la información concerniente a un álbum	Las fotos (con su descripción) de un álbum se consultan en tiempo real al servidor de web services de ésta red.

4.2. Diagrama de la Solución

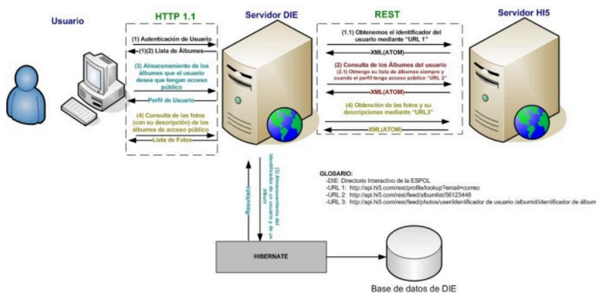

Figura 6: Diagrama de funcionamiento de los servidores de hi5 con nuestra aplicación.

4.3. Detalle Técnico

- Servicios Web basados en REST:
  - La obtención del identificador de un usuario de hi5:
    - http://api.hi5.com/rest/profile/lookup?email=**correo**
  - La obtención de la lista de álbumes:
    - http://api.hi5.com/rest/feed/albumlist/**identificador de usuario**
  - La obtención de las fotos con su respectiva descripción:
    - http://api.hi5.com/rest/feed/photos/user/**identificador de usuario** /albumid/**identificador de álbum**
- Mensaje de intercambio de datos en XML
  - Para elaborar el requerimiento de petición.
  - Para retornar el dato que se origina después de la ejecución de un método en el servicio web.

Para la comunicación con los Servicios web basados en REST, y para el procesamiento del mensaje de respuesta se utilizaron las siguientes librerías.

Tabla 4: Librerías utilizadas para la comunicación con los servicios web de hi5

Librería	Versión
activation	1.1.1
jaxws-api	2.0
wsdl4j	1.5.2
commons-logging	1.1
commons-httpclient	3.0
commons-codec	1.3
jdom	1.0
mail	1.4

5. Comunicación con los servicios web de YouTube

Para entender como realizamos la comunicación entre los servicios web de YouTube y nuestra aplicación debemos conocer el concepto de Redifusión Web

5.1. Redifusión Web

- Es el reenvío de contenidos desde una fuente original (sitio web de origen) hasta otro sitio web de destino (receptor), que a su vez se convierte en emisor puesto que pone a disposición de sus usuarios los contenidos recién recibidos
- Componentes en un sistema de redifusión:
    - Sitio web origen
    - Agregadores: permiten que los usuarios se suscriban y vean los contenidos de la fuente web.
    - Sitio web destino
    - Usuarios finales
    - Formatos de Redifusión(codificadas en XML)
        - RSS y ATOM

5.1.1. RSS [6]

RSS es una familia de formatos de fuentes web codificados en XML. Se utiliza para suministrar a suscriptores, información actualizada de manera frecuente. El formato permite distribuir contenido sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS. A pesar de eso, es posible utilizar el mismo navegador para ver los contenidos RSS. Las últimas versiones de los principales navegadores permiten leer los RSS sin necesidad de software adicional. RSS es parte de la familia de los formatos XML desarrollado específicamente para todo tipo de sitios que se actualicen con frecuencia y por medio del cual se puede compartir la información y usarla en otros sitios web o programas. A esto se le conoce como redifusión web.

5.1.2. ATOM[7]

Es un fichero en formato XML similar al RSS, usado para la Redifusión Web. Se lo desarrolló para utilizarlo como estándar en la redifusión de contenidos web y, para que sea un formato de redifusión más flexible. No terminó con el problema, es más, actualmente conviven los tres formatos.
Tanto ATOM como RSS tienen el mismo objetivo que es permitir la distribución de contenidos y noticias de sitios web.
Un documento ATOM puede contener mayor cantidad de información, de forma más compleja y consistente que un documento RSS.

5.2. Requerimientos que se implementaron

Tabla 5: Requerimientos YouTube

Requisito	Descripción
1. Autenticación con la red social	El usuario ingresa su correo y contraseña que posee como datos de sesión para validar si pertenece o no a ésta red social
2. Consulta de todos los videos que posee un usuario	Se visualizaran todos los videos que tiene el usuario esta red, para que luego elija aquellos que tendrán acceso público.
3. Almacenamiento de solo aquellos videos que un usuario desee que tengan acceso público.	El almacenamiento no es físico, solo se guarda ciertos valores, tales como: El identificador del usuario, identificador del video, titulo del video, entre otros.
4. Mostrar todos los videos asociados a un usuario	Se visualizaran todos los videos que el usuario los configuro como público

5.3. Diagrama de la Solución

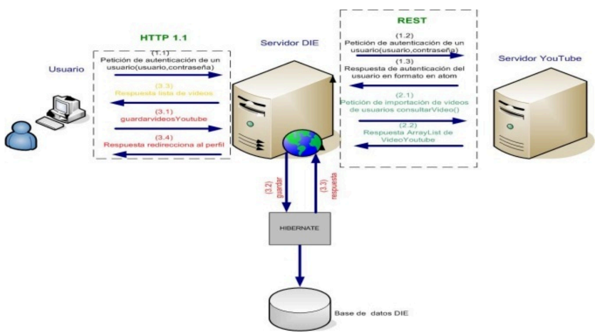

Figura 7: Diagrama de funcionamiento de los Servidores de YouTube con nuestra aplicación

5.4. Detalle Técnico

Para que nuestra aplicación se pueda comunicar con los servidores de YouTube es necesario registrarla en http://code.google.com/apis/youtube/dashboard/, para ello hay que tener un correo de gmail o una cuenta en YouTube.

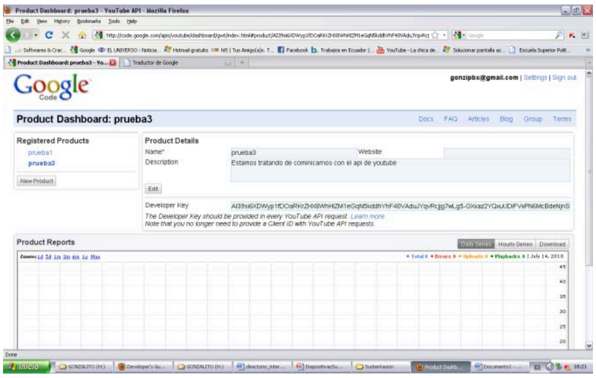

Figura 8: Obtención del API Key de YouTube

Para la comunicación con los Servidores de Youtube utilizamos **Google Data APIs (Gdata library)** la cual proporciona un protocolo simple y estándar para la lectura y la escritura de datos en la web.

Tabla 6: Librerías utilizadas para la comunicación con los servidores de YouTube.

Librería	Versión
gdata-youtube	2.0
gdats-youtube-mets	2.0
gdata-core	1.0
gdata-media	1.0

6. Interfaz de la Aplicación

Para ingresar a nuestra aplicación es necesario que se estudiante de le ESPOL, ya que se autentica con su respectivo usuario y clave.

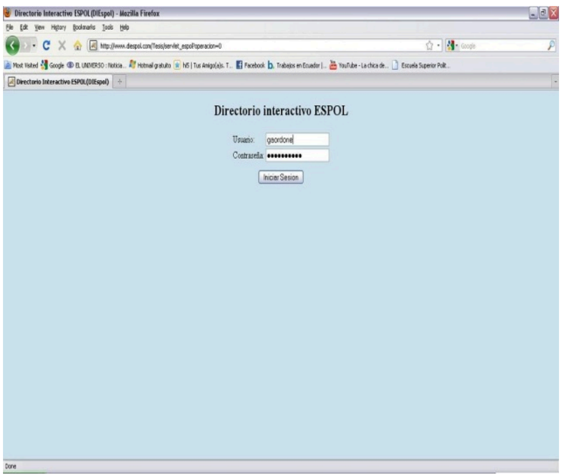

Figura 9: Inicio de sesión

6.1. Primera Pantalla de Ingreso al Sistema

Si es la primera vez que ingresamos al sistema, solo se mostrará la información que exista en las bases de datos de la ESPOL.

Para importar los álbumes de Facebook y Hi5, además de los videos de Youtube se debe configurar los datos de la cuenta que posee el usuario en éstas redes sociales.

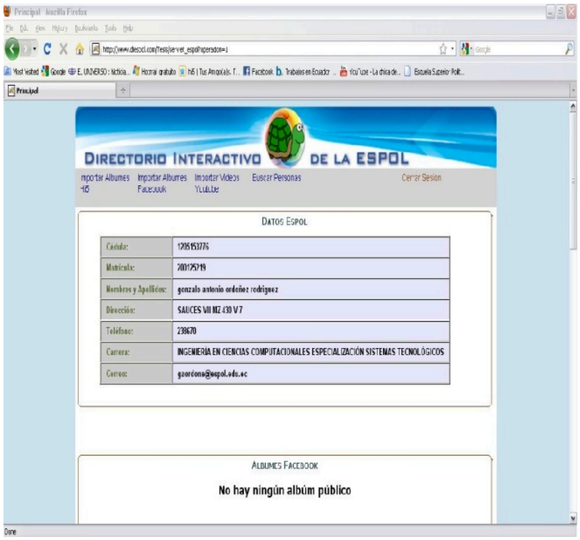

Figura 10: Primera Pantalla de ingreso al sistema.

6.2 Importar Álbumes de Facebook

Para importar álbumes de Facebook damos clic en "Importar Álbumes de Facebook"

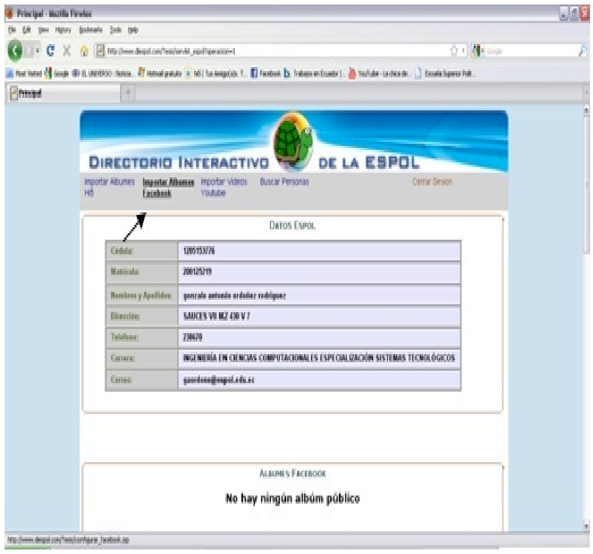

Figura 11: Importar álbumes de Facebook

En la siguiente pantalla se debe dar clic en "Connect" e, iniciar sesión con el correo y contraseña de la cuenta virtual de ésta red social.

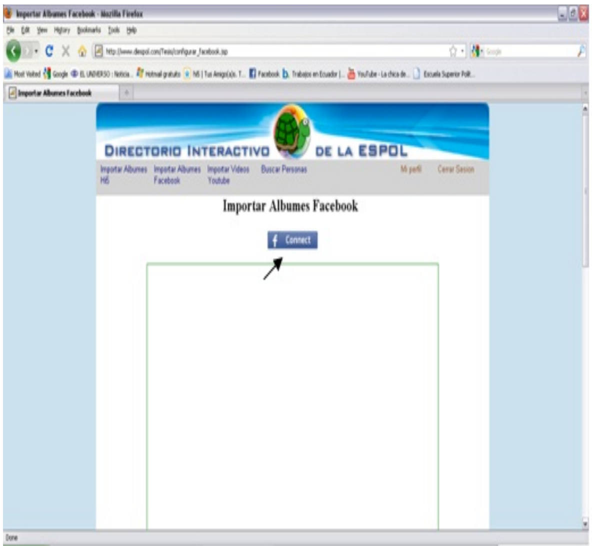

Figura 12: Conectar a la aplicación de Facebook

Nos aparecerá una ventana de Facebook que indica que la aplicación "Directorio Interactivo de la ESPOL", tendrá acceso a nuestros álbumes para importarlos.

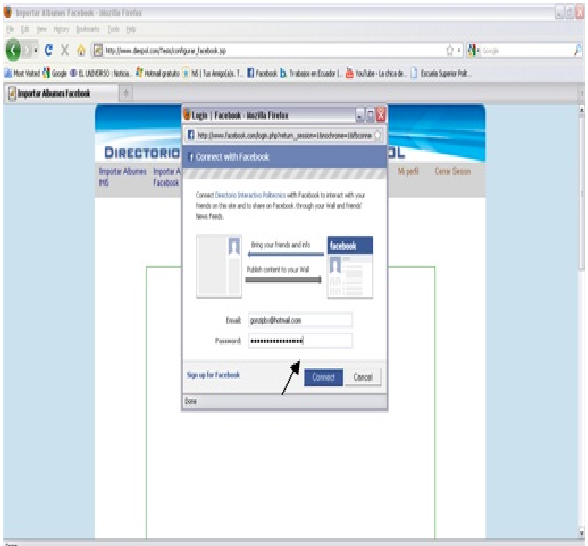

Figura 13: Ventana de autenticación con Facebook

Una vez conectados, se visualizarán todos los álbumes que tiene el usuario. Procedemos a elegir aquellos álbumes que deseamos que sean públicos para todos los usuarios del "Directorio Interactivo de la ESPOL" y, presionamos el botón Guardar.

Figura 14: Álbumes del usuario en Facebook

Luego nos re-direccionará a la ventana principal, donde se visualizarán los álbumes públicos (seleccionados anteriormente) y que podrán ser vistos por los demás usuarios.

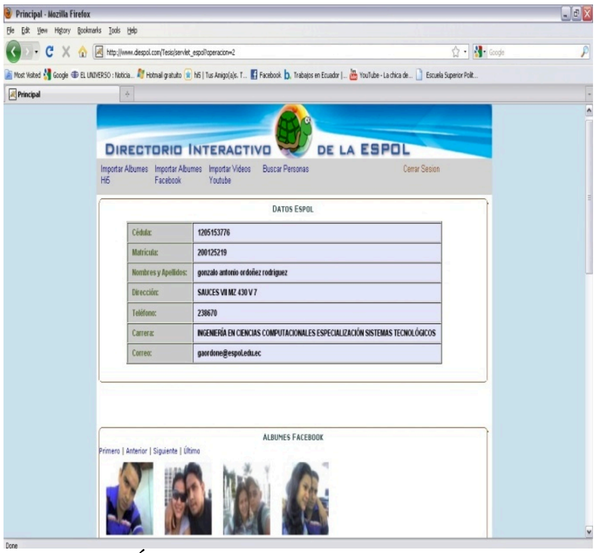

Figura 15: Álbumes de Facebook configurados con acceso público en el Directorio Interactivo de la ESPOL.

De una manera similar se podrán configurar los álbumes de Hi5 y los videos de YouTube.

7. Conclusiones

- La finalidad de que una red social publique su API, es para generar un mayor tráfico hacia su dominio, y aprovechar el interés de desarrolladores en agregar valor al sitio, por medio de terceras aplicaciones.
- El API de la red social Hi5 es inestable y se debe revisar paulatinamente los cambios en sus espacios de nombres.

- El envió del requerimiento y procesamiento del mensaje de respuesta (XML) lo realizamos sin utilizar ninguna librería facilitada por Hi5

- Facebook lidera el mercado de las redes sociales, porque su aplicación posee un alto grado de usabilidad, además su documentación es clara, y su API puede ser accedida por diversos lenguajes de programación (JavaScript, Php, java, Perl, Phyton, C#, entre otros).
  - FQL es un lenguaje muy parecido a SQL. Debe permitir realizar JOIN, para obtener mejores resultados al momento que se realizan las consultas.
  - El lenguaje FQL permite expandir el uso del API, debido a que no limita al desarrollador a consumir métodos predefinidos.
- El Api de google para Youtube fue el menos complejo de aprender y utilizar porque Code Google tiene una documentación bien detallada para el uso de las librerías
  - Code Google: es una comunidad de google para desarrolladores. Permite interactuar con muchas redes sociales por medio del lenguaje java. Los desarrolladores lo preferimos porque permite descargar librerías, que están bien documentadas y son muy fáciles de usar.
- Es favorable que se sigan definiendo y que se respeten los estándares de desarrollo de aplicaciones web. Facilitan el aprendizaje, programación y comunicación entre distintas aplicaciones.

8. Recomendaciones

- Planificar las actividades a realizar, establecer los recursos y tiempos de entregas en los avances, para evitar incumplimientos.
- Comprender los distintos protocoles de comunicación (SOAP, REST, RPC, etc.), con el fin de obtener el mayor beneficio posible
- Comprender la forma correcta de interacción con los APIs de las redes sociales
- Respetar los estándares de desarrollo web, con el fin de obtener una mayor velocidad de renderización, escalabilidad, y mantenimiento.
- Utilizar mecanismos que permitan que la mayor cantidad de procesamiento recaiga en el cliente, para que el servidor pueda atender una mayor cantidad de peticiones.
- Alojar la aplicación en un Servidor Web con excelentes características, para que los usuarios interactúen de forma rápida con la aplicación

9. Agradecimiento

Damos sinceras gracias a Dios, nuestro Señor, por darnos la vida y los recursos necesarios para alcanzar ésta meta anhelada.
A nuestras familias, por brindarnos su apoyo y confianza.
A nuestros amigos y compañeros que han sido parte de nuestra vida estudiantil.
A nuestros profesores y Director de Tesis, que han tenido mucha paciencia para enseñarnos y enrumbarnos correctamente.
A todos ellos, sinceras gracias por apoyarnos en la obtención del título de Ingeniería en Ciencias Computacionales.

10. Referencias

(1)Wikipedia, Servicios Web, http://es.wikipedia.org/wiki/Servicios_Web, última visita 5 Enero 2010.

(2)W3C, SOAP Versión 1.2, http://www.w3c.es/Traducciones/es/TR/2003/REC-soap12-part0-20030624/, última visita 8 Enero 2010.

(3)ESPOL, webService directorioEspol, https://www.academico.espol.edu.ec/services/directorioEspol.asmx, última visita 15 Enero 2010.

(4)Facebook, Facebook Query Language (FQL), http://wiki.developers.facebook.com/index.php/FQL, última visita 16 Enero 2010.

(5)Facebook, Tablas FQL, http://wiki.developers.facebook.com/index.php/FQL_Tables, última visita 16 Enero 2010.

(6)Wikipedia, RSS, http://es.wikipedia.org/wiki/RSS , última visita 30 de octubre del 2009

(7)Wikipedia, Formato de Redifusión ATOM, http://es.wikipedia.org/wiki/Atom_%28formato_de_redifusi%C3%B3n%29 , última visita 30 de octubre del 2009