

**Seminario de Graduación "Desarrollo de Aplicaciones basadas en la Web 2.0"**

# Tesina de Seminario

## DIRECTORIO INTERACTIVO DE LA ESPOL

**GONZALO ANTONIO ORDÓÑEZ RODRÍGUEZ  
STALIN ALBERTO ARROYABE MERCHAN**

# 1.- Problemas – Motivos – Causas

- La universidad no posee una herramienta o aplicación tecnológica que permita:
  - Fomentar la relaciones interpersonales entre estudiantes
  - Conocer a nuevos compañeros.

## 2.- Solución

- Aplicación web llamada “Directorio Interactivo de la ESPOL”, que permita:
  - Publicar información de un estudiante en su perfil de usuario.
    - ❖ La información es extraída desde las redes sociales más populares de nuestro medio (Facebook, Hi5, Youtube), y desde la base de datos universitaria
      - ✓ Facebook y Hi5.- se importaran los álbumes de la cuenta de usuario(previa configuración).
      - ✓ Youtube.- se importaran los videos de la cuenta de usuario (previa configuración).

# 3.- Ventajas de la Solución Planteada

- La mayoría de los estudiantes dedican un pequeño, pero significativo tiempo a la interacción con las redes sociales, colocando información personal a cada instante.
  - Esta información puede ser accedida mediante APIs y Servicios Web, respetando ciertas políticas de seguridad.
- Un usuario no tendrá que ingresar gran cantidad de información para tener personalizado su perfil en el "Directorio Interactivo de la ESPOL".

# 4.-Objetivos de la Solución

- Ayudar a enlazar más estrechamente los lazos compañerismo y de amistad entre los estudiantes, mediante el acceso libre a la información personal de cualquiera de ellos.
  - Está información solo puede ser visualizada por estudiantes, porque la única forma de ingreso es mediante usuario y contraseña ESPOL.
- No perder el rastro a los estudiantes politécnicos(activos e inactivos)
- Investigar, aprender, analizar, aplicar y evaluar las diferentes formas o mecanismos de comunicación entre aplicaciones web
  - Se toma en cuenta el uso de estándares, protocolos e interfaces de programación avanzada (APIs).

# 5.- Teorías de la web 2.0 utilizadas en el desarrollo de nuestra Solución

## 5.1 Web 2.0

- Se basa en la colaboración y el intercambio ágil de información entre los usuarios.
- El Web 2.0 es una actitud y no precisamente una tecnología.
- Tecnologías que dan vida a un proyecto Web 2.0
  - Respeto a los estándares como el XHTML.
  - Separación de contenido del diseño con uso de hojas de estilo.
  - Sindicación de contenidos.
  - Ajax (javascript asincrónico y XML), Flash, Flex, etc.
  - Lenguajes de Programación para páginas dinámicas.
  - Dar control total a los usuarios en el manejo de su información.
  - Proveer APis o XML para que las aplicaciones puedan ser manipuladas por otros.
- Ejemplos de sitios basados en la Web 2.0:
  - Facebook, Hi5. Myspace, Youtube, etc.
  - Blogs
  - Wikipedia
  - Entre otros

# 5.- Teorías de la web 2.0 utilizadas en el desarrollo de nuestra Solución

## 5.2.1 Servicios Web(REST)

- REST es un estilo de arquitectura para generar aplicaciones de cliente-servidor. Focalizado en Recursos.
- Sus siglas significan “Transferencia del Estado Representacional”
- Es una técnica de arquitectura de software para sistemas distribuidos de tipo hipermedia
- REST no es un estándar. No existe una especificación formal de REST.
- Principios básicos del protocolo:
  - Utiliza los métodos HTTP(POST, GET, PUT, DELETE) en la manipulación de recursos
  - No mantiene estado en la comunicación: el cliente envía en el mensaje todos los datos necesarios para que el servidor procese la respuesta
  - Expone URIs en forma de directorios
    - ❖ <http://www.xyz.org/discusion/temas/{id-de-tema}>
  - Transfiere XML, JavaScript Object Notation (JSON), o ambos
    - ❖ Es el estado actual del recurso, al momento en que el cliente de la aplicación realiza la petición
- Quienes utilizan REST:
  - Hi5, Facebook, Amazon, Ebay, entre otros.

# 5.- Teorías de la web 2.0 utilizadas en el desarrollo de nuestra Solución

## 5.2.2 Servicios Web(SOAP)

- SOAP es una especificación de protocolo para intercambiar datos entre dos extremos. Focalizado en Operaciones
- Sus siglas significan "Protocolo de Acceso Simple a Objetos"
- Es un estándar adoptado y definido por la W3C
- Utiliza un documento (WSDL) que especifica los servicios que el Servidor se compromete a ofrecer al cliente, siempre que éste los solicite de la forma adecuada
  - WSDL Es un fichero en donde se describe los tipos de datos, operaciones, excepciones y parámetros del servicio web basado en SOAP.



# 5.- Teorías de la web 2.0 utilizadas en el desarrollo de nuestra Solución

## 5.3 API

- Significa “Interfaz de programación de aplicaciones”
- El principal propósito de una API consiste en proporcionar un conjunto de funciones de uso general
- Los desarrolladores las utilizan porque se evitan el trabajo de programar mucho mas de lo necesario
- Ejemplos de APIs
  - Microsoft Framework .NET
  - Java EE
  - Javascript SDK de Facebook
  - Entre otras

# 5.- Teorías de la web 2.0 utilizadas en el desarrollo de nuestra Solución

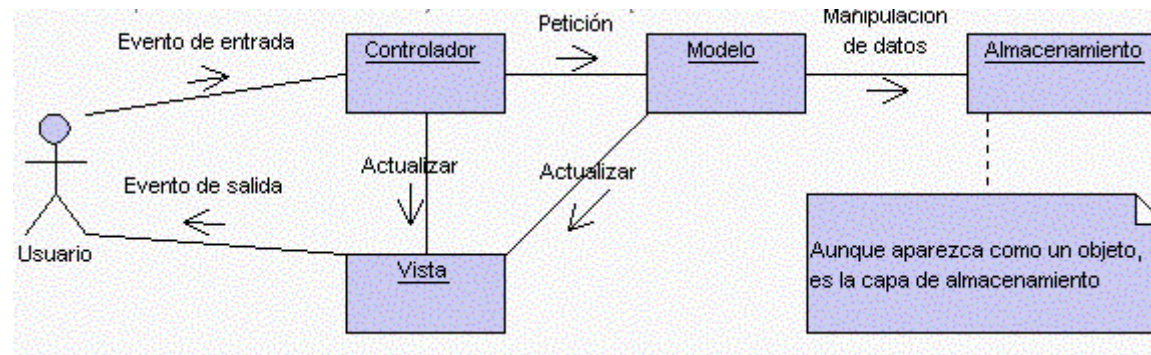
## 5.4 Redifusión Web

- 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.- Teorías de la web 2.0 utilizadas en el desarrollo de nuestra Solución

## 5.5 Modelo Vista Controlador

- Es un estilo de arquitectura de software que posee 3 componentes:
  - Modelo: datos y reglas de negocio
  - Vista: muestra la información del modelo al usuario
  - Controlador: gestiona las entradas del usuario



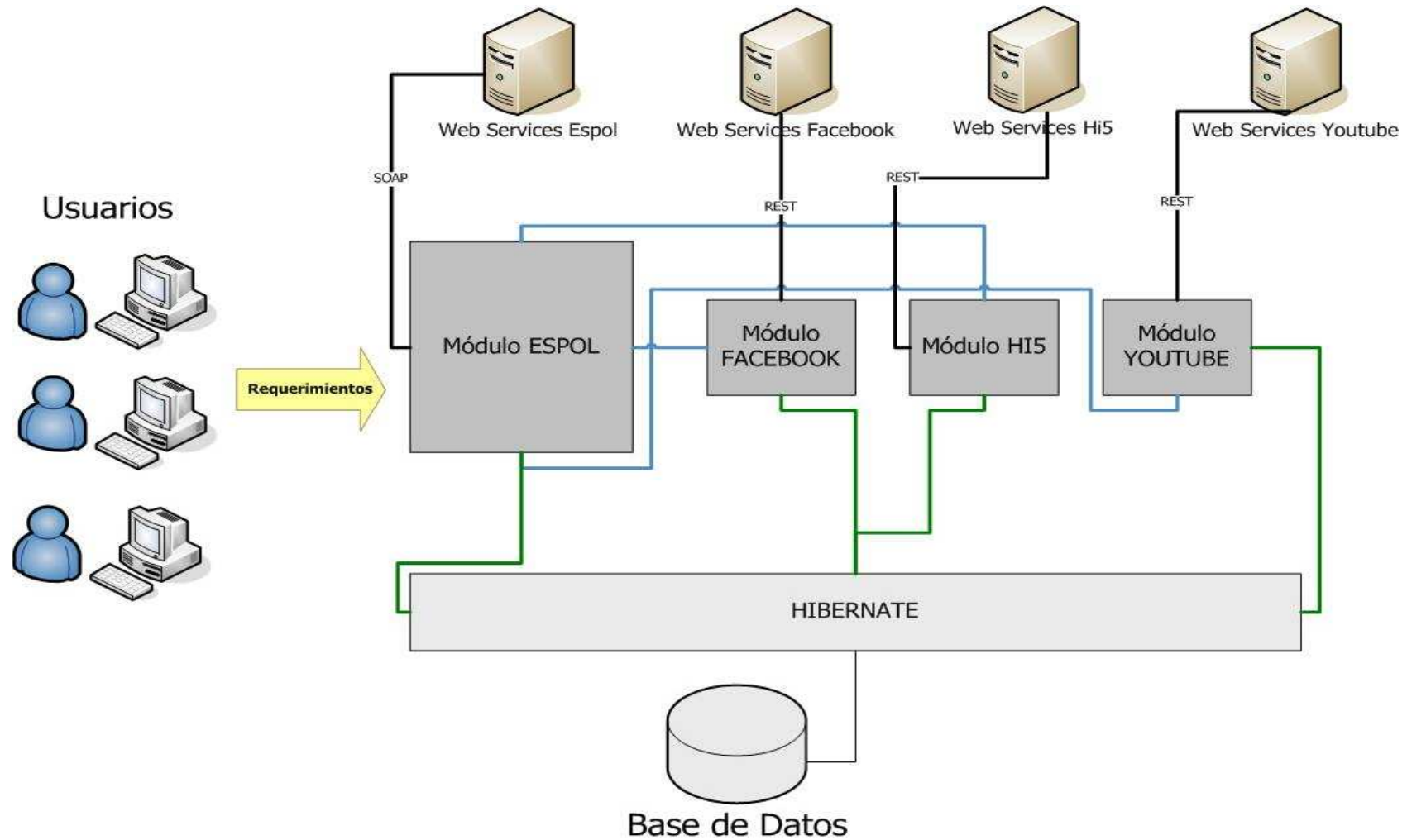
- Frameworks (Marcos de trabajos) basados en MVC para desarrollar aplicaciones web
  - Java: Grails, Spring, entre otros.
  - Php: Symfony, Zend Framework, entre otros.
  - Microsoft: Asp.net

# 5.- Teorías de la web 2.0 utilizadas en el desarrollo de nuestra Solución

## 5.6 Motores de Persistencia

- Es un mecanismo para conectar los objetos de negocio con la base de datos.
- El motor de persistencia traduce los datos de registros a objetos y de objetos a registros
- Ventaja de utilizarlo:
  - Las aplicaciones de software son mas flexible y reusabilidad.
  - No se depende del lenguaje de manipulación de datos, y por eso se puede emigrar de motor de base de datos
- Ejemplos de motores de persistencia:
  - Software Libre:
    - ❖ Hibernate (Usado en este proyecto)
  - Comerciales
    - ❖ ORM.net

# 6.-Descripción Modular de la Solución



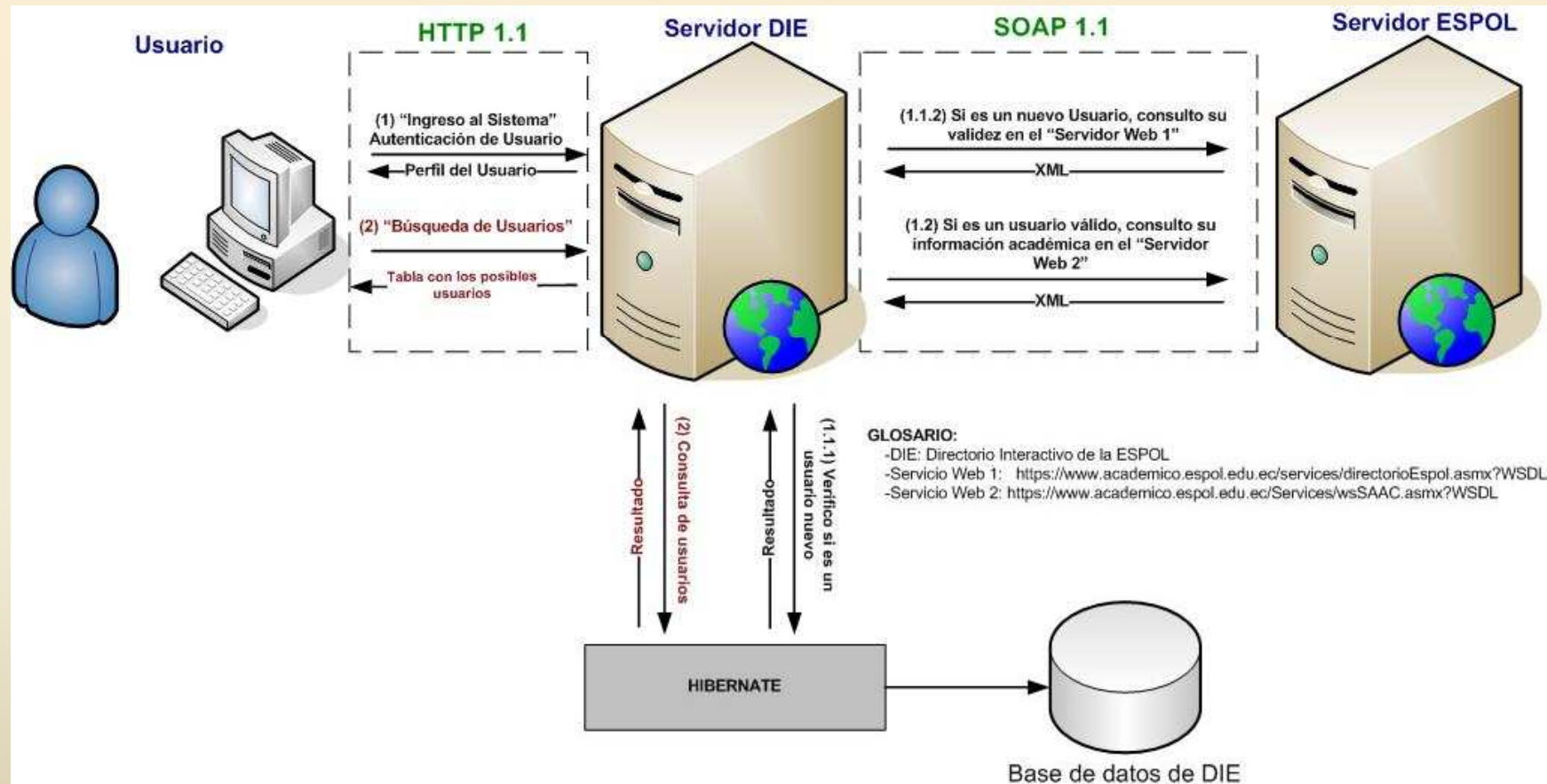
# 6.1 Módulo de ESPOL

## 6.1.1 Requerimientos que implementa

Requisito	Descripción
1. Ingreso al sistema	Permite validar que solo puedan acceder estudiantes politécnicos a la aplicación. Si es la primera vez que un usuario ingresa con sus datos, se valida en el web services ESPOL, caso contrario se lo hace en la base de datos de nuestra aplicación.
2. Búsqueda de usuarios	Permite consultar por nombres y apellidos, para luego listar los usuarios y da la opción de visualizar su perfil

# 6.1 Módulo de ESPOL

## 6.1.2 Diagrama de Funcionamiento



# 6.1 Módulo de ESPOL

## 6.1.3 Detalle Técnico

- Servicios Web basados en SOAP, junto a su Descriptor WSDL
- 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 SOAP, y para el procesamiento del mensaje de respuesta se utilizaron las siguientes librerías:

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



# 6.1 Módulo de ESPOL

## 6.1.3 Detalle Técnico

```
POST /Services/wsSAAC.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/InformacionAcademicaEstudianteGet"
<?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>
    <InformacionAcademicaEstudianteGet
      xmlns="http://academico.espol.edu.ec/webservices/">
      <identificacion>string</identificacion>

      <Matricula>string</matricula>
    </InformacionAcademicaEstudianteGet>
  </soap:Body>
</soap:Envelope>
```

Mensaje de petición en la consulta de información de un usuario

```
<soap:Envelope><soap:Body><InformacionAcademicaEstudianteGetResponse><InformacionAcademicaEstudianteGetResult>

<xs:schema id="NewDataSet"><xs:element msdata:IsDataSet="true"
msdata:UseCurrentLocale="true" name="NewDataSet"><xs:complexType><xs:choice
maxOccurs="unbounded" minOccurs="0"><xs:element
name="Table"><xs:complexType><xs:sequence>

  <xs:element minOccurs="0" name="APELLIDOS" type="xs:string"/>
  <xs:element minOccurs="0" name="NOMBRES" type="xs:string"/>
  <xs:element minOccurs="0" name="DIRECCION" type="xs:string"/>
  <xs:element minOccurs="0" name="TELEFONO" type="xs:string"/>
  <xs:element minOccurs="0" name="NOMBRE_CARRERA" type="xs:string"/>
  <xs:element minOccurs="0" name="EMAIL" type="xs:string"/>
  <xs:element minOccurs="0" name="CARRERA" type="xs:string"/>

</xs:sequence></xs:complexType></xs:element></xs:choice></xs:complexType></xs:element><
/xs:schema>

<diffgr:diffgram><NewDataSet><Table diffgr:id="Table1" msdata:rowOrder="0">

  <APELLIDOS>ARROYABE MERCHAN</APELLIDOS>

  <NOMBRES>STALIN ALBERTO</NOMBRES>

  <DIRECCION>GUAYACANES MZ 72 VILLA 46</DIRECCION>

  <TELEFONO>2820706 </TELEFONO>

  <NOMBRE_CARRERA>INGENIERÍA EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN
SISTEMAS DE INFORMACIÓN</NOMBRE_CARRERA>

  <EMAIL>sarroyab@espol.edu.ec </EMAIL>

  <CARRERA>INGENIERÍA EN CIENCIAS COMPUTACIONALES
</CARRERA>

</Table></NewDataSet></diffgr:diffgram></InformacionAcademicaEstudianteGetResult></InformacionAcademicaEstudianteGetResponse></soap:Body></soap:Envelope>
```

Mensaje de respuesta en la consulta de la información de usuario

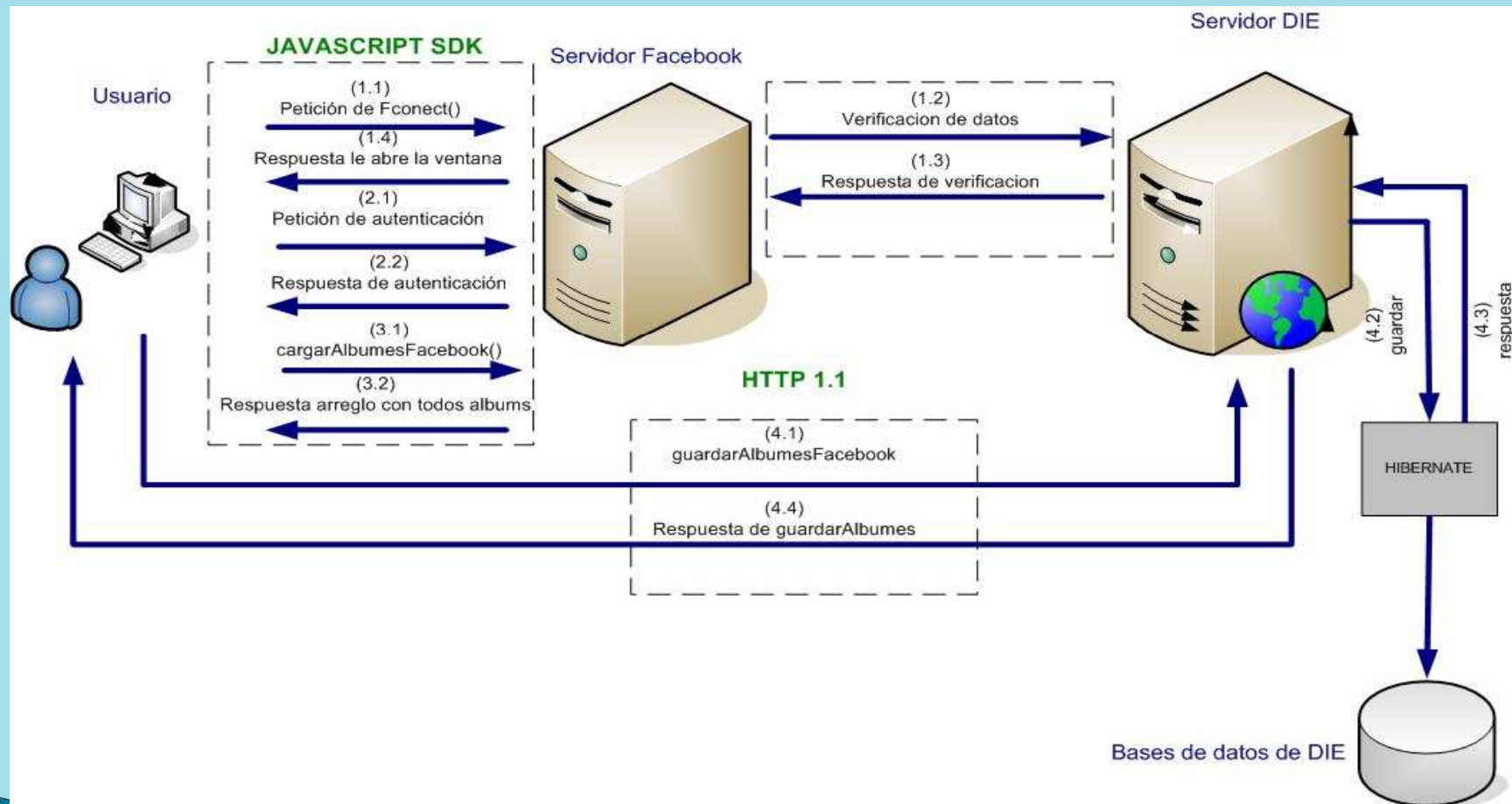
# 6.2 Módulo de Facebook

## 6.2.1 Requerimientos que implementa

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

# 6.2 Módulo de Facebook

## 6.2.2 Diagrama de Funcionamiento



# 6.2 Módulo de Facebook

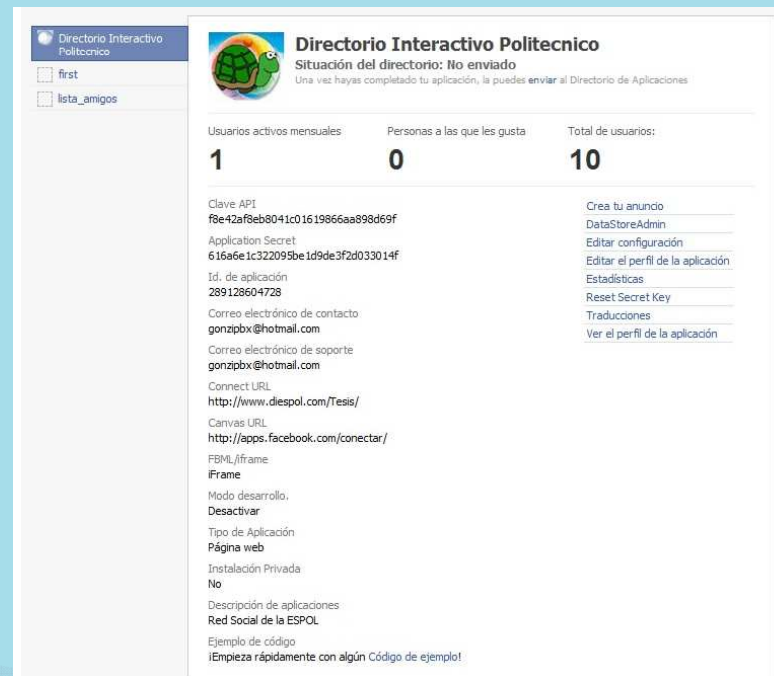
## 6.2.3 Detalle Técnico

### ▶ JavaScript SDK

- Es una manera simple de implementar entrada y registro sin tener que preocuparse sobre los detalles del protocolo

### ▶ Registrar la Aplicación

- Para obtener una solicitud de identificación para su sitio. Una vez que tenga el ID de la aplicación, podemos llamar mediante JavaScript al fconnect que no es mas que un botón de facebook que nos permite iniciar sesión



The screenshot shows the Facebook Developer console for an application named "Directorio Interactivo Politecnico". The status is "Situación del directorio: No enviado". The console displays various application details and statistics:

Usuarios activos mensuales	Personas a las que les gusta	Total de usuarios:
1	0	10

Clave API: f8e42af8eb8041c01619866aa898d69f  
Application Secret: 616a6e1c322095be1d9de3f2d033014f  
Id. de aplicación: 289128604728  
Correo electrónico de contacto: gonzpdx@hotmail.com  
Correo electrónico de soporte: gonzpdx@hotmail.com  
Connect URL: http://www.diespol.com/Tesis/  
Canvas URL: http://apps.facebook.com/conectar/  
FBML/iFrame: iFrame  
Modo desarrollo: Desactivar  
Tipo de Aplicación: Página web  
Instalación Privada: No  
Descripción de aplicaciones: Red Social de la ESPOL  
Ejemplo de código: ¡Empieza rápidamente con algún Código de ejemplo!

On the right side, there are several management links: Crea tu anuncio, DataStoreAdmin, Editar configuración, Editar el perfil de la aplicación, Estadísticas, Reset Secret Key, Traducciones, and Ver el perfil de la aplicación.

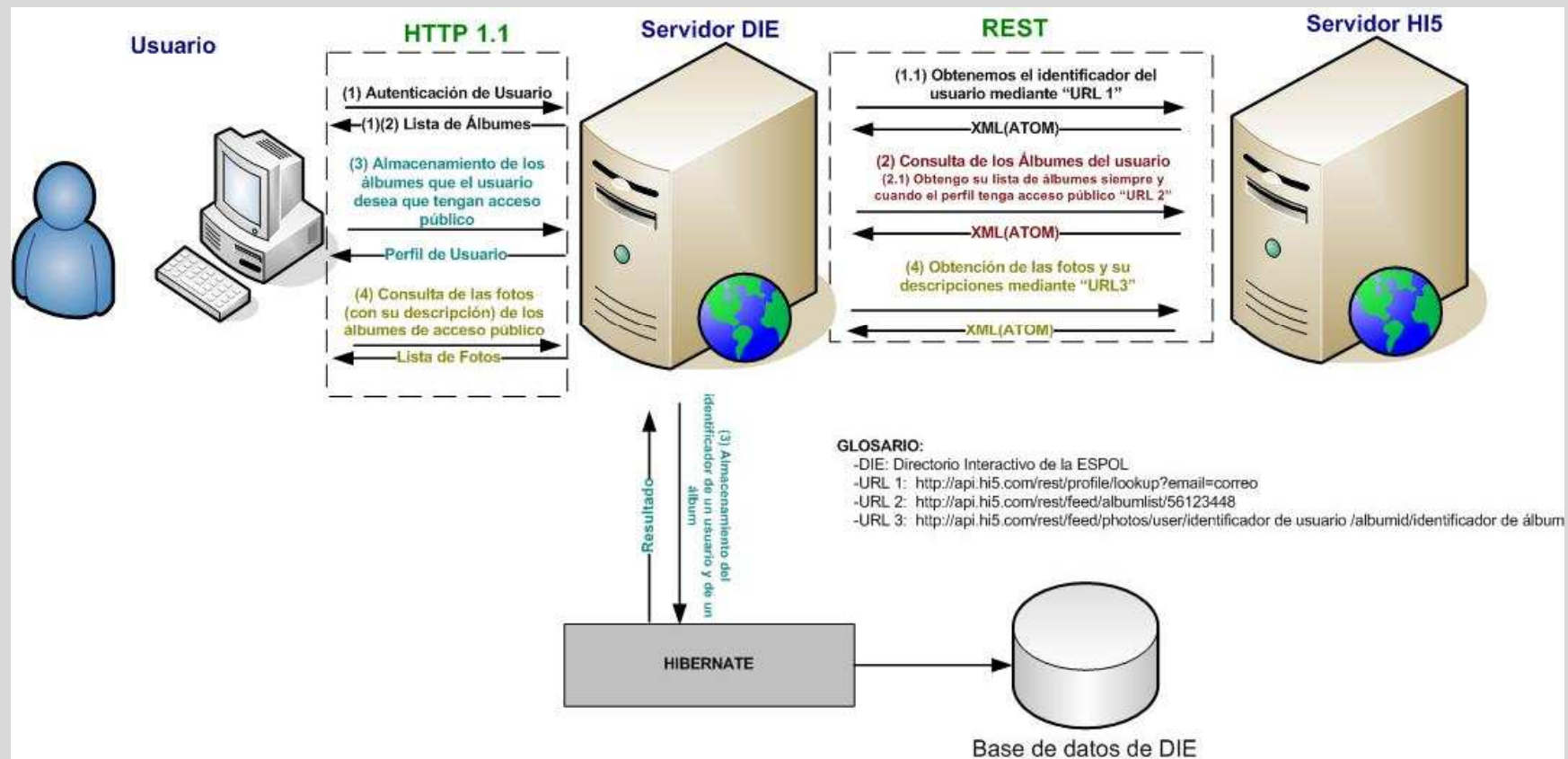
# 6.3 Módulo de Hi5

## 6.3.1 Requerimientos que implementa

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.

# 6.3 Módulo de Hi5

## 6.3.2 Diagrama de Funcionamiento



# 6.3 Módulo de Hi5

## 6.3.3 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](http://api.hi5.com/rest/feed/albumlist/identificador%20de%20usuario)
  - La obtención de las fotos con su respectiva descripción:
    - ❖ [http://api.hi5.com/rest/feed/photos/user/identificador de usuario /albumid/identificador de álbum](http://api.hi5.com/rest/feed/photos/user/identificador%20de%20usuario/albumid/identificador%20de%20album)
  
- 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:

# 6.3 Módulo de Hi5

## 6.3.3 Detalle Técnico

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



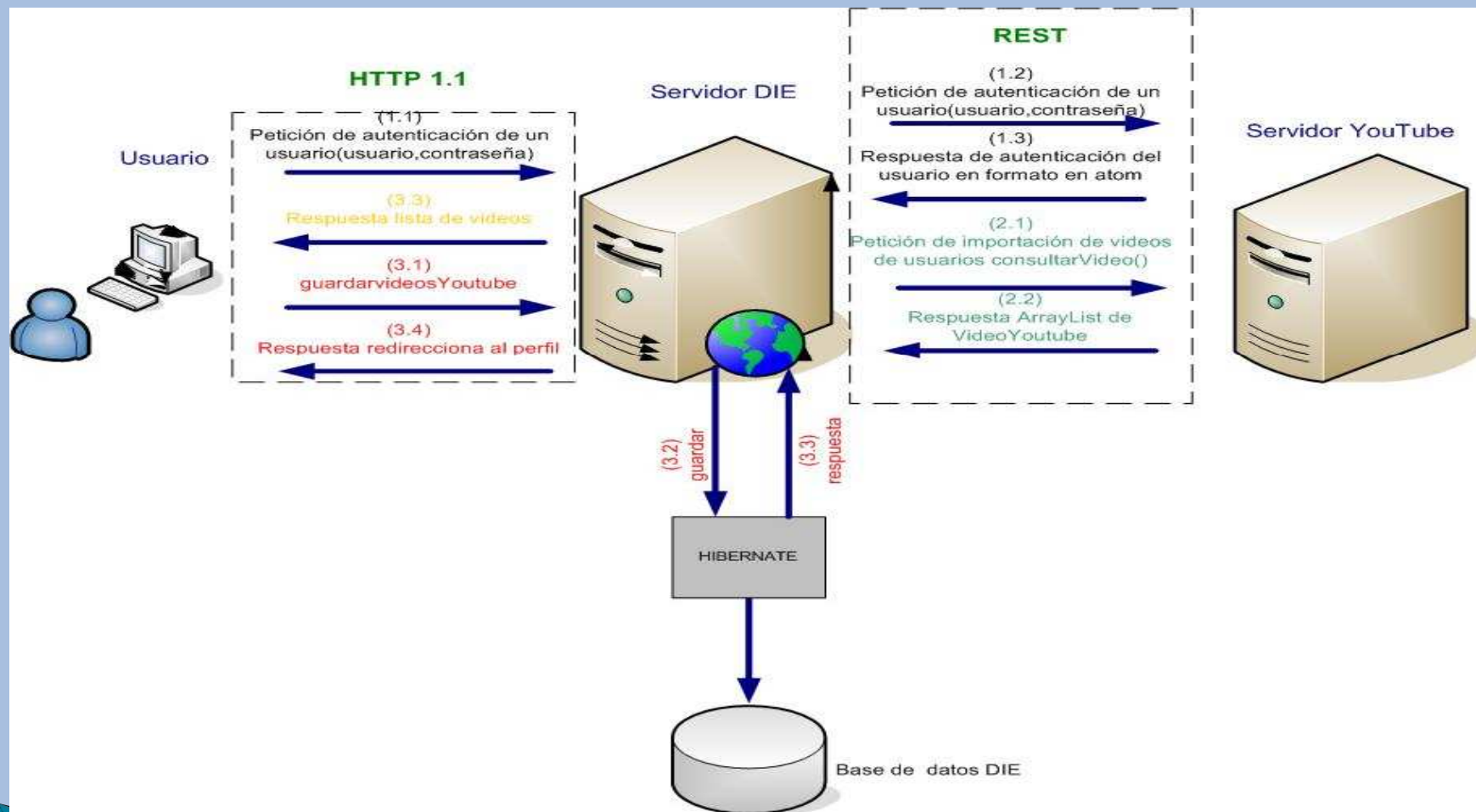
# 6.4 Módulo de Youtube

## 6.4.1 Requerimientos que implementa

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

# 6.4 Módulo de Youtube

## 6.4.2 Diagrama de Funcionamiento



# 6.4 Módulo de Youtube

## 6.4.3 Detalle Técnico

- ▶ Obtención del Api Key <http://code.google.com/apis/youtube/dashboard/>

The screenshot shows the Google Code Product Dashboard for a product named 'prueba3'. The interface includes a navigation menu with links for Docs, FAQ, Articles, Blog, Group, and Terms. The main content area is divided into several sections:

- Registered Products:** A list containing 'prueba1' and 'prueba3', with a 'New Product' button below.
- Product Details:** A form showing the product name 'prueba3', a description 'Estamos tratando de comunicarnos con el api de youtube', and a developer key 'AI39si6XDWyp1fDOaRkzZH8WhHIZM1eGqN5kddhYhF40VAduJYqvRcjjg7wLg5-GXxaz2YQxuUDIFvXPN6McBdeNjNs'. An 'Edit' button is also present.
- Product Reports:** A section with a 'Daily Series' tab selected, showing a line graph for 'Total', 'Errors', 'Uploads', and 'Playbacks' on July 14, 2010. The graph shows zero activity for all categories.

The browser's address bar shows the URL: <http://code.google.com/apis/youtube/dashboard/gwt/index.html#product/AI39si6XDWyp1fDOaRkzZH8WhHIZM1eGqN5kddhYhF40VAduJYqvRcjjg7wLg5-GXxaz2YQxuUDIFvXPN6McBdeNjNs>. The taskbar at the bottom shows several open applications, including 'Inicio', 'GONZALITO (H)', 'Developer's Gu...', 'GONZALITO (H)', 'directorio\_inter...', 'DispositivosSu...', 'Sustentacion', 'Product Dash...', and 'Documento1...'.

# 6.4 Módulo de Youtube

## 6.4.3 Detalle Técnico

- ▶ 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.

Librería	Versión
gdata-youtube	2.0
gdata-youtube-meta	2.0
gdata-core	1.0
gdata-media	1.0

Las funciones que utilizamos:

```
lista_videos=consultarVideo(usuario, contraseña)
YouTubeService service = new YouTubeService("gdataSample-YouTubeAuth-1", developerKey);
service.setUserCredentials(username, password);
lista_videos=printUploads(service);
printVideoFeed(service, http://gdata.youtube.com/feeds/api/users/default/uploads) //aquí obtenemos las entradas de videos, Luego cada entrada la enviamos a la siguiente función, pero esta nos retorna un arraylist de videosYoutube del usuario
printVideoEntry(VideoEntry entry) //en esta función nosotros cada entrada de video que nos carga la guardamos en una clase VideoYoutube la misma que tiene esta función llama a obtener id_video
```

# 7.- Aplicación en Funcionamiento

VIDEO

# 8.– Recomendaciones

- Planificar las actividades a realizar, establecer los recursos y tiempos de entregas en los avances, para evitar incumplimientos.
- Comprender los distintos protocolos 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.
- Alojarse la aplicación en un Servidor Web con excelentes características, para que los usuarios interactúen de forma rápida con la aplicación

# 9.- Conclusiones

- REST y SOAP permite la comunicación entre aplicaciones, en una red de computadoras
- REST:
  - Es un estilo de arquitectura para generar aplicaciones de cliente-servidor.
  - No es un estándar. No existe una especificación formal de REST.
  - Focalizado en Recursos
- SOAP :
  - SOAP es una especificación de protocolo para intercambiar datos entre dos extremos.
  - Es un estándar adoptado y definido por la W3C
  - Focalizado en Operaciones
- 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.
- Para la comunicación con el servicio web de la ESPOL utilizamos SOAP
- El API de la red social Hi5 es inestable y se debe revisar paulatinamente los cambios en sus espacios de nombres.
  - El envío del requerimiento y procesamiento del mensaje de respuesta (XML) lo realizamos sin utilizar ninguna librería facilitada por Hi5

# 9.- Conclusiones

- 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, Python, 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.



# GRACIAS POR SU ATENCIÓN

## ¿ PREGUNTAS ?