ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

"ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN MARCO DE TRABAJO PARA LA GENERACIÓN RÁPIDA DE APLICACIONES WEB MULTIMEDIA INTERACTIVAS"

TESIS DE GRADO

PREVIA A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS MULTIMEDIA

PRESENTADA POR:

DAVID FERNANDO CHANG VILLACRESES

CARLOS ANTONIO VILLAVICENCIO MOREIRA

GUAYAQUIL – ECUADOR

AÑO

2007

AGRADECIMIENTO

A nuestras familias, por su apoyo y soporte.

A Xavier, nuestro director que nos guió en el desarrollo de este trabajo.

A Dios, por ayudarnos a salir de la Universidad.

A los servicios web.

DEDICATORIA

A nuestros padres y familiares.

A la comunidad de desarrollo.

Al lector.

TRIBUNAL DE GRADUACIÓN

_____
Ing. Holger Cevallos Ulloa
PRESIDENTE

_____
Msc. Xavier Ochoa
DIRECTOR DE TESIS

_____
Ing. Carmen Vaca
VOCAL PRINCIPAL

_____
Ing. Guido Caicedo
VOCAL PRINCIPAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral".

(Reglamento de exámenes y títulos profesionales de la ESPOL)

David Fernando Chang Villacreses

Carlos Antonio Villavicencio Moreira

RESUMEN

Este trabajo tiene como objetivo crear un marco de trabajo que pueda ser usado para la generación de aplicaciones Web interactivas utilizando e integrando servicios Web existentes. Estos servicios Web permiten extraer información de distintas clases de medios. Herramientas similares a la que se propone diseñar no han sido explotadas en el ámbito del desarrollo web. Este marco de trabajo está orientado a ser utilizado por desarrolladores web y la intención es sembrar una semilla para la evolución de la comunidad informática.

Inicialmente se realiza una exploración inicial de los servicios web más comunes, los lenguajes de programación y scripting más comunes para la implementación de soluciones web, se explora el concepto de Ajax y Mashups, y las herramientas que están surgiendo para la creación de sitios dinámicos.

Luego se describen los servicios y fuentes de información relevantes en el desarrollo del marco de trabajo, clasificadas por tipos de medio y forma de extracción de la información de estos servicios. Se eligen las fuentes a incluir en el marco de trabajo y se justifica el uso de cada una de ellas.

Se analiza cómo se va a resolver el problema y todas las entidades que se relacionan con el marco de trabajo, y se propone el diseño de la solución, su modelo de funcionamiento, y el comportamiento dinámico del mismo. Se implementa el diseño antes mencionado utilizando los lenguajes de programación seleccionados para el

desarrollo óptimo del marco de trabajo, y se detalla la lógica y funcionamiento de cada una de las clases que componen el sistema.

Para exponer las capacidades de este marco de trabajo se implementa también una aplicación web multimedia interactiva, un buscador integrado que será sometido a pruebas, con las cuales se demuestre que se cumplen los objetivos propuestos inicialmente y beneficie la comunidad en línea.

ÍNDICE GENERAL

ABREVIATURAS

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CSS	Cascaded Style Sheets
GUI	Graphics User Interface
JSON	JavaScript Object Notation
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
PHP	Hypertext Preprocessor
REST	Representational State Transfer
SWF	ShockWave Flash
URL	Uniform Resource Locator
XML	eXtensible Markup Language

ÍNDICE DE FIGURAS

INTRODUCCIÓN

En la actualidad existen muchas soluciones web para las necesidades de la comunidad internauta. Desde que se creó el Internet, el mundo del desarrollo web ha sufrido muchos cambios e innovaciones.

Las aplicaciones que se encuentran en el Internet son cada vez más interesantes e innovadoras, cada vez se pueden encontrar tipos de servicios más creativos, y estos servicios manejan múltiples tipos de medios diferentes.

Sería entonces lógico plantear que mientras más se populariza el Internet, además de crearse un mayor número de capas de información y conocimiento, poco a poco puede ser expandida la recopilación del conocimiento humano, en muchas áreas y de muchas formas diferentes.

Los buscadores web, como el ya conocido Google, son los principales indexadores de toda esta vasta cantidad de conocimiento, y proveen diferentes métodos de búsqueda, clasificándolas por relevancia a los intereses de cada usuario, agrupando temas similares, y filtrando las búsquedas por el tipo de medio que los usuarios desean encontrar, sean éstos desde simples páginas web, hasta productos que estén a la venta en línea.

Gracias a estos indexadores, encontrar algo relacionado a lo que un usuario desee es más sencillo, pues existe la posibilidad de que más usuarios o sitios ofrezcan

información relevante, o que usuarios con intereses similares al del usuario que realiza la búsqueda hayan contribuido a alguna base de conocimiento.

A pesar de esto, es muy difícil encontrar sitios en donde se ofrezca información de diferentes tipos de medios, en otras palabras, usualmente los sitios están enfocados a un tipo de medio en particular, como búsqueda de videos, o búsqueda de imágenes.

La meta principal de este documento es crear un marco de trabajo, también conocido como framework, que permita correlacionar distintos medios, construir un primer pilar para crear sistemas que obtengan información ya existente, proveniente de los tantos sitios especializados.

El resultado final permitirá crear de manera más sencilla la parte informativa de las aplicaciones web que consulten los servicios en la red, y permitan a desarrolladores futuros tomar esta solución y crear capas subsecuentes sobre los formatos estandarizados provistos.

Esto servirá como semilla para la comunidad y le permitirá dar un paso adelante en la presentación al usuario final de información relevante para el mismo.

CAPITULO 1

Herramientas y Aplicaciones Multimedia Web

Introducción

La variedad de medios y de servicios en la red es impresionante. Teniendo sitios que combinen muchos tipos de medios y además servicios cada vez más interesantes y creativos, la variedad de lo que se puede encontrar en la red crece a pasos agigantados. Existen nuevas tecnologías que se están generalizando y ciertos formatos en los que es posible encontrar esta información. En este capítulo se verán todos estos conceptos y se echará un vistazo a las nuevas tendencias de aplicaciones en la red.

1.1 Tipos de medios en línea

Un medio es generalmente conocido como una forma de presentar un contenido. Existen muchos tipos de medios en línea, todos estos de libre acceso público, y su acceso es tan sencillo como el presionar un botón de "Buscar".

A continuación se describirán de manera breve los tipos más comunes de información que se puede encontrar en línea, para dar claridad a la utilización de los términos a utilizar en las secciones posteriores.

1.1.1 Texto

El tipo de medio más simple y popular de información en la red. La mayoría de las páginas web resultan finalmente en un texto legible para el usuario, y los motores de búsqueda se enfocaron inicialmente en encontrar búsquedas contextuales dentro del texto contenido en estas.

Existen tipos de codificación especial para el texto disponible, como el Unicode[1], que proveen soporte a todos los caracteres o símbolos de los alfabetos de idiomas no occidentales, como el griego o el japonés. Cada página web especifica el tipo de codificación en el que ésta se encuentra escrita. Por ejemplo, páginas provenientes de Asia que utilicen caracteres, o símbolos no occidentales propios de su alfabetización en su texto contenido, generalmente estarán en formato Unicode, y las páginas estadounidenses o latinas, que poseen caracteres romanos, se encontrarán en formatos más estandarizados, como UTF-8.

1.1.2 Imágenes

Formalmente, una imagen es un medio bidimensional completamente visual y auto descriptivo que nos provee una representación no tangible de temas, objetos o personas. Las imágenes son medios estáticos de

[1] Unicode Consortium. 1996. *The Unicode Standard.* Version 2.0. Reading: Addison-Wesley. ISBN 0-201-48345-9.

dimensión rectangular, que se encuentran embebidas en la mayoría de los sitios web en la actualidad. Éstas ayudan a explicar y soportar de mejor manera lo descrito en un texto, pues son auto descriptivas y existen varios tipos de formatos para las mismas. Las imágenes pueden encontrarse en distintos formatos. Los formatos más conocidos de imagen son JPG, GIF y PNG, debido a su alto nivel de compresión. La mayoría de los navegadores soportan visualizar estos formatos, incluyendo GIFs que contienen animaciones, o secuencias de imágenes estáticas.

1.1.3 Audio

Audio también puede ser encontrado en Internet, principalmente de dos formas, como streaming[2] o como archivos libres.

El audio digital es simplemente una señal que describe que tan alto o bajo reproducir un sonido o tono. Un sonido real es muestreado de a intervalos de tiempo para monitorear los cambios de intensidad de sonido, y luego reproducido en un parlante, para ser escuchado.

En internet es posible encontrar archivos libres de audio, esto quiere decir que son archivos previamente grabados, comprimidos en algún formato de audio, disponibles para descarga.

[2] Streaming es una forma de publicar contenido multimedia en la red, se reproduce el archivo mientras se lo descarga para ganar tiempo.

1.1.3.1 Códecs de Audio

Para escuchar un archivo de audio es necesario poseer los programas para decodificar la señal comprimida. Estos programas se conocen como decoders o decodificadores. Así mismo para crear un archivo de audio se deben poseer los archivos necesarios para crear un formato dado de compresión. Esta combinación de codificador y decodificador se conoce como *códec*.

Los formatos, de audio de mayor aceptación en el medio actualmente son el MP3, AC3 y WMA, principalmente por su alto nivel de compresión, y su popularidad, pues la mayoría de los usuarios posee los archivos necesarios para su reproducción de audio.

1.1.3.2 Streaming de Audio

Existen formas de escuchar audio directamente desde la red, pues gracias al aumento de la tecnología, el ancho de banda y velocidad de conexión de los usuarios finales ha aumentado de manera proporcional. De esta forma es posible recibir continuamente por la red una señal de un tipo de medio, y esta señal es reproducida en tiempo real en el equipo del usuario.

Este tipo de reproducción en tiempo real se conoce como *streaming*.

Muchos sitios proveen streaming de audio para escuchar noticias en tiempo real, o para mostrar extractos de archivos previos a ser descargados. El streaming de audio es también utilizado para servicios de mensajería o de conexión telefónica para comunicación por voz por internet con personas en otras partes del globo. Hoy en día es posible encontrar servicios de música en línea que permiten escuchar canciones de los intérpretes preferidos de un usuario por medio de streaming.

1.1.4 Video

Un video es básicamente una secuencia de imágenes, acompañado de una pista de audio sincronizada a las mismas.

Existen un sinnúmero de Códecs de video disponibles para su compresión, y gracias a los avances de la tecnología, la compresión de estos Códecs es cada vez mejor, lo que permite que los archivos de videos ocupen cada vez menos espacio conservando la misma calidad de imagen, y sean así más rápidos de descargar.

La mayoría de los videos encontrados en la red se encuentran en un formato de alta compresión, y una calidad regular de imagen, con la

finalidad de que su peso no sea exagerado y se lo pueda apreciar con una calidad aceptable. Los videos pueden ser encontrados en la red como archivos libres y como vínculos de streaming, al igual que los archivos de audio, todos estos archivos pueden ser encontrados en Google o Yahoo por medio de una búsqueda, y en servidores especializados con bases de datos de archivos de este tipo de medio.

1.1.4.1 Códecs de Video

A diferencia de los archivos de audio, existen muchos más Códecs de video en el medio, debido a que éstos están en constante mejora. Y si bien es cierto que existen formatos estándar, existen muchas variantes para cada uno, cada una con sus ventajas y desventajas. Debido a que existen tantos Códecs de video, los formatos en los que se puede encontrar un video en la red como un archivo libre pueden variar mucho. Pues cada video posee formatos de codificación de video y audio definidos por el usuario que lo codificó.

Las extensiones de archivo más populares de los archivos de video son de tipo AVI, MPG o WMV, no significando que existan Códecs definidos para cada tipo de extensión. Los Códecs más conocidos de video en la actualidad para descarga

son el DivX y Windows Media Video. No es muy relevante conocer todos los formatos disponibles de video, pues los reproductores generalmente ofrecen soporte para los formatos más conocidos.

1.1.4.2 Streaming de Video

Al igual que con el audio, existe en la red streaming de video. Dado a que el video es un tipo de medio más pesado que el audio, se requiere de un ancho de banda mayor para reproducir video en tiempo real. Existen en la red servicios que ofrecen a los usuarios facilidades para subir sus propios videos en archivos libres a la red, para ser compartidos y vistos como streaming, de manera muy similar a los repositorios de imágenes, estos repositorios de video ofrecen facilidades para que los usuarios compartan y editen las propiedades de estos videos en línea, asignándoles tags, comentarios y descripciones.

En la actualidad son muy populares los streaming de video, que utilizan la tecnología Macromedia Flash para comprimir los mismos y son transmitidos a la computadora del cliente a medida que se lo va viendo.

1.1.5 Animaciones

Inicialmente las animaciones eran conocidas como secuencias de imágenes estáticas. El formato de imagen GIF soporta esta tecnología, permitiendo visualizar imágenes que den la ilusión de movimiento. Con el tiempo las animaciones se han vuelto un medio más en la red. Y nuevos estándares como Flash permiten crear animaciones dinámicas que mezclan todos los medios descritos anteriormente en aplicaciones multimedia interactivas con los usuarios.

Estas nuevas animaciones pueden ser encontradas en la red embebidas en páginas web. La mayoría de las animaciones hechas en Flash se encuentran en el formato SWF, un tipo de archivo especial desarrollado por Macromedia, la compañía creadora de esta tecnología. Las páginas con texto creado con esta tecnología no pueden ser indexadas pues el formato SWF es un formato comprimido que no permite acceso a sus medios internos.

Estos archivos son visualizados y ejecutados en los navegadores de los usuarios y es necesario un plugin o complemento de Macromedia para su reproducción en un Browser.

1.1.6 Archivos de Descarga Directa

La mayoría de estos archivos se pueden reproducir una vez descargados al equipo del usuario con el programa adecuado, en su defecto en la red con el plugin necesario para cada Browser. Estos archivos pueden ser encontrados con parámetros de búsqueda avanzados en los buscadores más conocidos y pueden contener más de un tipo de medio de los descritos anteriormente.

Los archivos descargables pueden alojarse en múltiples servicios web y los archivos más comunes y descargados en la actualidad son los programas utilitarios descargables, archivos comprimidos, las animaciones web y documentos especiales. Todos estos archivos tienen extensiones especializadas para ser ejecutados por el programa para el cual son creados.

Existen sitios web que ofrecen hosting gratuito para almacenamiento de archivos en general, como Rapidshare, pero los archivos subidos a este tipo de servidores no son indexados por los buscadores pues en algunos casos, poseen fecha de expiración, y porque la mayoría de estos servicios no son orientados a compartir información, a no ser que el usuario que subió el archivo provea el vínculo.

1.1.6.1 Archivos de otros protocolos

Existen otros tipos de sitios que proveen vínculos a archivos de protocolos especiales, que si son orientados a compartir archivos, por medio de redes Peer to Peer, como BitTorrent y Gnutella. Estas redes se especializan en compartir archivos de diferentes usuarios, haciendo que cada usuario se comporte como servidor de archivos y cliente al mismo tiempo. Para compartir archivos con estas tecnologías es necesario un cliente creado especialmente para estos protocolos. Los buscadores como Yahoo proveen indexación a las páginas que dan vínculo a los archivos descargables por algunos de estos protocolos.

1.2 Formatos de datos en el Web

Es necesario tener estándares en los cuales la información debe de ser presentada, de lo contrario el desarrollo de cualquier aplicación sería un caos, pero tampoco puede existir un formato único, ya que cada uno es creado con el fin de satisfacer ciertos requerimientos y necesidades, tanto para el desarrollador, como para el computador.

1.2.1 Formatos parcialmente estructurados

Los formatos parcialmente estructurados, o semi-estructurados están orientados a ser leídos tanto por navegadores y lenguajes de programación, como por programadores, y llevan este nombre porque sus estructuras están separadas por encabezados o tags legibles, es decir, la mayoría de estos tags están en lenguaje natural, de manera organizada y jerárquica.

1.2.1.1 Formato HTML

El lenguaje de marcas para hipertexto es el formato más general y común en la red. Este formato es el utilizado por todos los sitios web y permite dar formato a las páginas que son visitadas por los navegadores. Una página web está compuesta por lo general de una cabecera y un cuerpo, dentro de los cuales se introduce código de formato HTML para dar estructura al documento, para embeber distintos medios y para proveer vínculos a otros sitios. Todo contenido debe estar dentro de tags para ser reconocido por el navegador, caso contrario, éste lo interpretará como texto ordinario del contenido del sitio. En un sitio HTML es posible embeber tags con lenguajes de

scripting para insertar funciones, y es posible además darle formato a un sitio por medio de hojas de estilo.

Existen formatos estándares simplemente para definir el diseño de una página web. El estándar más aceptado en este medio es el CSS, que consiste en un lenguaje simplificado, que utiliza medio de llaves para separar diferentes formatos, siendo éstos miembros de clases o poseen identificadores propios. De ésta forma, es posible cambiar todo el diseño de un sitio web con tan solo alterar la sección del CSS de la página por otra. Este formato utiliza palabras en inglés para describir el formato a tener de cada sección de la página, y posee ciertos comandos para manipular algunos eventos básicos como cambios dinámicos en una sección, o efectos del ratón.

1.2.2 Formatos estructurados

Los formatos estructurados son formatos descriptivos que nos permiten dar forma o estructura a un objeto cualquiera de manera estandarizada. Se los denomina estructurados, porque están orientados a ser interpretados por los lenguajes de programación, ahorrando caracteres al ser usados, es decir, minimizan el nivel de procesamiento en lo más posible. Dos de los formatos más utilizados para la web son el recientemente popularizado JSON y el XML.

1.2.2.1 Formato XML

El XML *(Extensible Markup Language)* es un formato de código web muy parecido al HTML al ser descrito con tags, pero en realidad el HTML es derivado del XML porque los tags usados son definidos, en cambio XML simplemente permite la representación de datos en dicho formato. Es también descriptivo, jerárquico y comprensible por los navegadores al detallar cada elemento con diferentes banderas y propiedades en lenguaje natural. Ajax, una tecnología para desarrollo web, se basa mucho en el formato XML para crear componentes y muchos tipos de comportamiento en las páginas web.

Un ejemplo de la descripción de los datos de una persona en XML puede ser el siguiente:

```
<?XML version="1.0" encoding="UTF-8"?>
<persona>
  <nombre>Juan</nombre>
  <apellido>Pérez</apellido>
  <direccion>
    <ciudad>Guayaquil, GYE</ciudad>
    <calles>Tulcan y Calicuchima</calles>
  </direccion>
```

```
  <telefonos>
    <telefono>04 2 123-456</telefono>
    <telefono>09 9 987-654</telefono>
  </telefonos>
</persona>
```

1.2.2.2 JSON

JSON (JavaScript Object Notation) es un formato de representación de contenidos u objetos, orientado a su fácil y rápida interpretación por una computadora, y para su ágil parseo[3].

Es sencillo de entender, y a diferencia de XML, está orientado a ser interpretado por un computador más rápidamente, pues se utilizan llaves para separar grupos de contenidos, y textos descriptivos para nombrarlos.

Un ejemplo de una descripción en JSON, basado en el ejemplo de XML anterior, sería:

```
{
    "nombre": "Juan",
    "apellido": "Perez",
```

[3] Parsing es el proceso de interpretar datos y aplicarles una gramática definida previamente para que el resultado final sean los mismos datos en otro formato.

```
    "direccion": {
        "ciudad": "Guayaquil, GYE",
        "calles": "Tulcán y Calicuchima"
    },
    "telefono": [
        "04 2 123-456",
        "09 9 987-654"
    ]
}
```

1.2.2.3 PHP Serializado

El formato de PHP Serializado es muy similar al JSON, y es un formato inherente de PHP, por ende, las librerías nativas proveen facilidades para decodificarlo de manera sencilla mediante la función unserialize[4]. En este formato se especifican los tipos de dato y longitudes de los valores de las variables, de acuerdo a la siguiente lista:

a - array

b - Boolean

d - Double

[4] Se puede obtener más información de la función unserialize en el manual online de php: http://www.php.net/unserialize

i - Integer

o - Common object

r - Reference

s - String

C - Custom object

O - Class

N - Null

R - Pointer reference

U - Unicode string

Un ejemplo de este formato, de manera análoga a los anteriores es:

```
a:4:{
        s:6:"nombre";         s:4:"Juan";
        s:8:"apellido";         s:5:"Perez";
        s:9:"direccion";
        a:2:{
          s:6:"ciudad";
          s:14:"Guayaquil, GYE";
          s:6:"calles";
          s:20:"Tulcan y Calicuchima";
         }
```

```
        s:8:"telefono";
            a:2:{
                i:0; s:12:"04 2 123-456";
                i:1; s:12:"09 9 987-654";
            }
}
```

Es importante notar que los elementos descritos en un arreglo también son descritos, de manera que dos elementos del arreglo representan un solo valor en el mismo.

1.2.2.4 Formato YAML

El formato YAML[5] es un formato surgiente, orientado a ser leído por seres humanos de manera natural, e interpretado por las computadoras de manera estructurada, por ende, muchos lenguajes de programación ofrecen soporte para YAML. Este formato antecede caracteres especiales a líneas de texto, y cada línea representa el valor de un parámetro, las estructuras están denotadas por indentación, y las secuencias denotadas por líneas. A continuación un ejemplo de este formato:

[5] http://www.yaml.org/

```
persona:
- nombre    : Juan
   apellido  : Pérez
  direccion :
        - ciudad : Guayaquil, GYE
          calles : Tulcan y Clicuchima
  telefono  :
        - 04 2 123-456
        - 09 9 987-654
```

1.3 Fuentes de Información en la red.

Se han descrito los diferentes medios disponibles en la red, ahora se explicarán en breves rasgos las diferentes fuentes de información en la red, las cuales pueden enfocarse a uno de los medios antes descritos, o incluso, en algunos casos, proveer combinaciones de varios medios.

1.3.1 Blogs

Una forma innovadora de mostrar contenidos y noticias en una página web, se conoce como Blog. El tipo de medio en el cual la mayoría de los blogs se enfocan es el texto, y es un sitio personal, en el cual uno o más usuarios pueden de manera sencilla poner diferentes artículos de cualquier índole. Los artículos, llamados posts, pueden ser desde un simple diario personal, hasta una noticia que llame la atención,

pudiendo cada post contener medios distintos, como fotos, sonidos e incluso videos. Cada usuario contribuye de forma única a la red de información, y es indexado a su vez por los buscadores. Los posts de cada blog son opcionalmente marcados con tags, o identificadores, provistos por el usuario que crea el post, lo cual facilita aún más la búsqueda entre los mismos. Estos posts además tienen fechas de publicación y de modificación, las cuales son guardadas cada vez que se crea o modifica un post, y pueden ser comentados por los visitantes del blog.

Existen varios servicios que facilitan al usuario poco experimentado a crear un blog, a tal punto que no es necesario saber absolutamente nada de código web para crear un sitio propio. Dos ejemplos de servicios como éstos incluyen Blogger[6] y Windows Live Spaces[7], de las dos más grandes empresas de software en la red, como lo son Google y Microsoft.

1.3.1.1 Blogs Colaborativos

Existen nuevos servicios de blogging que se están popularizando de manera rápida, éstos son conocidos como

[6] http://www.blogger.com

[7] http://spaces.live.com

blogs colaborativos. A pesar de que los servicios convencionales de hecho ofrecen soporte para que más de un usuario sea parte de los colaboradores de un sitio, recientemente han surgido servicios orientados a publicar todos los posts en un sólo sitio general, atribuyéndose de los valores antes vistos, como fecha de publicación, o tags relacionados, un número de visitas, es decir, cada post tiene entre sus propiedades cuántos usuarios lo han leído, de ésta forma es posible filtrar los posts más interesantes, si el visitante del sitio así lo desee. Ejemplos de estos innovadores blogs que combinan sindicación y democracia, son servicios como Digg[8] y Reddit.

1.3.2 Enciclopedias

Otro tipo de servicio que se encuentra en la comunidad internauta, son las enciclopedias en línea. La mayoría de las enciclopedias son sitios pagados por compañías que, de manera similar a las enciclopedias encontradas en cualquier biblioteca, proveen información fidedigna de algún tema en específico. Las enciclopedias pagadas han perdido poco a poco su popularidad como fuente de información en línea, a medida que sitios

[8] http://www.digg.com

libres y páginas especializadas proveían información detallada, pero menos confiable, de temas de interés para los usuarios. Pues los servicios de indexación y búsqueda como Google o Yahoo proveían estos sitios como resultados relevantes. Algunas de estas enciclopedias requieren suscripción para acceder a sus bases de conocimiento.

1.3.2.1 Enciclopedias Colaborativas, Wikis

Debido a la necesidad de información fidedigna para los usuarios y la usual falta de formalidad de la mayoría de los artículos encontrados en la web, nacieron las enciclopedias colaborativas, mayormente conocidas como Wikis [9] . La característica principal de estos servicios, es que cualquier usuario, sin necesidad de registro alguno, puede eliminar, editar o publicar nuevos artículos a la base de conocimientos de un Wiki. Las modificaciones realizadas son monitoreadas por administradores, y éstos aprueban, niegan o sugieren cambios para las modificaciones en la información.

Las ventajas de los wikis sobre las enciclopedias son grandes, desde la facilidad y rapidez de su crecimiento, dado a que los

[9] http://en.wikipedia.org/wiki/Wiki

editores son los mismos internautas, hasta la exactitud de los artículos encontrados en el medio. La desventaja principal de estas fuentes de información, es que al estar sujetas a cambios de cualquier usuario, los artículos pueden ser víctimas de vandalismo. La enciclopedia colaborativa más popular en la actualidad es Wikipedia.

1.3.3 Feeds

Algunos sitios, como los blogs, además de ser encontrados en formato HTML, generalmente poseen tipos de sindicación para resumir sus contenidos y separarlos en artículos, algunos formatos estandarizados pueden ser el RSS o el ATOM. Es posible suscribirse a los contenidos de estos formatos sindicados, conocidos como Feeds de un Blog, de manera que un usuario pueda recibir actualizaciones en tiempo real de las publicaciones de un blog o modificaciones de un sitio web. Más adelante en este capítulo se explicarán en detalle estos formatos de sindicación.

1.3.4 Información geográfica y mapas

En la red es posible además, realizar búsquedas de localidades físicas. Estos servicios están popularizándose debido a su utilidad en la realidad, a tal punto que la mayoría de aplicaciones web que mezclan servicios, se enfocan a información geográfica. Un punto geográfico, o geoPoint, indica una ubicación en un mundo real de manera descriptiva, es decir, contiene la dirección e información de longitud y latitud necesaria para ubicar un hito en el globo terráqueo. Algunos geoPoints incluso pueden tener un pequeño código web descriptivo. Servicios Web populares de búsqueda de geoPoints proveen mapas interactivos para realizar las búsquedas de éstos hitos.

1.3.5 Repositorios de Medios

Algunos de los servicios en auge proveen repositorios de distintos tipos de medios, esto quiere decir, que existen servicios que brindan espacios en línea para que un usuario pueda alojar sus archivos en línea, y en algunos casos, compartirlos. Algunos de estos repositorios incluso proveen facilidades para los usuarios de crear fuentes de constante actualización con sus medios usando feeds.

Sabiendo acerca de los repositorios y feeds orientados mayormente a texto, a continuación se describen los tipos más conocidos de repositorios,

para los otros medios. Y de acuerdo al tipo de medio, como son conocidos estos servicios.

1.3.5.1 Flogs y Photofeeds

Las imágenes pueden ser encontradas, además de en páginas simples, en los ya mencionados servicios de repositorios de imágenes. Estos repositorios sirven de almacenamiento en línea para usuarios que deseen respaldar o compartir sus imágenes. Existen diferentes tipos de repositorios, y la mayoría proveen formas para ver las imágenes de un usuario a manera de galería, y de buscar imágenes relacionadas a un tema en específico. Algunos repositorios están orientados a respaldar álbumes de fotografías, y algunos están orientados a compartir las imágenes de un usuario. Servicios como estos generalmente requieren un registro previo de manera gratuita. Los servicios más populares para almacenamiento de imágenes en la red son Flickr, para almacenar fotografías, Picasa Web Albums para publicar álbumes de fotografías de manera más privada, y DeviantArt, que permite subir imágenes de dibujos, arte o fotografía de los usuarios registrados. En el Capítulo 2 se revisará un poco más a estos servicios.

Existen repositorios de imágenes que además de cumplir la función de almacenarlas y visualizarlas a manera de galería, proveen servicios de posting o publicación, y permiten a los usuarios añadir comentarios a las imágenes publicadas. Estos servicios, al comportarse como Blogs de Fotografías, son conocidos como Flogs. Existen sitios que ofrecen facilidades para sindicar las imágenes publicadas en feeds que vinculen directamente a las imágenes, estos sitios se conocen como Photofeeds.

1.3.5.2 Audio Podcast

Así como el caso de los Flogs, o blogs de imágenes, también existen sitios web que ofrecen artículos en formato sindicado, como RSS, en los cuales hay información necesaria que vincula a archivos de audio, sean estos streaming o archivos libres. Estos archivos son conocidos como Podcasts y requieren suscripción para ser escuchados. El podcast de Audio es uno de los medios más populares para distribuir noticias, y existe un auge en el número de usuarios que prefieren crear feeds de audio podcasts en lugar de mantener un Blog de texto.

1.3.5.3 Video Podcast y Repositorios de Video

Existen servicios gratuitos en la red, que además de ofrecer almacenamiento a manera de repositorio, permiten a los usuarios crear blogs con sus propios videos, estos pueden ser tanto por streaming o por archivos libres. Estos tipos de sitios son conocidos como Video logs o Vlogs, como son conocidos de manera abreviada. Así mismo, si estos videos son provistos en un formato sindicado, de manera que permitan que un equipo suscrito se los descargue automáticamente, estos Video logs son conocidos como Video Podcasts. Uno de los servicios más populares que permiten a los usuarios subir videos para ser visualizados se llama YouTube. Este servicio será descrito más adelante en este capítulo.

Cabe recalcar, que existen nuevas tecnologías que permiten publicar videos de diferentes fuentes, combinando incluso blogs colaborativos con podcasts de video, dando posibilidad a los usuarios de dar su criterio sobre los videos publicados por otros. Un ejemplo poco conocido en la red, pero bastante interesante por su tecnología, de estos nuevos servicios se llama Videosift[10].

[10] http://www.videosift.com

1.4 Formas de Obtener Información en la red

Existen varias formas de obtener información de los sitios web cuyo contenido es predominantemente textual. Es posible obtener documentos estructurados, o simplemente clasificarlos por su contenido. Entre estas tecnologías, dos de las más usadas son web crawling y scraping, provistas por distintos servicios web. A continuación se explicarán éstos términos, para comprensión del lector a futuro.

1.4.1 Servicios Web

A groso modo, un servicio web, como su nombre lo indica, provee utilidades para que dos equipos se comuniquen a través de la red.

Todos los medios antes vistos son alojados en servidores distintos, y como se pudo apreciar, algunos de estos sitios proveen servicios web para organizarlos e incluso para compartirlos. Independientemente de la forma de desarrollo, un servicio web debe proveer facilidad para que cualquier cliente pueda obtener la información que necesita. Los servicios más utilizados por los internautas son los de clasificación de información y de búsqueda.

1.4.1.1 Estándares de comunicación de los servicios web

Los servicios web muchas veces proveen un API de desarrollo para el uso de programadores, para que éstos interactúen con la información provista por los mismos. Los APIs son un conjunto de herramientas para la utilización de la información de un servicio en particular. La mayoría de los servicios de búsqueda en la red proveen APIs para consultar la información en sus bases de datos. Los APIs de búsqueda son en su mayoría abiertos, es decir, que pueden ser utilizados para el desarrollo sin necesidad de pagar por su utilización. Los APIs de las compañías más visitadas por lo general requieren de un registro gratuito para su uso. Existen APIs para desarrollo en diferentes lenguajes de programación, y estos lenguajes se comunican usando algunos formatos especiales.

1.4.1.1.1 XML-RPC

XML-RPC son las siglas de XML Remote Procedure Call. En otras palabras, es un protocolo que utiliza XML para realizar llamadas de transporte de datos entre servicios web. Lo importante de este servicio es que define algunos tipos de datos, y a diferencia de

metodologías RPC antiguas, que necesitaban de grandes cantidades de información para configurar un intercambio de datos, XML-RPC permite hacer en muchas menos páginas de código que los otros formatos antiguos. Utiliza el lenguaje descriptivo de XML para crear descripciones de métodos y parámetros para funciones. XML-RPC está siendo reemplazado poco a poco por nuevas tecnologías basadas en esta.

1.4.1.1.2 SOAP

Simple Object Access Protocol, es el sucesor de XML-RPC, es decir, que permite enviar pedidos a un servidor, normalmente por HTTP, el protocolo usado por los navegadores para acceso web, y recibir una respuesta inmediata del mismo. Generalmente, los servicios web que se comunican por SOAP proveen como respuesta, sus resultados de búsqueda en formatos estandarizados, como XML. SOAP está basado en XML-RPC, pero combina otras pequeñas diferencias en su sintaxis que provienen de otros estándares.

1.4.1.1.3 REST

REST, a diferencia de los protocolos anteriores, enfocados a llamadas de procedimientos remotos, o funciones, se enfoca a llamadas de direcciones web con funcionalidades específicas. De esta forma, al enviar una petición REST, en vez de enviar una lista de funciones con parámetros, se envían direcciones de sitios web, cuyos posibles directorios son las funciones respectivas. A diferencia de RPC, si es necesario procesar varios parámetros, es necesario enviarlos todos por separado, en vez de dentro de un solo pedido.

1.4.1.1.4 Otros estándares

Otros estándares proveen resultados en algunos de los formatos estructurados antes mencionados, adicionalmente a los estándares vistos en esta sección, como JSON-RPC. Su funcionamiento es muy similar a los descritos anteriormente, pero su sintaxis es distinta, por lo cual es posible realizar ciertas operaciones adicionales, como llamadas bidimensionales entre cliente y servidor, entre otras.

1.4.1.2 Servicios de Indexación y Web Crawling

Muchas compañías se han especializado en desarrollar servicios para suplir la mayoría de las necesidades de la comunidad. Normalmente todo se inició como búsquedas simples por toda la red, pero ahora se han segmentado por el tipo de medio, básicamente: Información textual, que puede ser tan didáctica como una enciclopedia o solo para enterarse de las más recientes noticias y pueden provenir de fuentes oficiales o de fanáticos adictos a la información; imágenes, las cuales serían fotografías de personas, lugares o hechos, diseños digitales o arte; videos y audio que nunca dejan de llamar la atención en toda la red. Como ya se ha tratado anteriormente, las compañías más visitadas en la red son Google y Yahoo.

Los servicios web para extracción de información de empresas grandes, como Google y Yahoo, utilizan web crawlers. Éstos son bots, o programas automatizados en línea, que navegan por la web de manera constante, y van visitando los sitios web examinando su contenido en alguna base de datos. Cada vez que encuentran un vínculo en un sitio, los web crawlers lo añaden a su lista de sitios por visitar, y así continúa el ciclo. Es de esta forma como algunos indexadores como Google obtienen la información textual de las páginas.

Los resultados obtenidos son clasificados, generalmente por categorías, y separados con palabras claves para su indexación y acceso futuro. Luego de ser categorizados son almacenados en los cachés de búsqueda de los servicios de búsqueda respectivos.

Existen servicios web que permiten generar imágenes de vistas previas de páginas web, para mejorar aún más la experiencia del usuario en la búsqueda. Algunos Web Crawlers como Alexa o Exalead, realizan capturas al recorrer los sitios, y de esta manera son capaces de proveer vistas previas de los sitios web de manera conjunta con los resultados de búsqueda.

Existen scripts que permiten generar estas previstas en tiempo real, y existen servicios, la mayoría de estos pagados, que ofrecen solamente previstas de sitios web en diferentes resoluciones. Los servicios gratuitos más conocidos en este último caso son ThumbShots y Girafa[11]. Es posible además, realizar crawling de los servicios de búsquedas. Es decir pueden existir web crawlers de web crawlers. Estos sitios se encargan de generar búsquedas a los servidores de los indexadores constantemente, y almacenan sus resultados en sus propias bases de datos, permitiéndoles aprovecharse de los

[11] http://www.girafa.com http://www.thumbshots.com

resultados de los servidores de búsqueda populares y generar resultados combinados. A este tipo de servicio de búsqueda se lo conoce como *meta crawling*. Las ventajas de este tipo de crawlers son lógicamente, el poder apreciar los resultados encontrados en varios motores de búsqueda en un mismo sitio. A pesar de esto, como los meta crawlers realizan las búsquedas finales en sus respectivos caches, los cuales a su vez se basan en resultados del cache de los otros servicios de búsqueda, los finalmente devueltos por los meta crawlers pueden llegar a ser un poco viejos. Metacrawler[12] es el exponente más alto de este tipo de servicios.

Algunos servicios web proveen búsquedas de muchos tipos de medios, como imágenes y videos, aunque su enfoque es mayoritariamente en el texto de las páginas web. Algunos servicios web incluso proveen otros recursos multimedia interactivos para el usuario, como lo son los mapas globales, estos servicios incluso permiten a un usuario buscar una dirección física y encontrarla en un mapa en la red, o buscar hitos como tiendas o centros comerciales en los mapas interactivos, mostrando una imagen satelital o un mapa vial. Es

[12] http://www.metacrawler.com/

posible marcar los sitios de interés, de manera similar a como se lo haría en un mapa real.

1.4.2 Scraping

El scraping es una tecnología, orientada más a obtener información actualizada y estructurada de un sitio web. El scraping puede ser aplicado a cualquier tipo de sitio. El scraping consiste simplemente en identificar el formato de diseño, o una estructura jerárquica en el código HTML de un sitio web, y de acuerdo a éste generar una estructura definida.

Esta tecnología puede ser usada incluso para reconocer sitios dinámicos, al recorrer dos instancias de una misma página, pero con diferentes artículos, de manera que es posible reconocer que partes de un sitio cambian de un mismo formato. Uno de los servicios más inteligentes de scraping de la actualidad es Dapper[13], que nos permite incluso generar feeds RSS, y estructuras de sitios web en algunos de los formatos antes mencionados como JSON o XML.

Una restricción de las tecnologías de scraping es que es necesario conocer que representa cada elemento en una página dada, es decir, se necesita conocer el formato de la página previamente, por ejemplo, en un resultado de búsqueda de Google es necesario saber que parte del contenido es un

[13] http://www.dappit.com

resumen de un resultado o de otro. Esto puede ser logrado con la ayuda de un usuario que pueda identificar cada uno de los campos, o con un sistema de reconocimiento automatizado para sitios con formatos ya conocidos.

1.4.3 Sindicación

Existen además, formatos de sindicación para obtener información de manera estructurada de algunos sitios web. Anteriormente, se mencionó que los blogs siguen formatos específicos de sindicación, a los cuales es posible suscribirse. Todo esto de manera transparente para los usuarios. Un estándar muy popular para la información publicada en un blog o sitio en general es el RSS, que son las siglas para *"Really Simple Syndication*", otro estándar es el menos conocido ATOM, que nació de una versión previa del RSS actual, y que también es usado hoy en día por algunos sitios.

1.4.3.1 Serialización RSS y ATOM

Un feed es una descripción de una página en un formato simplificado reconocido por los navegadores, que contienen tags o banderas en XML para indicar el inicio o fin de un post, el título del feed, y más información relevante del blog o página.

Como se mencionó anteriormente RSS y ATOM son dos estándares diferentes para la sindicación, aunque ambos mantienen una estructura derivada del XML. A continuación un pequeño ejemplo de un archivo RSS versión 2.0 de un blog:

```
<?xml version="1.0" encoding="utf-8"?>
  <rss version="2.0">
   <channel>
    <title>Titulo del Blog</title>
    <link>http://www.blog.com/r2/</link>
    <description>Descripcion</description>
    <language>en-us</language>

    <item>
     <title>Titulo de un post</title>
       <link>http://www.blog.com/1</link>
       <description>Post</description>
       <pubDate>Fri,05Jan2007
5:00</pubDate>
    </item>

    <item>
           ...
    </item>

   </channel>
  </rss>
```

De esta forma, cada usuario con un blog de forma transparente está publicándose a sí mismo de una manera fácil de visualizar y de interpretar, es decir, se convierte en una fuente más de información. El blogger, o dueño de un blog, está creando así una base para poder obtener información de su sitio, y lo cual permite indexar su información, para ser buscada en la red, y también para ser, en un momento dado, relacionada con otros sitios similares.

1.5 Integración de la información (Mashups)

Dado que existen muchos APIs de desarrollo, muchos programadores web han optado por desarrollar aplicaciones que mezclen diferentes APIs, para crear nuevas aplicaciones o servicios completos, algunos muy innovadores y otros muy prácticos. El resultado de estas mezclas de dos o más APIs de desarrollo es conocido como un Mashup.

1.5.1 Mashup

Conceptualizando la idea anterior, los Mashup son sitios web que han combinado contenido y funcionalidad de otros sitios mayores, los cuales proveen públicamente sus APIs. Con estas herramientas es posible

mostrar información de manera inteligente de sitios que pueden ser rivales como Google y Yahoo, pero de manera transparente para el usuario final, quien después de todo es el mayor beneficiado, aunque muchas veces no es capaz de confiar en la capacidad de un Mashup.

Al inicio se fueron integrando pocas APIs, como ejemplo se puede citar una búsqueda de imágenes de un servicio de repositorio por medio de un mapa mundial de Google Maps, pero cada vez se unen más APIs o cualquier otro medio como RSS feeds, con el fin de obtener la mayor cantidad de información posible y siempre innovando para no caer en monotonía. Existen sitios que ofrecen alojamiento e incluso bases de datos enteras para que los desarrolladores ingresen sus proyectos que mezclan diferentes APIs.

1.5.2 Tipos de Mashups

Los sitios web están dejando de ser un conjunto de páginas estáticas para mutar en casi sistemas operativos, no sólo mostrando texto e imágenes sino otros medios como videos, audio, gráficos, etc. Con todas estas nuevas tecnologías, la red ofrece servicios y sitios cada vez más interesantes. Gracias a la facilidad de desarrollo de los APIs de los distintos servicios, los mashups son cada vez más creativos y útiles.

Es posible encontrar sitios completamente dinámicos que ofrecen los medios vistos anteriormente, combinando animaciones, streaming incluso hasta de manera colaborativa, es posible encontrar páginas orientadas a mezclar todos los servicios de interés para un usuario y buscadores integrados. La mayoría de los servicios antes mencionados utilizan Ajax y la mayoría están en etapa de pruebas, o como se suele decir para el software, están en fase beta, Ajax se explicará en la sección siguiente de éste capítulo. A continuación se verán los principales tipos de mashups que se encuentran en la red.

1.5.2.1 Homepages

Tal como su nombre lo dice, son páginas web donde el usuario puede iniciar su navegación por la red, realizando las funciones más comunes y necesarias como revisar su correo, enterarse de las últimas noticias de fuentes oficiales o blogs y realizar búsquedas de medios en la web. Muchas compañías han apostado en desarrollar sus páginas de inicio, Google, Yahoo y Microsoft (Microsoft Network, ahora conocido como Live) quieren inducir a sus usuarios de correo electrónico y búsquedas a utilizar sus homepages.

Algunos ejemplos que ilustran este tipo de aplicaciones son Netvibes, y Yourminis; son páginas de inicio donde cada usuario puede crear y modificar su aspecto a placer, permitiéndole al usuario revisar correos, feeds y buscar en la red información en los medios más populares, sin dejar pasar por alto que desarrolladores ajenos a los creadores aportan a la comunidad ofreciendo módulos o plugins para mejorar la experiencia en la navegación.

1.5.2.2 Buscadores Integrados

Además de soluciones cada día más populares, existen soluciones innovadoras y tal vez nunca antes vistas, como el buscador de Quintura, cuyo estilo de interacción en cuanto a la búsqueda se ve innovado al estilo actual, al basarse en una metáfora completamente nueva, la nube de ideas, es decir, a medida que se busca un elemento, Quintura se adapta al contexto de cada búsqueda y muestra previstas y más palabras relacionadas en la nube, para refinar la búsqueda.

Cabe recalcar, que utilizan entornos de desarrollo o Frameworks de uso libre, de otras compañías para implementar sus aplicaciones, por ejemplo, tanto Netvibes como Quintura

utilizan los motores de búsqueda de Google y de Yahoo para encontrar resultados en la red.

1.5.2.3 Servidores de objetos embebibles

Encontrar medios embebibles, listos para insertar en un código web, como videos e incluso galerías de imágenes, es común en la red. Algunos de estos provistos por los mismos servicios web. Gracias a la facilidad del scripting remoto, es decir, que un script puede ser llamado de otro sitio fuera de una página, existen aplicaciones enteras embebibles, como diálogos para comunicación instantánea y videos con todos sus controles. Un usuario común con conocimientos básicos de HTML puede de manera sencilla incluso embeber un reproductor en su sitio y armar una página con todos los medios descritos anteriormente, gracias a los innovadores servicios que día a día surgen en el mercado.

1.5.2.4 Servicios Folksonómicos

Es común también, que la mayoría de los sitios, en general los repositorios y buscadores orientados a medios no textuales,

clasifiquen la información en sus servidores con tags, además de buscarlos por descripción o título.

Un tag folksonómico es simplemente una palabra que se relaciona contextualmente a un medio. El tag permite al buscador clasificar de mejor manera los medios y facilita su búsqueda en una gran base de información. Los tags son añadidos por los usuarios que suben los medios a los servicios, así que es posible también encontrar medios con tags que no estén relacionados a los mismos, si el usuario que los sube a los servidores así lo desea.

1.5.2.5 Comunidades Virtuales

Como la vida en la red es cada vez más común, y los sitios están orientados a compartir información, están surgiendo servicios de comunidades virtuales. Estos servicios permiten a los usuarios conectarse unos con otros por medio de relaciones grupales, desde amistades a relaciones familiares. Muchos de los servicios orientados a comunidades proveen facilidades para blogging e incluso para subir archivos de otros tipos de medios, como audio, imágenes o videos.

Los servicios más reconocidos de repositorios de medios también permiten asociarse por medio de relaciones, pues el mercado los obliga a estar a la par con otros servicios web.

Las comunidades virtuales también permiten realizar búsquedas, pero de perfiles personales, es decir, que a medida que el Internet sigue expandiéndose, más y más datos personales son migrados a la red.

1.5.2.6 Otros tipos de mashups populares

Existen otros mashups orientados a funcionalidades específicas, algunos proveen simples funcionalidades sólo para probar el uso de los APIs de desarrollo disponibles, y otros integran nuevos comportamientos a servicios ya existentes.

Generalmente, los mashups se enfocan en el uso de mapas, pero sus funcionalidades añadidas a éstos son mayormente específicas y limitadas. Es posible, por ejemplo, encontrar direcciones de las páginas amarillas con el uso de mapas, la integración de directorios de contactos con direcciones predefinidas o hasta asociar datos policíacos con información geográfica o direcciones de localidades de ciertas ciudades para

obtener el nivel delincuencial en cierta zona de una ciudad o país.

El uso de Calendarios también es común en los mashups. Encontrar soluciones creativas que asocian calendarios personales de los usuarios para generar bases y recordatorios en tiempo real es casi común en algunos homepages ejecutivos. E incluso existen los conocidos calendarios de fechas importantes como cumpleaños que proveen recordatorios a los usuarios. Otras categorías importantes incluyen mensajería instantánea, o chat, integración con servicios telefónicos, realización de compras en línea y búsquedas por términos específicos o relacionados.

Existen muchas más tendencias apareciendo en la red, pero para propósitos de este documento, son éstos términos los que los permitirán explicar de mejor manera la implementación del framework en los capítulos siguientes.

1.6 Tecnologías para el desarrollo de Mashups

Existen muchas tecnologías disponibles para poder desarrollar un Mashup, tomando en consideración la facilidad de desarrollo, y curva de aprendizaje recorrida. Existen lenguajes de scripting, bases de datos y tecnologías

innovadoras que pueden ser utilizadas para crear soluciones escalables y estables. Se describirán a continuación las posibles herramientas a utilizar, describiendo las ventajas y desventajas de cada una.

1.6.1 Lenguajes de Programación

El scripting nos permite embeber códigos y funciones en páginas web, para crear comportamientos especiales en las páginas web. Existen muchos lenguajes de scripting disponibles y soportados por los diferentes Browsers, a continuación los más comunes.

1.6.1.1 JavaScript

JavaScript es uno de los lenguajes de scripting más usados en la red orientado a desarrolladores que están acostumbrados a la sintaxis Java, que a su vez está basada en la sintaxis de C++. Éste lenguaje era utilizado para validaciones sencillas en los sitios web, pero ahora su uso se ha popularizado a raíz del uso de Ajax como modelo de programación.

Casi todos los Browsers pueden interpretar códigos JavaScript embebidos en un HTML. JavaScript provee funcionalidades

para añadir texto en un documento HTML, e incluso para mostrar cuadros de diálogo en los navegadores.

Un ejemplo de inclusión de un simple código JavaScript que muestra un texto en una página se aprecia a seguir:

```
<script type="text/javascript">
document.write(" Hola! ");
</script>
```

JavaScript se ejecuta en el Browser al mismo tiempo que las sentencias se van descargando del servidor. JavaScript también puede ser incluido aparte sólo como un archivo de código fuente, este archivo debe llevar extensión .JS. JavaScript no provee soporte para herencia, pues está basado en prototipos, que sirven como plantillas.

1.6.1.2 JSP

JavaServer Pages es una tecnología que se ejecuta en el lado del servidor obedeciendo peticiones de un cliente. A diferencia de JavaScript, todo JSP debe ser antes compilado, por lo cual no puede ser considerado un lenguaje de scripting por

completo.

JSP utiliza funciones de Java y cada instrucción debe estar dentro de un tag, a diferencia de JavaScript.

La ventaja de JSP sobre JavaScript es que la mayoría de los Browsers pueden deshabilitar JavaScript, pero es imposible deshabilitar JSP al ser éste de más bajo nivel y ser ejecutado remotamente.

A continuación un ejemplo de un tag en JSP.

```
<%
out.println(" Hola! ");
%>
```

Existen soluciones mucho más sencillas a este lenguaje, en cuanto a scripting se refiere. Uno de los lenguajes de scripting mayormente aceptados es PHP.

1.6.1.3 PHP

Un popular lenguaje de programación para la creación de aplicaciones cliente-servidor [14] para la web. Provee ciertas facilidades adicionales orientadas al desarrollo, como funciones de parsing o interpretación de formatos externos. Es un lenguaje de programación reflectivo, es decir, se puede editar el código mientras se lleva a cabo la ejecución del mismo, y no es necesario declarar el tipo de una variable para su uso.

Al ser un lenguaje para aplicaciones cliente-servidor, es necesario que en el servidor esté instalado el Apache Server y ejecutándose.

Un ejemplo de la sintaxis del lenguaje se presenta a continuación, similar al ejemplo de JSP.

```
<?php
echo ' Hola! ';
?>
```

[14] En las aplicaciones cliente-servidor, el cliente (computadora de un usuario del sistema) ejecuta código almacenado en el servidor, retornando una respuesta hacia el cliente donde nuevamente se inicia el ciclo.

El formato PHP serializado es la forma más sencilla de obtener información de la red para este lenguaje, pues existen funciones que permiten importar por completo al lenguaje estos formatos web antes vistos. Existen además, como se comentó previamente, funciones de interpretación web de formatos externos a éste lenguaje como JSON y XML

PHP siempre está en constante actualización, y Zend [15] Enterprise, su compañía representante, provee soporte gratuito y aplicaciones de desarrollo para este lenguaje.

Las versiones más recientes de PHP soportan el Unicode por completo, a diferencia de muchos otros lenguajes de scripting. Los archivos de código de PHP llevan la extensión del nombre del lenguaje: ".php".

1.6.1.4 ASP

ASP es la tecnología de scripting que provee Microsoft. De manera similar a PHP y JSP, ASP puede estar embebido en un

[15] http://www.zend.com

HTML con tags especiales. ASP es también soportado por la mayoría de navegadores.

Aquí un ejemplo que efectúa lo mismo que los scripts previos.

```
<%
Response.Write(" Hola! ")
%>
```

Generalmente, ASP está escrito en lenguajes activos estandarizados, como VBScript, que utiliza la sintaxis de Visual Basic.

Es posible definir el lenguaje a utilizar con ciertas directivas adicionales, y es posible instalar plugins de terceros para utilizar otros lenguajes como Perl y Ruby.

1.6.1.5 Perl / CGI

Perl es otro lenguaje de programación, que se ha popularizado en la red.

Perl es un lenguaje de scripting que está popularizándose cada vez más en la red. No es un lenguaje soportado por todos los

navegadores, pero su facilidad de uso le da ciertas ventajas sobre otros lenguajes.

Todo en Perl es un objeto, y al igual que PHP, no es necesario declarar un tipo de variable.

A diferencia de los otros lenguajes de scripting. Los archivos de código Perl sólo pueden ser usados externamente, siempre extensión CGI o PL, o embebido dentro de otro lenguaje distinto de scripting, es decir, Perl no tiene tags HTML propios.

A continuación un extracto de código Perl en un archivo aparte.

```
print " Hola! ";
```

Existen alternativas a Perl, una de las alternativas más populares se llama Ruby.

1.6.1.6 Ruby

La sintaxis de Ruby es similar a las de Perl y Python, otros lenguajes de programación orientados a escritorio. Ruby nació en manos de un desarrollador en Japón y provee maneras

inteligentes de mostrar la información, para aumentar la productividad del código, pero es un poco más pesado que los otros lenguajes. Al igual que Perl, Ruby no puede ser ejecutado embebido en un HTML, sino en archivos CGI, pero existen filtros que permiten su ejecución. Esto se conoce como eRuby, y debe ser invocado por el navegador previo a su uso.

A continuación el mismo ejemplo visto en los dos otros lenguajes utilizando un filtro eRuby que identifica los tags de ASP como de Ruby:

```
<%
puts ' Hola! '
%>
```

Ruby no provee soporte para Unicode, solamente para UTF-8. Pero su facilidad de aprendizaje poco a poco ha popularizado su utilización.

1.6.2 Bases de Datos

De manera similar a los lenguajes de scripting, existen bases de datos populares para desarrollo de sistemas web. Una de las bases de datos con

mayor soporte es MySQL, debido a que es gratuita y soporta muchos lenguajes de scripting, entre los cuales están los lenguajes descritos anteriormente. Para ejecutar bases de datos es necesario alojarlas en un servidor, o en un servicio que soporte el tipo de base a utilizar. A continuación, se tratarán tres de los sistemas más populares para el manejo de bases de datos.

1.6.2.1 MySQL

MySQL es un motor de base de datos, multiplataforma, que se ha popularizado en los últimos años con el desarrollo de sitios web. MySQL es soportado por una larga lista de lenguajes de programación, incluyendo PHP. Es de fácil uso, y de fácil administración, posee extensivas fuentes de documentación, tanto oficiales como no oficiales, a tal punto de que se han publicado varios libros para su aprendizaje por desarrolladores. Al ser gratuita es la opción más aceptada para su uso en la red.

1.6.2.2 Oracle

Oracle es una colección de herramientas de manejo, control y edición de bases de datos. Permite al usuario generar bases de datos complejas y manejarlas de manera gráfica. Oracle es comercializada por la compañía del mismo nombre, y provee

capacitación para el uso y manejo de cada versión de sus bases de datos en el mercado.

1.6.2.3 Microsoft SQL Server

Al igual que Oracle, Microsoft ofrece sus sistemas de manejo de bases de datos, SQL Server provee funcionalidades de manejo simple como Oracle, pero su uso está orientado a empresas de escala mediana y pequeña. Existen también las llamadas ediciones Express, que proveen funcionalidades básicas a menor costo.

1.6.2.4 PostgreSQL

PostgreSQL es un proyecto open source, de código abierto, que no es administrado por una compañía única, sino por una gran comunidad de programadores y compañías. Bajo la licencia BSD[16] PostgreSQL se ha convertido en otra alternativa para motores de bases de datos.

[16] BSD (Berkeley Software Distribution) es un sistema operativo basado en UNIX distribuido por la Universidad de California.

1.6.3 Interfaces gráficas dinámicas (AJAX)

Las páginas web siempre se han caracterizado por ser simples formularios con uno o más botones, y si se quiere mayor intercambio de información entre el usuario y la computadora, se procede a cargar una nueva página sobre la página anterior.

El desarrollo de lenguajes de scripting, o de algoritmos para páginas en línea es un hecho desde hace algunos años atrás, pero fue hace poco que empezaron a surgir sitios innovadores que muestran contenido de manera dinámica, similar a una aplicación de escritorio. Muchos de estos lenguajes se están popularizando y el más popular de estas nuevas convenciones se llamó AJAX.

¿Qué es Ajax? Pues son las siglas en inglés que describen "*Asyncronous JavaScript and XML"* que en español quiere decir: JavaScript Asincrónico y XML, esto significa que las páginas creadas con esta arquitectura, son dinámicas, manipulables y se comportan de manera mucho más versátil que las ya casi obsoletas páginas que existen, que deben cargarse por completo una y otra vez al navegar.

Ajax utiliza JavaScript, un código en el lenguaje Java, para la ejecución remota a nivel del servidor, o fuera de nuestros equipos, para modificar la

página que se está viendo de manera dinámica, es decir, que sólo se carga lo que se necesita ver, en la misma página.

La terminología Ajax nació a manos de Adaptive Path[17], por su presidente, Jesse James Garret. El definió a Ajax de la siguiente forma:

Ajax no es una tecnología. Es en realidad varias tecnologías, cada una floreciendo en su propio rumbo, avanzando y juntándose en nuevas y poderosas formas.

Ajax incorpora:

- *Presentaciones basadas en estándares usando XHTML y CSS;*
- *Presentación dinámica de información e interacción utilizando el Document Object Model;*
- *Intercambio de datos y manipulación utilizando XML y XSLT;*
- *Recuperación asíncrona de datos usando XMLHttpRequest;*
- *Y JavaScript juntando todo lo anterior.*

[17] http://www.adaptivepath.com/publications/essays/archives/000385.php

Ajax permite presentar la información de mejor manera, pues permite a las aplicaciones web comportarse como aplicaciones de escritorio, lo cual ayuda al usuario a sentirse mejor con el sistema, además de facilitar al diseñador a crear entornos web que se ven más atractivos, llamativos e interesantes. Y como solamente se carga lo que se necesita ver, se optimiza ancho de banda, lo cual mejora aún más la experiencia del usuario al reducir estos tiempos de espera a los que los usuarios de la red están acostumbrados.

Es decir, más que una herramienta de desarrollo, Ajax es un esquema de trabajo para creación de páginas web dinámicas.

Compañías como Google y Microsoft están invirtiendo cantidades abrumadoras de dinero y esfuerzo en soluciones Ajax, ejemplos de estos diferentes entornos son Picasa Web Albums, Google Mail, Windows Live Mail, Google Docs and Spreadsheets, entre los más utilizados. Estos componentes forman ya parte de la vida cotidiana de algunos profesionales, y de usuarios finales. En la figura 1.1 se observa un poco más como funciona Ajax.

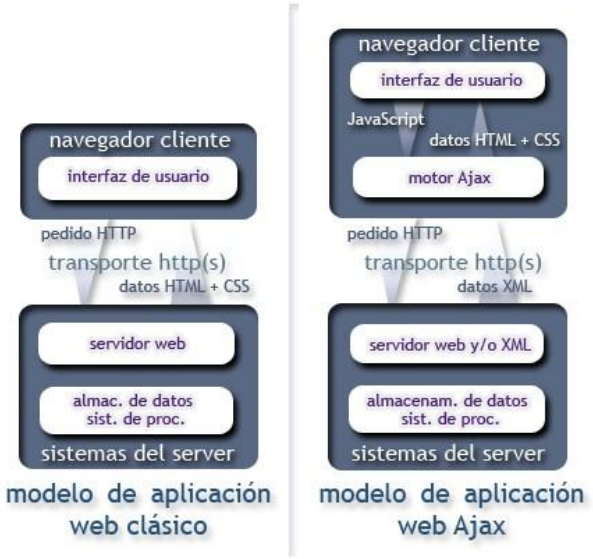


Figura 1.1: Funcionamiento de AJAX vs. El modelo clásico.

Cabe recalcar que este gráfico ilustra la interacción de JavaScript con XML, pero un desarrollador puede optar por utilizar el lenguaje de scripting que desee. En general, Ajax es un esquema para aplicaciones web dinámicas.

Muchos servicios web actualmente utilizan Ajax para ejecutar sus aplicaciones, y algunas utilizan toolkits de desarrollo Ajax para facilitar la creación de sus interfaces gráficas

1.6.3.1 Herramientas para desarrollo de aplicaciones en Ajax

Dado que Ajax es utilizado por sobre todo para generar interfaces gráficas amigables, es natural el desarrollo de Toolkits, o capas de desarrollo para facilitar la realización de estas interfaces.

Estos toolkits permiten la inclusión sencilla de widgets, o componentes gráficos, en sitios web, desde botones hasta editores de texto completos. Estos web toolkits son de código abierto, es decir, son de libre descarga para el uso de desarrolladores.

1.6.3.1.1 Herramientas provistas por compañías

Google, Yahoo y Microsoft son ejemplos de compañías que proveen toolkits para trabajar de manera más sencilla con sus respectivos APIs de desarrollo. Los toolkits son respectivamente:

1.6.3.1.1.1 Google Web Toolkit

Google Web Toolkit[18] es el marco de desarrollo de interfaces de Google. Provee los componentes que Google ofrece en sus aplicaciones web más conocidas, como Gmail y Google Docs and Spreadsheets. Además provee fácil inclusión a componentes de servicios exclusivos de Google como mapas en línea de Google Maps o calendarios de Google Calendar.

Es interesante mencionar que al instalar Google Web Toolkit, la compatibilidad con todos los navegadores más usados como Internet Explorer, Opera y Firefox es automática. Google provee ejemplos y una documentación breve de cada componente, pero provee otras herramientas al nivel de desarrollo que permiten verificar si existen errores en los sitios realizados con este paquete de utilidades.

[18] http://code.google.com/webtoolkit/

1.6.3.1.1.2 Yahoo UI Library

Yahoo UI Library [19] es la contraparte para generación de interfaces Ajax de Yahoo. Provee todas las herramientas necesarias para generar interfaces como las encontradas en los servicios de esta compañía, y facilidades generar diálogos y menús de manera sencilla. Las ventajas de Yahoo sobre Google para el desarrollo en sus respectivos Toolkits, son básicamente las comunidades de desarrollo que permiten a los desarrolladores colaborar entre sí para solucionar problemas al codificar las interfaces.

1.6.3.1.1.3 Microsoft ASP.Net Ajax

Microsoft ASP.Net AJAX[20] es la competencia a los dos sistemas anteriores provisto por Microsoft para desarrollo de aplicaciones web en Ajax, orientado a trabajar principalmente con su

[19] http://developer.yahoo.com/yui/

[20] http://ajax.asp.net/

lenguaje de scripting, ASP, pero es también posible añadirle funcionalidades adicionales para poder desarrollar interfaces utilizando otros tipos de lenguajes de scripting como PHP o Perl. Su nivel de desarrollo provee ventajas principalmente porque es posible utilizar herramientas de desarrollo como el Visual Studio para crear las aplicaciones, pero dejando esta propiedad de lado, no provee características que opaquen a los otros toolkits.

1.6.3.1.2 Herramientas independientes

Es posible encontrar algunos toolkits de desarrollo de interfaces gráficas independientes, es decir, fueron desarrollados por comunidades de desarrollo sin afiliación a ninguna empresa. Algunos de estos proveen facilidades incluso mejores que los toolkits de las compañías antes descritas. Algunos de los toolkits más populares en esta categoría se describen brevemente en los siguientes apartados.

1.6.3.1.2.1 Dojo Toolkit

Dojo Toolkit es probablemente el toolkit de Ajax más famoso y completo, gracias a su comunidad de soporte y sus interfaces diseñadas para ser agradables a la vista.

Su uso es sencillo en comparación a los toolkits antes descritos, y es importante destacar que este toolkit incluye ciertos componentes de los toolkits anteriores, al ser ellos de libre uso. De esta manera Dojo provee soporte para widgets que no existen en casi ningún otro toolkit como creación de gráficos vectoriales, y mapas. Su utilización es tan sencilla como simplemente utilizar los widgets con tags, similar a los HTML ya existentes.

Compañías grandes como IBM y America Online, ya han ofrecido su apoyo a Dojo, al darse cuenta de su potencial.

1.6.3.1.2.2 Ruby on Rails

Ruby on Rails es otro framework muy aceptado en la comunidad de desarrollo Ajax, al ofrecer las facilidades de codificación de Ruby, con las tecnologías Ajax más elementales. Ruby, a diferencia de Dojo Toolkit, no se enfoca principalmente en la interfaz de usuario, sino en la facilidad de codificación inherente de Ruby, para creación de aplicaciones avanzadas con poco código. Ruby on Rails utiliza los mismos efectos de una librería llamada Scriptaculous, especializada en los mismos, y necesita ser instalado en el equipo que lo ejecute para funcionar.

1.6.3.1.2.3 jQuery

JQuery es un framework orientado al desarrollo rápido y sencillo de sitios con JavaScript y CSS. Permite acceder a las clases de los CSS y crear efectos. JQuery es interesante debido a su gran soporte y comunidad de desarrolladores, que trabajan constantemente en plugins o adiciones a

la librería. JQuery es mucho menos pesado que los otros Frameworks, al ser un solo archivo de texto comprimido, lo cual permite el desarrollo de sitios Ajax más livianos. Permite además el llamado a archivos externos, que pueden estar escritos en cualquier lenguaje de scripting. El sitio oficial de jQuery ofrece además ejemplos y herramientas para desarrolladores. Una contraparte de este framework, es que el desarrollador deberá saber la sintaxis de CSS, de JavaScript y de jQuery, a pesar de esto, debido a la comunidad de desarrollo, la curva de aprendizaje es muy corta. Algunos plugins disponibles para el jQuery igualan o superan las funcionalidades ofrecidas por otros Frameworks. JQuery es compatible con todos los navegadores y no requiere de una aplicación instalada en el servidor, a diferencia de Ruby on Rails.

Conclusiones

Como se puede ver, existen muchos tipos de medio en el internet, y una cantidad abrumadora de servicios. Poco a poco surgen, gracias a la aparición de nuevas

tecnologías como Ajax nuevas aplicaciones para los servicios ya existentes y soluciones nuevas que son cada día más sorprendentes. Las formas de mostrar los resultados son cada vez más creativas, y las formas de interactuar con estos sistemas son cada vez más intuitivas para el usuario. En este capítulo se ha visto como encontrar la información pura en la red, pero no se ha hablado a fondo acerca de los servicios que existen en la actualidad. En el capítulo 2, estos servicios serán examinados y se explicará en mayor detalle la realidad actual del Internet.

CAPITULO 2.

Análisis de Fuentes de Información disponibles y su integración

Introducción

En el Capítulo 1 se formó una base teórica que abarcó a los servicios web, qué son, qué es lo que hacen y la innovación que los ha hecho populares en su género. En este capítulo se entrará en más detalle a los servicios web existentes y tipos de medios que manejan así como la interfaz que proveen para su utilización por parte de desarrolladores para la implementación de mashups o como es el propósito de este trabajo, la creación de un Framework.

2.1 Fuentes de información relevantes

A continuación se describirán compañías que ofrecen servicios web y a su vez interfaces para desarrolladores llamadas APIs. Dado que existen en la red muchos servicios similares, los siguientes son los más relevantes basados en el medio en el cual se enfocan, y en el soporte y documentación que ofrecen. Se describirán las fuentes de información provenientes de web crawlers y de motores de búsqueda, para cada medio, y finalmente explicaremos las fuentes alternas a los buscadores que vimos en el capítulo anterior, como scraping y sitios sindicados, pero antes que todo se hablará del funcionamiento de los API que comparten características muy similares en todos los servicios.

El API es el conjunto de interfaces de programación de alto nivel para poder acceder e interactuar con el servicio web[21]. Ciertos servicios requieren que el desarrollador se registre en el sitio web del servicio para obtener un identificador que será llamado API key y le permitirá tener "permiso" de usar el servicio web al nivel de programación.

Para establecer una comunicación con todos estos los servicios se utiliza la arquitectura REST, SOAP o demás estándares de comunicación de que se habló en el Capítulo 1. Los parámetros que recibe esta interfaz son proporcionados y bien documentados por el servicio web así como también el texto que se retorna una vez enviada dicha petición a la cual se le denomina respuesta. Esta respuesta aparece sólo como texto pero con un formato estructurado como XML, JSON y PHP serializado. Hay servicios, principalmente los grandes como Yahoo y Google, ofrecen al desarrollador un gran beneficio al no limitarse a una arquitectura ni a un sólo formato de respuesta.

En resumen, los parámetros necesarios de búsqueda son adjuntados a una dirección URL determinada enviada por HTTP, una vez procesada en los servidores, el servicio envía un texto de respuesta con la información solicitada

[21] Un API (Application Programming Interface) es una interfaz de código fuente provista para soportar pedidos a servicios del mismo, hechos por un programa externo.

que deberá ser interpretada por el desarrollador. La respuesta con formato estructurado que es retornada contiene meta-datos sobre la búsqueda, como el número de coincidencias existentes en sus índices y luego se presentan los resultados en sí, donde se muestran los datos como título de la página, texto de resumen, ubicación URL, entre otros.

Entre los principales parámetros que se le puede enviar al API se pueden mencionar:

• API key necesario para poder utilizar los servicios en calidad de desarrolladores.

• Los términos de búsqueda o keywords.

• El número de resultados por página y el número de página a recibir.

• El formato de respuesta que puede ser XML, JSON o PHP serializado.

• Formatos específicos para cada tipo de búsqueda, por ejemplo para búsquedas web: tipo de archivo, para imágenes: resolución, etc.

Una vez conocido como se realizará la comunicación con cada servicio web, es necesario hablar de las fuentes más relevantes que actualmente existen y las características que ofrecen o no. Estas fuentes serán clasificadas por el tipo de medio enfocado, pese a que la misma compañía se dedique a más de uno, serán citados en su respectiva clasificación.

2.1.1 Fuentes de búsqueda de Texto

Existen múltiples servicios web para la mayoría de medios descritos en el capítulo anterior, algunos se concentran en explotar un servicio en específico, mientas que otros servicios, provistos por compañías grandes como Yahoo y Google, ofrecen búsquedas de varios medios por separado, y en casos específicos un API para desarrolladores.

Algunos de estos servicios son muy populares, mientras que otros, al ser nuevos en la red, no son muy conocidos por la mayoría de los internautas. A continuación se describirán algunos de los servicios más originales y relevantes para estos tipos de medios.

2.1.1.1 Yahoo Search

Yahoo es una compañía norteamericana que ofrece múltiples servicios web y según datos del año 2006 de Alexa y Netcraft, compañías que monitorean el tráfico de la red, es el portal web más visitado de toda la comunidad cibernauta en el mundo, al igual que Google su primer servicio fue el Web Search que con el tiempo ha seguido mejorando.

Su buscador nos ofrece más opciones de búsqueda que Google como por ejemplo: qué tan pronto ha sido actualizada la página,

si tiene contenido licenciado por Creative Commons[22] y filtrar resultados por el país al que pertenece. Yahoo también intenta integrar la mayor cantidad posible de servicios web relacionados a la búsqueda del usuario, como imágenes, videos, etc. Los resultados que se obtienen difieren de los de Google al utilizar otras heurísticas para dar relevancia a las páginas. Yahoo también ofrece búsqueda por términos similares, es decir, permite encontrar páginas relacionadas y expresiones similares a las encontradas.

2.1.1.2 Google Search

Su servicio realiza una búsqueda en la web basado en los términos introducidos por el usuario, mostrando los resultados más relevantes al inicio. Google Web Search adicionalmente ofrece más características como ver un extracto de la página que contiene al texto buscado, visitar páginas que ya no existen gracias al caché de Google, traducir páginas web, ver páginas similares a un resultado específico y poder usar los resultados

[22] Existen muchos tipos de licencias Creative Commons, todas protegen los derechos del autor bajo distintos parámetros. http://creativecommons.org/licenses/

en conjunto con otros servicios de Google[23]. Se pueden realizar búsquedas avanzadas que nos permiten encontrar resultados mucho más específicos, con parámetros que Google mismo provee. Algunos de estos parámetros permiten realizar búsquedas dentro de un mismo sitio, filtrar los resultados por idioma, y hasta por tipos de archivo.

Tipos de búsquedas específicos también son objetivo de Google, de esta forma los usuarios que deseen leer noticias publicadas en el segmento del internet llamado blogs puedan limitarse a obtener resultados recientes o no tanto. Blog Search es un servicio web que permite encontrar posts publicados en blogs basados en los términos de búsqueda ingresados, estos resultados también pueden ser mostrados en formato RSS puesto que es una combinación ideal y preferida entre usuarios lectores.

Google ofrecía antiguamente un API llamado Google SOAP Search API, pero lamentablemente el 5 de Diciembre del 2006 Google decidió dejar de dar soporte a este API, aunque el

[23] Google provee estos servicios gratuitamente, pero sólo existen APIs para búsquedas e interacción con mapas. Referir al Anexo A para encontrar información de ciertos APIs provistos por Google.

servicio no se ha suspendido del todo para que los desarrolladores que lo estaban utilizando quienes aun pueden seguir haciéndolo pero limitados al saber que es un API descontinuado. Google AJAX Search API es el API actual de búsquedas en Google, es mucho más fácil de usar que el Google SOAP Search y provee búsquedas para casi todos los servicios de Google en un solo componente, incluyendo el Web, Imágenes, Video y Blogs. Además provee los componentes de búsqueda que se pueden agregar a un sitio de forma sencilla. La contraparte del uso de este API es que es una búsqueda patrocinada por Google y es necesario ubicar en la página siempre un formulario con la marca de Google. Este API no ha sido recibido bien por muchos desarrolladores ya que no brinda las flexibilidades del anterior API como por ejemplo la numerosa cantidad de parámetros que éste podía recibir.

2.1.1.3 SearchMash

SearchMash [24] es un sitio completamente desarrollado por Google, es un buscador dinámico hecho en Ajax, como un proyecto de poca fama de los desarrolladores de Google. En

[24] http://www.searchmash.com

otras palabras es un Google Search con otra interfaz gráfica y otro nombre. SearchMash provee algunas facilidades además de la simple búsqueda como tener a la mano siempre los resultados paralelos de la web, imágenes, videos, blogs y artículos de Wikipedia, pero su enfoque es la búsqueda en texto. SearchMash permite a los desarrolladores utilizar la interfaz REST para obtener respuestas sólo en formato JSON con los resultados asociados a los keywords y parámetros de búsqueda. Al no ser un API oficial, es muy práctico, pues no se necesita de un API key para buscar con el motor de Google y se puede obtener resultados para los medios en los que el mismo puede buscar con un formato único de respuesta.

2.1.1.4 Microsoft Live Search

Live Search, como es ahora denominado lo que una vez fue MSN Search, posee una interfaz muy amigable, totalmente basada en AJAX con herramientas muy útiles para hacer la búsqueda más sencilla de manejar. Así como los anteriores, Microsoft provee servicios de búsqueda para imágenes, feeds, videos, entre otros menos relevantes, así como guardar un caché del internet como lo hace Google y Yahoo para poder obtener páginas fuera de servicio.

La principal desventaja de los servicios de Microsoft es que siempre son restringidos e inflexibles para los desarrolladores. Su API se llama WebSearch SDK [25] que son muchas herramientas que deberán ser instaladas en la computadora del desarrollador. Las respuestas siempre son retornadas en formato XML y se utiliza el protocolo SOAP para el intercambio de información. Microsoft al ofrecer también plataformas de desarrollo y lenguajes de programación, intenta favorecer a sus respectivas marcas aunque si se da soporte a utilidades open source pero en menor escala.

2.1.1.5 Technorati

Technorati[26] es un servicio independiente orientado a búsqueda de blogs. El motor de Technorati permite de manera práctica buscar frases o noticias a los cuales los bloggers apuntan en sus posts. Technorati además provee un API de búsqueda para desarrolladores y un wiki para consultar la base y el uso de las herramientas provistas por su motor. Permite búsquedas por el texto contenido en los posts, así como búsqueda por los tags asignados a los mismos, en los servicios de blogging que

[25] http://search.msn.com/developer/

[26] http://www.technorati.com/

soporten estos atributos. Entre los resultados nos provee de información del blog como Título, URL, feed y propietario e información del post resultante como el resumen del texto de resultado, permalink[27], tags[28] y fecha de publicación.

2.1.1.6 Wikipedia

Wikipedia[29] es una enciclopedia gratuita y colaborativa, en la cual usuarios anónimos y registrados pueden editar libremente artículos de cualquier índole, contribuyendo así a una base de datos inmensa de conocimiento. Wikipedia poco a poco se ha convertido en una de las fuentes más confiables de información, debido a que es actualizada y moderada constantemente por editores oficiales de la compañía ya que como es costumbre, existen personas inescrupulosas que se dedican a hacer vandalismo editando los artículos con el propósito de desinformar a la comunidad internauta.

---

[27] Permalink es la forma abreviada de decir Permanent Link o Vínculo permanente, que es una dirección web para acceder directamente a un post de un blog o similar.

[28] Un tag es literalmente una etiqueta, una palabra que la comunidad cree que es la más apropiada para describir algo, en este caso un post de un blog.

[29] http://es.wikipedia.org/

Para ayudar al mantenimiento del sitio, los usuarios pueden tener acceso al API creado para obtener información de los servidores MediaWiki, donde se encuentra alojado el sitio web de Wikipedia, este API provee de una gran variedad de funciones, parámetros y formatos de retorno de datos. Pero lo que realmente es útil para los desarrolladores que necesitan obtener cualquier artículo tal como se muestra en Wikipedia, es la característica de exportación de contenidos o Special:Export, que puede ser enviado utilizando la interfaz REST con un URL de petición sencillo de construir y la el retorno de datos es en formato XML con toda la información necesaria de cualquier artículo existente, queda en responsabilidad del desarrollador el interpretar la respuesta.

2.1.1.7 Metacrawler

Como ya se mencionó en el Capítulo 1, Metacrawler es un crawler de crawlers, que indexa contenido de los índices de Google, Yahoo, Microsoft Live y Ask. Los resultados de búsqueda pueden ser ordenados por relevancia o por sitio web. Sus servicios son búsquedas Web, de imágenes, audio, videos, noticias, páginas blancas y amarillas. Las búsquedas de Audio y Video son realizadas de fuentes no oficiales y oficiales de la

web, estos resultados pueden mostrar la página con el streaming del video oficial o el link directo del archivo de audio o video en caso de tener codificación no streaming.

2.1.2 Fuentes de búsqueda de Imágenes

Existen muchos repositorios de imágenes en el internet además de los buscadores que se enfocan en imágenes generales alojadas en sitos web. Muchos de estos repositorios son orientados a fotografías como Flickr, un servicio de Yahoo, y otros orientados a compartir las imágenes de los usuarios en comunidades. Debido a la popularidad de los servicios más conocidos algunos servicios competidores son usualmente poco conocidos por los internautas.

2.1.2.1 Yahoo Image Search

Yahoo también ha adaptado sus servicios de búsqueda con una orientación a imágenes, Yahoo Image Search. Con Image Search se obtienen resultados de imágenes, a simple vista de mejor relevancia que los de Google, puesto que este último retorna imágenes indeseables también. Provee fácil acceso a filtros como resolución de imagen, resultados en colores o

blanco y negro y búsquedas sugeridas, con mucha utilidad para obtener resultados más específicos.

2.1.2.2 Google Image Search

Google Image Search utiliza el poder del buscador de Google para realizar una búsqueda solo de imágenes, la misma que se puede hacer más especifica si son enviados parámetros como tipo de archivo, resolución o censura. El resultado de la búsqueda provee vínculos y thumbnails [30] de las imágenes encontradas. Existe además un proyecto experimental de Google conocido como Google Image Labeler[31] Este proyecto permite a los internautas asignar tags a las imágenes aleatorias de forma amena, por medio de un juego competitivo entre dos personas, haciendo así la búsqueda aún más relevante, con contenido realmente relevante.

[30] Un thumbnail es una imagen de poca resolución cuya finalidad es poder ofrecer una vista previa de la imagen original a quien la visualiza.

[31] http://images.google.com/imagelabeler.

2.1.2.3 Picasa Web Albums

Picasa Web Albums es la versión en línea de una de las aplicaciones de escritorio más importantes de Google, Picasa. Esta aplicación permite organizar las imágenes del disco duro de un usuario de manera automatizada, y generar álbumes con dichas imágenes.

El servicio Picasa Web Albums está integrado con Picasa y ofrece la posibilidad de subir los álbumes desde el equipo a la red de manera casi automática[32]. Al ser desarrollado en Ajax, su interfaz es prácticamente una extensión de su aplicación de escritorio hermana.

A diferencia de la mayoría de repositorios de imágenes de la red, la interacción del usuario con el servicio está enfocada a la creación de álbumes y no del mantenimiento de fotografías digitales o imágenes por separado, y es posible comentar estas fotos y buscar entre ellas por tags y descripciones asignados por el usuario, pero a pesar de todas estas posibilidades, Picasa Web Albums no es orientado a comunidades virtuales.

[32] http://picasa.google.com/web/learn_more_picasa.html

2.1.2.4 Flickr

Flickr es también una comunidad online cuyo principal objetivo es compartir fotografías y almacenarlas en un gran repositorio común desde el punto de vista del usuario, como toda comunidad permite la interacción entre usuarios, etiquetar las fotografías y realizar comentarios. Flickr actualmente se encuentra en fase gamma de desarrollo, que es una fase más estable que la fase beta. La compañía que desarrollo Flickr ha sido comprada por Yahoo en el año 2006 para formar parte de esta gran integración de servicios.

Flickr al no haber formado parte de Yahoo en un inicio no usa exactamente el mismo API estándar que Yahoo, pero se puede decir que el funcionamiento es muy parecido ya que igual se utiliza la interfaz REST con parámetros y las respuestas son estructuradas. La principal diferencia es que la respuesta proporciona información más técnica sobre cada fotografía, pero del mismo modo Flickr posee la documentación suficiente para que con esa información se puedan construir otros datos como: la ubicación de la imagen, la página original de Flickr de la fotografía, el usuario propietario de la misma, etc. Para construir estos datos puede que sea necesario múltiples

peticiones REST para ir obteniendo las piezas del rompecabezas.

2.1.2.5 Microsoft Live Image Search

Live Search de Microsoft ofrece búsqueda de imágenes, de manera similar que como lo hacen Google y Yahoo, es posible encontrar las imágenes con vista previa y con una breve descripción de su formato y dirección, al igual que con los otros servicios. No existe una funcionalidad especial para este motor de búsqueda que otro buscador no ofrezca, pero su presencia es necesaria para no quedarse detrás de la competencia.

2.1.2.6 DeviantArt

DeviantArt [33] es un servicio popular y público para subir imágenes. Lo que diferencia este servicio de otros servicios de imágenes antes mencionados como Flickr, es que las imágenes subidas a DeviantArt deben ser fotografías, pinturas, dibujos o creaciones en general del usuario que las sube. Estas fotos pueden ser entonces comentadas por otros miembros, e incluso

[33] http://www.deviantart.com

pueden ser subidas a la red con derechos de propiedad intelectual para el usuario.

Es importante recalcar que DeviantArt no es solamente un repositorio de imágenes, sino también una comunidad virtual, y es por esto que cada usuario posee una página asignada que puede ser personalizada con sus propios contenidos, y cada una de estas páginas posee galerías en donde un usuario puede mostrar sus fotos al público, o mostrar sus fotos favoritas de otros usuarios.

Existen servicios más avanzados para los usuarios, mayormente de personalización, los cuales ya no son gratuitos. DeviantArt no provee un API de búsqueda como Flickr, a pesar de su popularidad.

2.1.2.7 Riya

Riya[34] es otro repositorio de imágenes, no tan popular como Flickr y DeviantArt. Riya ofrece posibilidades de subir, comentar y compartir las fotos subidas en los servidores, de manera muy similar a Flickr, y también provee un API de

[34] http://www.riya.com

desarrollo con funciones similares a las de Flickr para creación de aplicaciones web utilizando sus servicios de almacenamiento y búsqueda.

Lo que hace que Riya sea un servicio innovador en comparación a la mayoría de los servicios de alojamiento de imágenes, es un algoritmo de reconocimiento de rostros aplicado a las fotografías del usuario, este algoritmo está implementado para aprender de las caras de las personas que aparecen en las imágenes, para su posterior reconocimiento automático, e incluso la búsqueda de imágenes con un rostro en específico en múltiples servicios además de los de Riya, como el antes mencionado Flickr.

2.1.3 Fuentes de búsqueda de Audio

La mayoría de los servicios web relacionados con Audio, están orientados a las búsquedas de artistas y de música en específico. Otros servicios web de Audio, en cambio, permiten subir archivos de ciertos formatos para realizar podcasts, o realizar búsquedas de podcasts.

Uno de los servicios de búsqueda de información acerca de artistas de música más populares es Last.fm.

2.1.3.1 Yahoo Podcast Search y Yahoo Music

Yahoo ofrece búsqueda de podcasts, pero no ofrece un API de desarrollo para búsqueda dentro del mismo. Debido a las diferentes restricciones legales de derechos de autor, en ciertos países es ilegal buscar música para descarga en la red, es por esto que los servicios de búsqueda de audio son orientados a Podcasts.

Yahoo posee también un servicio de búsqueda de música por autores, en donde es posible buscar música, pero por las razones antes expuestas, estos archivos estarán limitados para el usuario, al ser un extracto de las canciones. En países donde las leyes no son tan estrictas con los derechos de autor, como China, es posible buscar canciones completas de música utilizando Yahoo China mp3 search.

2.1.3.2 Buscadores de archivos de audio

Existen en internet varios buscadores dedicados a indexar archivos audio como mp3, wma, ogg, etc. Ejemplos de estos sitios son AltaVista, Astraweb, Lycos, MP3Search entre muchos, los cuales poseen web crawlers que recorren la red en busca de éste tipo de archivos para luego ofrecerle al usuario

una interfaz de búsqueda. Debido a la controversia legal sobre derechos de autor, estos sitios prefieren mantenerse en las sombras con su servicio al margen de la ley o inclusive cambiar su modelo de negocios para vender música en convenio con las disqueras, es por esto que no ofrecen un API que permita a desarrolladores usar sus servicios para la obtención de sus resultados.

2.1.3.3 Last.fm

Last.fm[35] es un servicio que permite a los usuarios registrarse a un sistema de preferencias musicales. Last.fm ofrece una base de información de Artistas y grupos musicales en la red, y hace recomendaciones de éstos de acuerdo a las preferencias de música de los usuarios. Es posible realizar búsquedas de artistas e incluso es posible escuchar por streaming ciertas canciones de los mismos. Los usuarios pueden también editar artículos de artistas y añadir información de artistas nuevos, en caso de no estar en la base de Last.fm, y de esta forma contribuyen al desarrollo del sitio, de manera similar a un wiki.

[35] http://www.last.fm

Last.fm también ofrece estaciones de radio streaming, y para poder escucharlas es necesario descargar una aplicación de escritorio. Cuando una canción es reproducida en esta aplicación, el usuario puede especificarle al sistema si dicha canción es de su agrado o no, aprendiendo así este las preferencias del usuario. Es posible además, instalar plugins de Last.fm a los reproductores del sistema del usuario, de manera que el sistema aprende qué tipo de música éste prefiere escuchar.

No es posible descargar canciones de Last.fm, pues esto sería una violación de los derechos de autor de los artistas, pero es posible pagar en Last.fm para poder escuchar siempre estaciones en línea de los artistas favoritos de un usuario. Last.fm si provee un API de búsquedas, con las funciones respectivas para obtener información de artistas y usuarios, en formato XML.

2.1.4 Fuentes de búsqueda de Videos

Podemos encontrar una gran variedad de servicios orientados a realizar búsquedas de videos en la red, sean estos para streaming o de archivos descargables. A continuación algunos ejemplos de estos motores:

2.1.4.1 Google Video Search

Google también compite con las comunidades de videos online con Google Video, el resultado de ésta búsqueda ofrece información breve del video así como un thumbnail. Al seleccionar un resultado, el video es visualizado en un reproductor Flash[36] que utiliza streaming para descargarlo y reproducirlo a la vez. Adicionalmente los videos también están marcados con etiquetas añadidos por los usuarios y demás opciones de comunidades online como videos relacionados, calificaciones, etc. Es posible además, obtener un feed RSS de un conjunto de resultados para ser usados en un lector de feeds. Otra opción que se le ofrece a los usuarios es la posibilidad de descargar videos, en realidad este servicio es descargar una referencia al video que sigue alojado en la red y en conjunto con un reproductor ejecutable provisto por Google se realiza una conexión remota y la visualización del video en tiempo real.

[36] Flash es una tecnología propietaria de Adobe y se utiliza mayormente para el desarrollo de aplicaciones multimedia y es muy popular por su flexibilidad y resultados finales.

2.1.4.2 YouTube

YouTube es un servicio web que permite almacenar y visualizar videos. Se compone de una comunidad de internautas que suben sus videos a los servidores de YouTube, clasificándolos con tags, dejando los videos libres para comentarios públicos. Con cada video se pueden hacer operaciones interesantes como ser valorado, agregado a una lista de reproducción personal, comentado o reportado como inadecuado o ilegal, etc. Además se puede copiar el link HTML del video y pegarlo en artículos de blogs o foros, lo cual permite una reproducción directa sin la necesidad de visitar la página del video.

Es interesante recalcar que YouTube es una de las más populares comunidades de videos de la red, aprovechando este éxito, en Noviembre 13 del 2006, YouTube fue comprado por Google, pero sigue trabajando de forma independiente[37] y no debe ser confundido con Google Video.

[37] http://www.theage.com.au/news/Business/Google-closes-A2b-YouTube-deal/2006/ 11 / 14 / 1163266548827.html

YouTube, similar al caso de Flickr, posee su propio API que fue desarrollado antes de su integración a Google. YouTube API permite a los desarrolladores enviar parámetros de búsqueda como los keywords por medio de la interfaz REST y la respuesta obtenida viene en formato XML con información relacionada a los resultados encontrados como el autor, duración, calificación, tags, URL, etc.

2.1.4.3 Yahoo Video Search

Yahoo Video Search es en cambio la adaptación orientada a videos. Yahoo en realidad busca videos alojados en la red y también de videos que forman parte de la comunidad de Yahoo, ésta comunidad solo para usuarios registrados permite una socialización entre usuarios al realizar compartir videos, calificarlos y comentarlos. Un aspecto negativo de Yahoo Video Search es que no ofrece muchas garantías en los resultados ajenos a la comunidad Yahoo, porque algunos de estos resultados ya no existen en sus páginas originales o son codificados en formatos de difícil uso para los usuarios promedio y como consecuencia no son del agrado de los mismos.

2.1.4.4 Microsoft Live Video Search

El servicio de búsqueda de videos de Microsoft es interesante porque hasta ahora sólo es realizado basado en un índice de diferentes fuentes oficiales y comunitarias tales como MySpace, YouTube, Google y Yahoo video, entre muchas.

Muy aparte del servicio de búsqueda de videos, MSN Video es un portal que ofrece una colección de videos oficiales para que sus usuarios puedan reproducir y crear listas de reproducción. Se encuentra en fase de desarrollo beta y está limitado únicamente a ver videos y navegarlos entre categorías.

Pero Microsoft también ha entrado en la competencia de las comunidades virtuales orientadas a videos. SoapBox, parte de MSN Video, es una comunidad que como todas permite subir, compartir videos, comentarlos y socializar. También posee un sistema folksonómico de etiquetación, lo nuevo que significa que se puede ver una nube de tags que indican la popularidad y cantidad de videos de esa clasificación. Su interfaz gráfica esta realizada completamente en AJAX lo que aumenta su usabilidad en todo sentido, todas las opciones son amigables, están siempre a la mano sin necesidad de recarga de toda la página.

SoapBox aún se encuentra en fase beta de desarrollo y la codificación de video es formato Flash al igual que la mayoría. Al ser un servicio poco conocido su contenido es muy pobre aún debido a la poca colaboración de usuarios registrados, los mismos que deben contar con una invitación del servicio Live.

2.1.4.5 Stage 6

Los creadores de la codificación de video digital DivX, ahora entran en la competencia de las comunidades virtuales de videos. Actualmente Stage6 está en estado Alfa de desarrollo y permite realizar lo que la mayoría de comunidades permiten hacer, subir y compartir videos, calificar y clasificar dependiendo del contenido y postear comentarios referentes a los mismos [38]. Además es posible crear canales, que son colecciones de videos de contenido similar compiladas por los usuarios.

La tecnología utilizada para la compresión de video es DivX obviamente, y la calidad es significativamente superior a Flash. La principal desventaja es que esta compresión requiere mayor

[38] http://stage6.divx.com/

espacio físico de almacenamiento por lo que la transmisión tarda más. Su página web tiene una integración con el DivX Web Player, que se utiliza para la reproducción de videos. El mismo software permite subir videos a los servidores de Stage 6 permitiendo compartir y someter a crítica a la comunidad online.

Los videos que los usuarios suben mantienen una propiedad intelectual de quien los sube, y además es posible obtener una ganancia monetaria por esos videos, en cada profile de usuario nuevo se deberá indicar si se desea vender los videos y luego el costo de cada video. DivX obtiene un pequeño cargo tarifario por cada venta realizada, de esta forma se incentiva a la dirección y producción amateur de video para ir ganando terreno en comunidades online que actualmente se consideran semilleros de nuevos talentos.

2.1.4.6 Revver

Revver[39] es un servicio análogo a YouTube, en el cual un usuario puede subir los videos y estos videos pueden ser

[39] http://www.revver.com

comentados por otros usuarios de la comunidad. Lo que destaca a Revver sobre los demás servicios de repositorios de videos, es que al igual que con Stage 6, es posible ganar una cantidad económica por los anuncios adjuntos a los mismos. Estos anuncios se comparten en proporción de cincuenta por ciento al servicio, y cincuenta por ciento al usuario. Revver añade al final de cada video un tag con vínculos llamado RevTag, el cual contabiliza el número de clicks, y obtiene una ganancia económica del mismo.

2.1.4.7 Metacafé

De origen israelita, Metacafé es otra comunidad de videos online, la tercera más grande para Octubre del 2006 según Comscore. Principalmente dedicada al ocio, los usuarios pueden subir, descargar y compartir videos cómicos, comerciales virales [40] entre muchos. Entre sus servicios se ofrece un software descargable para Windows que permite la clasificación y organización local de videos, así como subir y descargar los mismos cuando la computadora se encuentre en

[40] Comerciales virales se refieren a una forma de usar marketing para llegar al público, en este caso con un video, y dejar que ellos también propaguen el mismo en su entorno, de tal forma que la propagación crece exponencialmente en el mercado, como un virus.

período de inactividad aprovechando esos recursos que serían desperdiciados.

La controversia presentada contra Metacafé se ha dado por la censura de comentarios ofensivos, racistas y para adultos, que son dados de baja sólo con la calificación de otros usuarios, aunque cuando se activan las opciones de filtrado ofrecidas estos comentarios son omitidos.

2.1.5 Fuentes de Scraping

Existen sitios que proveen información interesante, pero que pueden no estar completamente indexados en un buscador. Estos sitios como algunas fuentes de noticias, o páginas poco conocidas que estén en constante actualización, o que contengan información adicional que no pueda ser extraída por un simple web crawler ni tampoco sindicadas, pueden ser examinadas utilizando scraping para extraer su contenido.

Existen sitios especializados en realizar scraping de ciertas páginas web. Pero desgraciadamente, como el scraping requiere del conocimiento de una página, es casi imposible crear tecnologías de scraping para todas las páginas web de una forma generalizada. Páginas que requieren interacción con el usuario para que se defina el patrón de elementos a

examinar en el proceso de scraping están surgiendo poco a poco. Un ejemplo reciente de estas tecnologías es Dapper.

2.1.5.1 Dapper

Dapper es un servicio web que permite al usuario obtener información de cualquier fuente de internet en formato XML el mismo que puede ser convertido por Dapper mismo en otros formatos populares como RSS, HTML, JSON y gadgets para homepages populares, etc.

En sólo unos minutos, un usuario puede utilizar la interfaz de Dapper para navegar en sus sitios favoritos y seleccionar los deseados como fuentes de información, luego para cada sitio se deberá resaltar las secciones de la página a modo de entrenamiento para Dapper, así el sistema aprende y correlaciona lo resaltado para poder generar un documento de salida personalizado con campos también definidos por el usuario. Estos módulos de scraping de sitios específicos son llamados dapps, y son publicados en una comunidad para su utilización por otros usuarios. Si un módulo no es usado por un período específico, este modulo es borrado del sistema.

Como su lema lo dice, "Es como legos", la creatividad es el límite que tiene el usuario para poder crear su página de contenido personalizado y actualizado con sus fuentes favoritas. Los servicios de Dapper son muy flexibles ya que al tener la capacidad de presentar la página de resultados en formatos como JSON o XML, los desarrolladores tienen un ilimitado y no restringido campo de información en la red. Desgraciadamente, como estos dapps son únicos, a pesar de que retornan un formato generalizado, son todos definidos por el usuario que los creó, y no son utilizables como una fuente confiable de información, por no seguir un mismo formato, y por su fecha de expiración.

2.1.5.2 Directorios

Servidores Web de administración de servidor, tales como Apache, Internet Information Server, etc. generan una página web con el contenido de todos los archivos que existen en un directorio del árbol del sistema siempre y cuando no tenga una página índice (index.html, index.php, index.asp) que se cargue automáticamente cuando un usuario externo trata de acceder a dicha dirección a través de un navegador de internet. Por ejemplo para el sitio www.misitio.com/directorio/ si existe un

archivo index.htm éste será cargado automáticamente al acceder a esta dirección por el navegador, caso contrario se presentará una página de apariencia simple con vínculos a los archivos que se encuentran en aquel directorio.

Estas páginas también son examinadas por los web crawlers con su contenido generado como HTML, con técnicas avanzadas de búsqueda, como decirle al motor que me retorne páginas cuyo título diga "índex of" y su extensión sea "HTML" podemos obtener sólo estos resultados deseados. Es en este punto donde entra el rol del scraping, leer el contenido estático de una de estas páginas de directorios, analizar el contenido o asumir uno y comenzar a extraer los vínculos hacia los archivos del directorio y almacenarlos en algún tipo de dato abstracto (como una lista) para posterior lectura. Un ejemplo de esta técnica consistiría en buscar directorios con archivos de audio, obtener los archivos en una lista y transformar la misma en un playlist o lista de reproducción que puede ser enviada a un reproductor de medios.

2.1.6 Fuentes de Sindicación

Sindicación web es hacer disponible una sección de un sitio web para que todos puedan acceder al mismo y leerlo a través de este medio llamado web feeds de forma siempre actualizada y disponible. Los sitios web que lo ofrecen normalmente presentan un icono o vínculo al archivo del feed para captar la atención del usuario interesado, pero dada la gran cantidad de sitios web en la red, los buscadores de feeds son la solución ideal a la necesidad de encontrar este tipo de fuentes de información. Estos buscadores recorren la red con crawlers indexando únicamente la dirección del web feed y otra información relevante como un extracto de la noticia, autor, etc. Un ejemplo de estos buscadores es Microsoft Live Search en la categoría de Feed Search, ofrece buscar texto y muestra al usuario únicamente direcciones de web feeds. Technorati, en cambio, además ser un buscador textual con campo de acción limitado a la blogosfera, también es capaz de informar cual es la dirección del feed que sindica el contenido de ese blog y de dónde provino el resultado encontrado.

No sólo texto se puede encontrar por medio de feeds. Actualmente, los datos sindicados se han convertido en un medio muy importante para publicar contenidos de audio y video. Estas publicaciones son conocidas como podcast de audio y podcast de video, o vidcast, respectivamente.

Estos feeds pueden ser escuchados públicamente o pagados. De igual forma existen buscadores que permiten encontrar estas sindicaciones principalmente las públicas, ejemplos de estos buscadores se presentan a continuación.

2.1.6.1 Odeo

Odeo [41] es uno de los servicios más innovadores para la realización de búsquedas de podcasts. El sitio ofrece un ambiente dinámico realizado en Ajax, y mantiene una interacción informal con el usuario comparado con otros servicios similares. Odeo permite a los usuarios realizar búsquedas de podcasts, los cuales están separados por categorías y tags, y establecer preferencias entre los resultados. Es posible escuchar los resultados de búsqueda por streaming, y además añadirlos a una lista de reproducción, para poder escucharlos luego. Es posible además, publicar estos podcasts en otros sitios, pues Odeo provee facilidades para embeber estos podcasts, ofreciendo varios diseños de reproductores, e incluso permite descargar algunos de estos archivos como mp3

[41] http://odeo.com

al equipo. Un API para este servicio está en constante desarrollo.

Las búsquedas dentro de los podcasts que indexa Odeo son provistas por Google, con su opción de filtrado por sitio. Pero es posible obtener información detallada en JSON, al poseer el identificador de un artículo de podcast en la base de Odeo.

2.1.6.2 Pluggd

Pluggd es otro servicio para búsqueda de Podcasts de Audio, muy similar al servicio que provee Odeo. Al igual que con su competidor, es posible agregar preferencias a los resultados obtenidos, y es posible realizar búsquedas por tags y por categorías.

La búsqueda de Pluggd es contextual, es decir, puede encontrar resultados en podcasts que no necesariamente tengan la palabra de búsqueda. Además, ofrece una tecnología llamada HearHere, para realizar búsquedas dentro de los archivos de audio de los podcasts, en donde el usuario puede especificar un tema en particular, y el motor identificará en que partes de archivo de audio se trata de el tema que el usuario especifique.

2.1.6.3 iTunes

ITunes, de propiedad de Apple, además de ser un reproductor de multimedios y una tienda online de música, ofrece el servicio de búsqueda de audio, lo cual permite obtener resultados de mp3 de variada selección y orientación, pudiendo ser canciones o podcasts reales como noticieros.

Gracias a un URL de búsqueda es posible obtener todos los resultados relacionados a palabras clave específicas en formato JSON, especificando el tipo de audio que se desee obtener, sea este un podcast o un archivo de música.

Cabe recalcar que los archivos resultantes que pertenecen a canciones musicales provistos por este servicio son solo muestras, y para su descarga completa es necesario pagar un valor monetario, mientras que los podcasts son libremente descargables.

2.1.7 Otras fuentes de información de importancia

Además de las descritas anteriormente, es importante marcar la existencia de otras fuentes relevantes, que proveen información en los medios antes descritos, pero orientadas a enfoques distintos a información en la red. Un

ejemplo de estos tipos de fuentes son la información geográfica, o información de fechas, que almacenan lugar y tiempo, respectivamente.

2.1.7.1 Fuentes de información geográfica

Estas son las fuentes de información que contienen información relevante de ubicaciones, y permiten buscar hitos, lugares, e incluso direcciones físicas. Las más importantes fuentes de este tipo de información son Yahoo Maps, Google Maps y Microsoft Virtual Earth.

2.1.7.1.1 Yahoo Maps y Yahoo Geocode Search

Yahoo provee varias formas de crear mapas interactivos, utilizando una animación en Flash, o utilizando un componente AJAX. Ambos son trabajados de manera similar y es posible añadir hitos definidos por el usuario a los mismos, con la limitación de la necesidad de un plugin en el caso de los mapas en Flash. Es posible ver los mapas en varios tipos de vista, como lo son la vista satelital, la vista vial, que muestra el mapa de calles y avenidas de un sitio, y la vista híbrida, una mezcla de ambas. Para utilizar el API de Yahoo es necesario un API key. Para utilizar un mapa es necesario crear un

objeto del mismo utilizando funciones de JavaScript provistas por el servicio.

Yahoo Geocode Search es el servicio de Yahoo orientado a buscar ubicaciones almacenadas en la base de Yahoo. Cada petición devuelve una lista de ubicaciones relacionadas a los términos de búsqueda definidos, en formato JSON, XML, entre otros. Podemos encontrar, por ejemplo, restaurantes dentro de una ciudad, o la ubicación geográfica de una capital, con información de latitud y longitud. Esta información puede después ser mostrada en un mapa que provee los controles necesarios para su navegación geográfica, como el zoom. El API de Yahoo Geocode Search es sencillo de utilizar, al sólo necesitarse llamar a un URL con los parámetros de búsqueda.

2.1.7.1.2 Google Maps

Google también provee un API de desarrollo para aplicaciones con mapas, y ofrece el mismo manejo sobre éstos que su contraparte de Yahoo. Los mapas de Google sólo están disponibles en componentes AJAX.

A pesar de esto, al ser Google más popular en la comunidad, los desarrolladores tienden a preferir este API al de Yahoo. El trabajo con los APIs de Google y Yahoo es casi idéntico, y por tanto, es posible combinar ciertas bondades de cada uno, como la búsqueda de localidades de Yahoo con los mapas de Google.

Google ha desarrollado una herramienta de escritorio utilizando este motor geográfico, llamada Google Earth, que permite visualizar edificios y monitorear tráfico vehicular utilizando la base de información de sus mapas. Así mismo, Google desarrollo otros servicios web basados en este motor, como son Google Moon y Google Mars, para visualizar satelitalmente la Luna y Marte. Google provee un motor de búsqueda de localidades, pero a diferencia de Yahoo Geocode Search, es necesario preparar un XML de pedido para recibir otro XML con los formatos de respuesta. Este servicio está restringido a XML y provee resultados muy similares a los de Yahoo. El API de trabajo con los mapas de Google es casi idéntico al de Yahoo, con

ligeras diferencias en nombres de funciones e inicialización de los mapas.

2.1.7.1.3 Windows Live Local

Es un tercer servicio de mapas, y fuente de información geográfica [42]. Este servicio, además de proveer las bondades de los servicios anteriores en AJAX, permite la visualización de ciertas ciudades en 3D, algo que Google realiza en Google Earth, una aplicación de escritorio que utiliza la base de los mapas. De manera similar, para visualizar los edificios y lugares en 3D, es necesaria la adición de un plugin para cada navegador. Windows Live Local permite la creación de hitos de interés definidos por el usuario, y alojarlos en su cuenta. Este servicio es un poco más pesado para el ordenador que los anteriores, pero es un poco más atractivo visualmente. Además de estas propiedades, Microsoft ofrece un servicio llamado Bird's Eye, o Ojo de Pájaro, que permite visualizar ciertos sitios desde vista aérea con una perspectiva angular a la tierra, es decir, podemos apreciar mejor la altura de los elementos del

[42] http://maps.live.com/

sitio, como edificios y paisajes, algo que no puede realizarse con una vista satelital donde todo es visto casi plano. Microsoft provee un API para este servicio llamado Microsoft Virtual Earth SDK. Una contraparte de este servicio, es su compatibilidad pobre con muchos navegadores, al ser propiedad de Microsoft, su desarrollo está orientado a las tecnologías soportadas por su navegador, el Internet Explorer.

2.2 Integración de Servicios a través de un Framework

Se han descrito las múltiples fuentes de medios en la red, y sabemos que existe una necesidad de mezclar varias de estas fuentes, para así poder unificar la información disponible.

2.2.1 Justificación, el problema y su solución

Los medios encontrados en repositorios multimedia de distintas compañías, o esparcidos por distintos servidores en línea, o de páginas independientes necesitan ser unificados, este es el problema a resolver. Para una búsqueda dada, es necesario visitar diferentes fuentes para encontrar información relevante. Si bien la información encontrada puede ser útil, encontrarla junta lo es aún más. Esto está evidenciado en el auge

de los mashups y de las aplicaciones dinámicas antes vistas. Además, si se desea encontrar diferentes resultados de un mismo tipo de medio, existirán resultados repetidos que necesitarán ser examinados más de una vez, resultando en información redundante.

En internet se puede observar pocos mashups que realicen búsquedas simultáneas, y la mayoría lo único que ofrece es clasificar los resultados por motor de búsqueda, añadirlos a una gran lista o mezclarlos sin ninguna restricción. Pocos son los buscadores (como Metacrawler y Quintura) que realizan alguna innovación para el usuario y que ofrecen de forma transparente una utilidad inigualable al combinar servicios distintos.

En otras palabras, todos estos distintos tipos de medios son similares en resultados, pero tienen muy distintas formas de acceso. Como se puede apreciar, es cada vez más necesario el poder trabajar con estos medios de diferentes fuentes, como si fueran una sola. El propósito de este trabajo es el de unificar estas fuentes, y de proveer una capa para permitir a los desarrolladores futuros el uso generalizado y sencillo de los resultados de estas. Este Framework será una solución para este nuevo problema. A continuación se describirán los servicios que serán considerados para el desarrollo del framework que permitirá esto.

2.2.2 Alcance

En el alcance serán detalladas las fuentes iniciales de información a utilizar, de las detalladas anteriormente, y se justificará el uso de cada una gracias a sus ventajas o desventajas.

2.2.2.1 Fuentes de información iniciales

Hasta ahora se han revisado los servicios web más relevantes y preferidos por los desarrolladores para la implementación de una diversidad de mashups, a continuación se explicará qué servicios serán los soportados por el framework a realizar en este trabajo, escogidos por su popularidad en la red, y por proveer facilidades de desarrollo.

Las compañías más populares que deben ser incluidas en el Framework, de manera auto justificada, por su gran popularidad en la red y por su amplio soporte de búsquedas de distintos tipos de medios, son Yahoo para web, image y video search, Google para web, image, blogs y video, Live para web, image y feed search, Flickr, YouTube, Technorati, ITunes y Odeo.

Las búsquedas de Google serán realizadas a través de SearchMash ya que el formato utilizado en sus respuestas (JSON) es muy conveniente y de una interpretación más sencilla que XML, es decir, el costo computacional es reducido. Así mismo SearchMash, como ofrece soporte para Web, Imágenes, Blogs y Video, se convertirá en la fuente proveniente de Google. Adicional a los medios antes mencionados y gracias a la capacidad de poder incluir parámetros exclusivos de Google, es posible realizar búsquedas de archivos descargables como documentos, programas y audio, esto será beneficioso para mantener un repositorio de estos medios. SearchMash al ser un proyecto secundario y que no comparte de la misma popularidad del buscador principal de Google, correrá el posible riesgo de ser descontinuado. A pesar de esto SearchMash es el servicio más completo provisto oficialmente por Google que se adapta a los intereses y necesidades para el desarrollo del framework de este trabajo.

Queda descartado el uso de Ajax Search API debido a que no es muy flexible para desarrolladores sino a usuarios avanzados propietarios de sitios web personales. De igual forma SOAP Search API, al no contar con soporte actualmente por parte de

Google, no puede ser utilizado. Además que SOAP Search API solo provee facilidades para búsqueda Web y no de los demás tipos de medio que son el objetivo de este trabajo y para los cuales Google posee servicios de búsqueda. Existen desarrolladores independientes que han implementado soluciones para continuar utilizando el SOAP Search API pero siempre existe el riesgo que Google elimine definitivamente la interfaz quedando inutilizables todas estas soluciones, un ejemplo de este tipo de implementaciones es Evil API [43] planeado a ser un reemplazo extraoficial del SOAP Search API. El Evil API realiza scraping de los resultados de búsqueda de Google aunque esto pueda resultar en contra de los términos de servicio de Google.

Adicional a los servicios de búsqueda de videos en SearchMash, YouTube es una opción que no puede ser omitida. Su API no es complejo de usar, pero dado que las respuestas son en formato XML son más costosas de procesar e interpretar. A pesar de esto, la infinidad de videos que están almacenados es vasta y crece significativamente día tras día.

[43] http://evilapi.com/

Como se ha revisado anteriormente, el API de Yahoo provee soporte para recibir una respuesta en formato JSON y los parámetros de búsqueda brindan una mayor flexibilidad para el filtrado de resultados específicos. A diferencia de Google, Yahoo fomenta el uso de sus APIs tradicionales sin descartar la evolución de los mismos o creación de nuevos y mejores servicios. Los desarrolladores siempre han tenido una afinidad a usar el API de Yahoo por ser sencillo de usar en comparación a otros APIs, los ejemplos de su utilización son innumerables y se los puede apreciar en la página oficial del Yahoo API. En estos ejemplos son mashups que combinan servicios de Yahoo con Google, Microsoft, entre otros, dependiendo del propósito del Mashup y la capacidad del desarrollador.

Flickr también será incluido en el framework, al ser el uno de los repositorios de imágenes más utilizados globalmente, su API es muy útil pero precisamente orientado a búsquedas sino a todo tipo de manipulación del repositorio y de las cuentas de usuario. Por ejemplo para realizar una búsqueda de imágenes se obtiene una respuesta con campos de identificación de fotos relacionadas, luego para poder obtener los datos como URL, resumen, resolución es necesario más peticiones al servicio de

Flickr, haciendo más costoso su obtención de resultados. Pese a todo esto es necesario usarlo debido al posicionamiento y aceptación que tiene en la red.

Otro servicio que se va a utilizar es Microsoft Live, aunque sus API no es muy flexible en comparación a los demás (por ejemplo, siempre retorna 10 resultados, no más ni menos), porque no provee soporte directo a no ser por utilizar su framework, no deja de ser una buena fuente de resultados de web, imágenes y feeds. Los resultados de web e imágenes tienen una relevancia similar a la de Google y son significativos. Su servicio de feed search sí es muy favorable a la utilidad del framework, puesto que es posible obtener directamente los feeds con algo de información y meta datos del sitio donde provienen.

Para fortalecer los resultados de búsqueda de blogs y feeds se utilizará el popular motor de Technorati, el mismo que proporciona el permalink del post junto a más información relacionada por resultado como el feed (en RSS y/o ATOM), URL, título y dueño del blog, entre otros. Los resultados de Technorati siempre están en competencia con Google blog

search con el fin de indexar más rápido cada nuevo post de la blogosfera.

De la misma forma, se han escogido dos de los servicios de búsqueda de audio más conocidos, Odeo y iTunes de Apple, dado que estos dos servicios proveen posibilidades para minar sus datos. Para el uso de la información de Odeo es necesaria la búsqueda de páginas utilizando otro motor de búsqueda web, como Google o Yahoo, y una vez obtenidos los detalles necesarios, es posible obtener información detallada de archivos de audio alojados en este servicio.

Para iTunes, en cambio, la búsqueda se realiza de manera sencilla, obteniendo resultados en formato JSON, los cuales serán integrados fácilmente al framework.

Cabe recalcar que es posible descargar archivos de audio y de video de algunas de estas fuentes. Esto aplica a algunos de los servicios que además de proveer reproducción en línea, proveen vínculos a los archivos originales. Cuando esto sea posible, el framework a desarrollar identificará estos archivos, posibilitando trabajar con estas direcciones a futuro.

Web scraping no es la excepción en el desarrollo del framework, su único aporte será utilizado para los resultados de directorios libres, se utilizará una técnica para abrir cada página y obtener la lista de archivos de la misma, para el caso específico de audio, con todas las mp3 de un sitio se generará una lista de reproducción en formato XSPF[44].

El framework estará abierto para obtener cualquier fuente de sindicación RSS y ATOM, es por eso que estos formatos deberán ser interpretados y almacenados. Las fuentes pueden ser infinitas, desde los feeds encontrados por los motores de búsqueda como Technorati o inclusive introducidos por el usuario. El formato de los feeds ha tenido gran aceptación y tiene un largo futuro en la web, es por eso que no debe ser menospreciado y considerar en su lugar su posible expansión.

Además de todas estas funcionalidades, el framework soportará inicialmente la adición sencilla de mapas. Se decidió el soporte para Yahoo y Google Maps, por su compatibilidad con los navegadores. La tecnología a utilizar para el desarrollo de

[44] XSPF es un formato de lista de reproducción basado en la estructura XML y donde sus elementos listados son las ubicaciones de los archivos de audio que la conforman

mapas es AJAX, en ambos casos, por su mayor compatibilidad con los navegadores. Se permitirá la creación de hitos por el usuario en un mapa dado, el enfoque de la vista en una ubicación dada, y la búsqueda de direcciones válidas y su adición automática al mapa. Para la búsqueda de información geográfica y de direcciones se utilizará el motor de Yahoo, Yahoo Geocode Search, pues sus respuestas pueden estar en varios formatos, incluyendo JSON y XML contrario del servicio de Google que sólo provee soporte para XML con un procesamiento previo de un XML de petición requerido.

2.2.2.2 Facilidades de desarrollo del Framework

Habiendo analizado lo que el internet nos ofrece actualmente, sus servicios y herramientas, se ha podido realizar una selección minuciosa de lo que realmente se ajusta a las necesidades y objetivos de este trabajo. En la sección anterior se habló de los servicios a considerar, en esta sección se describirá de forma general el alcance del framework a desarrollar ya que se entrará en detalle en el siguiente capítulo que abarcará el diseño.

El framework utilizará diversas fuentes de servicios web para realizar búsquedas tomando en cuenta el tipo de medio. Los resultados serán almacenados en objetos de tipo de dato abstracto que representarán los diversos tipos de medio que se manejarán, estos son textual, imágenes, video, audio y archivos independientes. Estos objetos serán denominados repositorios y los datos de resultados guardarán un formato adecuado y estándar para su fácil utilización y organización.

El sistema también será orientado para el uso de motores de búsqueda integrados, o para el desarrollo de homepages, que utilizarán las capacidades del framework combinándolo con una interfaz amigable y usable para el usuario, quien busca encontrar las mejores opciones de resultados a una búsqueda.

El framework estará inicialmente desarrollado en PHP, dado que es de libre uso y provee funciones de interpretación y codificación web muy útiles para el manejo de los formatos con los que trabajará el framework. Además de esto, todo el código y comentarios estarán en español, debido a que este es el idioma de los autores del mismo.

Las ventajas que ofrece el framework se mencionan a continuación:

- Un conjunto de funciones de búsqueda listas para ser usadas por desarrolladores.
- Evitar perder tiempo buscando y aprendiendo a utilizar diferentes APIs los cuales obviamente no se utilizan de la misma manera.
- El código del framework estará abierto a contribuciones de desarrolladores.
- Aunque no se provea de soporte para todos los servicios web actuales, queda abierta la posibilidad de agregar servicios web conociendo el funcionamiento del API.
- Es posible agregar más tipos de medio al repositorio de información, para su manejo individual, y cualquier otra funcionalidad ya que la implementación procura ser lo más fácil de entender posible.
- Incluso dentro de las funciones provistas por el framework, se intenta lo más posible reducir la interacción con las funciones del mismo, para su fácil aprendizaje, a diferencia de algunos APIs disponibles en el medio.
- Facilitará el desarrollo de interfaces de motores de búsqueda integrados, basados en términos específicos.

- Su uso será compatible con interfaces Ajax, lo cual facilitará el desarrollo futuro de interfaces gráficas que interactúen con el framework, para crear nuevas soluciones web.
- Existe la posibilidad de la creación a futuro de una interface de desarrollo para aplicaciones que utilicen el framework, así como la posibilidad de añadirlo a IDEs de desarrollo existentes.

Así mismo, existirán limitaciones para el framework a desarrollar, la mayoría de estas relacionadas a la temprana etapa del framework.

- Una desventaja podría ser que el comportamiento gráfico y lógico de la aplicación debe correr completamente por parte del usuario, pues no existe una interfaz para el desarrollo.
- Una limitación del framework es que sólo está orientado a minar las fuentes de datos en cuestión.
- Al estar en español, la barrera idiomática impide el desarrollo y uso del mismo a desarrolladores que no lo entiendan, una versión a futuro podría estar en idiomas distintos al mismo.

- No existe un manejo de relacionamiento contextual de los datos, la relación general en el framework será el query de búsqueda del sistema y sus parámetros.
- Estará desarrollado inicialmente en un lenguaje de programación específico, PHP, con posibilidad de soporte futuro a otros lenguajes.

Conclusiones

Como se puede apreciar, en internet existen una infinidad de servicios web 2.0 orientados a un tipo de medio o que abarcan más de uno, es decir multimedios. Tantos son estos servicios que usados de manera independiente, los usuarios tienden a perderse y olvidarse de todo a lo que están registrados o prefieran usar. De entre tanto servicios deben ser escogidos los que más se adaptan a las necesidades del framework, claro que es posible agregar nuevos conforme la tecnología de información avanza lo cual solo se puede lograr con una programación bien diseñada. El diseño, la piedra angular del desarrollo de cualquier software, será cubierto en el siguiente capítulo.

CAPITULO 3.

Análisis y Diseño del Framework

Introducción

Ya se ha descrito que servicios se utilizarán y el alcance del proyecto. En este capítulo se detallará más a fondo el diseño planteado para el Framework, el mapa de casos de uso planteados para el mismo, y la función que el Framework cumplirá para el desarrollo de nuevas soluciones.

3.1 FindJira Framework

Los marcos de trabajo, o Frameworks, como se conocen en el mundo de desarrollo de software, son una estructura, o un conjunto de pasos a seguir, sobre el cual otro proyecto de software o aplicación se puede desarrollar. Un ejemplo conocido en el mundo marcos de trabajo o Frameworks es el .NET framework, de la compañía Microsoft. Todos los Frameworks son creados con el afán de facilitar la cantidad de código que se necesita para desarrollar una nueva aplicación, y consecuentemente para reducir lo más posible el tiempo de desarrollo de la misma. Por ejemplo, un equipo de desarrollo puede utilizar el .NET Framework para desarrollar una aplicación bancaria que se enfoque en el funcionamiento lógico de la misma con los componentes ya provistos, en vez de preocuparse en el funcionamiento de un botón o una lista de datos que sean necesarios para la presentación de la información.

Cómo se mencionó en el capítulo anterior, el propósito de este trabajo es desarrollar un framework que integre una variedad de servicios web existentes y almacenar los resultados comparándolos con los ya encontrados y clasificándolos por tipos de medio. La principal meta es filtrar esta información, pulirla y crear un framework que permita sacar provecho de ella para así crear nuevas aplicaciones, que no son del todo conocidas, como aplicaciones que relacionen blogs con videos similares, o fotos de diferentes fuentes con música relevante, que de una forma u otra, se relacionen entre sí. Al final se deberá ofrecer al usuario del framework, el desarrollador, una interfaz única para el acceso a los medios ofrecidos y la capacidad de añadir nuevos servicios de una manera sencilla conociendo el formato de la respuesta que dicho servicio entregue.

El nombre de este framework se ha decidido llamarlo FindJira, un nombre nuevo, llamativo y fácil de recordar para los internautas. Derivado de "buscar" ("find") y "zilla" de "Mozilla" que a su vez es una referencia del personaje ficticio "Godzilla", cuya pronunciación original en japonés es "Gojira"[45], este nombre nació para darle un nombre único y fácil de recordar a un proyecto que realizaba búsquedas durante una cátedra a la que asistieron los autores de esta tesis.

[45] El nombre "Gojira" se dice venir de la combinación de dos palabras: gorira 'gorila' y kujira 'ballena'. “Godzilla” es una traducción fonética de este nombre.

3.2 Análisis

Primeramente se revisará en detalle las capacidades del framework, lo que el desarrollador podrá realizar con el mismo con los casos de uso, la ubicación del sistema posicionado entre las capas que interactúan como consecuencia a una petición, y por los pasos en los que deberán pasar los datos desde y hasta el usuario siguiendo una secuencia establecida.

3.2.1 Casos de Uso del Framework

FindJira Framework permitirá al desarrollador que lo utilice realizar un conjunto de actividades que faciliten su interacción con los servicios disponibles. En la figura 3.1 muestra de forma visual las acciones principales que el desarrollador puede realizar utilizando el Framework.

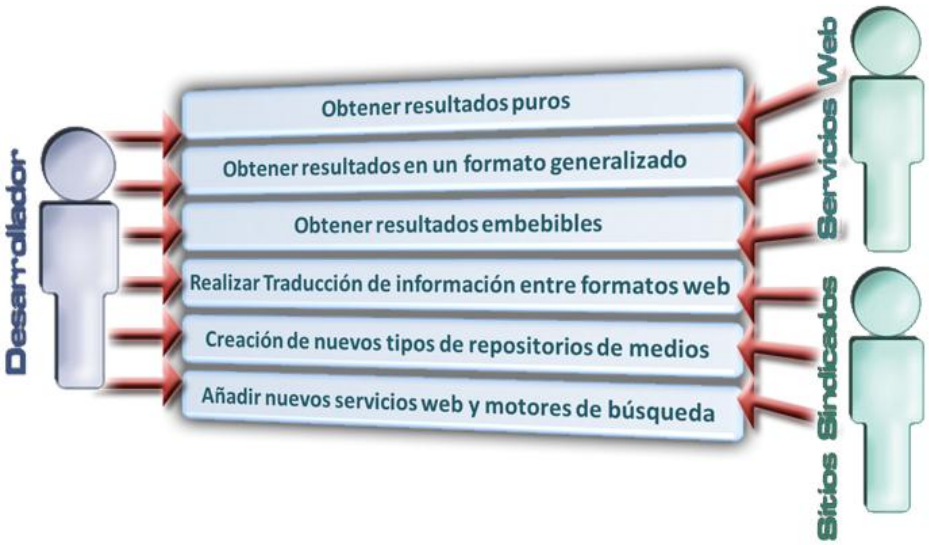


Figura 3.1: Diagrama de casos de uso de FindJira Framework.

Veamos en detalle los casos de que permite el Framework:

- **Obtener resultados puros -** El framework proveerá al desarrollador la facilidad de obtener los resultados de los servidores de búsqueda mapeados a un arreglo asociativo[46], así como también el texto de resultado nativo de cierto servicio permitiendo el sencillo manejo de los datos obtenidos de diferentes búsquedas.

- **Obtener resultados en un formato generalizado** - Gracias a la utilización de un repositorio en un formato común[47], los desarrolladores que utilicen el Framework podrán obtener resultados de medios y fuentes diferentes en un mismo formato para utilizarlos de manera eficiente y sencilla. A este formato se le ha denominado "formato FindJira" debido a la convención que se utiliza en este framework.

- **Obtener resultados embebibles -** Los repositorios de medios proveerán, de manera inteligente, los resultados listos para ser insertados en el web. Con un simple comando de impresión, los desarrolladores podrían incluso crear arreglos dinámicos enteros de medios distintos. Cabe recalcar que entre los resultados siempre se podrá acceder al campo de vista previa o thumbnail

[46] Un arreglo asociativo es un arreglo común, en donde cada elemento del mismo posee una clave o identificador, numérico o textual.

[47] Un repositorio es un lugar central donde la información es almacenada y mantenida.

- **Realizar Traducción de información entre formatos web** - El Framework proveerá funciones para decodificar algunos de los diferentes formatos web antes vistos, como JSON, XML y RSS, así como los formatos de todos los servicios disponibles soportados por el mismo para luego ser llevados al formato FindJira.

- **Creación de nuevos tipos de repositorios de medios** - Gracias a las propiedades de la herencia[48], será sencillo para un desarrollador añadir un nuevo tipo de medio al sistema, y crear una base de medios definidos por el desarrollador para un uso en especial, actualmente se proveen de los medios básicos que son: texto, imágenes, audio, video y archivos de propósito general.

- **Añadir nuevos servicios web y motores de búsqueda** - De manera similar al caso anterior, un desarrollador con conocimientos básicos de orientación a objetos no tendrá dificultades para añadir un nuevo motor de búsqueda al sistema, con las funciones de conexión pertinentes. Tan sólo siguiendo la documentación y conociendo el formato y campos de respuesta del nuevo servicio logrará añadirlo al framework.

[48] La herencia es una propiedad del Paradigma de Orientación a Objetos que permite que una clase esté basada en las propiedades de otra, evitando el desarrollo de código duplicado.

3.2.2 Actores del Sistema

Los actores del sistema son los entes que van a interactuar de una forma u otra con el Framework, a continuación se describen estos actores para esclarecer su papel en el sistema, y la forma de interacción con cada uno.

- **Usuario: Desarrollador** - Este actor utilizará las funciones que proporciona FindJira framework listadas anteriormente entre los casos de uso, para él será transparente lo que suceda en el interior del framework, como la comunicación con los servicios y la interpretación de resultados.

- **Servicios Web** - Son los servidores que ofrecen los servicios web antes mencionados, su trabajo es recibir una petición de búsqueda con palabras claves entre otros parámetros y devolver una respuesta con los resultados de su base de datos de índices. Cada servicio es un actor en el sistema, e interactúa con el mismo sólo durante el corto lapso de obtención de información, luego, este enlace es roto y todos los datos son interpretados por FindJira. Otros servicios web no orientados a búsqueda como los servicios orientados a scraping, los servicios de mapas y los servicios de vistas previas de páginas también pueden ser soportados por una aplicación desarrollada con el Framework.

- **Sitios Sindicados** - Son todos los sitios que provean un feed tanto RSS o ATOM, en cualquiera de sus versiones. Pueden ser considerados actores

al proveer de la información necesaria para que el framework pueda leer cualquier información disponible de dicho sitio.

3.2.3 Usuarios finales: Desarrolladores

FindJira framework está orientado a los desarrolladores que tengan la iniciativa de generar aplicaciones web multimedia interactivas, sin la necesidad de aprender los APIs de los servicios web más populares. En el siguiente capítulo se explicará la implementación del framework, pero por ahora se puede adelantar que estará implementado utilizando el lenguaje web PHP versión 5, este lenguaje ha sido seleccionado por su versatilidad y flexibilidad en la programación, por su capacidad de orientación a objetos y así como también la gran cantidad de recursos publicados dentro y fuera de la red.

El usuario interactuará con el framework a través de una clase llamada handler que en la siguiente sección de diseño se explicará con todo el detalle del caso. Esta clase le proveerá al usuario una interfaz con el framework en la cual podrá realizar las siguientes acciones:

- Setear los parámetros necesarios dependiendo del tipo de fuente de información a utilizar. Dado que el propósito es facilitarle el trabajo al usuario, para realizar búsquedas únicamente deberá enviar las palabras

clave que se desean buscar a través de todos (o parte) de los servicios residentes en el framework. Otros parámetros el desarrollador debe enviarle a la interfaz cuando la fuente de información es un servicio web son:

  - El número de resultados por página.
  - El índice del resultado inicial de todos los resultados almacenados.
  - Resolución de la imagen de la vista previa o thumbnail.
  - Resolución del objeto embebible, el cual se aplica no a todos los tipos de medio, solo a imágenes, video, audio y mapas.
  - Parámetros de búsqueda avanzada: Restricción de sitio web, título de la página, texto en el URL de la página, tipo de archivos y texto no deseado en la búsqueda, si un servicio determinado los soporta.

- En el caso de un feed, será necesaria la dirección del mismo para realizar la extracción de información.

- Enviar la orden de realización de la búsqueda. Una vez que todos los parámetros hayan sido definidos.

- Extraer resultados de cada tipo de medio.

En cambio si lo que se desea es agregar nuevos servicios a la funcionalidad del framework, es posible para el desarrollador hacerlo si conoce el lenguaje PHP orientado a objetos y los siguientes aspectos sobre el servicio que desea ingresar:

- URL de petición al servicio web usando la interfaz REST.
- Conocer el formato estructurado de la respuesta de dicho servicio, básicamente saber si retorna XML o JSON.
- Cualquiera que fuera el formato, es necesario encontrar la relación entre los campos de la respuesta con los del formato FindJira, la mayoría de estos campos son comunes pero con nomenclatura diferente que utiliza cada servicio. Una vez encontrada dicha relación se procederá a la asignación entre campos hacia el formato FindJira.
- Incluir en la clase handler, la interfaz del framework, el objeto que hace referencia al nuevo servicio agregado, siguiendo los patrones de los servicios existentes.

3.2.4 Descripción de los servicios del sistema

- **Servicio de Yahoo -** Con este servicio se realizarán búsquedas web, de imágenes y videos. Se utilizará la interfaz REST para realizar las

peticiones al servidor. Se prefiere la respuesta con formato JSON cuyo arreglo será recorrido y traducido al formato FindJira.

- **Servicio de SearchMash** - Con este servicio se realizarán búsquedas web, de imágenes, videos, de blogs, wikis y una búsqueda especial parametrizada para obtener resultados web con directorios de archivos generales. Se utilizará la interfaz REST para realizar las peticiones al servidor. La respuesta obtenida siempre será JSON cuyo arreglo será recorrido y traducido al formato FindJira. Con este servicio también se hará uso del código HTML para realizar embeds de los reproductores de video de Google Video y YouTube.

- **Servicio de YouTube** - Con este servicio se realizarán búsquedas de videos. Se utilizará la interfaz REST para realizar las peticiones al servidor. La respuesta obtenida siempre será XML cuyo arreglo será recorrido y traducido al formato FindJira. Con este servicio también se hará uso del código HTML para realizar embeds del reproductor de video de YouTube.

- **Servicio de Flickr** - Con este servicio se realizarán búsquedas de imágenes. Se utilizará la interfaz REST para realizar las peticiones al servidor. Se prefiere la respuesta con formato JSON cuyo arreglo será recorrido y traducido al formato FindJira. Con este servicio también se harán peticiones secundarias para obtener meta datos de cada resultado

como nombre del archivo, formato, resolución, etc.

- **Servicio de Microsoft Live** - Con este servicio se realizarán búsquedas web, de imágenes y de feeds. Se utilizará la interfaz REST para realizar las peticiones al servidor. La respuesta obtenida siempre será XML cuyo arreglo será recorrido y traducido al formato FindJira.

- **Servicio de Technorati** - Con este servicio se realizarán búsquedas de blogs y de feeds. Se utilizará la interfaz REST para realizar las peticiones al servidor. La respuesta obtenida siempre será XML cuyo arreglo será recorrido y traducido al formato FindJira.

- **Servicio de Odeo** - Con estos servicios se realizarán búsquedas de podcasts. Se utilizará la interfaz REST para realizar las peticiones al servidor. La respuesta obtenida siempre será XML cuyo arreglo será recorrido y traducido al formato FindJira. Inclusive se utilizará un reproductor integrado al framework para reproducir cada resultado.

- **Servicio de iTunes** - Con estos servicios se realizarán búsquedas de podcasts. Se utilizará la interfaz REST para realizar las peticiones al servidor. La respuesta obtenida siempre será JSON cuyo arreglo será recorrido y traducido al formato FindJira. Inclusive se utilizará un reproductor integrado al framework para reproducir cada resultado.

- **Servicio de lectura de feeds** - Este servicio realizará la lectura e

interpretación de la dirección de un web feed formato RSS o ATOM, la misma que será traducida al formato FindJira de similares características al de los repositorios, solo que esta vez contendrá información de los posts del feed.

- **Servicios de creación de mapas interactivos** - Estos servicios proveerán la facilidad de inserción de uno o más mapas interactivos, sean estos de Yahoo o de Google. Utilizan el motor de búsquedas de localidades de Yahoo, y permiten a los desarrolladores añadir varios elementos o hitos a un mapa, con descripciones en formato HTML. Es importante recalcar que los hitos en los mapas deben tener una dirección específica para ser buscados con dicho motor, en caso de no ser así, pueden ser añadidos individualmente al tener información detallada de longitud y latitud.

- **Servicio de scraping de archivos de directorios** - Se utilizará la técnica de scraping para obtener el listado de archivos en un directorio público hacia un arreglo asociativo de similares características al formato FindJira. Este arreglo podrá ser luego filtrado por extensión de cada archivo como por ejemplo seleccionar sólo los archivos de audio extensión MP3 y generar un playlist de formato XSPF listo para su reproducción, así podremos reproducir de manera continua todas las MP3 de un directorio público.

3.2.5 Ubicación del Framework

Desde el usuario objetivo que se encuentra al final de la cadena, es decir la persona que necesita información y realizará las búsquedas a través de una interfaz que utilice FindJira framework, hasta el internet que es la fuente de toda información en línea, los datos viajan a través de varias etapas a las que se denominan capas, estas capas se comunican entre sí y a su nivel, el mismo que cada vez es más bajo. FindJira framework actuará en la mitad del camino, como puente entre el desarrollador y hacia la comunicación con los APIs de los servicios web que indexan contenidos de internet.

Figura 3.2: Ubicación del framework entre el usuario y el internet.

- **Usuario** - Usuario meta al que estará dirigido todo servicio que entrega información, está constantemente informado de los servicios que se encuentran en la red y es su decisión cual le conviene más para obtener la información que necesita.

- **Aplicación -** Es una aplicación que utilizará el FindJira Framework y se comunicará con el usuario. La tecnología que mejor se ajusta lo que se necesita para crear un ejemplo que demuestre el potencial del FindJira Framework es Ajax. Esta capa es la que implementará el desarrollador que utilice el framework.

- **FindJira Framework -** El marco de trabajo que obtendrá la información de algunos servicios y la tendrá disponible para que sea utilizada oportunamente a través de una interfaz, este capítulo se concentrará en explicar su diseño.

- **Internet** - La fuente de todo tipo de información en línea.

- **Web API - I**nterfaces que ofrecen compañías para acceder a los resultados de búsqueda, el API de cada servicio es único y están orientadas a desarrolladores.

- **Servicios Web -** Son las compañías que brindan servicios de búsqueda a través de toda la red, las cuales se explicaron en detalle en el capítulo anterior.

3.2.6 Diagrama físico del sistema

Es necesario tener una perspectiva visual de los elementos físicos que interactúan entre sí a través de las redes que conforman el internet. Los elementos relacionados e involucrados en la participación de entregar información al usuario se describen en la figura 3.3.

Figura 3.3: Ubicación física del framework.

- **Usuario -** Un usuario que visite un sitio basado en el Framework. Describiendo brevemente este gráfico, podemos ver que dicho usuario tiene acceso a la red, la cual está representada con una nube de información. Al mismo tiempo, este usuario interactuará con una

aplicación hecha en el framework, representada con los engranes del gráfico, y la cual puede ser una interfaz, como una homepage, algún buscador integrado, e incluso un servicio ejecutado desde el equipo del usuario mismo.

- **FindJira** - El Framework se comunicará a su vez con los diferentes servicios web, como es conocido, a través del Internet, y retornará los resultados obtenidos al usuario. Es decir, por medio del Framework, es posible que el usuario se pueda comunicar con los servicios web, sin interactuar con estos de manera directa.

- **Servicios web** - Ellos siempre están disponibles para ser utilizados por la infinidad de mashups que existen en la red, no les interesa saber realmente a quién le entregan la información sino llevar sus estadísticas de preferencia entre el público. En este caso particular estarán atentos a recibir una petición de búsqueda y entregar su respuesta por la misma vía.

3.2.7 Flujo de Datos

Anteriormente se mencionó que la información viaja a través de varias capas, esta misma información deberá en cada etapa transformarse a un nuevo nivel que le corresponderá interpretar la siguiente capa, en esta sección se explicará el flujo que siguen estos datos desde que son

invocados. Cada una de las etapas en el flujo representa un modo de interacción entre capas.

Figura 3.4: Flujo de datos desde y hacia el framework.

- **Indexación de las Fuentes de Información** - Como se ha descrito anteriormente, cada fuente indexa sus propias bases de datos, es decir, cada servicio web se encarga de ordenar y clasificar los distintos medios disponibles. El acceso a estos datos clasificados es por medio del API de desarrollo.

- **FindJira Framework** - El framework constará de módulos para la interpretación de respuestas, clasificación de resultados por tipo de medio y servicio, almacenamiento de multimedia en repositorios dedicados y comunicación con los API de los servicios soportados. En este punto los datos de los servicios web se analizan y se transforman para formar parte del repositorio acorde a su género.

- **Comunicación** - Las peticiones deben viajar en un formato, y las respuestas con los resultados en otro, todos estos protocolos deberán ser respetados y automatizados en esta etapa. El desarrollador conocerá las funciones que puede utilizar y es su responsabilidad el cómo utilizarlas.

- **GUI** - La parte más importante en el sentido de la interacción entre el hombre y la máquina. Para el desarrollo de este framework, se asume al usuario un internauta común. En este trabajo se desarrollará un ejemplo del uso de FindJira framework, siguiendo todas las pautas de interacción necesarias para generar un ambiente intuitivo, dinámico y amigable, en otras palabras, usable. FindJira Search representará el ejemplo a realizar como un portal de búsqueda en todos estos medios, es decir, se comportará como un Mashup de búsqueda web generalizada.

3.3 Diseño

El diseño es la fase más importante en el desarrollo de cualquier sistema computacional, en este punto se establecen las ideas surgidas a partir de la investigación y el análisis.

3.3.1 Modelo de Funcionamiento

El framework contará con un funcionamiento previamente diseñado de la siguiente manera: El usuario necesita realizar una búsqueda en el framework a través del GUI[49], es cuando se genera la petición y es enviada a los servidores de los servicios web, los mismos que retornan una respuesta estructurada, el framework obtiene dicha respuesta, la interpreta y almacena estos resultados en un repositorio de resultados. Una vez concluidas estas operaciones el usuario estará listo para realizar la extracción de resultados del repositorio los mismos que siempre estarán disponibles hasta realizar otra búsqueda. Todas estas operaciones son cruciales, y necesitan ser trabajadas de manera individual para cada servicio que el framework soporta. Gracias al enfoque orientado a objetos, que será descrito más adelante, será más sencilla la adición de nuevos servicios de comportamiento similar y de diferentes estructuras de retorno de resultados.

[49] GUI, Graphic User Interface o Interfaz Gráfica de Usuario, es el medio por el cual un usuario final interactúa con el sistema, principalmente con el uso de un ratón y teclado, mostrado los componentes conocidos como cuadros de texto y menúes, de manera intuitiva y sencilla.

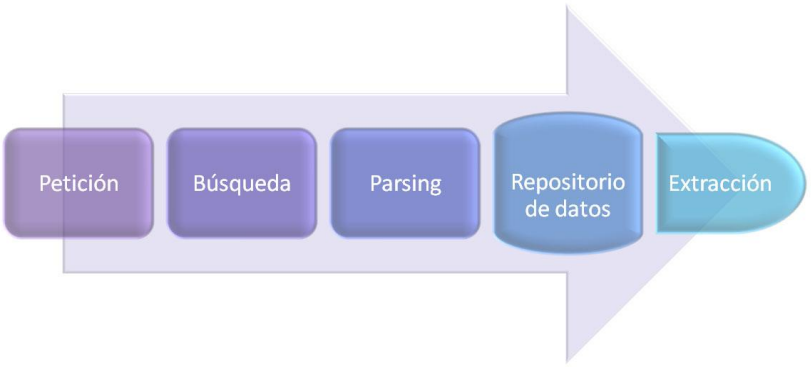


Figura 3.5: Modelo de funcionamiento del framework.

- **Petición -** En este módulo se generará la cadena de petición de búsqueda para los diferentes servicios soportados por FindJira Framework. Se procesarán todos los comandos y filtros antes explicados para las búsquedas diferentes de acuerdo a las especificaciones del usuario y al final se obtendrá un string listo para consultar en el servicio web respectivo. Cabe recalcar que cada servicio de búsqueda tiene parámetros únicos y, en algunos casos, las peticiones deben ser realizadas respetando los formatos al pie de la letra. En el caso de generar una cadena de petición errónea, el motor de búsqueda puede retornar al framework una búsqueda nula, es decir sin resultados, o puede retornar una estructura jerárquica similar a las respuestas, pero describiendo un error.

- **Búsqueda -** A este módulo le corresponde la interacción con los diferentes APIs de desarrollo, enviando peticiones a los servidores

respectivos y esperando las respuestas con los resultados pertinentes, estos resultados serán almacenados temporalmente en su formato de retorno nativo, los mismos que ya se explicaron en el Capítulo 1. Es importante resaltar que si algún servicio web llegase a fallar, y se obtiene una respuesta de error, no deberá ser tomada en cuenta. Para facilidad de desarrollo, se implementará una sola función de búsqueda utilizando la interfaz REST la cual es compatible con todos los servicios. De esta forma, un desarrollador no necesita implementar la interacción con un servidor nuevo, sino que simplemente debe modificar los parámetros respectivos en su servicio. Dado que cada servicio de búsqueda provee un límite de resultados al retornar, el Framework deberá proveer la opción de buscar más allá de dicho límite, para eliminar estas diferencias y trabajar con todos los servicios de manera unificada.

- **Parsing -** Cada módulo de parsing, o de interpretación, se encargará de decodificar la respuesta, sea esta XML, RSS o JSON, dependiendo de la estructura jerárquica de la respuesta de cada uno de los servicios a ser utilizados. Al final cada resultado se encontrará traducido al formato FindJira y esta lista de resultados será enviada al manejador del sistema, y consecuentemente, al repositorio. Además de interpretar los campos que se incluyen en la respuesta nativa, este módulo se encargarán de incluir ciertos valores adicionales necesarios para la utilización en el web de estos medios, como por ejemplo, reproductores embebibles o vistas

previas de sitios web.

- **Repositorio de Datos -** Es aquí donde se podrá encontrar el resultado de minar los datos de todas las diferentes fuentes, en un esquema organizado, que permitirá su sencilla extracción para añadirlos como contenido web a los sitios que utilicen el framework. Es posible añadir más tipos de medio al repositorio, e incluso crear repositorios de varios medios en conjunto. Este comportamiento dependerá de la interfaz del framework, que se encargará del manejo inteligente de la información dentro de cada repositorio. Una abstracción de los tipos de medios más conocidos que serán contenidos en el framework como repositorios independientes se aprecian en la figura 3.6.

Figura 3.6: Ejemplo de un repositorio que puede soportar FindJira Framework.

- **Extracción -** Una vez obtenidos los datos, y almacenados en el repositorio respectivo, será posible el uso de las funciones para extraer y consultar elementos del mismo. De igual manera, encontraremos las funciones que permitirán crear objetos multimedia, como reproductores y listas de reproducción, a partir de los datos en formato embebible de los repositorios, para uso sencillo de los sitios que utilicen FindJira framework. En general, son las funciones de extracción del repositorio los que serán invocadas por el desarrollador que use el framework para su propia solución a través del uso de un manejador que será la interfaz del sistema.

3.3.2 Diseño Estático del Sistema

El sistema del Framework poseerá tres elementos principales: manejador, repositorio y servicios. Cada uno de estos elementos agrupará algunas de las funcionalidades antes descritas, de manera sencilla para el uso de un desarrollador. En esta sección se describirá en detalle cada uno de estos grandes módulos, y los métodos que contiene cada uno, para así esclarecer el comportamiento interno del Framework. En la figura 3.7 se muestra este esquema de clases de manera más comprensible.

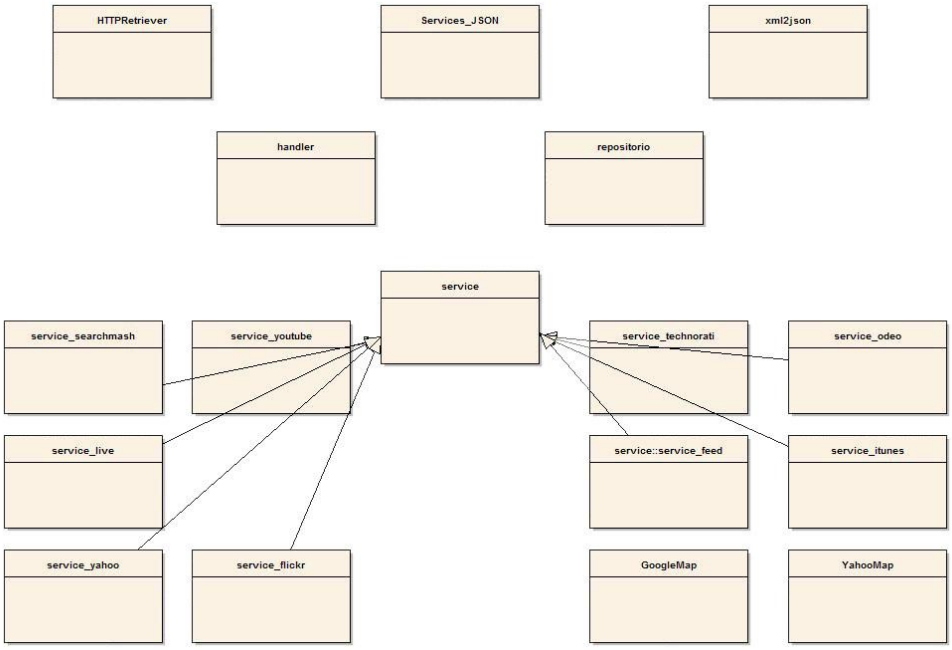


Figura 3.7: Diagrama Simplificado de Clases.

3.3.2.1 Clase Manejador

El manejador o handler es una clase que proveerá de una interfaz al desarrollador que desee utilizar el framework para crear sus aplicaciones web. Estas interfaces permitirán que los procesos internos se ejecuten de forma transparente para el usuario sin preocuparse de revisar lo implementado.

En el constructor del manejador se inicializarán todos los objetos que representan a los servicios web presentes y los repositorios de los diversos tipos de medios. Además se

configurará cada servicio con su máximo número de resultados por página permitido por su correspondiente API.

Se contará con un método que preparará la búsqueda, en la cual se configurarán todos los parámetros de búsqueda, sean avanzados o no. Como se mencionó anteriormente estos parámetros son:

- Palabras clave relacionadas a un tema a buscar en el framework.
- El número de resultados por página a mostrar.
- El índice del resultado inicial de todos los resultados almacenados.
- Resolución de la imagen de la vista previa o thumbnail.
- Resolución del objeto embebible, el cual se aplica no a todos los tipos de medio, solo a imágenes, video, audio y mapas.
- Búsqueda en servicios, con el número de resultados obtenidos no limitados por los mismos.
- Parámetros de búsqueda avanzada que únicamente se aplican a Yahoo y SearchMash.
    - Restricción de sitio web.
    - Título de la página.
    - Texto en el URL de la página.

- Tipo de archivos.
- Texto no deseado en la búsqueda.

El siguiente método es realizar la búsqueda, el mismo que realizará de forma simultánea o independiente para cada tipo de medio presente en el handler. La búsqueda que es realizada dentro de cada medio involucra a todos los servicios afines a dicho tipo de medio, estos servicios depositarán su respuesta en el repositorio correspondiente. Para citar un ejemplo, si se ejecuta el método de buscar imágenes, los servicios de Yahoo, SearchMash, Live y Flickr realizarán la búsqueda y depositarán sus resultados en el repositorio de imágenes.

En esta clase se implementa el método de buscar más allá del límite provisto por cada servicio y tratarlos a todos de manera similar, este método realiza búsquedas consecutivas mediante una iteración y todos los resultados se agregan al repositorio general del tipo de medio correspondiente.

Si se desea obtener información de feeds RSS o ATOM, en cambio, es necesario realizar una serie de pasos similares, es

necesario inicializar un objeto del servicio de feeds, el cual es compatible con todos los formatos disponibles. A este servicio se le debe especificar el URL del feed a utilizar. En caso de no conocer la URL del feed en cuestión, es posible llamar a un método que descubra de manera automática si un URL posee feeds relacionados.

De manera similar a los servicios web de búsqueda, en el caso de los feeds se recupera la información del mismo. Este método es genérico y permite obtener un feed en un formato generalizado. Cada post en un feed es un elemento de repositorio, de esta forma, obtenemos la información de los posts en el orden de lectura de cada feed.

3.3.2.2 Clase Servicio

Debido a que existen estas distintas formas de manejo para cada uno de los servicios, existirá una clase "Servicio", la cual poseerá todas las funcionalidades comunes para los servicios, y dará herencia a otras clases para los diferentes motores soportados en esta versión inicial del Framework, las cuales poseerán los detalles necesarios para el óptimo manejo de cada uno de los motores.

Estas clases heredadas de "Servicio" estarán implementadas de manera expandible, es decir, que para añadir un nuevo motor al framework, o funcionalidades nuevas a los motores ya soportados, no se necesitará más que añadir otra clase con las mismas propiedades de "Servicio", o la funcionalidad respectiva a uno ya existente.

En el constructor de la clase se inicializan los atributos de esta clase para dejarlos en blanco, excepto por los que representan la paginación de resultados y la resolución de vistas previas y objetos embebibles, los cuales se inicializan en valores fijos. El campo que guarda la dirección del servicio de vistas previas de páginas web también se inicializa con un servicio pre-definido.

Las funciones de asignación y obtención (get y set) de los principales parámetros de esta clase involucra: El query o palabras claves de búsqueda, resolución de la vista previa y del objeto embebible, URL del servicio de vistas previas de páginas web, el número de resultados por página que se utilizará para realizar la petición así como el máximo número de resultados por página que permite el servicio, el número de página de resultados inicial que se desea en la respuesta y el índice del resultado inicial que será devuelto igualmente.

Un método sumamente importante es el que realiza la conexión, para efectos mnemotecnia éste método se llama "buscar", su única función es recibir un URL bien definido y navegar a esa dirección obteniendo la respuesta del sitio a esta petición hacia una cadena de texto, éste método retorna esta cadena que contiene, para efectos del framework, la respuesta de un servicio web.

Cada clase que hereda de la clase servicio, es decir, cada servicio web, implementarán métodos de búsqueda de los diferentes tipos de medio que soporta dicho servicio: web, imágenes, video, blog, etc. Dentro de cada uno de éstos métodos se utilizará a su vez un método auxiliar que genera el URL de petición acorde al tipo de medio y luego se capturará en un atributo de la clase a la respuesta en su formato nativo haciendo uso del método "buscar" antes mencionado. Al final de éste método se interpretará y traducirá esta respuesta nativa hacia el formato FindJira que se guardará en un atributo de clase, para esta traducción se utilizará un método de parse.

Los métodos de parse, uno para cada medio descritos en el párrafo anterior, decodificarán la respuesta nativa desde JSON, XML o RSS hacia un arreglo asociativo del lenguaje PHP. Este arreglo será recorrido y con el conocimiento previo de la estructura de la respuesta se realizarán las asignaciones pertinentes generando el formato FindJira. Los campos del formato FindJira serán descritos cuando se detalle la clase Repositorio. Para poder completar la mayor cantidad de campos posibles para el estándar FindJira, será necesaria la utilización de métodos auxiliares, ya que todas las respuestas de los servicios no son exactamente iguales y algunas carecen de campos o los mismos pueden ser obtenidos a partir de otros campos de la misma respuesta. Para cada servicio inclusive es necesaria la utilización de métodos auxiliares exclusivos para el servicio, estos métodos al ser muy técnicos serán detallados en el siguiente capítulo. A continuación se explicará cada una de éstas funciones auxiliares de la clase padre Servicio:

Obtener web thumbnail - Retorna el URL correspondiente a la imagen vista previa de un sitio web, es necesario conocer de un servicio que se encargue de esto y también el URL base para

la formación de este thumbnail deberá estar inicializado como atributo de clase.

Obtener información de URLs - El framework posee algunas funciones adicionales de manejo de cadenas. Dado un URL estos métodos retornan el nombre de un archivo, su extensión, su dirección de dominio, o la ruta donde están alojados, estas funciones triviales son necesarias para obtener ciertos tipos de información de los archivos encontrados.

Obtener imagen embebible - Retorna una cadena de caracteres correspondiente al código HTML que incrusta una imagen. Este método recibe la ubicación de la imagen y la dimensión deseada.

Obtener javimoya video download - Este método retorna el URL para poder descargar un video de formato streaming (YouTube, Google Video) utilizando el servicio de www.videodownloader.net autoría de Javimoya. Es importante recalcar que el URL retornado no corresponde a la descarga directa del archivo sino un link a la página antes mencionada donde se podrá descargar el video deseado.

Obtener parámetros de un URL - Retorna el contenido asociado a un parámetro de un URL que va ser utilizado en una interfaz REST. Este método recibe el URL a ser analizado y el nombre del parámetro a ser retornado. Por ejemplo: si se envía "http://www.misitio.com/?param1=a&param2=b&param3=c" y se desea el contenido de param2 entonces se retornará "b".

Obtener reproductor mp3 embebible - Retorna una cadena de caracteres correspondiente al código HTML que incrusta un reproductor mp3. Este método recibe la ubicación del archivo de audio y la dimensión deseada.

Generar lista XSPF - Retorna una cadena de caracteres correspondiente al código XML / XSPF que lista los archivos de audio que forman parte del playlist. Este método recibe un arreglo de elementos con formato FindJira (no son necesarios todos los campos, sólo título, thumbnail y ubicación) y la cadena retornada puede ser utilizada para generar un archivo físico o mostrarlo en pantalla.

Obtener archivos de un directorio - Esta función realiza un web scraping del URL recibido, abre ésta página web, analiza el contenido y retorna un arreglo con el formato FindJira (sólo en campo de ubicación para su descarga) con los archivos de

cierto formato encontrados en este directorio. Para el análisis éste método se basa en que los directorios mantienen una misma estructura, se empieza a analizar y extraer los hipervínculos después de la ocurrencia del texto "parent directory", los que representan la lista de archivos alojados en el directorio en cuestión. Al final, a cada archivo se le antepone como prefijo la ruta del directorio para que en arreglo a ser retornado contenga elementos con ruta completa, completamente útil si es necesaria su reproducción.

Obtener el icono representativo de un sitio - Este método obtiene el favicon, o ícono de cualquier página web, si existe. Recibe la dirección de una página cualquiera y se encarga de obtener el dominio y la dirección de la imagen en cuestión.

Obtener los feeds de un sitio - Esta función se encarga de descubrir automáticamente la existencia de feeds RSS o ATOM en un sitio web. Recibe como parámetros el URL que se desea examinar, y esta función retornará, en caso de que existan, las direcciones de los feeds para dicha página en un arreglo.

Dependiendo de cada servicio, si el modo de trabajo con el API del mismo lo requiere, existirán funciones adicionales a las descritas anteriormente. Por ejemplo, el API de Flickr, en el cual es necesario un segundo llamado al servicio para obtener información adicional de una imagen.

Para optimización de uso de recursos, una vez que cada objeto encargado de un servicio haya cumplido su función, es decir, una vez que haya cumplido su ciclo de vida obteniendo, analizando y almacenando las respuestas del servicio al que representa, será eliminado del sistema.

Los feeds tienen un manejo bastante similar a los servicios web. La diferencia de la clase de manejo de feeds es básicamente la lógica del servicio. Mientras con un buscador se obtiene un conjunto de resultados, en un feed se extrae el contenido del mismo. Así mismo, mientras los resultados de un servicio web están ordenados por relevancia, los resultados de un feed estarán ordenados por fecha de publicación, con los posts más recientes al inicio. Los servicios de feeds serán casi idénticos a los de los servicios anteriores, pero por este cambio de contexto

lógico, las funciones de buscar serán llamadas funciones de extracción.

Es importante recalcar, que el servicio de feeds deberá interpretar muchos tipos de estándares, sean estos ATOM, RSS o una versión especial de alguno de ellos. Algunos tags en estas versiones son diferentes, pero para facilidad de desarrollo, serán trabajados de igual manera. Existe una capa adicional que procesa estos feeds, creando un arreglo de resultados unificado, y esta capa se comunica con el servicio de feeds para convertir estos datos a formato FindJira.

3.3.2.3 Servicios de mapas

Los servicios de mapas poseen un sistema de manejo distinto a los servicios anteriormente descritos. Para la creación de un mapa independiente, sea este de Yahoo o Google, es necesaria la creación de un objeto de servicio de mapa individual. Esto se debe a que a diferencia de los servicios anteriores, los mapas necesitan tener una persistencia en el framework, y cada servicio de mapa provee funciones de agregación y manejo de

hitos, que deben estar relacionadas a un único mapa. Es posible crear objetos de servicios de búsqueda individuales para cada búsqueda a realizar, pero este manejo de objetos no es estrictamente necesario en los buscadores y en los feeds, como lo es con los mapas.

Los servicios de mapas proveen funciones para trabajar con los puntos geográficos en cada mapa y permiten añadir información especial a los mismos utilizando las funciones de sus APIs respectivos. A diferencia de las simples peticiones que se pueden encontrar en los servicios de búsqueda, los servicios de mapas interactúan con el framework por medio del uso de funciones. Es decir, un objeto de servicio de mapas, del framework maneja un objeto individual de mapas de un API en específico.

Para ambos servicios de mapas soportados, como ya se especificó, el manejo es similar. El uso de funciones JavaScript es obligatorio en estos APIs de desarrollo, al ser necesario manejar un objeto de mapas. Las funciones del framework

hacen este manejo imprimiendo código JavaScript de manera similar a imprimir código web. Es por esto que cada objeto de mapa debe ser inicializado siempre al inicio de una página web. Estas funcionalidades están limitadas al uso que los APIs de Yahoo y Google proveen, pues una vez mostrado el mapa, éste debe ser recargado para cada modificación. Es posible añadir hitos con información HTML en ellos para luego ser mostrados en el mapa.

También es factible añadir un hito al proveer una dirección, esto sólo es válido para direcciones viales en la base de Yahoo Geocode Search.

Los métodos necesarios para poder embeber un mapa en un sitio utilizando el framework son los siguientes, en su orden de uso en un sitio:

Imprimir Código en JavaScript - Esta función incluye el JavaScript necesario de cada servicio de mapas en una página web para poder llamar a las funciones provistas por los APIS de mapas. Esta función debe ser llamada al inicio, entre los tags HTML <head> y </head>.

Centrar mapa - Esta función centra el campo de visión del mapa a una longitud y latitud específica, con un nivel de zoom dado. Una restricción de los APIs de mapas es que en caso de no tener puntos válidos en un mapa, éste debe tener una posición inicial, o no será dibujado. Es decir, que si no se han especificado puntos válidos, está función deberá llamarse de manera obligatoria.

Añadir una dirección - Esta función utiliza el motor de Yahoo GeoPoint search para buscar direcciones con latitud y longitud. En caso de encontrarse una dirección vial en dichos resultados, éstos son añadidos a los puntos inválidos. En caso de no hallarse una dirección, los resultados obtenidos son añadidos como puntos inválidos.

Añadir un GeoPoint - Esta función es utilizada para añadir un punto con longitud y latitud específicas. Es posible enviarle como parámetro un código HTML que describa el hito en cuestión. Todos los GeoPoints son añadidos automáticamente a los puntos válidos.

Mostrar un Mapa - Esta función muestra el mapa en la página, con los parámetros ya especificados, y añadirá todos los hitos almacenados en los puntos válidos de manera automática.

Lógicamente, esta función deberá llamarse cada vez que se desee refrescar un mapa, una vez modificados los puntos válidos del mismo.

3.3.2.4 Clase Repositorio de Datos

Finalmente, se cuenta con un manejo unificado de información, la clase "Repositorio". Esta clase tendrá los métodos de inserción y extracción de resultados provenientes de cualquier servicio del framework y que mantengan el formato FindJira. Los campos del formato FindJira serán descritos un poco más adelante pero por ahora es importante citar que entre los mismos se encuentra su ubicación original o URL y el servicio o fuente de información de donde provino este resultado, por ejemplo Google, Yahoo o Technorati. Esta clase es general, por lo que las instancias de la misma representarán los repositorios para cada tipo de medio

En el constructor de la clase simplemente se indica que no se tienen elementos en el repositorio y que el atributo que los guardará está vacío. Es importante esta inicialización para

evitar problemas de sobre-escritura de datos antiguos. Aunque también es responsabilidad del handler hacer uso correcto de otro método que limpia y elimina los datos del repositorio actual.

Para insertar un resultado nuevo en el repositorio se verificará si el repositorio está vacío, de ser así, el nuevo repositorio contendrá el elemento recientemente insertado. Si el repositorio ya contaba con elementos, entonces se verificará si el nuevo elemento ya estaba contenido en la lista comparándolo con su identificador único que es su ubicación URL y en su campo origen se añadirá la fuente de información de donde provino, de este modo si ya se contaba con una fuente, ahora tendrá dos fuentes de origen, y así en lo sucesivo.

Como los resultados que nos proporciona cada servicio vienen ordenados por relevancia, este campo también es almacenado en el formato FindJira, y sirve para ubicar los nuevos resultados que se insertan en el repositorio. Para que todos los resultados se puedan posicionar en base a una secuencia de resultados es

imprescindible que repositorio almacene inicialmente los resultados de un servicio previamente configurado como "base". Cada nuevo resultado se posicionará al final del orden de relevancia de los existentes en el repositorio. Por ejemplo, si se ha realizado una búsqueda web, al inicio el repositorio contendrá todos los resultados de Yahoo, luego que se realiza la búsqueda en el servicio de Google, el "primer" resultado es repetido, es decir ya existía en el repositorio, no se lo inserta pero se modifica su campo origen citando la nueva fuente (Google) y el "segundo" resultado no existía en el repositorio, entonces deberá ubicarse a la cola de los resultados posicionados como "segundos". De igual forma cuando se realice la búsqueda en el servicio de Live, si su "segundo" resultado no estaba contenido en el repositorio se posicionará en la correspondiente cola, en este caso, al luego del segundo resultado de Yahoo y del segundo resultado de Google.

La técnica de ingresar el elemento será descrita con detalle en el siguiente capítulo. En todo caso, explicando el funcionamiento brevemente, el repositorio será segmentado en grupos de resultados de igual tamaño, entre los cuales será

insertado el nuevo elemento. Este será insertado al final del primer segmento o grupo de elementos, y una vez insertado el elemento entrante, todos los segmentos se volverán a integrar conservando el mismo orden. Al final de este proceso, se generarán nuevamente los índices de relevancia únicos para el repositorio, cabe recalcar que lo índices de relevancia adquiridos por servicio web nunca se alteran, y nunca deberán existir elementos repetidos en el repositorio.

Otro método auxiliar es una inserción masiva, en la cual se inserta un listado de elementos de formato FindJira. Este listado es recorrido y cada elemento es insertado siguiendo el algoritmo antes explicado.

Así mismo se tiene métodos para la extracción todos los resultados almacenados en el repositorio, si se desea extraer sólo una parte de ellos, se deberá indicar como parámetros los índices inicial y final que correspondan a la relevancia dentro del repositorio.

Existirá un solo tipo de clase "Repositorio", de la cual se heredarán los diferentes repositorios orientados a cada tipo de medio. De manera similar a una tabla simple, cada elemento del repositorio tendrá índices asociados indicando el orden del mismo en el repositorio, y cada detalle de un elemento contendrá una descripción del contenido del mismo.

3.3.2.4.1 Campos del Repositorio

Cada tipo de medio tendrá un objeto repositorio de similares características y contendiendo el arreglo de resultados respectivos. Se definirá un formato genérico para el repositorio, a continuación se describirán algunos de los campos a considerar para su creación, y el nombre que llevará cada campo en el Framework.

Titulo del resultado, "titulo" - El titulo que el servicio web le ha otorgado a cada resultado, normalmente proviene del título de la página web, el título del post en el caso de ser un blog, el nombre del video, el nombre del archivo de imagen, entre otros.

Ubicación externa URL, "raw_url" - Es la ubicación web donde se encuentra el archivo asociado al resultado retornado.

URL para visualizar, "display_url" - Es el URL sin muchos parámetros sólo con el propósito de visualización en una interfaz gráfica.

URL de caché, "cache_url" - Es el URL donde se encuentra almacenada una copia del archivo original, este campo es devuelto principalmente por servicios de buscadores que mantienen una copia de las páginas indexadas en sus bases de datos.

Feed, "feed" - Es la ubicación del archivo de sindicación en el caso que esté asociado al resultado original.

Descripción o Extracto, "resumen" - Es una porción del texto asociado a una página o artículo, por el cual se lo consideró como resultado. También puede ser la descripción de una imagen o video.

URL del thumbnail, "thumb_url" - Ubicación del archivo de la vista previa, a veces proporcionado por el

servicio y en otras utilizando el servicio externo que captura imágenes de sitios web.

URL del thumbnail (recortada), "crop_url" - Ubicación del archivo de la vista previa, es la misma imagen del campo anterior sólo que ha sido recortada proporcionalmente para no provocar una distorsión visual.

Thumbnail embebible, "embed_url" - Código HTML que es capaz de incrustar la vista previa.

Origen, "origen" - Una sola palabra para describir al servicio web de donde provino un resultado. Más luego en el repositorio, si más de una fuente es mezclada en un mismo repositorio, un resultado puede tener varias fuentes asociadas, estas fuentes serán escritas en este campo separadas con comas.

URL contenedor, "container_url" - Ubicación de la página web que contiene el medio encontrado si se trata de imágenes, videos u otros medios que requieran de la misma. Por ejemplo mostrar un video de YouTube, Google video o una fotografía de Flickr en sus respectivas páginas.

Tamaño del archivo, "file_size" - Tamaño en bytes del archivo encontrado, principalmente si se tratase de una imagen, no todos los servicios proveen de este campo.

Formato del archivo, "file_format" - Formato del archivo encontrado, principalmente para medios como imágenes, audio y video, aunque no todos los servicios proveen de este campo, puede ser obtenido extrayendo la extensión del archivo.

Resolución, "width" y "height" - Campos que describen la resolución original del medio visual como imágenes y videos.

Duración, "duracion" - Tiempo en segundos que dura la reproducción de un medio, este campo no es provisto siempre por los buscadores.

URL de descarga, "download_url" - Ubicación en la que es posible descargar el resultado encontrado.

HTML embebible, "embed" - Código HTML que es capaz de incrustar un reproductor de medios cargado con el resultado en cuestión.

Índice del servicio, "indice_ser" - Índice que representa la posición del resultado actual (en el servicio web del cual el resultado es originario) de entre todos los encontrados para las palabras clave buscadas. Este campo no podrá ser alterado.

Índice del repositorio, "indice_rep" - Índice que representará la posición del resultado actual en el repositorio de FindJira, este campo será alterado en el proceso de inserción de un resultado a la lista de elementos que contiene el repositorio.

Gracias a la flexibilidad del lenguaje PHP, si un desarrollador necesita otros campos para su lógica personalizada, es capaz de agregarlo sin mayor dificultad modificando el código fuente.

3.3.3 Dinámica del Sistema

Hasta este punto ya se ha revisado como se piensa implementar el sistema, y qué servicios son los que el Framework soportará. A continuación, se revisará como el sistema trabajará internamente. La secuencia en que los módulos del sistema interactúan se puede apreciar en la figura 3.8.

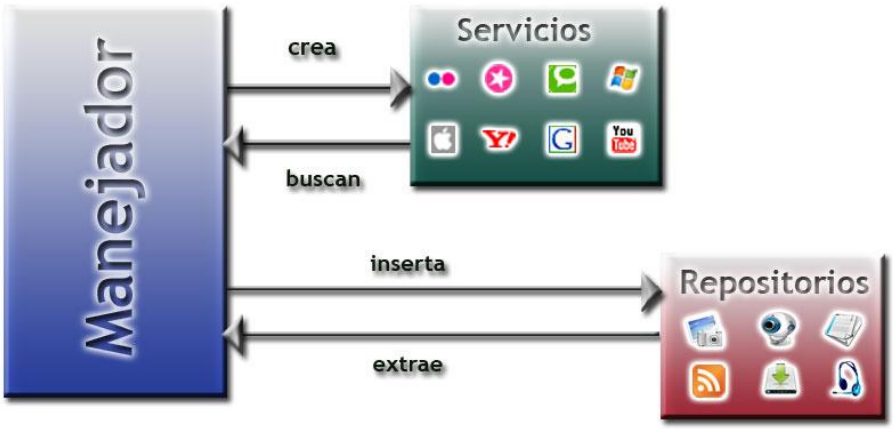


Figura 3.8: Diseño Dinámico del Framework.

Primeramente, para obtener resultados de los servicios web en formato puro, o raw, es necesario crear un objeto del servicio a minar en el código del handler. Éste objeto será el encargado de realizar los pedidos en el formato de dicho servicio. Por ejemplo, si un desarrollador desea realizar búsquedas en el servicio de Yahoo, este deberá crear un objeto del servicio respectivo, después de esto, deberá especificar el query de búsqueda, y opcionalmente, el número de resultados a obtener, o desde que resultado se desea obtenerlo, y si es requerido, especificar algún filtro de búsqueda provisto por Yahoo para la misma. Una vez hecho esto, el desarrollador deberá llamar a la función de búsqueda del medio correspondiente, es decir, deberá especificar si desea buscar imágenes, web, video, etcétera. Esta información es enviada al servicio, y una vez llamada la función de búsqueda, éste preparará con los datos provistos un

pedido o URL de request en el objeto de Yahoo. La respuesta retornada por Yahoo está en formato JSON, para el caso de la implementación del Framework, para otros servicios el formato puede variar. Para obtener esta respuesta en formato nativo, es posible llamar a la función respectiva del servicio, y se obtendrá una cadena de caracteres con el contenido sin procesar de la respuesta.

El objetivo del Framework, es el obtener una respuesta estandarizada. Es por esto que una vez realizados los pasos anteriores, es también posible obtener la respuesta en Formato FindJira, es decir, el formato estandarizado descrito en los apartados anteriores. Para crear la respuesta en este formato es necesario que cada módulo de cada servicio tenga implementado un traductor o parser de cada uno de los formatos propios al formato FindJira. Dado que estos resultados "raw" son simples cadenas, es necesario convertirlas de su formato de origen a un formato comprensible para el lenguaje, en este caso, un arreglo asociativo, o de niveles jerárquicos de PHP. Esta transformación ya fue descrita cuando se mencionó la clase de servicio, y depende, de librerías internas a PHP, o en su defecto, librerías externas para el manejo de formatos no soportados por este lenguaje.

Este resultado en formato FindJira es devuelto al handler para su manipulación. En este momento, la lógica del handler implementado en el Framework insertará este resultado en un objeto repositorio persistente creado previamente para cada tipo de medio. La inserción de cada una de estas clases de repositorio se encargará de mantener datos de las distintas fuentes organizados. Los datos en el repositorio, como se conoce, estarán listos para su inserción en la web.

El handler podrá, en este punto realizar más búsquedas, limpiar los repositorios o extraer información de los mismos. Para éste último caso, se utilizarán los métodos de la clase repositorio que podrán extraer todos los elementos o un rango de los mismos. Éstos resultados pueden entonces ser trabajados con una interfaz dinámica realizada en Ajax o alguna otra tecnología web para su inserción o uso apropiado en una interfaz.

Para el caso de los mapas, el handler simplemente invocará, en un orden dado, a las funciones descritas cuando se habló de dicho servicio en un orden dado. El handler entonces se comunicará con el servicio de mapas, el cual a su vez hará uso del API de su servicio respectivo, actualizando

sus puntos geográficos válidos, y una vez llenos estos datos previos, se mostrará el mismo llamando a la función respectiva. El handler podrá llamar otros mapas utilizando nuevos objetos de mapas, e incluso refrescar un mapa actualizado volviendo a llamar a las funciones que los muestran.

Conclusiones

Como podemos ver, el Framework provee una solución inteligente y moldeable a las necesidades de cada usuario y desarrollador. Esta solución permitirá obtener resultados organizados, y utilizará clases que permitirán trabajar con la información en línea de manera sencilla, además de comportarse de manera expandible y permitir adicionar módulos a la misma. La aplicación del Framework es tan vasta como la creatividad de los usuarios que lo utilicen, además de proveer una base para desarrollo de futuros marcos de trabajo basados en este, o que incluso, se creen nuevos marcos que mejoren sus falencias.

Hemos tratado el diseño en este capítulo para ayudar a esclarecer cualquier duda de la forma de trabajo y la forma de recopilación de información del Framework, pero no hemos tratado los detalles de la Implementación del mismo. En el capítulo siguiente podremos apreciar algunas de las funcionalidades del Framework, para entender su

funcionamiento detallado. Examinaremos, además, su uso con el desarrollo de un ejemplo interactivo, es decir, de una interfaz que utilice el Framework como base, para mostrar el potencial del mismo.

CAPITULO 4.

Implementación de FindJira Framework

Introducción

En los capítulos anteriores se describió la funcionalidad del Framework y su alcance, en el Capítulo 3 se describió el diseño del sistema, como éste interactúa con los servicios web diferentes y además se detalló cómo funciona internamente el mismo. En este capítulo se verán los detalles de la implementación del sistema, cómo trabaja el mismo con cada servicio, y qué módulos lo componen, al ver el diagrama de clases del mismo.

4.1 Herramientas y Lenguaje de Programación

Para la realización de este framework se ha dicho que se va a utilizar el lenguaje de programación web PHP versión 5. Pero ¿por qué PHP existiendo tantos lenguajes actualmente? PHP ha sido seleccionado debido a la gran cantidad de recursos publicados dentro y fuera de internet y muy buena calidad de soporte en todas partes, la flexibilidad que ofrece este lenguaje al momento de programar es muy grande, existe una gran cantidad de funciones integradas en las librerías nativas del núcleo de PHP que facilitan mucho el trabajo y optimizan recursos computacionales. Su fácil adaptación al motor de bases de

datos MySQL hace de este lenguaje más potente aún permitiendo crear aplicaciones web con sistemas de bases de datos de una manera sencilla en comparación con otros lenguajes web, que necesitan instalar complejos o pesados Frameworks. PHP además es multiplataforma y portable, el entorno de desarrollo puede ser fácilmente configurable gracias a herramientas que instalan todas las utilidades necesarias en una estación de trabajo como XAMPP[50] el cual instala entre otras aplicaciones al servidor Apache, necesario para ejecutar el código PHP, el motor de bases de datos MySQL con su aplicación de administración phpMyAdmin y el mismo PHP 5.

Con la utilización de PHP 5 se utilizará el paradigma orientado a objetos para el desarrollo de FindJira framework, ya que un buen diseño de clases, atributos y métodos permitirá una fácil expansión futura del sistema debido a que todo se encuentra dividido por módulos, organizado por capas y con una evidente relación entre los componentes del mismo.

4.2 Convenciones y Técnicas de Implementación

En la fase de implementación es necesario avanzar de forma secuencial al momento de trabajar con cada módulo, se empezó con el módulo de servicios y la mayoría de sus funciones que permiten la conexión con los diversos servicios

[50] http://www.apachefriends.org

web y su correcta traducción al formato FindJira. Una vez terminada esta implementación se procedió a desarrollar el repositorio cuya principal función es la inserción adecuada de un elemento entre los existentes siguiendo los patrones explicados en el capítulo anterior. Finalmente se desarrolló el handler que integra todas las funciones de interfaz del framework con el usuario que lo utilizará para la posterior generación de aplicaciones web.

Fue además necesaria la utilización de código fuente ya desarrollado para propósitos específicos y con licencia open source, el mismo que fue adaptado para su utilización en FindJira framework, los dos principales módulos que se utilizaron fueron:

- HTTP Retriever - Es una clase escrita en PHP que permite la conexión con un sitio web y obtener su respuesta, en una cadena de caracteres. Esta clase provee funcionalidades adicionales que no son necesarias de utilizar en la versión actual del framework.

- XML 2 JSON - Esta clase permite la conversión entre los formatos XML a JSON utilizando de por medio arreglos asociativos nativos del lenguaje PHP. Fue necesaria la utilización de esta clase para convertir las respuestas que vienen en formato XML hacia el formato FindJira.

- Validate Syntax - Esta clase permite validar cualquier tipo de cadena URL,

siguiendo una gramática que permite incluso URLs con direcciones de puertos o parámetros en el mismo.

- Unicode - Esta clase permite la conversión entre los formatos UTF8 y Unicode, para ser mostrados propiamente en el web. Permite la conversión de caracteres especiales Unicode, como caracteres asiáticos, a formatos de texto entendibles por los navegadores. De no realizarse estas conversiones, ciertos caracteres se mostrarían en las páginas como combinaciones ilegibles de caracteres romanos.

- BMP - Esta clase provee la funcionalidad necesaria para que PHP pueda manejar formatos de imágenes de mapas de bits. Es integral en la utilidad de recortar imágenes, que luego podrían ser usadas para generar vistas previas de resultados.

Para mantener la legibilidad del código fuente se han fijado una serie de convenciones y estándares a seguir, esto es realmente útil en el trabajo colaborativo y hará del mantenimiento y portabilidad del código una tarea más sencilla. Las convenciones de codificación establecidas son las siguientes:

- El nombre de los archivos de clases externas al sistema tendrán el prefijo "class_".
- El nombre de los archivos de clases del sistema conservarán el mismo nombre

de la clase, como es el caso de "service", "repositorio", "handler" y todas las clases heredadas de la clase servicio.

- El nombre de los archivos que representan alguna utilidad al sistema y la misma no está integrada a módulo alguno, tendrán el prefijo "útil_".
- Los métodos para obtención y asignación de valores en los atributos tendrán los prefijos "get_" y "set_" respectivamente.
- Los métodos para obtención de tipos abstractos de datos u otro tipo de procesamientos llevarán el prefijo "obtener_".
- Los métodos que forman parte del núcleo de la clase y que marcan el funcionamiento de la misma llevarán nombres representativos, por ejemplo "buscar".
- Todo método que manipule código HTML que será incrustado posteriormente contendrá la palabra "embed".
- Todo comentario será realizado con caracteres de comentarios de línea "//" mientras los comentarios de bloque "/* */" servirán para evitar que bloques de código se ejecute y su utilización se aplicará únicamente al código fuente.
- Cada método retorna lo especificado en la documentación, en caso de falla retornará nulo "NULL".

4.3 Implementación del Sistema

En el capítulo anterior se habló del diseño del sistema y de los grandes módulos que conforman el framework, estos son manejador, servicios y repositorio. Al tratarse de la implementación de un framework, se ha realizado un diseño de tal forma que se encuentre en la interfaz con el usuario todas las funciones necesarias para su utilización. En el capítulo anterior se describió la interacción de las clases repositorio y service con en handler y a su vez la herencia de cada uno de los servicios web existentes con la clase service.

Se pudo observar también que las clases auxiliares como "HTTPRetriever" y "XML2JSON" se encuentran separadas de todas las demás clases. Además, se representan todas las clases para cada servicio que soporta el framework como clases que heredan de "service". Los servicios de mapas geográficos son autónomos y funcionan independientemente por lo que únicamente serán llamados a ser ejecutados por el "handler".

4.3.1 Clase Manejador o Handler

El "**constructor**" de esta clase inicializará todos los atributos de la misma, en el caso que estos atributos sean objetos, pues se crearán las instancias de los mismos. Existirá un atributo global para un query genérico. Se definirán en esta clase todos los objetos de todos los servicios a ser

utilizados por el desarrollador. Los objetos de repositorios también deberán estar definidos en el handler.

Los métodos que implementa el handler son los siguientes:

El método "**limpiar repositorios**" destruye el contenido de cada repositorio creado en el constructor de la clase, dejándolos listos para ser llenados con nueva información de futuras búsquedas. Este método deberá ser llamado antes de realizar una búsqueda.

El método "**preparar búsquedas**" recibe como entrada todos los atributos de la clase con excepción de los objetos de repositorios y servicios, estos atributos se asignan y si es necesario dentro de cada objeto que realizará el servicio de búsqueda, por ejemplo: Yahoo y Google utilizan los parámetros avanzados como site, inurl e intitle. Para esto se utilizan los métodos auxiliares que asignan atributos masivamente a todos los servicios: "setear queries", "setear resultados por página", "setear resultado inicial", "setear dimensiones thumbnail", "setear dimensiones embed".

El método "**do_all_search**" ejecuta los métodos de búsqueda por cada tipo de medio. Estos métodos como por ejemplo: "**do_image_search**" realiza la búsqueda en los servicios que soportan imágenes los cuales son SearchMash, Yahoo, Live y Flickr, cada respuesta de estos servicios será

insertada en el repositorio de imágenes. Así para cada tipo de medio con los servicios que soportan a cada uno, los cuales son: do_web_search, do_feed_search, do_video_search, do_audio_search, do_wiki_search, do_indexof_search.

Los métodos de "**extraer repositorio**" para cada tipo de medio, retornan los elementos contenidos en el repositorio de dicho tipo de medio basado en los índices iniciales y finales que son obtenidos como parámetros de entrada de este método. Si no se especifican estos parámetros, pues se retornarán todos los elementos del repositorio solicitado. Si se requiere un comportamiento adicional a los provistos anteriormente por el framework, es la capa del handler la que deberá alojarlo.

Finalmente "**search_n_medio**" permitirá la búsqueda iterada para exceder los límites que cada servicio presenta para cada tipo de medio.

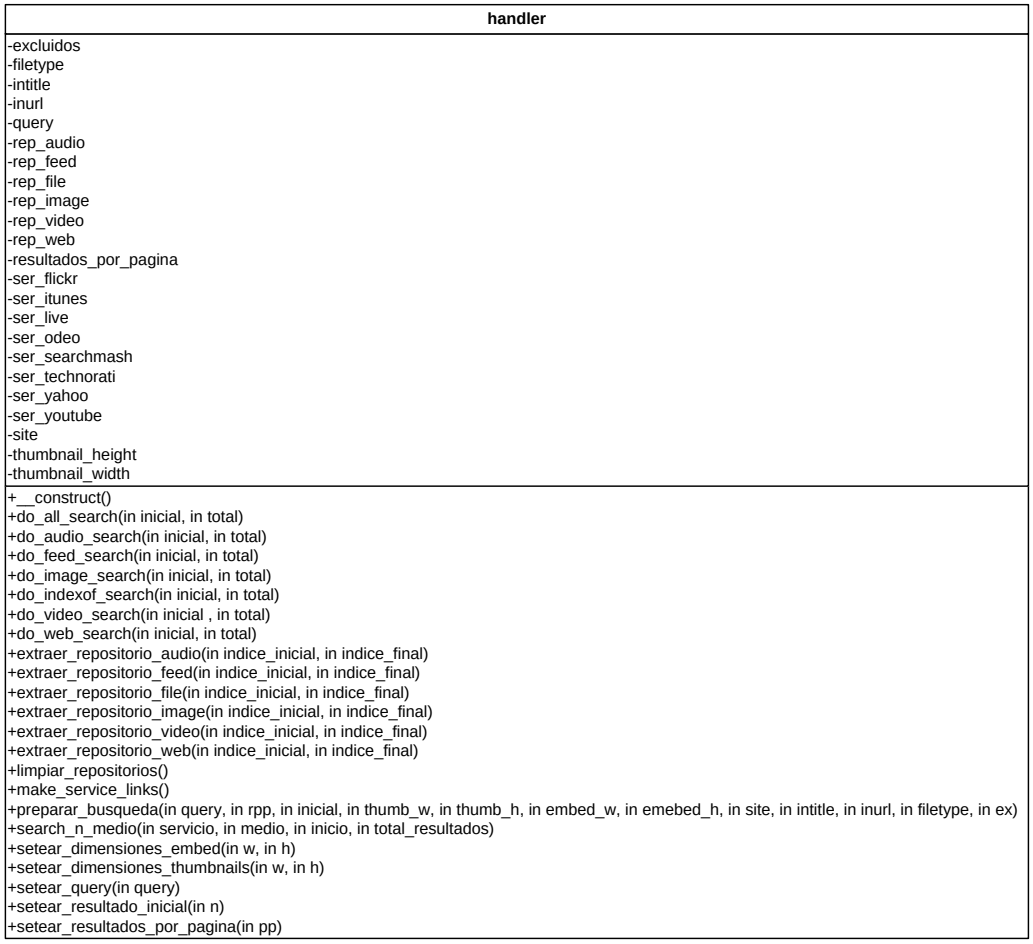


Figura 4.1: Clase Manejador.

4.3.2 Clase Servicio o Service

Como sabemos, ésta es la clase en donde el framework maneja las principales fuentes de información soportadas. Cada clase heredada de "service" tendrá sus atributos orientados a su servicio web

correspondiente. Los atributos antes descritos incluyen el query, la petición y las respuestas en formato raw, los thumbnails de las páginas, la información embebible y el arreglo en formato FindJira para un medio respectivo.

El método "**get respuesta FindJira"** retorna el arreglo asociativo que contiene la respuesta traducida del formato nativo del servicio de búsqueda o feed en formato FindJira. Es importante citar que retorna todos los elementos de la respuesta cuyo número puede variar dependiendo de los parámetros que se introdujeron previamente a la búsqueda.

Los métodos de obtención y asignación simples de atributos simples que representan parámetros del framework afectan a todos los atributos de la clase exceptuando a las respuestas. Es así que se tienen variables de servicio como respuesta_raw y respuesta_json anteriormente descritos.

El método "**buscar**" requiere de la clase auxiliar HTTPRetriever para la conexión con el servidor del servicio web, se utiliza un URL que recibe como parámetro y se abre esa ubicación, luego se retorna la respuesta en formato nativo, es decir tal como el servicio la envía, sin validación de

errores. En el manejo de feeds, como sabemos, la función análoga a la búsqueda es llamada "feed_retrieve". Este resultado será almacenado en el atributo respuesta_raw, para ser parseado por las funciones del framework.

Los demás métodos que tiene la clase "service" son de soporte para la misma, éstos ya se explicaron brevemente en el capítulo anterior, más en detalle estos métodos son:

Obtener web thumbnail - Retorna el URL correspondiente a la imagen vista previa de un sitio web, para esto se añade la cadena de la ubicación de dicho servicio (por ejemplo: Girafa) que se conoce previamente. En caso de no existir una vista previa de un sitio, de manera automática se encola la dirección pedida en el mismo y se obtiene una imagen genérica.

Obtener información de URLs - Es un conjunto de funciones para el manejo de strings en URLs, en ellas es posible:

- Obtener el nombre del archivo incluyendo la extensión del mismo, basado en la ubicación de la última barra "/".
- Obtener el nombre del archivo sin la extensión del mismo, basado en la ubicación de la última barra "/" y del punto que delimita la extensión.
- Obtener sólo la extensión del archivo basado en el último punto que

delimita la extensión.

- Obtener un URL limpio quitándole en caso de que lo presente, la barra "/" al final de la cadena.
- Obtener el dominio de un archivo basado en la aparición de la cadena "http://" y de la barra "/" que aparece después de la primera.

Obtener imagen embebible - Retorna el código HTML que incrusta la imagen y dimensión que recibe como parámetros de entrada, es decir:

```
<img src="[archivo]" width="[ancho]"
height="[alto]">
```

Obtener javimoya video download - Este método está implementado de forma similar a obtener web thumbnail sólo que esta vez con la ubicación del servicio de javimoya video download. El URL obtenido proveerá vínculos de descarga para los vídeos de servicios como YouTube.

Obtener parámetros de un URL - Dado un URL y un nombre de parámetro, analiza el URL en búsqueda del nombre del parámetro y siguiendo la estructura de interfaz REST, el valor del parámetro se encuentra luego del símbolo igual "=" a continuación del nombre del parámetro.

Obtener reproductor mp3 embebible - Similar a obtener imagen embebible, se retorna el código HTML del reproductor basado en la ubicación del archivo, es decir:

```
<embed src='flash_player/mp3player.swf'
width='[ancho]' height='[alto]' bgcolor='#FFFFFF'
type='application/x-shockwave-flash'
pluginspage='http://www.macromedia.com/go/getflas
hplayer' flashvars='file=[ubicación del
archivo]&autostart=false' />
```

Éste código permitirá embeber un reproductor flash con cualquier mp3 disponible, y será de utilidad para los resultados de los servicios como Odeo y iTunes, que proveen vínculos para sus archivos de Podcasts. Es posible obtener reproductores provistos por los mismos servicios, como es el caso de Odeo, pero para mantener la estandarización del framework, se decidió utilizar un reproductor genérico que forme parte de los archivos del mismo, como se especificará más adelante en las utilidades adicionales.

Generar lista XSPF - Retorna una cadena de caracteres con la estructura XSPF, de una lista de reproducción de medios, y cuyos elementos son el resultado de recorrer el arreglo que se recibe como entrada al método.

Ésta lista puede ser de utilidad para reproductores que soporten este formato, como el Windows Media Player, y algunos reproductores web.

Obtener archivos de un directorio - Realiza el scraping de una página web, primeramente se obtiene el texto de respuesta de una ubicación web que representa un directorio libre, este HTML es recorrido en búsqueda de los hipervínculos que aparecen luego del texto "Parent directory", para esto obtenemos la cadena contenida dentro de " href=" " y " "> " ya que estas páginas al ser generadas por el servidor apache o cualquier otro, mantienen una homogeneidad en el código. Finalmente se valida que el archivo tenga extensión "mp3" y es añadido a un arreglo de formato FindJira que será retornado en este método. Utilizando la función para crear listas de reproducción, es posible crear directorios de reproducción con el framework de manera sencilla.

Obtener favicon url - Este método retorna el URL completo del archivo "favicon.ico" asociado a una página web que recibe como parámetro de entrada. El ícono de un sitio puede ser utilizado luego para representar feeds u otras fuentes de información en interfaces gráficas interesantes. En caso de no existir un ícono en la dirección especificada, se obtendrá un código indicándolo.

Obtener los feeds de un sitio - Realiza el scraping de una página web, recorre el texto de una página web en búsqueda de los tags "<link

alternate=" que contienen información de los feeds asociados a la página web que se recibe como parámetro de entrada, es decir, realiza el proceso de autodiscovery de feeds. Retornará un arreglo con los vínculos de los diferentes formatos de feeds provistos por un sitio. En caso de no encontrar dichos feeds, se obtendrá un arreglo vacío.

Las clases que heredan de service, conservan los mismos métodos antes mencionados, pero es necesario crear nuevos métodos que permitirán específicamente aprovechar los beneficios del servicio web. Estos métodos son: "elaborar url", métodos de "search" y de "parse".

El método "**elaborar url**" construye el string de petición al servicio web con todos los parámetros de búsqueda avanzada pertinentes, es importante recalcar que las palabras claves de búsqueda deberán estar codificadas para URL, es decir, que cada caracter especial deberá estar en un formato entendible para los servicios. Un ejemplo de esta codificación, es el caracter de espacio, que codificado a URL es representado por la cadena "%20".

Los métodos "**search**" realizarán las búsquedas para cada tipo de medio que el servicio soporte, por ejemplo Yahoo soporta web, imágenes y videos, entonces se deberá crear los métodos web_search, image_search y video_search respectivamente. Dentro de estos métodos se elabora el URL con el método antes descrito y con el URL que corresponde al API del servicio. Luego de elaborar el URL se captura en el atributo "respuesta_raw" la respuesta nativa que retorna el método "buscar". Finalmente esta respuesta será traducida con el método "parse" hacia la respuesta en formato FindJira.

Los métodos "**parse**" transformarán la respuesta en formato nativo como XML a JSON y luego a un arreglo PHP asociativo que será recorrido secuencialmente. Cada elemento se reasignará con nuevas claves en un arreglo temporal siguiendo el estándar FindJira. Como ya se explicó anteriormente, como algunos campos vienen con nombres diferentes es necesario renombrarlos para seguir el estándar planteado por el Framework. En el caso de otros campos, deben ser obtenidos producto de un procesamiento entre valores de otros campos de la respuesta o con los métodos auxiliares, como por ejemplo: generar el thumbnail para páginas web. A veces es necesaria una segunda llamada al servicio web para obtener toda la información necesaria para llenar los elementos a retornar.

De manera análoga a los métodos "search", existirá un método "parse" para cada tipo de medio, y para el caso de los feeds, un parse especial para el arreglo obtenido de la decodificación de los feeds utilizando una clase externa al servicio.

Para las respuestas en formato JSON, la decodificación inicial a un arreglo es manejada por las funciones internas de PHP. Para el caso de las respuestas en XML, se utiliza una clase externa descrita en el diagrama del sistema, que permite la conversión de formato XML a una cadena JSON, ésta es manejada de forma análoga a un servicio JSON.

En el caso de los feeds, la función análoga de decodificación depende también una clase externa. Dicha clase se encarga de primero reconocer el formato del feed fuente, sea este RSS o ATOM, y luego convertir la información de los posts a un arreglo asociativo de PHP con los índices llevando el nombre de los tags XML de los feeds, este arreglo es renombrado para llevar el formato FindJira.

4.3.2.1 Utilidades de Google Video / YouTube

Principalmente para dar soporte al servicio de Google Video, se han implementado los siguientes métodos:

- Obtener ID del video de Google - Este método retornará el ID del video de un URL, para esto se retorna el valor del parámetro "docId" provisto en el mismo. Este ID permite crear objetos embebibles con dicho video.
- Obtener Embed de Google Video - Retorna el código incrustable del objeto reproductor de video de Google.
- Obtener Embed de YouTube - Retorna el código incrustable del objeto reproductor de video de YouTube, dado que entre los resultados de Google video ahora se presentan mezclados con resultados de YouTube.
- Obtener container YouTube - Este método sólo está implementado en la clase del servicio de YouTube y únicamente retorna la ubicación de la página web de YouTube donde se puede ver el video dado un ID de video de la respuesta nativa de YouTube.

Como podemos ver en los anexos, para embeber un video de YouTube o de Google Video, es necesario crear un objeto embebible, utilizando el ID y las dimensiones del reproductor.

Las funciones antes descritas se encargan de crear todas estas cadenas, que luego serán adjuntadas al elemento respectivo del repositorio, esto permitirá a un desarrollador insertar videos de manera instantánea con los elementos del repositorio sin procesamiento adicional de cadenas.

4.3.2.2 Utilidades de Flickr

Los métodos que dan soporte a las búsquedas de Flickr son los siguientes:

- Obtener Flickr info - Dado un ID y SECRET este método genera el URL listo para la petición de información sobre una fotografía en Flickr.
- Obtener Flickr sizes - Dado un ID este método genera el URL listo para la petición de información de resoluciones y tamaños sobre una fotografía en Flickr.
- Obtener Flickr URL - Dado un ID, SECRET, FARM, SERVER, resolución y extensión, este método retorna la ubicación de la página web de Flickr donde se puede ver la fotografía.

Los métodos de Flickr necesitan ser llamados más de una vez para obtener toda la información necesaria, debido a las restricciones del API. Referir a los anexos para mayor detalle del formato de Flickr.

4.3.2.3 Utilidades de Microsoft Live Search

Obtener monarch key es el único método que da soporte a las búsquedas de Live Search, todas las respuestas de Live Search mantienen el mismo formato XML pero hay un campo variable que guarda todos los elementos de resultados, entonces este método retorna el índice de dicho campo que siempre se llama "FEDERATOR_MONARCH".

4.3.2.4 Utilidades de Odeo

Similar a los servicios antes explicados, en Odeo también son necesarios métodos que dan soporte a las búsquedas, estos son:

- Obtener Odeo ID - Este método retornará el ID del audio de un URL, para esto se retorna el valor del parámetro "audio" provisto en el URL.

- Obtener Odeo URL - Dado un ID, este método genera el URL listo para la petición de información sobre un podcast en Odeo.
- Obtener Container Odeo - Dado un ID, este método retorna ubicación de la página web de Odeo donde se puede escuchar el podcast.
- Obtener Embed Odeo - Este método no es más usado desde que se utiliza un reproductor externo para mostrar al usuario cada podcast, pero de todos modos está implementado. Retorna el código HTML del objeto incrustable que es el reproductor de audio de Odeo.

La búsqueda de Odeo es provista por motores externos, es decir, que Odeo no tiene un motor de búsqueda especializado. A pesar de esto, es posible obtener la información de los elementos de Odeo en formato JSON una vez obtenido el código de un elemento, utilizando una dirección especial con dicho código.

4.3.2.5 Servicios de Mapas Geográficos

Los servicios de mapas poseen otros tipos de comportamiento distintos a los de los otros servicios en el framework. Estos

servicios no extienden de las funcionalidades de la clase service. Los atributos provistos por estas clases son los puntos geográficos a insertar en los mapas y ciertos parámetros para mostrar o no ciertos componentes en los mapas embebibles. Los API Keys para el servicio de búsquedas y mapas son también atributos de estas clases.

Los servicios soportados por el Framework para inserción de mapas son, como se explicó anteriormente, los servicios de Yahoo y Google. Ambos funcionan exactamente de la misma forma y es necesario llamar a las mismas funciones tal como se explicó en el capítulo anterior. Dichas funciones imprimen código JavaScript, provisto por los APIs, y es necesaria la llamada a la función que muestra los mapas para refrescar los cambios hechos en los mismos. Es necesario el uso de un API key para poder incluir un mapa en un sitio. Estos servicios necesitan ser modificados para ser usados por los desarrolladores.

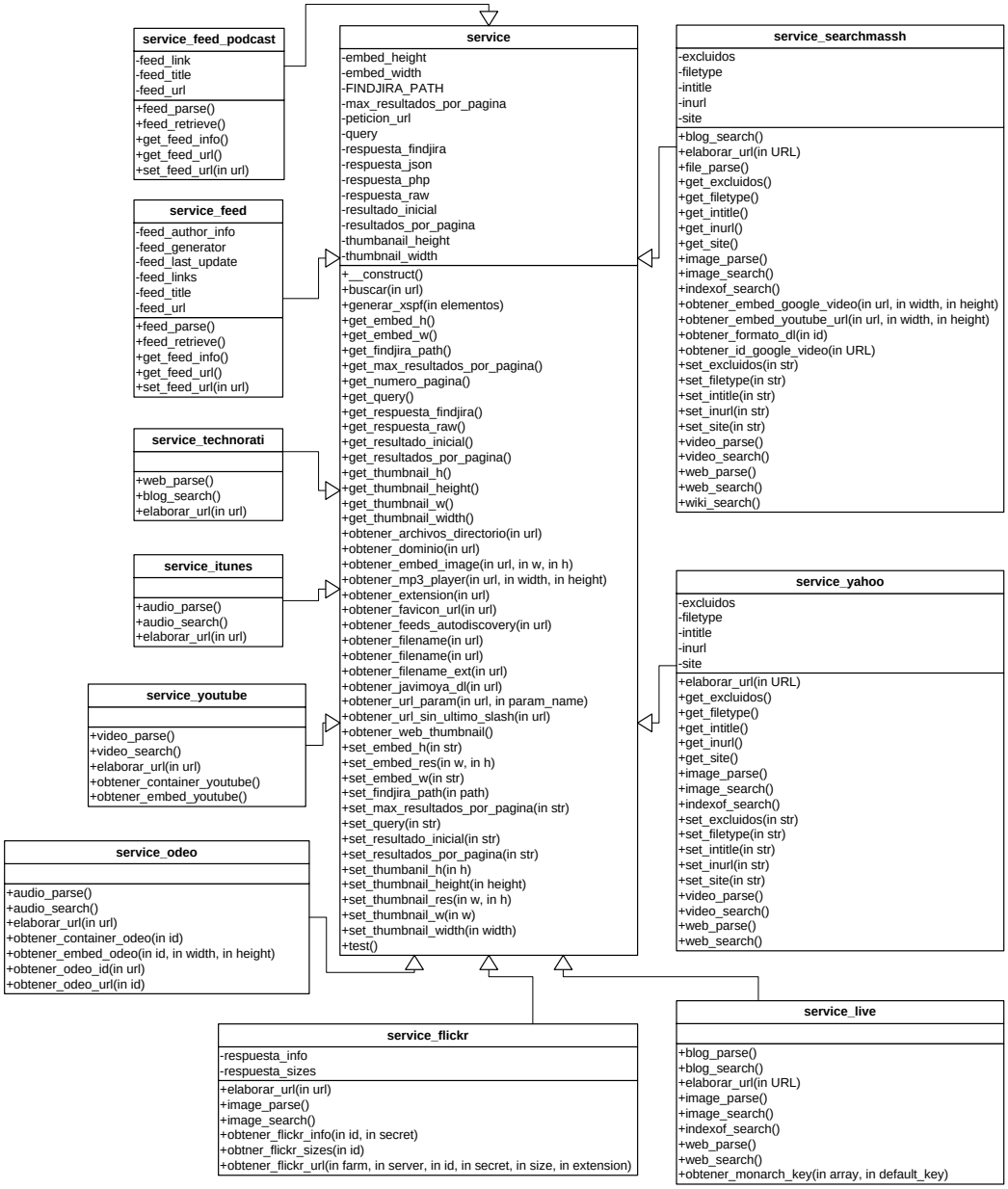


Figura 4.2: Clase Servicio.

4.3.3 Clase Repositorio de Datos

La clase de donde se almacenarán la información de todas las fuentes del framework. El método "**insertar**" permite añadir un elemento al repositorio. Como se describió en el Diseño, todos los elementos son insertados de manera que se respete el orden de sus motores respectivos. Cada elemento es verificado por su URL de fuente. En caso de ya existir un elemento de una fuente dada, pero de otro conjunto de datos, es considerado como un resultado repetido, y en lugar de ser insertado al repositorio, su campo de origen será alterado de manera que se describan las dos o más fuentes de información de dicho elemento.

Es posible la inserción de un arreglo de elementos de repositorio, al llamar a la función "**insertar_batch**". Estos métodos de inserción siguen el orden de los resultados de sus fuentes de información, es decir, que respetan la relevancia de los resultados provistos por los buscadores, o en el caso de los feeds, respetan la fecha de creación de los distintos posts.

De manera análoga, es posible consultar el repositorio mediante funciones de extracción. Estas son llamadas "**extraer_elementos**", que retornan todos los elementos del repositorio, "**extraer_batch**" y "**extraer_batch_externa**", retornan los elementos contenidos entre dos índices específicos, y en el caso de la segunda, dado un arreglo de

entrada que servirá para que dicha función sea llamada externamente sin necesidad de crear un objeto de tipo repositorio.

Finalmente, la función "**limpiar**" vaciará el repositorio.

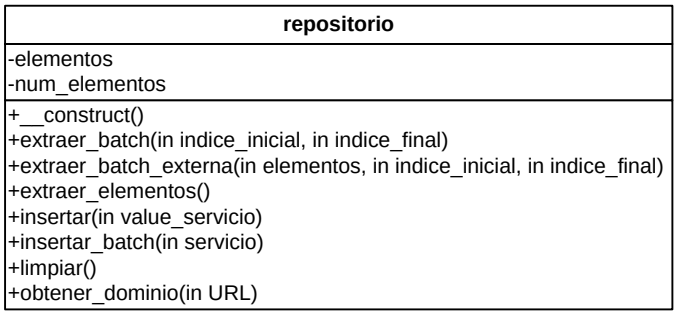


Figura 4.3: Clase Repositorio.

4.3.4 Funciones auxiliares y de soporte

Las funciones auxiliares y de soporte incluyen todas las funcionalidades externas a la lógica framework, cuya utilidad fue necesaria en la fase de implementación. Estas funciones pueden ser código fuente libre que ha sido descargado de la red así como herramientas ejecutables tales como el reproductor mp3. El reproductor de audio desarrollado en flash de uso libre soporta formatos comunes como mp3 y permite la reproducción de archivos en línea. Este reproductor es utilizado para los medios de audio embebibles que provean un vínculo a archivos de audio, como los podcasts.

Además se hace uso de clases externas principalmente para la conversión de información. Dichas clases no están orientadas para ser manejadas por un desarrollador, sino para proveer una capa de soporte inferior al framework. A seguir se describen las clases orientadas al manejo de formatos estandarizados:

"class_HTTPRetriever"; esta clase permite obtener la información de página web, dado un URL, permite hacer GETs y POSTs utilizando el protocolo HTTP, y permite enviar parámetros adicionales en el URL. Esta clase es utilizada por el framework para recopilar información en bruto de la red.

"class_JSON"; esta clase permite el manejo del formato JSON. Esta clase provee conversión entre formatos JSON y PHP Serializable, y permite codificar strings en formatos web, compatibles con JSON. Todas estas funcionalidades fueron añadidas en la versión 5 de PHP, y por ende, no son utilizadas por la versión inicial framework. A pesar de esto, se decidió incluir estas clases en el mismo por cuestiones de compatibilidad con otras versiones anteriores de PHP.

"class_validateSyntax"; permite la validación de diferentes formatos de direcciones web, soporta casi todos los formatos conocidos, incluyendo direcciones con números de puertos y caracteres especiales. Esta función es utilizada por el framework para verificar distintos formatos de URLs provistos.

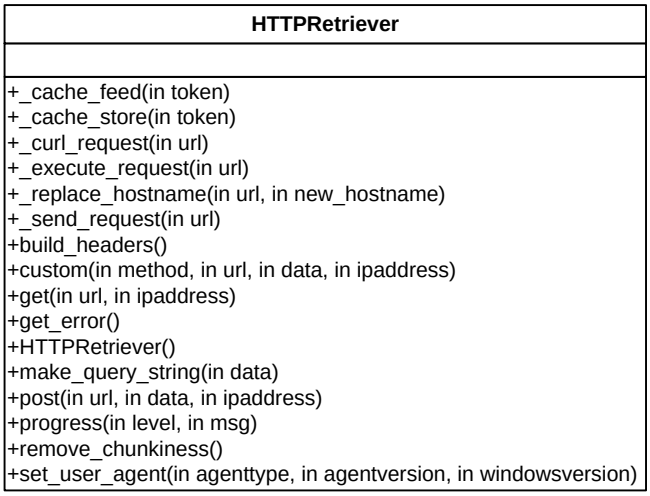
"class_xml2json"; Esta clase permite la conversión de XML a JSON. Esta función es utilizada para permitir al framework trabajar con servicios XML, y convertir esta información a una cadena JSON. Esta cadena puede ser usada para interpretación o conversión a arreglos por las funciones de PHP, como puede ser usada para exportar información en JSON.

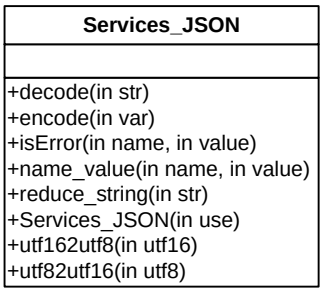
"feedsProcessingAPI"; Es un conjunto de utilidades de código abierto, que permite el sencillo manejo de feeds. Permiten la lectura de un feed a base del URL del mismo, para luego manipular su información y mostrarla en una página con un formato deseado. El framework utiliza una función modificada de esta clase, que permite obtener un arreglo de PHP en formato asociativo, para ser parseado en las funciones de servicio, en lugar de ser impresa en un sitio. Este arreglo asociativo posee el formato nativo de los feeds, y será parseado por el framework.

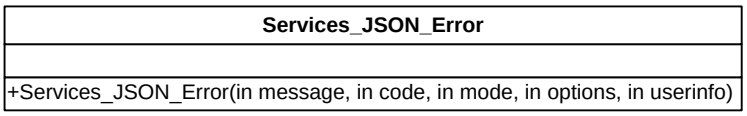
Actualmente están soportados en el Framework los feeds de tipo ATOM y RSS versiones 1 y 2.

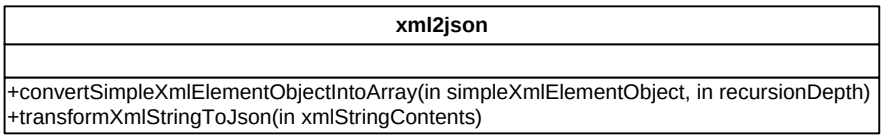
"util_recortar"; Permite obtener una vista previa de una imagen o un crop (o recorte de una región de interés) de la misma. Esto es de utilidad para mostrar vistas previas de imágenes encontradas, pero sin distorsión, es decir, con dimensiones proporcionales a la imagen original. Esta utilidad es compatible con todos los tipos de archivos permitidos para uso en internet.

"util_xspf"; Permite crear una lista de reproducción en formato XSPF de manera volátil, es decir, genera un archivo temporal con la lista de reproducción en dicho formato, dado un URL en específico. El archivo es retornado como una cadena en memoria. Esta utilidad adicional utiliza las funciones de service, y hace posible el uso de listas de reproducción instantáneas, a partir de directorios en línea.


HTTPRetriever
+_cache_feed(in token)
+_cache_store(in token)
+_curl_request(in url)
+_execute_request(in url)
+_replace_hostname(in url, in new_hostname)
+_send_request(in url)
+build_headers()
+custom(in method, in url, in data, in ipaddress)
+get(in url, in ipaddress)
+get_error()
+HTTPRetriever()
+make_query_string(in data)
+post(in url, in data, in ipaddress)
+progress(in level, in msg)
+remove_chunkiness()
+set_user_agent(in agenttype, in agentversion, in windowsversion)


Services_JSON
+decode(in str)
+encode(in var)
+isError(in name, in value)
+name_value(in name, in value)
+reduce_string(in str)
+Services_JSON(in use)
+utf162utf8(in utf16)
+utf82utf16(in utf8)


Services_JSON_Error
+Services_JSON_Error(in message, in code, in mode, in options, in userinfo)


xml2json
+convertSimpleXmlElementObjectIntoArray(in simpleXmlElementObject, in recursionDepth)
+transformXmlStringToJson(in xmlStringContents)

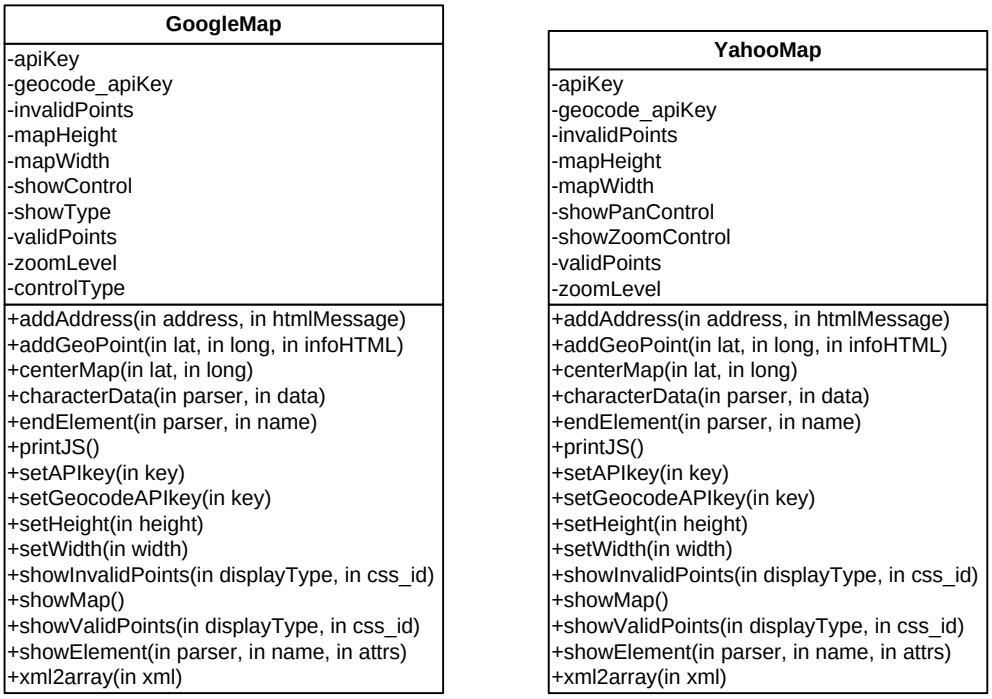


Figura 4.4: Clases auxiliares de Findjira Framework.

Conclusiones

Se ha visto en este capítulo el desarrollo del marco de trabajo y las posibilidades que ofrece el mismo para realizar las búsquedas en los diferentes servicios. Fue posible el desarrollo basado en la etapa de diseño sin sufrir cambios en la lógica del sistema o en el funcionamiento secuencial del mismo. Se sabe como obtiene la información y que lenguajes utiliza, pero aún no ha sido visto en acción siendo utilizado por una aplicación que explote todo lo que ofrece. En el Capítulo a seguir, se mostrará el desarrollo de un ejemplo utilizando el Framework, para así dejar completa la etapa de desarrollo. Este ejemplo será un concepto nuevo, basado en algunos buscadores

integrados, y será dinámico, al ser desarrollado mayormente usando toolkits de Ajax. Este ejemplo será llamado Iguana Search, y a pesar de que su nombre implica un buscador ordinario, éste ofrecerá muchos servicios innovadores no presentes en la mayoría de los buscadores actuales.

CAPITULO 5.

Implementación de una solución utilizando el marco de trabajo: Iguana Search

Introducción

En este trabajo se han descrito muchas formas de servicios web, y los muchos medios que estos servicios exploran e indexan. Se explicará el proceso de realización de un Mashup con el Framework, un buscador integrado. En este buscador se darán a conocer las funciones más interesantes del Framework, y se mostrará cómo el mismo interactúa con la aplicación desarrollada.

5.1 Análisis previo

Como la idea principal de este trabajo es de unificar diferentes servicios y medios, y como el FindJira Framework está orientado a la realización de búsquedas en los mismos para el desarrollo de mashups, el ejemplo a describir en éste capítulo deberá entrar en esta categoría. El buscador integrado deberá mostrar al usuario resultados de manera sencilla de entender, es decir, basados en los principios de la interacción hombre máquina. Al mismo tiempo, deberá proveer acceso rápido a nuevas búsquedas y facilidades para la visualización de

los detalles de ciertos resultados de interés, sin afectar la navegación del gran conjunto de resultados obtenidos.

Sin considerar los factores de navegabilidad y usabilidad, es muy importante también optimizar el uso del Framework, pues al trabajar con grandes cantidades de resultados, es necesaria una buena optimización del sistema para que la navegación y uso sean lo más rápidas posibles. Está claro que esto depende principalmente del ancho de banda del usuario, y de la capacidad de procesamiento de su equipo, pero es importante reducir en lo mayor posible estos tiempos para abarcar una mayor gama de usuarios. Se decidió nombrar al sitio de ejemplo Iguana Search, en alusión a las iguanas características de la ciudad de Guayaquil, ciudad natal de los desarrolladores y autores de este trabajo.

Para lograr todo esto, es necesario el diseño de una interfaz muy amigable y un sistema de comunicaciones entre capas eficiente. A continuación se describe todo el diseño del sitio y los detalles de implementación necesarios para cumplir estos objetivos.

5.2 Diseño de Iguana Search

El diseño del sitio se debe enfocar en la navegabilidad, debe estar centrado en los usuarios y en los estándares de interacción. Se debe considerar primero en los detalles a evitar. No es recomendable crear un sitio intrusivo, es decir, que moleste al usuario con ventanas emergentes o popups. También se quiere evitar la inclusión de ventanas que tengan que ser movidas constantemente para visualizar los detalles del sitio. En lo posible, cada detalle de interés debe ser visible al usuario de manera organizada. Y cada resultado deberá ocupar un espacio justo para poder así aprovechar el espacio en lo mayor posible, sin sobresaturar el sitio.

Cada opción única del sitio deberá ser fácil de entender, por ejemplo, el usuario deberá saber dónde está el cuadro de búsqueda, o donde puede visualizar los resultados de cada medio. Se deberá también, considerar las interfaces a las que el usuario está acostumbrado, y usarlas como metáforas o como base para una navegación más natural. El sitio deberá ser accesible en todas las resoluciones de pantalla de los usuarios comunes, es decir, desde 800x600 pixeles en adelante, y esto no deberá afectar gravemente la navegación del sitio.

5.3 Diseño preliminar de la interfaz del sitio

El diseño preliminar del sitio consiste en tres secciones principales inamovibles, la sección de búsqueda, la sección de resultados y la sección de detalles y contenido adicional. Cada una de estas secciones se remite a sus funciones y trabajarán en conjunto para mostrarle al usuario la información de la forma menos intrusiva y más entendible posible. En la figura 5.1 se describen éstas tres secciones de manera entendible.

Figura 5.1: Diseño de la disposición de las secciones del sitio.

La sección de búsqueda consiste en un simple campo de texto y un botón de búsqueda. Es en esta parte de la página donde el usuario puede ingresar su query de búsqueda, y éste query o cadena de palabras clave, empezará a mover las piezas de la página y el framework para armar un conjunto de resultados

relevantes. Esta sección deberá estar disponible en todo momento, sin molestar al usuario.

La sección de resultados de búsqueda deberá mostrar de manera poco intrusiva los resultados de los medios soportados. Debido a que los resultados pueden ser de diferentes medios, se han escogido dos formas de mostrar los mismos. A continuación se describe la forma de mostrar los resultados para medios textuales y no textuales, las dos clasificaciones escogidas de conjuntos de resultados.

Para los resultados de medios textuales, como sitios web y resultados de blogs y feeds, se mostrarán los resultados como se acostumbra en la mayoría de los navegadores. Esto consiste en ciertos detalles relevantes como el Título de la página, un breve resumen, la dirección del sitio. Otros detalles pueden ser incluidos en este grupo, como el motor de búsqueda en el cual fue encontrado, esto puede ser de interés en el caso del ejemplo. En esta imagen podemos apreciar un resultado de la lista de resultados de Yahoo! para la palabra clave "Guayaquil". Podemos apreciar los datos relevantes antes descritos en el resultado, se escogió este formato de presentación debido a que es el formato al que los usuarios internautas se han acostumbrado a ver en casi todos los

buscadores populares, como Google, Live y Yahoo. Se añadirá a los resultados una vista previa de las páginas web, cuando disponibles, para que el usuario pueda, en ciertos casos, discernir aún más los resultados de interés de manera visual.

Figura 5.2: Resultado de Yahoo.

Para los resultados de medios no textuales, como lo son los videos y las imágenes, se pensó mostrar vistas previas de los mismos, de ésta forma el usuario puede discernir inmediatamente que resultados son más relevantes para su búsqueda, siguiendo una metáfora de un simple álbum de fotos. Estos resultados se mostrarán a manera de galería y llevarán un texto con su título o nombre de archivo, para facilitar aún más su comprensión, o en caso que la imagen asociada aún no cargue en la página. En la imagen podemos apreciar una sección del conjunto de resultados de búsqueda de DeviantArt, para el mismo query que el ejemplo anterior.

Figura 5.3: Fragmento de una galería de resultados de imágenes de DeviantArt.

Finalmente, en el panel de detalles, se mostrarán detalles adicionales para los medios no textuales. En este panel se mostrarán las vistas previas de los mismos y será posible ocultar este panel con el click de un botón, para no afectar la navegación del sitio, si el usuario lo requiere. Este panel es algo innovador del sitio, y permitirá descargar al ordenador del usuario los resultados de los medios que lo permitan. En la mayoría de los servicios es necesario visitar la página que contiene a dicho resultado para poder ver estos detalles. El mostrar los resultados de manera inmediata ahorra tiempo de navegación para el usuario.

Los tres paneles antes descritos y la mayoría de sus contenidos estarán implementados en HTML y CSS, de manera que puedan ser manipulados de

manera sencilla y sean compatibles con las tecnologías Ajax escogidas. El panel de búsqueda estará siempre visible y permitirá al usuario realizar una búsqueda nueva en cualquier momento que desee. Así mismo, las bondades de CSS nos permiten definir una sola clase para todos los resultados y así asegurar la uniformidad de los resultados y la buena interacción del sitio.

5.4 Diseño preliminar de los funcionamientos en Ajax

Al ser una aplicación web dinámica, la página de Iguana search deberá tener funcionalidades implementadas en Ajax. Las funcionalidades principales del sitio son, como es lógico, la búsqueda y la presentación dinámica de los resultados en la misma página, y la comunicación con archivos PHP externos pertenecientes al Framework.

La librería escogida para la implementación de funcionalidades Ajax para este trabajo fue JQuery. La razón por la cual esta librería se escogió fue por su estructura liviana, la cual permite que el sitio cargue mucho más rápidamente, ofreciendo las mismas funcionalidades que otras librerías de desarrollo Ajax como Dojo, y además de una gran comunidad de soporte, que permitirá esclarecer cualquier duda de desarrollo. Además existen plugins para esta librería que facilitan el avance de la implementación de la parte interactiva de

un sitio enormemente, como facilidades para ocultar y mover secciones de la página con solo acceder a sus atributos CSS, o cuadros emergentes ya implementados no intrusivos para mostrar información adicional que no se desee mostrar en la página.

Es posible además refinar ciertos detalles de la búsqueda web, y de guardar algunas preferencias de usuario en la página. Los detalles de búsqueda avanzada incluyen el buscar en sitios específicos, en títulos de las páginas y el encontrar resultados en directorios web públicos que estén en línea, todas estas funcionalidades son accesibles vía Ajax y son opciones de búsqueda que son soportadas por el FindJira Framework. Las preferencias incluyen el escoger cuantos resultados mostrar en cada página, entre otras opciones que serán descritas en la sección siguiente.

5.5 Detalles de implementación

Iguana Search, como ya se ha mencionado, utilizará FindJira Framework el mismo que servirá para obtener las respuestas de los servicios de búsqueda web en un único formato que será leído constantemente para mostrarle los resultados al usuario final quien será el navegante que realice las búsquedas en la aplicación.

Entre los niveles del framework y la interfaz del usuario fue necesario implementar scripts de PHP, llamados php_handlers. Estos scripts crearán los objetos de las clases handler de FindJira con todos los parámetros que el usuario necesita para realizar la búsqueda. En este punto se utilizan las funciones del handler correspondientes para cada tipo de medio, ya que es necesario crear php_handlers para cada tipo de medio, por ejemplo el php_handler para el medio de imágenes luego de preparar la búsqueda con los términos y parámetros necesarios se llamará al método de realizar búsqueda de imágenes y luego se extraerá su repositorio a una variable de sesión de PHP que se mantendrá persistente mientras la aplicación se esté ejecutando en el navegador.

Los php_handlers también serán capaces de retornar un rango de elementos del repositorio del medio en cuestión, siempre y cuando se le envíen los parámetros adecuados, que servirá para mostrar resultados paginados en la interfaz gráfica, los mismos que ya deberán contar con un formato HTML que sea fácilmente integrable a la interfaz de usuario. La última funcionalidad que proveen los php_handlers y sólo para los medios de imágenes, audio y video es retornar, también en HTML, los detalles de un elemento en particular asociados con los campos del repositorio, como por ejemplo el reproductor incrustable, la página

de origen, la lista de servicios de común su ocurrencia y otros campos similares que se describieron junto al repositorio de datos.

La interfaz de usuario o GUI consiste en una página HTML estática cuya única programación se ejecutará del lado del cliente usando el lenguaje JavaScript. Esta GUI a parte de contener los efectos visuales, se enfocará en hacer peticiones HTTP utilizando la técnica AJAX a los php_handlers utilizando la siguiente dinámica: Cuando el usuario busca algo en el cuadro de texto, se llamará a una función JavaScript llamada buscar() que realiza una petición AJAX a los php_handler de cada tipo de medio enviándole el query ingresado, los php_handlers retornarán el número de resultados almacenados en el repositorio del medio, de esta forma si la respuesta del handler es exitosa se puede llamar a otra función JavaScript llamada imprimir() que se encargará de escribir la primera página de resultados, en caso contrario se le notificará en el GUI que no existen resultados relacionados para la búsqueda en cuestión. Esta función de impresión nuevamente realiza peticiones AJAX a los php_handlers de acuerdo al tipo de medio que está tratando pero ahora los parámetros son los índices inicial y final del rango de elementos del repositorio, es ahora cuando se muestra en pantalla el retorno formateado desde el php_handler y una barra de navegación entre las páginas de resultados del

repositorio, cabe recalcar que el número de resultados por página deberán ser especificados en el GUI mismo.

Los elementos del repositorio en el caso de imágenes, audio y video, deberán estar formateados correctamente tomando en cuenta los eventos para cuando el usuario le dé un click se muestren los detalles en un panel especifico para este propósito. De esta forma se llamará a otra función JavaScript llamada ver detalles() que recibirá el número de elemento del repositorio correspondiente para poder realizar una petición AJAX al php_handler respectivo solicitando los detalles, éste se los devolverá, como es costumbre, formateados en HTML listos para ser incrustados en el GUI.

Para armar la paginación se implementó una función de JavaScript llamada paginar() que recibe un índice inicial y final y el número de elementos del tipo de medio actual, esta función genera el HTML para la navegación entre páginas de resultados contenidos en el repositorio, resalta la página actual del conjunto de páginas resultantes e inclusive incrusta flechas para una fácil navegación horizontal. Basada en cálculos matemáticos se deduce los índices de la página anterior, siguiente y de los elementos del conjunto, así se van generando los

textos con los hipervínculos que apuntan a la función que imprime un rango de elementos antes mencionada.

Para búsquedas web fue necesario implementar una sección de búsqueda avanzada, donde una vez realizada la búsqueda web normal se le pregunta al visitante si desea refinar su búsqueda, en este punto se le pregunta parámetros adicionales como restringir la búsqueda a un solo sitio web, los tipos de archivos que se desean obtener en los resultados, las palabras que deberán ser excluidas de la búsqueda y el rango de acción de la búsqueda al título de la página, URL o en ambos más el contenido mismo de la página que es lo predeterminado. Adicionalmente se le ofrece al visitante la opción de búsqueda en directorios, que no es otra cosa que buscar sólo en directorios públicos de los cuales se hablaron en el Capítulo 1. Estas son las únicas opciones que se pueden proporcionar debido a que son comunes entre todos los servicios web, pese a que cada uno tenga sus propias opciones de búsqueda avanzada, estas serían muy exclusivas así que no se las podría usar en todos los servicios de la misma manera. Toda esta búsqueda avanzada trabaja de manera casi independiente a la implementación mencionada para la función buscar(). En este caso se usa una función llamada búsqueda avanzada() que recibe nuevamente los términos y parámetros para de esta forma reiniciar el

repositorio web únicamente con los nuevos resultados, perdiéndose en el proceso los resultados que aparecieron inicialmente.

Es importante destacar un detalle que se presenta utilizando las tecnologías de PHP y AJAX. Debido a que el visitante busca en todos los medios simultáneamente, es necesario ejecutar una petición AJAX para cada tipo de medio y como ya se mencionó, en todas estas peticiones se utilizan los php_handlers que acceden leyendo o escribiendo datos en la variable de sesión de PHP, la cual sólo puede ser accedida por una petición a la vez, en otras palabras, la variable de sesión funciona de forma mutuamente excluyente. Como estas peticiones toman su tiempo porque realizan las conexiones con los servicios web y procesan los datos, la latencia total de la búsqueda aumentaba debido a esta espera hasta que se presentaban los resultados al usuario. Para poder solucionar este grave problema se inicializó la sesión de PHP sólo en el momento de utilizar la variable. Además, justo después de su utilización, se ejecuta una función de PHP llamada session_write_close() que forza la finalización de la sesión antes que termine la ejecución del script, convirtiendo el manejo de la variable de sesión en un manejo similar al de una sección de código crítico. Además se envían unas cabeceras HTTP para evitar que los datos de la variable de sesión se pierdan. En resumen, se utiliza un mecanismo similar al de los semáforos. Esto permite que la exclusión mutua esté enfocada

sólo a la parte del código que lo necesita, tratando como sección crítica el acceso a la variable de sesión de PHP, esto hace más eficiente el código, al minimizar las condiciones de carrera en el código para el acceso a la sesión.

Para finalizar se implementó una pequeña sección de preferencias en la cual se le permite al usuario configurar el número de resultados por página que se mostrarán para dos tipos de medios como son los textuales (web y feed) y audiovisuales (imagen, audio y video). Para esta sección se utilizaron cookies del lado del cliente para mantener esta configuración, siempre escritas y leídas en JavaScript.

Los detalles de Interacción y visualización en Ajax de Iguana search son manejados, como se explicó anteriormente, con la librería jQuery. Esta librería permitió añadir pequeñas funcionalidades menores al sitio, mayormente para facilitar la interacción y mejorar la visualización del sitio. Estas funcionalidades incluyen el poder mostrar ventanas emergentes o popups para las páginas externas de Acerca De y los Términos de Uso. Esto es gracias al plugin llamado ThickBox, que permite mostrar diálogos sobre la página, opacando el contenido de la misma. Estos diálogos sólo se mostrarán si el usuario quiere leer contenidos no relevantes a su búsqueda, completamente opcionales. El poder

mover los paneles de búsqueda, las animaciones de desvanecimiento y el poder ocultar los detalles con un simple evento de click son también manejados por las funciones de jQuery, con una lógica sencilla, accediendo a los paneles por sus nombres otorgados en CSS y asignándoles la propiedad que se desea editar, de manera sencilla.

Adicional a la librería para la interfaz gráfica, se necesitó incluir algunos scripts de JavaScript que intentan arreglar ciertas limitaciones de los browsers, uno de estos es la capacidad de visualizar correctamente los archivos de imágenes de tipo PNG principalmente porque el Internet Explorer 6 no soporta la transparencia de estos archivos[51]. El otro script corrige en el mismo navegador un parpadeo molestoso que suele presentarse[52]. Aún así nunca se visualiza de manera idéntica en dos navegadores diferentes, es por esto que la interfaz de Iguana Search está optimizada para ser usado en el Mozilla Firefox, debido a su alta popularidad y a la compatibilidad que ofrece a los desarrolladores web.

Iguana Search como producto final no utiliza todo lo que FindJira framework ofrece, como utilizar el servicio de mapas satelitales, soporte de sindicación en

[51] http://homepage.ntlworld.com/bobosola

[52] http://ajaxian.com/archives/no-more-ie6-background-flicker

feeds, scraping de archivos en páginas web de directorios, generación de listas de reproducción en formato XSPF y obtención de los íconos representativos de cada sitio web. Este último se lo omitió, como los anteriores para evitar una sobrecarga de procesos en la interfaz resultando mucha espera por parte del usuario. A continuación se presentan algunas capturas de la aplicación de ejemplo que ilustran el diseño antes explicado.

Figura 5.4: Página de bienvenida de Iguana search.

Se puede apreciar la pantalla de bienvenida o splash del sitio. De manera intuitiva se presentan los controles básicos de búsqueda, los agradecimientos y las opciones adicionales. Esta es la página de inicio de Iguana Search.

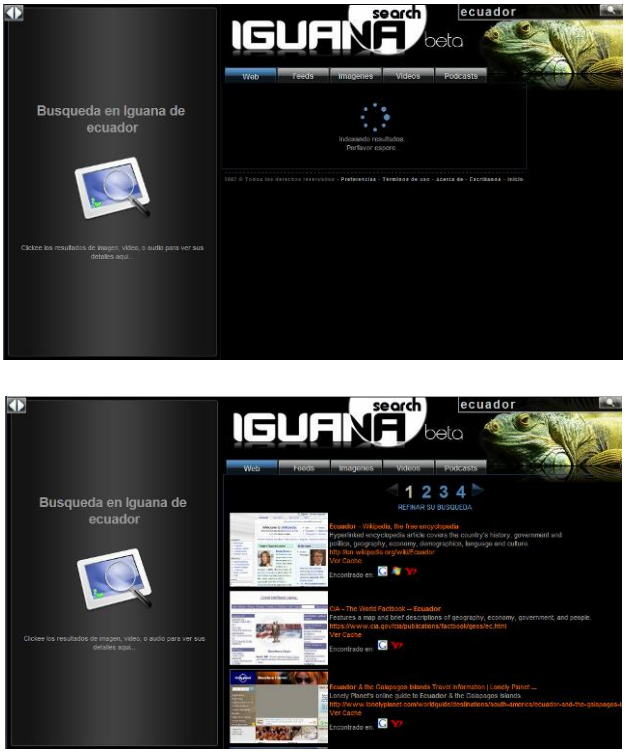


Figura 5.5: Indexación y resultados iniciales de un ejemplo de búsqueda.

Figura 5.6: Resultados de Feeds de un ejemplo de búsqueda. El panel de detalles se ha contraído.

El cuadro de texto para realizar búsquedas estará siempre accesible en la esquina superior derecha del sitio, para un fácil acceso al servicio. Se especifica la fuente de cada resultado, el sitio de origen del mismo y el URL de origen. Para los resultados textuales se muestra un pequeño resumen del sitio y para los medios no textuales, una prevista del mismo en el panel de detalles, éste panel puede ser oculto y mostrado a placer para obtener un mayor espacio para examinar el conjunto de resultados obtenidos de cada medio.

Figura 5.7: Resultados de Videos de un ejemplo de búsqueda. Se reproduce un resultado en el panel de detalles.

Al enviar una petición de búsqueda en los servidores, se cargan todos los paneles de medios y se muestran instrucciones sencillas en el panel de detalles. Al terminar de ejecutarse la petición, se llenan los paneles de resultados para cada medio. En los paneles de Webs y Feeds se pueden apreciar los resultados en lista, y en los paneles de Podcasts, Videos y Audio, los resultados en modo Galería, para los cuales es posible visualizar los resultados en el panel de detalles, mientras se navega en el sitio, de manera no intrusiva. El resultado activo es resaltado ligeramente para mayor navegabilidad. Además se provee acceso en todo momento al diálogo de preferencias, para modificar los parámetros de visualización de resultados, el diálogo de información del sitio, el diálogo de sugerencias y el diálogo de términos de uso.

5.6 Pruebas

La implementación de Iguana Search como una aplicación de ejemplo se la realizó desde el punto de vista de un usuario del framework para de esta forma medir las facilidades de uso del mismo. La implementación de los php_handlers fue necesaria para poder crear los objetos de tipo handler para cada tipo de medio y además y dar formato HTML a los elementos del repositorio para que sean fácilmente incrustados en la página que los lee, porque de esta forma fue diseñado el comportamiento de Iguana Search.

También queda de parte del desarrollador el diseño web y gráfico de la página a la que accede el usuario final, esto incluye las peticiones AJAX a los php_handlers, el manejo de resultados por páginas y cualquier otra innovación a este nivel.

El tiempo total de desarrollo de Iguana Search fue de dos semanas laborables, en donde toda la lógica del sitio es llevada por los php_handlers y el Ajax presente en el mismo funciona meramente para llamar a las diferentes funcionalidades de los mismos, para conseguir información del Framework y presentarla de manera dinámica. El tiempo de desarrollo del Framework tomó cuatro semanas. Excluyendo la lógica del Framework, el desarrollo de las funciones del mismo en un ambiente orientado sólo a la interacción con los

servicios tomó tres de estas cuatro semanas. En otras palabras, el Framework permite un significativo ahorro de tiempo en el desarrollo, añadiendo la posibilidad de crear nuevos servicios basados en la lógica del mismo.

Si el desarrollador no utilizara el framework, se debería sumar a todo lo anterior el tiempo de investigación y desarrollo de los mecanismos de conexión e interpretación de datos para cada servicio web que desease utilizar, pero no todo desarrollador se especializa en el manejo avanzado de PHP, los php_handlers son ejemplos sencillos y reutilizables para cualquier aplicación que use FindJira framework, nada complejos en comparación a la lógica interna del framework que es lo que se intenta facilitar.

Para hablar de cantidad de esfuerzo necesario para la implementación, el archivo de interfaz gráfica contiene alrededor de 1000 líneas de código entre código estático y JavaScript. Los archivos de php_handlers en total suman aproximadamente 800 líneas de código. En cuanto al framework, el número de líneas del núcleo del mismo es cercano a 1250, sin incluir los cada uno de los servicios que funcionan de manera diferente, los cuales suman casi 1800 líneas que es lo más significativo que puede ahorrarse un desarrollador, al implementar todos los servicios que se incluyen en el framework. Además de esto, el código del Framework está debidamente documentado para el sencillo

entendimiento de un desarrollador nuevo a la lógica del mismo, esto ayuda a evitar el consumo de tiempo en aprendizaje, en caso de que desee modificar el mismo, algo completamente optativo.

Se realizó una encuesta en el sitio acerca de algunos aspectos del mismo, y del desempeño del sitio frente a diferentes consultas. Con esta encuesta se logró evaluar el rendimiento del Framework frente a la demanda de los usuarios en la etapa preliminar de pruebas. Cabe recalcar que el desempeño del mismo es directamente proporcional al ancho de banda del servidor donde está alojado el sitio, pues depende de esta conexión para realizar las búsquedas.

Pruebas de Usabilidad

Se evaluaron un total de 100 encuestas a manera de muestra para las pruebas de usabilidad, se invitó a los usuarios que probaron el sistema a contestar informalmente a las siguientes preguntas:

- ¿Qué te pareció el diseño del sitio?
- ¿Qué te pareció la forma de usar del sitio?
- ¿Qué te pareció la velocidad del sitio?
- ¿Qué te gustaría que tuviera el sitio?
- ¿Encontraste los resultados que buscabas?
- ¿Qué tan rápida es tu conexión a internet?
- ¿Cuál es tu proveedor de Internet? (Ej: Satnet, Linkotel, etc)

Adicionalmente, se pide al usuario describir algún error, en caso de ocurrir el mismo.

Los resultados estadísticos de dichas encuestas para cada una de las anteriores preguntas se describen a continuación:

¿Qué te pareció el diseño del sitio?

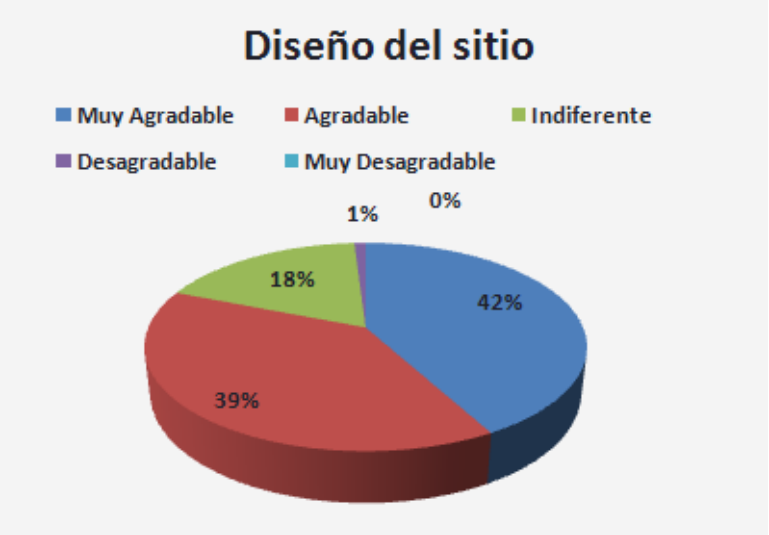


Figura 5.8: Resultados de Pruebas: Diseño del sitio

En general el diseño del sitio es aceptado por los usuarios que realizaron las pruebas. Las quejas recibidas en cuanto al diseño se enfocan en el uso del panel de detalles, pues puede ocupar mucho espacio cuando no está en uso. Pero gracias a la opción de poderlo ocultar estas quejas desaparecen. En cuanto a la combinación de colores oscuros, los usuarios se sienten cómodos al respecto, pues les permite visualizar los resultados de manera más llamativa. Los videos pueden ser visualizados de mejor manera con este esquema.

¿Qué te pareció la forma de usar del sitio?

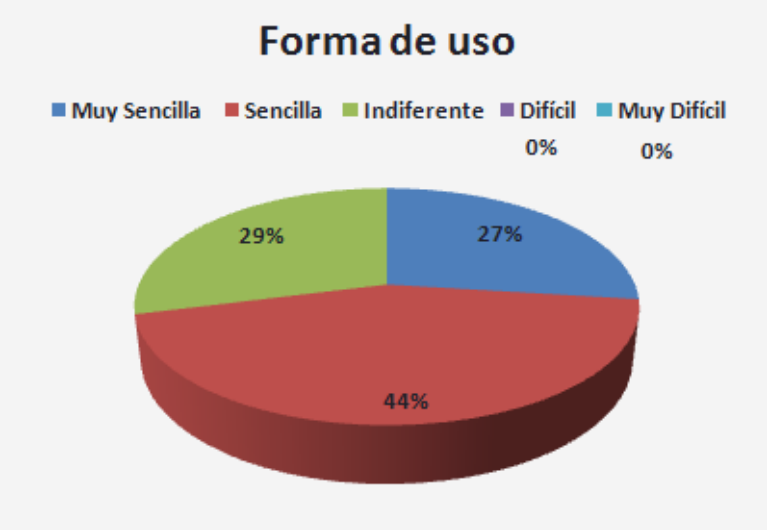


Figura 5.9: Resultados de Pruebas: Forma de uso del sitio

En general la opinión acerca de la forma de uso del sitio es positiva. Las críticas se enfocan en detalles como el uso del sitio, el uso del panel de detalles. O la ubicación del input para la búsqueda. Estas observaciones son sólo al inicio, pues una vez que el usuario se acostumbra a la interfaz, se siente a gusto en la misma.

¿Qué te pareció la velocidad del sitio?

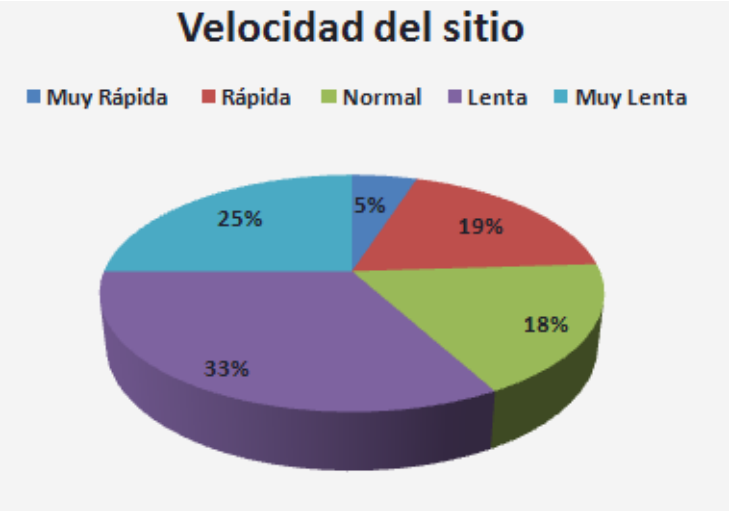


Figura 5.10: Resultados de Pruebas: Velocidad del sitio

La velocidad del sitio depende exclusivamente de dos factores: La velocidad de conexión del usuario, y la velocidad de conexión del servidor para realizar los requests a los diferentes servicios. Existen horas pico para el mismo, y para estas horas la velocidad de conexión del servidor puede ser bastante deficiente. Pero en horas de poca demanda, como en las noches o las mañanas, el sitio contesta de manera muy apropiada.

¿Qué te gustaría que tuviera el sitio?

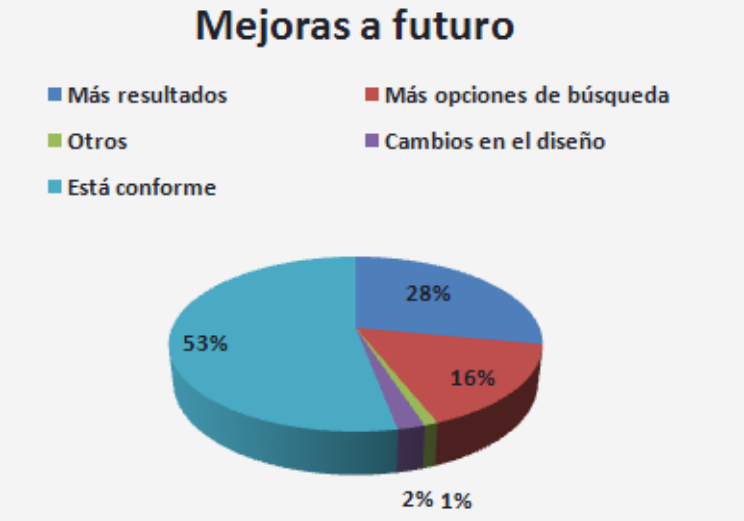


Figura 5.11: Resultados de Pruebas: Mejoras a futuro

La gran mayoría de los usuarios se encuentra conforme con el sitio en su estado actual. Pero las pocas críticas recibidas indican que el usuario desea más control sobre las opciones del sitio, como el número de resultados a mostrar, el número de páginas a obtener de los servicios y los medios que el sitio ofrece. Ligeros cambios en el diseño como el espaciado entre resultados son cambios menores a futuro que pueden ser agregados al sitio.

¿Encontraste los resultados que buscabas?

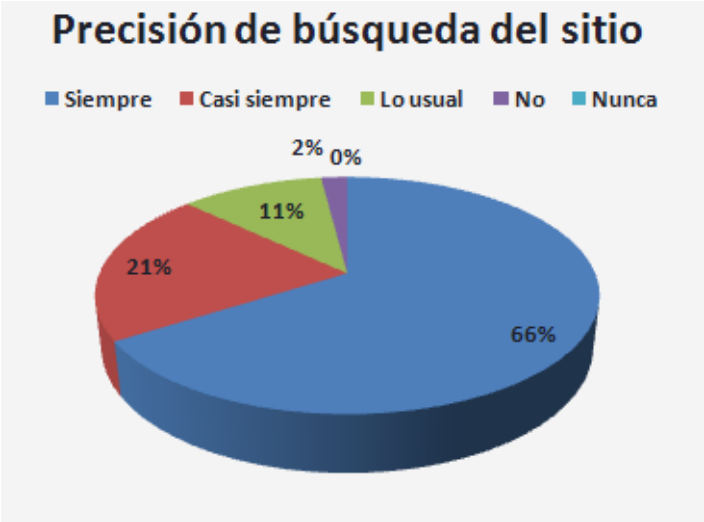


Figura 5.12: Resultados de Pruebas: Precisión de búsqueda del sitio

Esta encuesta nos ayuda a realmente evaluar si el Framework funciona correctamente o no y si provee los resultados deseados. La precisión del sitio depende exclusivamente de los resultados obtenidos de los distintos servicios a indexar. El Framework se encarga de organizarlos por relevancia, esto implica que resultados de cierto nivel de relevancia en los resultados de los servicios, seguirán manteniendo la mimas en el sitio. Las respuestas negativas (2%) están directamente relacionadas a errores ocurridos en el servidor.

¿Qué tan rápida es tu conexión a internet?

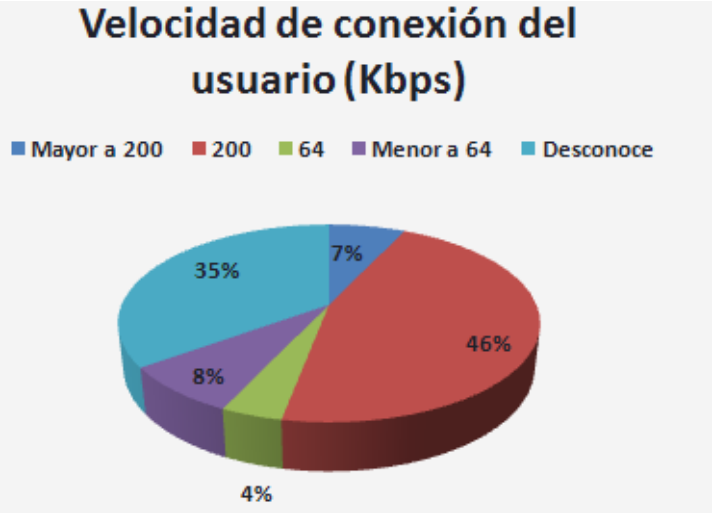


Figura 5.13: Resultados de Pruebas: Velocidad de conexión del usuario

La mayoría de los usuarios poseen una conexión aceptable para el uso del sitio. El sitio demanda un ancho de banda de por lo menos 64 Kbps para la experiencia óptima en el mismo, debido a las múltiples imágenes o video y audio streamings requeridos por el usuario. La conexión a los servicios está restringida por el ancho de banda del servidor donde el sitio se aloja. Como ya se mencionó, los problemas de velocidad de respuesta son exclusivamente del servidor, más no del sitio. Se buscará migrar el sitio a un servidor con un ancho de banda dedicado a futuro para suplir la demanda de pedidos.

¿Cuál es tu proveedor de Internet? (Ej: Satnet, Linkotel, etc.)

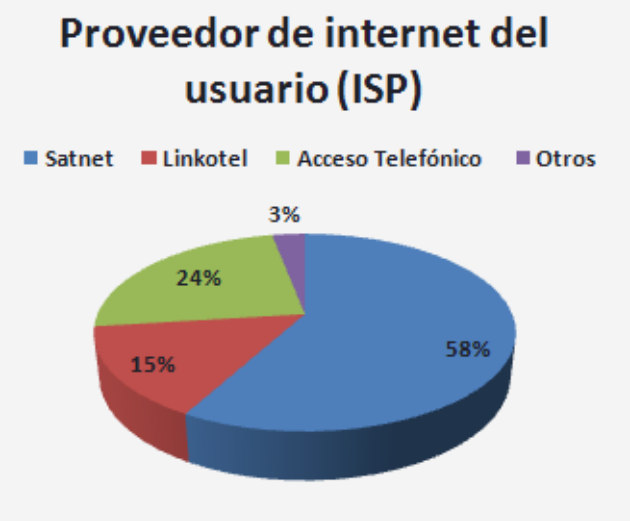


Figura 5.14: Resultados de Pruebas: Proveedor de internet del usuario

La mayoría de los usuarios que se quejan de la lentitud de carga poseen conexiones inferiores a 64Kbps, y son, generalmente, internautas con acceso a internet telefónico, o Dial Up. Se estima que la mayoría de usuarios que desconocen su ancho de banda tienen en promedio una conexión de 64Kps. El 92% de usuarios de Satnet encuestados poseen una conexión a Internet de 200Kbps. Esto nos indica que los usuarios de Satnet que tuvieron problemas de velocidad de conexión en el sitio (un total de 16 usuarios) fueron víctimas de un cuello de botella producido en el servidor, por el bajo ancho de banda del mismo, más no producido por la velocidad de procesamiento de la información en el Framework. Es el servidor el que obtiene e indexa los resultados de los servicios, en donde la respuesta de los servicios está limitada por el ancho de banda del servidor y la indexación es casi inmediata. Se comprobó que el

Framework funciona muy bien en servidores pequeños e incluso en una PC de escritorio que simule un servidor, como es el caso del equipo utilizado para las pruebas.

Con estas estadísticas queda demostrado que el Framework funciona de manera adecuada, al permitir desarrollar un sitio que en las condiciones requeridas para el mismo trabaja de manera óptima. Podemos ver fácilmente que es posible realizar un sitio completamente dinámico con el Framework y que es sencillo de interactuar con éste. Podemos apreciar también que el código del Framework funciona de manera óptima y que dicho código está optimizado para soportar la demanda de peticiones de los usuarios, claro está, bajo las limitaciones del servidor donde el Framework esté alojado. El sitio no ofrece todos los servicios que el Framework posee, como conexión a Feeds, por ejemplo. A pesar de esto podemos ver que el Framework está listo para ser utilizado para el desarrollo de Mashups interactivos.

Conclusiones

Se puede apreciar que el ejemplo realmente muestra la capacidad del Framework, y gracias a las bondades de Ajax fue posible la implementación de un Mashup innovador e interesante, el cual está destinado a la comunidad de desarrollo web y permitirá poner una base para el avance tecnológico en el desarrollo de aplicaciones que utilicen APIs externos. Se han cumplido con todos los objetivos planteados, al

crear un marco de trabajo para generar aplicaciones web multimedia, el cual está listo para ser usado por desarrolladores. Ahora son muchas las posibilidades para las aplicaciones realizables con el FindJira Framework, siendo un buscador integrado como el visto en el ejemplo, una de las más interesantes. Como se mencionó anteriormente, únicamente en la creatividad e innovación del desarrollador está el límite del el uso que recibirá el mismo.

CONCLUSIONES

Para el Framework:

Como se ha visto en el documento, hemos cumplido los objetivos propuestos, analizando los objetivos principales del documento, es apreciable que:

Se han analizado los problemas actuales, y se ha propuesto una solución a un problema actual de integración de medios en línea.

Se ha diseñado un esquema de programación, y un sistema útil para explotar los servicios disponibles.

Se logró integrar distintos medios existentes con el repositorio implementado, el cual permite la mezcla de información de uno o más medios, e incluso la creación de nuevos tipos de repositorios, gracias a las herencias y estructuras jerárquicas implementadas en el Framework.

El marco de trabajo fue optimizado para su uso en línea, y su eficiencia y robustez fue puesta a rigurosas pruebas con un sitio que explota sus propiedades más importantes, con éstas pruebas se pudo comprobar que el Framework está listo para su uso en el desarrollo web.

El uso del lenguaje PHP fue muy útil para el Framework, pues permitió el manejo de formatos web estándar como XML y JSON de manera sencilla, y gracias a la gran comunidad de desarrollo dedicada a este lenguaje, fue sencilla la implementación del Framework, pues la mayoría de los problemas encontrados tenían una solución ya

encontrada por otros miembros de la misma, y fue posible implementar una solución muy cercana al diseño planificado.

Se comprobó que los usuarios desean encontrar información previamente indexada, y gracias a las estadísticas obtenidas, demostramos que es sencillo encontrar información relevante si se busca en varios medios distintos a la vez, de mejor manera si provienen de varias fuentes, ahorrando tiempo de navegación y ancho de banda.

Comprobamos entonces, que existe una demanda para este tipo de soluciones, siendo FindJira Framework la semilla para muchas otras.

Para el ejemplo (Iguana Search):

El uso del Framework en conjunto con una aplicación multimedia interactiva en Ajax, permitió comprobar que el Framework está también listo para su uso en el desarrollo de mashups y su interacción con Ajax es óptima.

El ejemplo fue sencillo de desarrollar utilizando el Framework, lo cual nos permitió comprobar que efectivamente el uso del mismo ahorra esfuerzo de programación.

La construcción de un buscador integrado para la comunidad es un primer paso para el desarrollo de mashups innovadores, y nos permite comprobar que es posible crear nuevas soluciones para la comunidad con el uso del Framework.

RECOMENDACIONES

Para el Framework:

Es recomendable añadir un conjunto mayor de servicios existentes en el web al Framework, servicios cuyos APIs estén disponibles para su uso en la comunidad de desarrolladores, para ampliar el rango de mashups potenciales que se puedan desarrollar con el mismo.

Añadir soporte para bases de datos, para poder tener repositorios persistentes almacenados en un servidor, en el caso de que un desarrollador lo requiera.

Podría recomendarse el desarrollo de un Framework similar en otro lenguaje de programación distinto a PHP, como JSP o ASP, para el soporte sobre otros servidores web, como Tomcat o Internet Information Server.

Sería muy interesante el desarrollo de una aplicación basada en el Framework, para el desarrollo de la lógica de las aplicaciones desarrolladas con el mismo. Es decir, una interfaz gráfica para el uso o trabajo sobre el Framework.

Podría recomendarse el desarrollar versiones del Framework para acoplarse a librerías existentes de desarrollo de sitios dinámicos como Ruby on Rails, Dojo o JQuery, que

permitan integrar de manera aún más sencilla las propiedades del Framework a aplicaciones Ajax que estén basadas en estos Toolkits.

Se puede considerar el desarrollo de funciones de búsqueda, o servicios enteros que corran en el lado del cliente, de manera que la indexación del Framework no se realice en el servidor, para alivianar cargas de ancho de banda y procesamiento sobre el mismo.

Evidentemente, podemos recomendar el uso del Framework para el desarrollo de mashups, al haberse comprobado que es eficiente, usable, sencillo de usar y robusto.

Para el ejemplo (Iguana Search):

Sería muy interesante publicar el ejemplo en un hosting que soporte el Framework y en un dominio público, que provea el ancho de banda óptimo para el uso del Framework y soporte el procesamiento requerido por el mismo, para dejar un servicio de código abierto para la comunidad.

Se pueden añadir muchos más servicios que el Framework soporta al ejemplo como lectura y clasificación de feeds, búsqueda de mapas, e incluso nuevos servicios que puedan ser desarrollados en un futuro.

Gracias a que el diseño del sitio está completamente dirigido por CSS, puede ser posible el desarrollo de nuevas interfaces o formas creativas de mostrar información

al usuario, o el desarrollo de distintos esquemas de color o estilos para el sitio existente.

Es una idea interesante el desarrollar un php_handler que permita realizar consultas a los repositorios de búsquedas, devolviendo resultados en formatos estándar como JSON o XML, desarrollando así una especie de API de búsqueda, similar a los utilizados por el Framework, pero que provea los resultados indexados y embebibles de Iguana Search.

Cabe recalcar que es posible el optimizar aún más el funcionamiento interno del ejemplo. Variaciones posibles pueden ser el uso de JSON para realizar las consultas al Framework, en vez de PHP serializado, o el uso de cookies para evitar el acceso a variables de sesión. El uso de JSON puede hacer posible el optimizar aún más la latencia de las conexiones al servidor, y el uso de soluciones distintas a las variables de sesión pueden reducir la carga de recursos del sitio sobre el navegador.

ANEXOS

ANEXO A

Documentación de los APIs más comunes

A continuación se describen los parámetros, URLs de los servicios y las funciones más importantes de los APIs de los servicios web más comunes.

A.1 Yahoo Web Search Service

Página de Soporte
http://developer.yahoo.com/search/web/V1/webSearch.html

URL del servicio de búsqueda
http://search.yahooapis.com/WebSearchService/V1/webSearch

Tabla A.1.1 Parámetros del servicio de búsqueda para Web, Video e Imágenes

Parámetro	Valor	Descripción
appid	string	El ID del API, es necesario proveer este valor para obtener resultados.
query	string	Este es el query de búsqueda, soporta las palabras clave y formatos adicionales de Yahoo.
results	integer: default *10*, max *100*	El número de resultados que se van a retornar.
start	integer: default *1*	El numero de resultado desde el cual se van a empezar a listar.
language *(solo para web search)*	string: default *no value* (all languages)	El idioma en el cual se desea en el que estén los resultados.
output	string: *xml* (default), *json*, *php*	El formato en el cual estarán codificados los resultados.
callback	string	En el caso de las búsquedas en JSON, es posible envolver los resultados con una función, el nombre de dicha función es el valor a ingresar en este parámetro.

Tabla A.1.2 Principales Campos de la Respuesta

Campo	Descripción
ResultSet	Contiene todas las respuestas obtenidas, sus atributos son: • **totalResultsAvailable**: El número de resultados encontrados. • **totalResultsReturned**: El número de resultados retornados. • **firstResultPosition**: La posición del primer resultado obtenido.
Result	Contiene cada respuesta de manera individual.
Title	El título de la página web.
Summary	El resumen del texto asociado con el resultado.
Url	El URL del resultado.
ClickUrl	El URL de Yahoo que permite la salida a cada resultado.
DisplayUrl	El URL del resultado sin el prefijo http:// con el propósito de visualizar la dirección de la página.
MimeType	El contenido MIME de la página.
ModificationDate	La última fecha de modificación de la página (Formato UNIX timestamp).
Cache	El URL del resultado almacenado en el caché de Yahoo, y su tamaño en bytes.

A.2 Yahoo Image Search Service

Página de Soporte
http://developer.yahoo.com/search/image/V1/imageSearch.html

URL del servicio de búsqueda de imágenes
http://search.yahooapis.com/ImageSearchService/V1/imageSearch

Tabla A.2.1 Principales Campos de la Respuesta

Campo	Descripción
ResultSet	Contiene todas las respuestas obtenidas, sus atributos son: • **totalResultsAvailable**: El número de resultados encontrados. • **totalResultsReturned**: El número de resultados retornados. • **firstResultPosition**: La posición del primer resultado obtenido.
Result	Contiene cada respuesta de manera individual.
Title	El título de la imagen.
Summary	El resumen del texto asociado con el resultado.
Url	El URL del resultado.
ClickUrl	El URL de Yahoo que permite la salida a cada resultado.
RefererUrl	El URL de la página web que contiene la imagen.
FileSize	El tamaño del archivo en bytes.
FileFormat	Puede ser de tipo *bmp*, *gif*, *jpg*, or *png*.
Height	El alto de la imagen en pixeles.
Width	El ancho de la imagen en pixeles.

Thumbnail	El URL de la vista en miniatura.
Publisher	El creador de la imagen
Restrictions	Provee restricciones del servidor en el que se encuentra la imagen, puede ser: • Noframe que quiere decir que no puede ser mostrada dentro de un frame en un sitio. • Noinline quiere decir que no podría mostrarse la imagen pues el servidor tiene restricciones de acceso dependiendo de la referencia a la imagen.
Copyright	El dueño del copyright.

A.3 Yahoo Video Search Service

Página de Soporte
http://developer.yahoo.com/search/video/V1/videoSearch.html

URL del servicio de búsqueda de videos
http://search.yahooapis.com/VideoSearchService/V1/videoSearch

Tabla A.3.1 Principales Campos de la Respuesta

Campo	Descripción
ResultSet	Contiene todas las respuestas obtenidas, sus atributos son: • **totalResultsAvailable**: El número de resultados encontrados. • **totalResultsReturned**: El número de resultados retornados. • **firstResultPosition**: La posición del primer resultado obtenido.
Result	Contiene cada respuesta de manera individual.
Title	El título del video.
Summary	El resumen del texto asociado con el resultado.
Url	El URL del resultado.
ClickUrl	El URL de Yahoo que permite la salida a cada resultado.
RefererUrl	El URL de la página web que contiene la video.
FileSize	El tamaño del archivo en bytes.
FileFormat	Puede ser de tipo *avi*, *flash*, *mpeg*, *msmedia*, *quicktime*, o *realmedia*.
Height	El alto del keyframe [[ref keyframe]] en pixeles que el motor de Yahoo extrajo del vídeo, este es el ancho del video original reproducido al 100%.
Width	El ancho del keyframe en pixeles que el motor de Yahoo extrajo del vídeo, este es el ancho del video original reproducido al 100%.
Duration	La duración del archivo de video en segundos.
Channels	El número de canales de audio, por lo general 1 (mono) y 2 (stereo).
Streamming	Si el video se reproduce por streaming o no. (Streaming: true, o no: false)
Thumbnail	El URL de la vista en miniatura del video.
Publisher	El creador del video.
Restrictions	Provee restricciones del servidor en el que se encuentra el video, puede ser: • Noframe que quiere decir que no puede ser mostrada dentro de un frame en

	un sitio. • Noinline quiere decir que no podría mostrarse la imagen pues el servidor tiene restricciones de acceso dependiendo de la referencia al video.
Copyright	El dueño del copyright.

A.4 Google SOAP Search

Página de Soporte
http://code.google.com/apis/soapsearch/reference.html

URL del servicio de búsqueda de videos
http://api.google.com/search/beta2

Tabla A.4.1 Parámetros del servicio de búsqueda

Nombre	Descripción
key	Provista por Google, esta clave identifica al desarrollador de manera única, y permite monitorear las búsquedas realizadas por las aplicaciones del mismo.
q	El query de búsqueda, más adelante se describen los términos admisibles.
start	El índice del primer resultado de la búsqueda, comenzando en cero.
maxResults	Número de resultados de la búsqueda, el máximo permitido es 10.
filter	Permite el filtrado de los resultados con contenido similar, y el uso de parámetros.
restricts	Permite restringir el rango de la búsqueda a tópicos específicos.
safeSearch	Una variable que activa el filtrado de contenido adulto.
lr	Restringe la búsqueda a ciertos idiomas soportados por Google.

Tabla A.4.2 Parámetros para el query de búsqueda

Nombre de Propiedad	Ejemplo	Descripción
Incluír término	**Ecuador+Guayaquil**	Si un término es esencial para la búsqueda, es posible añadirlo con el signo "+" antes de dicha palabra.
Excluír término	**Sudamérica-Ecuador**	Si un término no es deseado para la búsqueda, es posible sustraerlo con el signo "-" antes de dicha palabra.
Frase exacta	**"día de la independencia"**	Es posible realizar búsquedas con la frase tal como se desea al incluirlas entre comillas.
Operador lógico OR	**vacaciones salinas OR playas**	Permite obtener páginas con resultados de la operación lógica OR.
Restricción de dominio	**becas site:www.espol.edu.ec**	Se pueden realizar búsquedas restringidas a un solo dominio con ésta propiedad.
Restricción por fecha	**Noticias daterange:2452122-2452234**	Es posible limitar los resultados según la fecha de publicación de la página utilizando el formato provisto.

		Las fechas se describen con enteros Julianos, es decir, describen el número de días pasados desde Enero 1, 4713 AC. Por ejemplo, la fecha Juliana de August 1 del 2001 es 2452122.
Búsqueda por título	**intitle:"Index Of"**	Google Search puede restringir los resultados a páginas que tengan en su título una cadena esecífica.
Búsqueda por título (todos los términos)	**allintitle: Index Of**	El query debe empezar con el término allintitle, y permite buscar resultados con todas las palabras descritas. El ejemplo de esta propiedad y el de la propiedad anterior son equivalentes.
Búsqueda dentro del url	**inurl:ecuador inurl:viajes**	Los términos que siguen al parámetro inurl deberán estar en el URL de la página de resultados.
Búsqueda dentro del url (todos los términos)	**allinurl: ecuador viajes**	Permite que todos los términos dados estén en el URL de resultados.
Filtrado por tipo de archivo	**Inteligencia Artificial filetype:pdf OR filetype:doc**	Permite realizar búsquedas y restringirlas sólo a documentos de cierto tipo.
Exlusión de tipo de archivo	**Inteligencia Artificial -filetype:doc -filetype:pdf**	Permite excluir ciertos tipos de archivo de los resultados de búsqueda.
Links de retorno	**link:www.espol.edu.ec**	Este término permite mostrar todas las páginas que vinculen al sitio especificado.
Resultados del Cache	**cache:www.eluniverso.com**	Permite buscar entre las páginas guardadas en el cache de Google.

Tabla A.4.3 Principales Campos de la Respuesta

Campo	Descripción
documentFiltering	Una variable que indica si se habilitó el filtrado en la búsqueda.
searchComments	Comentarios adicionales acerca de la búsqueda orientados para un usuario final. Un ejemplo es la exclusión de palabras comunes.
estimatedTotalResultsCount	El resultado estimado de todos los resultados en la búsqueda.
estimateIsExact	Un valor indicando si el estimado anterior es verdadero o falso.
resultElements	Un arreglo de items de <resultElement>. Corresponde a la lista completa de resultados.
searchQuery	El valor del query de búsqueda recibido
startIndex	Indica el índice del primer resultado obtenido.
endIndex	Indica el índice del último resultado obtenido.
searchTips	Un string con sugerencias del uso apropiado de Google.
directoryCategories	Un arreglo de ítems de <directoryCategory>.
searchTime	Un texto que representa un número de tipo de dato flotante describiendo el tiempo exacto en segundos que tardó la búsqueda.

summary	El resumen de la página en la descripción.
URL	El URL del resultado.
snippet	Un extracto del texto de la página que contiene los términos de la búsqueda.
title	El título de la página resultante, extraído de la cabecera HTML.
cachedSize	Indica el tamaño de la página en el cache de Google.
relatedInformationPresent	Booleano que indica que existen páginas relacionadas a este resultado.
hostName	Cuando más de un resultado viene de la misma página, el hostName es la dirección del sitio en común de ambos.

A.5 SearchMash

URL del servicio de búsqueda

http://www.searchmash.com/results/

Tabla A.5.1 Parámetros del servicio de búsqueda

Nombre	Descripción
[q]	El query de búsqueda se escribe directamente, se pueden utilizar los mismos parámetros de búsqueda web que se usan en Google SOAP.
images:[q]	Para realizar búsqueda de imágenes se antepone el prefijo **images:**
blogs:[q]	Para realizar búsqueda de blogs se antepone el prefijo **blogs:**
video:[q]	Para realizar búsqueda de videos se antepone el prefijo **video:**
wikipedia:[q]	Para realizar búsqueda de artículos de wikipedias se antepone el prefijo **wikipedia:**
i	El índice del primer resultado de la búsqueda, comenzando en 1.
n	Número de resultados de la búsqueda, el máximo permitido es 10

Tabla A.5.2 Principales Campos de la Respuesta

Campo	Descripción
estimatedCount	El resultado estimado de todos los resultados en la búsqueda.
moreResults	Un valor indicando si el estimado anterior es verdadero o falso.
event	Un código de manejo interno de SearchMash que describe el evento de búsqueda actual. No hay documentación al respecto.
query	El valor del query de búsqueda recibido, tiene a su vez dos campos o atributos, prefix y terms.
prefix	es el filtro que describe el medio en el que se realiza la búsqueda, puede ser uno entre: video, images, blogs, wikipedia
terms	Las palabras claves de la búsqueda.
results	El arreglo de resultados de búsqueda.
rawUrl	El URL nativo de la página que contiene el medio encontrado: Ej: http://www.espol.edu.ec/index.html
url	El URL en formato UTF-8.
site	El URL del resultado.

snippet	Un extracto del texto de la página que contiene los términos de la búsqueda.
title	El título de la página resultante, extraído de la cabecera HTML.
cacheUrl	El URL del resultado en el cache de Google.
displayUrl	La dirección del dominio en donde está el resultado: Ej: www.espol.edu.ec
imageWidth	El ancho de la imagen encontrada. (Sólo para imágenes)
imageHeight	El alto de la imagen encontrada. (Sólo para imágenes)
imageUrl	El URL de la imagen encontrada. (Sólo para imágenes)
thumbnailWidth	El ancho de la vista en miniatura de la imagen o el video encontrado. (Para imágenes y videos)
thumbnailHeight	El alto de la vista en miniatura de la imagen o el video encontrado. (Para imágenes y videos)
thumbnailUrl	El URL de la vista en miniatura de la imagen o el video encontrado. (Para imágenes y videos)
duration	La duración en segundos del video encontrado. (Sólo para videos)
author	El autor del video encontrado. (Sólo para videos)
videoUrl	El url del video encontrado. (Sólo para videos)

A.6 Flickr

Página de Soporte
http://www.flickr.com/services/api/

URL del servicio de búsqueda
http://api.flickr.com/services/rest/

Tabla A.6.1 Parámetros del servicio de búsqueda de Imágenes

Nombre	Descripción
method	Para realizar búsquedas en la base de datos de Flickr, se utiliza el método: flickr.photos.search
api_key	Provista por Flickr, esta clave identifica al desarrollador de manera única, y permite monitorear las búsquedas realizadas por las aplicaciones del mismo.
text	El query de búsqueda, se incluye en la búsqueda fotos en cuyo título, descripción o tags coinciden con lo descrito.
format	El formato en el cual estarán codificados los resultados, puede ser *rest, xmlrpc, json, php_serial o soap2*
per_page	Es el número de resultados mostrados por página, el default es 100 y el máximo permitido es 500
page	Es el número de página de resultados mostrada, si es omitido se mostrará la primera.

Tabla A.6.2 Principales Campos de la Respuesta

Campo	Descripción
pages	El total de páginas de resultados encontrados.
page	El índice de la página actual de resultados, comenzando en 1.

perpage	El total de resultados por página.
total	El número total de resultados encontrados.
photo	El resultado encontrado, posee atributos necesarios para la identificación de la misma, como id, owner y title.
id	El id de la imagen.
owner	El código del usuario que subió la imagen.
secret	El código secreto de la imagen.
server	El código del servidor en el que está ubicada la imagen.
farm	El código de la granja de servidores que contiene la imagen.
title	El titulo de la imagen.
ispublic	Variable que indica si la foto encontrada es pública, 1 = pública, 0 = privada.
isfriend	Variable que indica si el usuario que hizo el query está relacionado con el usuario que se especifica en el query, por medio del parámetro user_id.
isfamily	Variable que indica si el usuario que hizo el query está emparentado con el usuario que subió la foto, por medio del parámetro user_id.

Elaboracion de una imagen

Para elaborar el URL de una imagen de Flickr es necesario seguir las siguiente sintaxis:

```
http://farm{farm-id}.static.flickr.com/{server-
id}/{id}_{secret}_{size}.{extension}
```

La misma va a ser llenada con campos de la respuesta retornada, que son los siguientes:

Tabla A.6.3 Campos de la respuesta de detalle de imagen

Código	Descripción
farm-id	El código de granja de servidores retornado en la respuesta.
server-id	El código de servidor retornado en la respuesta.
photo-id	El código de la imagen retornado en la respuesta.
secret	El código secreto e la imagen retornado en la respuesta.
size	El tamaño en el que se desea la foto: s: cuadrado pequeño de 75x75 pixeles t: thumbnail, 100 pixeles en el lado más largo m: pequeño, 240 pixeles en el lado más largo -: mediano, 500 pixeles en el lado más largo b: grande, 1024 pixeles en el lado más largo (sólo para imágenes muy grandes) o: tamaño original, en este caso es obligatorio conocer la extensión original.
extension	La extensión del archivo. Puede ser: jpg, gif o png.

Elaboración de la página contenedora de la imagen

Para visualizar la página que contiene la imagen desde el sitio de Flickr, es necesario realizar otra petición al servicio con el método **flickr.photos.getInfo** enviándole el

api_key, **photo_id** y el **secret**. Igualmente se obtiene una respuesta cuyo campo **URLs** contendrá la información necesaria para encontrar el url deseado.

A.7 YouTube

Página de Soporte
http://www.youtube.com/dev_docs

URL del servicio de búsqueda
http://www.youtube.com/api2_rest

Tabla A.7.1 Parámetros del servicio de búsqueda de Imágenes

Nombre	Descripción
method	Para realizar búsquedas en la base de datos de Youtube, se utiliza el método: youtube.videos.list_by_tag
dev_id	Provista por Youtube, esta clave identifica al desarrollador, y permite monitorear las búsquedas realizadas por las aplicaciones del mismo.
tag	El query de búsqueda, se incluye en la búsqueda fotos en cuyos tags coinciden con lo descrito.
page	Retorna la página de resultados que se desea ver en la respuesta.
per_page	El número de resultados por página que se mostrarán en la respuesta.

Tabla A.7.2 Campos principales de la respuesta

Campo	Descripción
ut_response	Envuelve la respuesta de búsqueda. Indica si el resultado de la búsqueda es correcto (status="ok") o incorrecto (status="fail").
video_list	Envuelve la lista de videos resultantes de la búsqueda.
video	Envuelve un resultado en la búsqueda.
author	Envuelve el nombre de usuario del cliente que subió el video en cuestión.
id	Describe el id hexadecimal único que describe al video resultante. Este id puede ser usado para embeber el video en una página web.
title	El título del video.
length_seconds	La longitud del video en segundos.
rating_avg	Un número flotante del 1 al 5 que indica el promedio de estrellas que califican el video resultante.
rating_count	El número de veces que el video resultante ha sido calificado.
description	Descripción del video en cuestión.
view_count	El número de veces que el video ha sido visualizado.
upload_time	El tiempo en segundos que demoró el video en ser subido a los servidores de youtube.
comment_count	El número de comentarios que ha recibido el video por parte de los usuarios.
tags	Los tags que describen al video resultante y que se relacionan a los tags requeridos en la búsqueda.
url	El url del video resultante en cuestión.

thumbnail_url	El url de la vista en miniatura del video resultante.

Elaboracion de un objeto video

Para elaborar el objeto video de YouTube, listo para ser reproducido, es necesario seguir las siguientes sintaxis:
El objeto embebible sigue el formato siguiente en código HTML:

```
<object width="{ancho}" height="{alto}">
    <param name="movie"
value="http://www.youtube.com/v/{id}"></param>
    <param name="wmode" value="transparent"></param>
    <embed       src="http://www.youtube.com/v/{id}"
          type="application/x-shockwave-flash"
          wmode="transparent"
          width="{ancho} " height="{alto}" >
    </embed>
</object>
```

El mismo va a ser llenado con campos de la respuesta retornada, que son los siguientes:

Tabla A.7.3 Campos necesarios para el objeto embebible

Nombre del Código	Descripción
id	Describe el id hexadecimal único que describe al video resultante.
ancho	El ancho del objeto que se desea para su visualización, por defecto es 425 pixeles.
alto	El alto del objeto que se desea para su visualización, por defecto es 350 pixeles.

A.8 Wikipedia Export

Página de Soporte
http://www.mediawiki.org/wiki/Parameters_to_Special:Export

URL del servicio de búsqueda
http://{Dominio del Wiki}/wiki/Special:Export/{Nombre del Artículo}

En donde el dominio es la dirección base del wiki que contiene los artículos donde se realiza la búsqueda. Ejemplos de dominios son Wikipedia, Wikia, entre otros.

El nombre del artículo es el string con el cual se ha creado una entrada en la base del wiki, reemplazando los espacios con subguiones.

A partir de este punto se procede a adjuntar al string un parámetro de búsqueda que es el nombre del artículo perteneciente al wiki.

Tabla A.8.1 Principales Campos de la Respuesta

Campo	Descripción
sitename	El nombre del dominio del wiki donde está el artículo encontrado.
base	La página principal del wiki donde está el artículo.
generator	La versión del generador del artículo delwiki.
namespaces	Contiene un listado de sitios internos del wiki.
page	Page contiene la información de la página del artículo encontrado. Este puede ser un artículo o una des-ambiguación.
title	El título del artículo.
id	El id del artículo en cuestión.
revision	Contiene la información de la última edición del artículo.
timestamp	Contiene la información de fecha y hora exacta de la última edición del artículo.
contributor	Contiene la información del ID y el nombre de usuario del último usuario en editar el artículo.
comment	Comentarios del autor acerca de su última edición.
text	Contiene el artículo completo del wiki. Está en el formato desarrollado por Wikimedia para sus artículos.

A.9 Odeo

La búsqueda de podcasts en Odeo es provista por un buscador web, y es posible obtener los resultados por medio de inspección de los resultados obtenidos. Gracias al apoyo de Evan Williams, su creador, se encontró una forma de obtener información de un podcast en específico en formato JSON, una vez obtenido un ID, pues no existe una página de soporte.

URL del servicio de búsqueda

Se busca en Google o Yahoo, y de los URLs de los resultados es posible encontrar los ID de los archivos de audio alojados en Odeo.

Luego de esto es posible utilizar este URL.

http://odeo.com/audio/{ID del archivo de audio}/json
esta respuesta es individual para cada archivo de audio.

Tabla A.9.1 Principales Campos de la Respuesta

Campo	Descripción
image_url	URL de la imagen representativa de un podcast.
redirect_mp3_url	URL del mp3 archivado en los servidores de Odeo.
status	Indica el estado del podcast, si es en vivo o es archivado.

artist	Nombre del artista, en caso de existir.
title	El título del podcast.
external_url	URL del podcast en sevidores ajenos a Odeo.
id	Código identificador del podcast en Odeo.
good_sample_rate	Booleano que indica la calidad del archivo de audio.
duration	Duración en segundos del podcast.

A.10 ITunes

iTunes no provee un API de búsqueda, pero existe una manera de realizar requests a los servidores de Apple para obtener respuestas JSON. La búsqueda en ITunes fue cambiada por Apple durante el desarrollo del Framework.

URL del servicio de búsqueda anterior
http://www.apple.com/itunes/scripts/itmsSearch.php?searchTerm={query}&searchType={tipo de audio}

URL del servicio de búsqueda nuevo
http://www.apple.com/global/scripts/ajax_proxy.php?s=12&r={encode_url_de_los_parametros_de_busqueda}

Los parámetros de búsqueda deben ir codificados a url, y éstos deben tener la forma:

'&term={termino_de_busqueda}&entity={tipo_de_audio}&country=us&limit=40'

Donde entity puede ser: album, allTrack, musicVideo, movie, tvSeason, audiobookLockup o podcast.

Tabla A.10.1 Principales Campos de la Respuesta

Campo	Descripción
resultCount	El total de resultados obtenidos, como máximo 100.
image_url	URL de la imagen representativa de un podcast.
wrapperType	Si es un track o un album.
mediaType	Indica el tipo de audio que fué encontrado: podcast, music, etc.
artistName	Nombre del artista, en caso de existir.
artistLinkUrl	Link al sitio del artista, en caso de existir.
artworkUrl60	URL de la imagen del podcast, en resolución 60x60.
artworkUrl100	URL de la imagen del podcast, en resolución 100x100.
country	País de origen del archivo de audio.
currency	Unidad monetaria del archivo de audio.
discCount	Número de discos del artista.
discNumber	Número del disco al cual el archivo de audio pertenece.
itemExplicitness	Especifica si el archivo posee contenido explícito o no.
itemParentPrice	Precio del ítem en su unidad monetaria respectiva.

searchRelevance	Qué tan relevante es la búsqueda.
previewUrl	URL fuente del archivo de audio.
primaryGenreName	Categoría del archivo, ej:TV & Film, Rock, Pop
trackCount	Número de pistas en el album al que este audio pertenece.
trackNumber	Número de pista en el disco del archivo de audio.
trackTime	Tiempo de duración del archivo. En podcasts este es null.
title	El título del podcast.
itemName	Nombre de la cadena de podcasts a la cual pertenece este archivo.
id	Código identificador del podcast en Odeo.
good_sample_rate	Booleano que indica la calidad del archivo de audio.
duration	Duración en segundos del podcast.

A.11 Video Downloader

El servicio de descarga de videos de Javimoya, un blogger desarrollador, dueño del dominio www.javimoya.com. Este servicio permite la obtención de vínculos para la descarga de videos de múltiples fuentes como Google Video y YouTube, con simplemente proveer el URL del video en cuestión como parámetro.

URL del servicio de búsqueda
http://videodownloader.net/get/?url=[dirección_del_video]

La dirección del video es la de la página contenedora del video, como por ejemplo:

Para Google Video: http://video.google.com/videoplay?docid=[video_id]
Para YouTube: http://youtube.com/watch?v=[codigo]

A.12 Generador de Vistas Previas

Un servicio de generación de vistas previas recibe como parámetro una dirección cualquiera y retorna una imagen embebible con la vista previa del dominio dado. En caso de no existir una vista previa del sitio en cuestión, dicho sitio es encolado para su captura. Existen varios servicios, algunos de ellos gratuitos, que cumplen este propósito, entre los cuales se escogió Girafa para el Framework. A continuación se presenta una tabla con todos los servicios de uso libre disponibles.

Tabla A.12.1 Servicios de vistas previas y sus URLs de petición

Servicio	URL de petición
Thumbshots.de	http://www.thumbshots.de/cgi-bin/show.cgi?url=[url_del_sitio]
Artviper.com	http://www.artviper.net/screenshots/screener.php?&frameborder=5&

	url=[url_del_sitio]&q=70&h=90&w=120%0D%0A&sdx=0&sdy=0
WebSiteThumbnails.net	http://www.websitethumbnails.net/view.php?url=[url_del_sitio]
Thumbshots.org	http://open.thumbshots.org/image.pxf?url=[url_del_sitio]
WebDesignbook.net	http://webdesignbook.net/snapper.php?w=120&h=90&url=[url_del_sitio]
WebShotsPro.com	http://www.webshotspro.com/thumb.php?url=[url_del_sitio]
WebSnapr.com	http://images.websnapr.com/?url=[url_del_sitio]
img.SimpleAPI.net	http://img.simpleapi.net/small/?&url=[url_del_sitio]
Thumbzor.com	http://www.thumbzor.com/tel.php?url=[url_del_sitio]
M-Software.de	http://www.m-software.de/screenshot/Screenshot.png?scale=6&width=120&height=90&url=[url_del_sitio]
Girafa.com	http://msnsearch.srv.girafa.com/srv/i?i=%204ac8babd1cf73ace&s=MSNSEARCH&cy=en-us&r=[url_del_sitio]

A.13 Technorati

Un servicio de búsqueda en blogs y feeds. Technorati ofrece un API de desarrollo para que un desarrollador obtenga resultados de un término en específico. Es necesario un API key para el uso de este servicio. Los resultados obtenidos están en XML-RPC.

URL del servicio de búsqueda
http://api.technorati.com/search?key=[api_key]&query=[termino_de_busqueda]&start=[resultado_de_inicio]&limit=[numero_de_resultados]

Donde query es el parámetro de búsqueda, start, el índice de inicio y limit el número de resultados a mostrar.

Tabla A.13.1 Principales Campos de la Respuesta

Campo	Descripción
result	La cabecera de la respuesta obtenida
query	El parámetro de búsqueda
querycount	Número de Resultados que concuerdan con la búsqueda.
rankingstart	Resultado inicial
item	Elementos de la respuesta
weblog	Blog encontrado
name	Nombre del sitio o feed
url	URL del sitio o feed
rssurl	URL del feed RSS, si existe
atomurl	URL del feed ATOM, si existe
inboundblogs	Número de sub blogs de este dominio

A.14 Microsoft Live Web Search Service

URLs de los servicios de búsquedas

http://search.live.com/results.aspx
http://search.live.com/images/results.aspx
http://search.live.com/feeds/results.aspx

Tabla A.14.1 Parámetros del servicio de búsqueda para Web, Video e Imágenes

Parámetro	Valor	Descripción
q	string	Este es el query de búsqueda.
output	string: *xml* (default)	El formato en el cual estarán codificados los resultados.

Tabla A.1.2 Principales Campos de la Respuesta

Campo	Descripción
Documentset (Source = "FEDERATOR MONARCH")	Contiene todas las respuestas obtenidas, sus atributos son: • **total**: El número de resultados encontrados. • **count**: El número de resultados retornados. • **start**: La posición del primer resultado obtenido.
Document	Contiene cada respuesta de manera individual.
Title	El título de la página web.
Desc	El resumen del texto asociado con el resultado.
Url	El URL del resultado.
DisplayUrl	El URL que permite la salida a cada resultado.
Languaje	El lenguaje de la página web.
ImageUrl	El Url de la imagen (en resultados de imágenes).
ThumbnailUrl	El Url de la vista previa (en resultados de imágenes).

ANEXO B

API DE FINDJIRA FRAMEWORK

CLASES Y MÉTODOS

HANDLER

CLASE	**handler**
DETALLES	Esta clase es una interfaz entre el framework y el usuario, siendo parte aun del framework, permite una integración entre servicios y repositorios.

FUNCIÓN	**constructor**
DESCRIPCIÓN	Aquí se inicializan todos los componentes y se crean los objetos de tipo repositorio y servicios
USO	Automáticamente al crear un objeto de esta clase
RECIBE	n/a
DEVUELVE	n/a

FUNCIÓN	**limpiar_repositorios**
DESCRIPCIÓN	Vacía el contenido de todos los repositorios usados
USO	Cada vez que se desee limpiar el contenido de los repositorios
RECIBE	n/a
DEVUELVE	n/a

FUNCIÓN	preparar_busqueda
DESCRIPCIÓN	Prepara las variables para ser utilizadas en la búsqueda simultanea
USO	Antes de llamar las funciones de do_*_search
RECIBE	query numero de resultados por pagina resultado inicial de esa página ancho de vista previa alto de vista previa ancho de embed alto de embed restricción a sitio web buscar este texto en títulos buscar este texto en URL tipos de archivos deseados palabras excluidas de la búsqueda
DEVUELVE	n/a

FUNCIÓN	extraer_repositorio_video
DESCRIPCIÓN	Extrae el contenido del repositorio de video
USO	Cada vez que se solicite el arreglo que conforma el repositorio de video
RECIBE	índice inicial de elemento

	índice final de elemento
DEVUELVE	Un arreglo con el repositorio de video

FUNCIÓN	**extraer_repositorio_audio**
DESCRIPCIÓN	Extrae el contenido del repositorio de audio
USO	Cada vez que se solicite el arreglo que conforma el repositorio de audio
RECIBE	índice inicial de elemento
	índice final de elemento
DEVUELVE	Un arreglo con el repositorio de audio

FUNCIÓN	**extraer_repositorio_feed**
DESCRIPCIÓN	Extrae el contenido del repositorio de feed
USO	Cada vez que se solicite el arreglo que conforma el repositorio de feed
RECIBE	índice inicial de elemento
	índice final de elemento
DEVUELVE	Un arreglo con el repositorio de feed

FUNCIÓN	**extraer_repositorio_image**
DESCRIPCIÓN	Extrae el contenido del repositorio de imágenes
USO	Cada vez que se solicite el arreglo que conforma el repositorio de imágenes
RECIBE	índice inicial de elemento

	índice final de elemento
DEVUELVE	Un arreglo con el repositorio de imágenes

FUNCIÓN	**extraer_repositorio_web**
DESCRIPCIÓN	Extrae el contenido del repositorio web
USO	Cada vez que se solicite el arreglo que conforma el repositorio web
RECIBE	índice inicial de elemento
	índice final de elemento
DEVUELVE	Un arreglo con el repositorio web

FUNCIÓN	**extraer_repositorio_file**
DESCRIPCIÓN	Extrae el contenido del repositorio de archivos
USO	Cada vez que se solicite el arreglo que conforma el repositorio de archivos
RECIBE	índice inicial de elemento
	índice final de elemento
DEVUELVE	Un arreglo con el repositorio de archivos

FUNCIÓN	**do_all_search**
DESCRIPCIÓN	Realiza todas las búsquedas simultáneamente
USO	Ejecuta las búsquedas de todos los tipos de medios pero este proceso es muy costoso
RECIBE	n/a

DEVUELVE	n/a

FUNCIÓN	**do_video_search**
DESCRIPCIÓN	Realiza búsqueda de videos
USO:	Cada vez que se desee ejecutar una búsqueda de videos
RECIBE:	elemento inicial de resultados
	total de resultados deseados
DEVUELVE:	n/a

FUNCIÓN	**do_audio_search**
DESCRIPCIÓN	Realiza búsqueda de audio
USO	Cada vez que se desee ejecutar una búsqueda de audio
RECIBE	elemento inicial de resultados
	total de resultados deseados
DEVUELVE	n/a

FUNCIÓN	**do_feed_search**
DESCRIPCIÓN	Realiza búsqueda de feeds
USO	Cada vez que se desee ejecutar una búsqueda de feeds
RECIBE	elemento inicial de resultados
	total de resultados deseados

DEVUELVE	n/a

FUNCIÓN	**do_image_search**
DESCRIPCIÓN	Realiza búsqueda de imágenes
USO	Cada vez que se desee ejecutar una búsqueda de imágenes
RECIBE	elemento inicial de resultados
	total de resultados deseados
DEVUELVE	n/a

FUNCIÓN	**do_web_search**
DESCRIPCIÓN	Realiza búsqueda de páginas web
USO	Cada vez que se desee ejecutar una búsqueda de páginas web
RECIBE	elemento inicial de resultados
	total de resultados deseados
DEVUELVE	n/a

FUNCIÓN	**do_indexof_search**
DESCRIPCIÓN	Realiza búsqueda en directorios
USO	Cada vez que se desee ejecutar una búsqueda en directorios
RECIBE	elemento inicial de resultados
	total de resultados deseados

DEVUELVE	n/a

FUNCIÓN	**setear_queries**
DESCRIPCIÓN	Asigna el query en los objetos de servicios
USO	Es utilizada en preparar_busqueda
RECIBE	query
DEVUELVE	n/a

FUNCIÓN	**setear_queries**
DESCRIPCIÓN	Asigna el query en los objetos de servicios
USO	Es utilizada en preparar_busqueda
RECIBE	query
DEVUELVE	n/a

FUNCIÓN	**setear_resultado_inicial**
DESCRIPCIÓN	Asigna el resultado inicial en los objetos de servicios
USO	Es utilizada en preparar_busqueda
RECIBE	índice de resultado inicial
DEVUELVE	n/a

FUNCIÓN	**setear_dimensiones_thumbnail**
DESCRIPCIÓN	Asigna las dimensiones de la vista previa en los objetos de servicios

USO	Es utilizada en preparar_busqueda
RECIBE	ancho y alto
DEVUELVE	n/a

FUNCIÓN	**setear_dimensiones_embed**
DESCRIPCIÓN	Asigna las dimensiones del objeto incrustable en los objetos de servicios
USO	Es utilizada en preparar_busqueda
RECIBE	ancho y alto
DEVUELVE	n/a

FUNCIÓN	**setear_dimensiones_embed**
DESCRIPCIÓN	Hacer un hipervínculo del los servicios de un resultado dado un string separado con comas
USO	Es usado en los php_handlers para mostrar en el GUI
RECIBE	un string con palabras que representan los servicios separadas por coma
	el query
DEVUELVE	un string con formato HTML con hipervínculos de imágenes apuntando hacia la búsqueda en cada servicio.

FUNCIÓN	**search_n_medio**
DESCRIPCIÓN	Búsquedas iterativas para cada servicio y en cada una recibiendo un trozo de resultados
USO	Dado que cada servicio tiene sus límites, esta función llamada en los do_*_search basado en el

	límite que desea el usuario
RECIBE	objeto de servicio
	un string que representa el medio
	elemento inicial de resultados
	total de resultados deseados
DEVUELVE	n/a

ATRIBUTOS DE LA CLASE	
$query	Es una cadena de caracteres con las palabras claves de búsqueda introducidas por el usuario de la interfaz gráfica.
$resultados_por_pagina	Número de resultados que se mostrarán en una página.
$thumbnail_width	Ancho en píxeles de la resolución de la imagen de la vista previa.
$thumbnail_height	Alto en píxeles de la resolución de la imagen de la vista previa.
$site	Restringir la búsqueda a este dominio.
$intitle	Restringir la búsqueda a las palabras claves de éste título.
$inurl	Restringir la búsqueda a las palabras claves de ésta ubicación URL.
$filetype	Restringir la búsqueda a este tipo de archivos.
$excluidos	No mostrar resultados con estas palabras (separadas con barra vertical "|").
$rep_audio	Objeto que representa al repositorio de audio.
$rep_feed	Objeto que representa al repositorio de feeds.

$rep_image	Objeto que representa al repositorio de imágenes.
$rep_web	Objeto que representa al repositorio de páginas web.
$rep_video	Objeto que representa al repositorio de videos.
$rep_file	Objeto que representa al repositorio de archivos.
$ser_flickr	Objeto que representa al servicio de Flickr.
$ser_itunes	Objeto que representa al servicio de iTunes.
$ser_live	Objeto que representa al servicio de Live.
$ser_odeo	Objeto que representa al servicio de Odeo.
$ser_searchmash	Objeto que representa al servicio de Searchmash (Google).
$ser_technorati	Objeto que representa al servicio de Technorati.
$ser_yahoo	Objeto que representa al servicio de Yahoo.
$ser_youtube	Objeto que representa al servicio de Youtube.

REPOSITORIO

CLASE	**repositorio**
DETALLES	Provee de los métodos de manejo del repositorio de datos, tanto como inserción y lectura de elementos del mismo.

FUNCIÓN	**constructor**
DESCRIPCIÓN	Aquí se inicializan todos los componentes
USO	Automáticamente al crear un objeto de esta clase
RECIBE	n/a

DEVUELVE	n/a

FUNCIÓN	**extraer_batch**
DESCRIPCIÓN	Extrae un rango de elementos del repositorio
USO	Para extraer dicho rango usándose en orientación a objetos
RECIBE	índice inicial
	índice final
DEVUELVE	un arreglo con los elementos deseados en el rango

FUNCIÓN	**extraer_batch_externa**
DESCRIPCIÓN	Extrae un rango de elementos del repositorio
USO	Para extraer dicho rango, esta función puede ser utilizada desde afuera para cualquier arreglo
RECIBE	un arreglo con elementos a ser recorridos
	índice inicial
	índice final
DEVUELVE	un arreglo con los elementos deseados en el rango

FUNCIÓN	**extraer_elementos**
DESCRIPCIÓN	Extrae todos los elementos del repositorio
USO	Casi no es necesario, pero si se desea extraer todo el repositorio con este método se lo puede lograr
RECIBE	n/a

DEVUELVE	un arreglo con los elementos del repositorio

FUNCIÓN	**insertar**
DESCRIPCIÓN	Inserción de un elemento de tipo FindJira al repositorio de datos
USO	De uso interno ya que normalmente no se llama a esta función desde afuera de la clase
RECIBE	un elemento para ser parte del arreglo asociativo
DEVUELVE	true / false

FUNCIÓN	**insertar_batch**
DESCRIPCIÓN	Insertar un arreglo de elementos de tipo FindJira al repositorio de datos
USO	De uso interno ya que normalmente no se llama a esta función desde afuera de la clase
RECIBE	un rango de elementos para ser parte del arreglo asociativo
DEVUELVE	n/a

FUNCIÓN	**limpiar**
DESCRIPCIÓN	Vaciar el repositorio
USO	Cada vez que se desee borrar los datos contenidos en el repositorio
RECIBE	n/a
DEVUELVE	n/a

FUNCIÓN	obtener_dominio
DESCRIPCIÓN	Obtener un dominio a partir de un URL valido
USO	Método utilitario que retorna el dominio de un URL
RECIBE	un string que contiene un URL valido
DEVUELVE	un string solo con el dominio

CAMPOS DE LOS REPOSITORIOS

	Web	Feeds	Imágenes	Videos	Audio
Título.	x	x	x	x	x
Raw URL.	x	x	x	x	x
Display URL.	x	x	x	x	x
Caché URL.	x	x			
Feed URL.		x			
Container URL			x	x	x
Download URL				x	x
File size			x	x	x
File format			x	x	x
Resumen del resultado.		x		x	x
Width			x	x	
Height			x	x	
Duración				x	
Thumbnail URL.	x	x	x	x	x
Thumbnail URL (recortado).	x	x	x	x	x

Embed Thumbnail.	x	x	x	x	x
Embed HTML				x	x
Servicio de origen.	x	x	x	x	x
Índice del servicio.	x	x	x	x	x
Índice del repositorio.	x	x	x	x	x
ID del Post		x			
Fecha del Post		x			
Categoría del Post		x			
Título del Post		x			

SERVICE

CLASE	**service**
DETALLES	provee todos los métodos necesarios para la conexión con los servicios web así como otras herramientas para su parseo, esta clase es padre de las demás clases de servicios

FUNCIÓN	**constructor**
DESCRIPCIÓN	Aquí se inicializan todos los componentes
USO	Automáticamente al crear un objeto de esta clase
RECIBE	n/a
DEVUELVE	n/a

FUNCIÓN	**get_findjira_path**
DESCRIPCIÓN	Retorna el directorio donde se encuentra ubicado el FindJira framework
USO	Para obtención del parámetro
RECIBE	n/a

DEVUELVE	un string con el path

FUNCIÓN	**set_findjira_path**
DESCRIPCIÓN	Asigna la ubicación del directorio de FindJira framework
USO	Para asignación del parámetro
RECIBE	un string con el path
DEVUELVE	n/a

FUNCIÓN	**get_respuesta_findjira**
DESCRIPCIÓN	Retorna la respuesta parseada para que sea usada por handler y sea ingresada al repositorio
USO	Para obtención de la respuesta ya procesada del servicio
RECIBE	n/a
DEVUELVE	un arreglo de formato FindJira que contiene los resultados encontrados por este servicio

FUNCIÓN	**get_respuesta_raw**
DESCRIPCIÓN	Retorna la respuesta nativa de cada servicio
USO	Para obtención de la respuesta nativa del servicio
RECIBE	n/a
DEVUELVE	un string conteniendo la respuesta obtenida por conexión HTTP

FUNCIÓN	**set_query**
DESCRIPCIÓN	Asigna el query para la búsqueda
USO	De uso interno
RECIBE	un string con el query
DEVUELVE	n/a

FUNCIÓN	**get_query**

DESCRIPCIÓN	Retorna el query asignado actualmente
USO	De uso interno
RECIBE	n/a
DEVUELVE	un string con el query

FUNCIÓN	**set_thumbnail_width**
DESCRIPCIÓN	Asigna el ancho para la vista previa
USO	De uso interno
RECIBE	la medida del ancho en pixeles
DEVUELVE	n/a

FUNCIÓN	**set_thumbnail_height**
DESCRIPCIÓN	Asigna el alto para la vista previa
USO	De uso interno
RECIBE	la medida del alto en pixeles
DEVUELVE	n/a

FUNCIÓN	**get_thumbnail_width**
DESCRIPCIÓN	Retorna el ancho asignado actualmente para la vista previa
USO	De uso interno
RECIBE	n/a
DEVUELVE	la medida del ancho en pixeles

FUNCIÓN	**get_thumbnail_height**
DESCRIPCIÓN	Retorna el alto asignado actualmente para la vista previa
USO	De uso interno
RECIBE	n/a
DEVUELVE	la medida del alto en pixeles

FUNCIÓN	set_resultados_por_pagina
DESCRIPCIÓN	Asigna el numero de resultados por pagina de búsqueda
USO	De uso interno
RECIBE	el número de resultados deseados a recibir por respuesta del servicio
DEVUELVE	n/a

FUNCIÓN	get_resultados_por_pagina
DESCRIPCIÓN	Retorna el numero de resultados por pagina de búsqueda
USO	De uso interno
RECIBE	n/a
DEVUELVE	el número de resultados deseados a recibir por respuesta del servicio

FUNCIÓN	set_max_resultados_por_pagina
DESCRIPCIÓN	Asigna el máximo número de resultados por pagina de búsqueda
USO	De uso interno
RECIBE	el número máximo de resultados deseados a recibir por respuesta del servicio
DEVUELVE	n/a

FUNCIÓN	get_max_resultados_por_pagina
DESCRIPCIÓN	Retorna el máximo número de resultados por pagina de búsqueda
USO	De uso interno
RECIBE	n/a
DEVUELVE	el número máximo de resultados deseados a recibir por respuesta del servicio

FUNCIÓN	get_numero_pagina

DESCRIPCIÓN	Función necesaria para servicios que piden numero de pagina en lugar de resultado inicial por pagina
USO	De uso interno
RECIBE	n/a
DEVUELVE	un número equivalente al índice de pagina

FUNCIÓN	**set_resultado_inicial**
DESCRIPCIÓN	Asigna el índice de resultado inicial para la búsqueda
USO	De uso interno
RECIBE	índice de resultado inicial deseado en la respuesta del servicio
DEVUELVE	n/a

FUNCIÓN	**get_resultado_inicial**
DESCRIPCIÓN	Retorna el índice de resultado inicial para la búsqueda
USO	De uso interno
RECIBE	n/a
DEVUELVE	índice de resultado inicial deseado en la respuesta del servicio

FUNCIÓN	**set_thumbnail_w**
DESCRIPCIÓN	Asigna el ancho para la vista previa
USO	De uso interno
RECIBE	la medida del ancho en pixeles
DEVUELVE	n/a

FUNCIÓN	**set_thumbnail_h**
DESCRIPCIÓN	Asigna el alto para la vista previa
USO	De uso interno

RECIBE	la medida del alto en pixeles
DEVUELVE	n/a

FUNCIÓN	**get_thumbnail_w**
DESCRIPCIÓN	Retorna el ancho asignado actualmente para la vista previa
USO	De uso interno
RECIBE	n/a
DEVUELVE	la medida del ancho en pixeles

FUNCIÓN	**get_thumbnail_h**
DESCRIPCIÓN	Retorna el alto asignado actualmente para la vista previa
USO	De uso interno
RECIBE	n/a
DEVUELVE	la medida del alto en pixeles

FUNCIÓN	**set_thumbnail_res**
DESCRIPCIÓN	Asigna el ancho y alto para la vista previa
USO	De uso interno
RECIBE	la medida del ancho y alto en pixeles
DEVUELVE	n/a

FUNCIÓN	**set_embed_w**
DESCRIPCIÓN	Asigna el ancho para la vista previa del embed
USO	De uso interno
RECIBE	la medida del ancho en pixeles
DEVUELVE	n/a

FUNCIÓN	**set_embed_h**

DESCRIPCIÓN	Asigna el alto para la vista previa del embed
USO	De uso interno
RECIBE	la medida del alto en pixeles
DEVUELVE	n/a

FUNCIÓN	**get_embed_w**
DESCRIPCIÓN	Retorna el ancho asignado actualmente para la vista previa del embed
USO	De uso interno
RECIBE	n/a
DEVUELVE	la medida del ancho en pixeles

FUNCIÓN	**get_embed_h**
DESCRIPCIÓN	Retorna el alto asignado actualmente para la vista previa del embed
USO	De uso interno
RECIBE	n/a
DEVUELVE	la medida del alto en pixeles

FUNCIÓN	**set_embed_res**
DESCRIPCIÓN	Asigna el ancho y alto para la vista previa del embed
USO	De uso interno
RECIBE	la medida del ancho y alto en pixeles
DEVUELVE	n/a

FUNCIÓN	**buscar**
DESCRIPCIÓN	Conexion a un servidor remoto y obtener respuesta.
USO	Este método es el núcleo de todas las búsquedas y conexiones
RECIBE	la ubicación url del sitio a conectarse

DEVUELVE	un string con la respuesta (en caso de exito)
	false (en caso de falla)

FUNCIÓN	**obtener_web_thumbnail**
DESCRIPCIÓN	Obtener thumbnail del servicio de vista de websites
USO	Cada vez que se desee obtener una imagen de vista previa de un sitio web
RECIBE	el URL del website que se desea obtener su vista previa
DEVUELVE	el URL de la imagen del website

FUNCIÓN	**obtener_filename_ext**
DESCRIPCIÓN	Obtener el nombre del archivo basado en el ultimo slash
USO	De uso interno
RECIBE	un string de un URL valido
DEVUELVE	el nombre del archivo con su extensión

FUNCIÓN	**obtener_filename**
DESCRIPCIÓN	Obtener el nombre del archivo basado en el ultimo slash
USO	De uso interno
RECIBE	un string de un URL valido
DEVUELVE	el nombre del archivo sin su extensión

FUNCIÓN	**obtener_extension**
DESCRIPCIÓN	Obtener la extensión del archivo basado en el último punto
USO	De uso interno
RECIBE	un string de un URL valido
DEVUELVE	la extensión del archivo

FUNCIÓN	obtener_URL_sin_ultimo_slash
DESCRIPCIÓN	un URL sacándole el ultimo slash "/"
USO	De uso interno
RECIBE	un string de un URL valido
DEVUELVE	el mismo URL sin el ultimo slash

FUNCIÓN	obtener_dominio
DESCRIPCIÓN	Obtener un dominio a partir de un URL valido
USO	De uso interno
RECIBE	un string de un URL valido
DEVUELVE	el dominio del URL

FUNCIÓN	obtener_embed_image
DESCRIPCIÓN	Obtener el HTML de la imagen listo para ser embebido
USO	De uso interno
RECIBE	un string de un URL valido
	ancho
	alto
DEVUELVE	un string de código HTML formateado y listo para ser incrustado

FUNCIÓN	obtener_javimoya_dl
DESCRIPCIÓN	Obtener el URL de video usando el servicio javimoya
USO	De uso interno
RECIBE	un string de un URL valido
DEVUELVE	un URL con la ubicación de la descarga del video

FUNCIÓN	obtener_url_param

DESCRIPCIÓN	Obtener el contenido de un parámetro GET de un URL
USO	De uso interno
RECIBE	un string de un URL valido
	el nombre de un parametro que exista dentro de un URL
DEVUELVE	el contenido del parámetro GET

FUNCIÓN	**obtener_embed_mp3_player**
DESCRIPCIÓN	Obtener el objeto flash para reproducir mp3
USO	De uso interno
RECIBE	un string de un URL valido
	ancho
	alto
DEVUELVE	un string de código HTML formateado y listo para ser incrustado

FUNCIÓN	**generar_xspf**
DESCRIPCIÓN	Este método genera una estructura XML con formato XSPF
USO	Para ser llamada desde cualquier ubicación cuando se desee generar un XSPF
RECIBE	un arreglo con estructura FindJira, obligadamente debe tener el campo "download_url"
DEVUELVE	un string con el XML formato XSPF

FUNCIÓN	**obtener_archivos_directorio**
DESCRIPCIÓN	Este método lee un directorio, realiza el scraping de los archivos del mismo
USO	Para ser llamada desde cualquier ubicación cuando se desee obtener archivos de un directorio online
RECIBE	un URL de directorio (Index of)
DEVUELVE	un arreglo con los archivos contenidos

FUNCIÓN	obtener_favicon_url
DESCRIPCIÓN	Este método obtiene el favicon de un dominio
USO	Para ser llamada desde cualquier ubicación cuando se desee obtener el favicon de un dominio
RECIBE	un URL valido
DEVUELVE	el URL del archivo favicon

FUNCIÓN	obtener_feeds_autodiscovery
DESCRIPCIÓN	Este método obtiene arreglo de feeds de cualquier URL provisto (feed autodiscovery)
USO	Para ser llamada desde cualquier ubicación cuando se desee obtener los feeds asociados a un URL
RECIBE	un URL valido
DEVUELVE	un arreglo con los feeds asociados a un URL

ATRIBUTOS DE LA CLASE	
$query	Es una cadena de caracteres con las palabras claves de búsqueda introducidas por handler.
$peticion_url	Es la cadena que contiene la ubicación URL del servicio junto con los demás parámetros necesarios para la obtención de resultados adecuados.
$respuesta_raw	Texto de respuesta en formato nativo que retorna el servicio web.
$respuesta_findjira	Arreglo asociativo que contiene la transformación de la respuesta nativa al formato FindJira.
$respuesta_json	Texto de la respuesta en formato JSON, si el servicio lo provee así.
$respuesta_php	Texto de la respuesta en formato PHP asociativo.
$resultados_por_pagina	Número de resultados que se mostrarán en una página.
$max_resultados_por_pagina	Número máximo de resultados que se mostrarán en una página que es permitido por servicio.
$resultado_inicial	Es el índice del resultado con el que empezará a

	listarse los elementos resultados de la búsqueda.
$thumbnail_width	Ancho en píxeles de la resolución de la imagen de la vista previa.
$thumbnail_height	Alto en píxeles de la resolución de la imagen de la vista previa.
$embed_width	Ancho en píxeles de la resolución del objeto incrustable HTML.
$embed_height	Alto en píxeles de la resolución del objeto incrustable HTML.

SERVICIO DE FEEDS Y PODCASTS

CLASE	**service_feed**
DETALLES	Provee los servicios necesarios para leer un número de posts de feeds ATOM y RSS, y los convierte a formato de repositorio FindJira. usa clase externa para manejo de feeds y reconocimiento de formatos

FUNCIÓN	**set_feed_url**
DETALLES	setea url del feed
RECIBE	el URL completo del feed
RETORNA	n/a

FUNCIÓN	**get_feed_url**
DETALLES	obtiene el URL del feed
RECIBE	n/a
RETORNA	el URL completo del feed

FUNCIÓN	**get_feed_info**
DETALLES	obtiene toda la información del feed
RECIBE	n/a
RETORNA	un arreglo serializado con las variables del objeto feed

FUNCIÓN	**feed_retrieve**

DETALLES	se conecta a un feed y obtiene la información necesaria
RECIBE	n/a
RETORNA	n/a
INTERNO	modifica respuesta raw para tener el sitio retornado,
	modifica respuesta FindJira para convertir dicho sitio en un arreglo serializado

FUNCIÓN	**feed_parse**
DETALLES	parsea la información de un feed XML a JSON y de JSON a PHP serializado
RECIBE	n/a
RETORNA	arreglo en formato FindJira
INTERNO	asigna valores de variables especiales de la clase para ser extraídas después
	crea la resupuesta_php con el arreglo del feed
	parsea toda la información del feed al arreglo en formato FindJira

ESTRUCTURA DE UN SERVICIO WEB

En esta sección cubriremos el ejemplo del servicio web Searchmash que es el más extenso, pero es una gran base para quienes deseen ingresar otro servicio más al framework. Las funciones get y set iniciales no siempre son necesarias, para servicios simples como youtube o itunes no fue indispensable ya que el query no acepta estos parámetros de búsqueda, cosa que sí lo hacen Yahoo y Live. Luego de *elaborar_url* se implementan los metos de búsquedas por tipos de medio y a continuación los parsers respectivos para cada uno, ya que cada respuesta suele ser diferente en el mismo servicio. Al final se implementan métodos utilitarios que son necesarios para obtener elementos u objetos externos, principalmente son útiles en los métodos de parse.

CLASE	service_searchmash
DETALLES	provee los servicios necesarios para leer búsquedas de imágenes, videos, blogs y web de searchmash (resultados de Google, en formato JSON), y los convierte a formato de repositorio FindJira

FUNCIÓN	set_site
DETALLES	setea string para restringir búsqueda a un sitio
RECIBE	el string del sitio en cuestión
RETORNA	n/a

FUNCIÓN	get_site
DETALLES	obtiene string seteado previamente para restringir búsqueda a un sitio
RECIBE	n/a
RETORNA	el string del sitio o parte del mismo

FUNCIÓN	set_intitle
DETALLES	setea string para restringir búsqueda para buscar dentro del título de la pagina
RECIBE	el string en cuestión
RETORNA	n/a

FUNCIÓN	get_intitle
DETALLES	obtiene string para restringir búsqueda para buscar dentro del título de la pagina
RECIBE	n/a
RETORNA	el string en cuestión

FUNCIÓN	get_inurl
DETALLES	obtiene string para restringir búsqueda para buscar paginas con dicho string en su URL
RECIBE	n/a
RETORNA	el string en cuestión

FUNCIÓN	set_inurl
DETALLES	setea string para restringir búsqueda para buscar paginas con dicho string en su URL
RECIBE	el string en cuestión
RETORNA	n/a

FUNCIÓN	get_filetype
DETALLES	obtiene string para restringir busqueda a ciertos tipos de archivo
RECIBE	n/a
RETORNA	el string en cuestión

FUNCIÓN	set_filetype
DETALLES	setea string para restringir búsqueda a ciertos tipos de archivo
RECIBE	el string en cuestión
RETORNA	n/a

FUNCIÓN	get_excluidos
DETALLES	obtiene string para eliminar cadenas de la búsqueda
RECIBE	n/a
RETORNA	el string en cuestión

FUNCIÓN	set_excluidos
DETALLES	setea string para eliminar cadenas de la búsqueda
RECIBE	el string en cuestión
RETORNA	n/a

FUNCIÓN	elaborar_url
DETALLES	crea el URL de busqueda listo para crear un html get, con los parámetros necesarios
RECIBE	el URL del servicio para el cual se realizara la búsqueda
RETORNA	el URL completo listo para hacer el get y obtener

	la pagina de respuesta

FUNCIÓN	web_search
DETALLES	setea el URL para la busqueda de sitios, elabora el URL, lo convierte a formato FindJira y lo almacena en las variables de clase
RECIBE	n/a
RETORNA	n/a

FUNCIÓN	image_search
DETALLES	setea el URL para la búsqueda de imágenes, elabora el URL, lo convierte a formato FindJira y lo almacena en las variables de clase
RECIBE	n/a
RETORNA	n/a

FUNCIÓN	video_search
DETALLES	setea el URL para la búsqueda de videos, elabora el URL, lo convierte a formato FindJira y lo almacena en las variables de clase
RECIBE	n/a
RETORNA	n/a

FUNCIÓN	blog_search
DETALLES	setea el URL para la búsqueda de blogs, elabora el URL, lo convierte a formato FindJira y lo almacena en las variables de clase
RECIBE	n/a
RETORNA	n/a
NOTAS	el formato de blog_search es el mismo que el de web, por ende, se llama a web_parse

FUNCIÓN	blog_search
DETALLES	setea el URL para la búsqueda de blogs, elabora el URL, lo convierte a formato FindJira y lo almacena en las variables de clase
RECIBE	n/a

RETORNA	n/a
NOTAS	el formato de wiki_search es el mismo que el de web, por ende, se llama a web_parse

FUNCIÓN	**indexof_search**
DETALLES	setea el URL para la búsqueda, setea los parámetros necesarios para restringir la búsqueda a directorios web, elabora el URL, lo convierte a formato FindJira y lo almacena en las variables de clase
RECIBE	n/a
RETORNA	n/a
NOTAS	el formato de wiki_search es el mismo que el de web, por ende, se llama a web_parse

FUNCIÓN	**web_parse**
DETALLES	realiza la conversión del formato retornado por el servicio a formato FindJira
RECIBE	n/a
RETORNA	arreglo php serializado listo para ser añadido al repositorio

FUNCIÓN	**image_parse**
DETALLES	realiza la conversión del formato retornado por el servicio a formato FindJira
RECIBE	n/a
RETORNA	arreglo php serializado listo para ser añadido al repositorio

FUNCIÓN	**video_parse**
DETALLES	realiza la conversión del formato retornado por el servicio a formato FindJira
RECIBE	n/a
RETORNA	arreglo php serializado listo para ser añadido al repositorio

FUNCIÓN	**obtener_formato_dl**

DETALLES	obtiene el URL necesario para poder descargar videos desde Google video
RECIBE	id del video a concatenar con su URL
RETORNA	cadena completa, con el URL y el id del video

FUNCIÓN	**file_parse**
DETALLES	permitira parsear archivos retornados para su descarga, no soportada por searchmash
RECIBE	n/a
RETORNA	n/a

FUNCIÓN	**obtener_id_google_video**
DETALLES	permite obtener el id de un video del URL de google video
RECIBE	el URL completo
RETORNA	el id del video

FUNCIÓN	**obtener_embed_google_video**
DETALLES	ensambla la estructura html necesaria para poder embeber videos de google video, listos para ser impresos en código html
RECIBE	el id del video, y las dimensiones del reproductor
RETORNA	el código web en una cadena

FUNCIÓN	**obtener_embed_youtube_url**
DETALLES	ensambla la estructura html necesaria para poder embeber videos de youtube , listos para ser impresos en codigo html
RECIBE	el URL del video, y las dimensiones del reproductor
RETORNA	el código web en una cadena

COMO INGRESAR UN NUEVO SERVICIO AL FRAMEWORK

En la sección anterior se detalló cómo debe ser la estructura de un servicio web. Para ingresar un nuevo servicio web al framework, es necesario crear una clase con la estructura antes descrita, con los métodos de parsing adecuados a la respuesta que retornará el servicio, si la respuesta es XML y no JSON realizar la conversión necesaria previamente. Es posible analizar los servicios ya implementados para el comportamiento ante situaciones similares.

Los cambios directos al framework se los realiza en la clase *handler*, donde:

- Es necesario incluir la nueva clase.
- Crear su objeto en el constructor.
- Finalmente ubicar su comportamiento en el tipo de medio que aplica, siguiendo los esquemas de los servicios ya ingresados.

Es posible crear un nuevo handler basado en el ya existente, o modificar el actual, si la lógica de la aplicación del desarrollador necesita un enfoque distinto al actual.

BIBLIOGRAFÍA

[1] Michael Mahemoff (Junio 2006). *Ajax Design Patterns*. Primera Edición. 655 páginas.

[2] David Sklar, Adam Trachtenberg (Agosto 2006). *PHP Cookbook*. Segunda Edición. 810 páginas.

[3] Christopher Schmitt, Dan Cederholm (Agosto 2004). *CSS Cookbook.* Primera Edición. 252 páginas.

[4] Christian Gross (Diciembre 2006). *Ajax and REST Recipes: A Problem-Solution Approach*. Primera Edición. 360 páginas.

[5] Christian Heilmann (Julio 2006). *Beginning JavaScript with DOM Scripting and Ajax: From Novice to Professional.* Primera Edición. 512 páginas.

[6] Steve Ph.D. Holzner (Marzo 2006). *AJAX for dummies*. Primera Edición. 384 páginas.

[7] Richard Mansfield (Marzo 2005). *CSS Web Design for dummies*. 380 páginas.

[8] Nicholas Chase (Agosto 2006). The Ultimate Mashup - Web services and the semantic Web

[9] Unicode Consortium. 1996. The Unicode Standard. Version 2.0. Reading: Addison-Wesley.

[10] Programmable Web. *API List by category* [en línea]. <http://www.programmableweb.com/apilist/bycat> [Consulta: 14 octubre 2006]

[11] Web Mashup [en linea] <http://www.webmashup.com/> [Consulta: 15 octubre 2006]

[12] Google GData API. [en línea]. <http://code.google.com/apis/gdata/> [Consulta: 18 octubre 2006]

[13] Yahoo! Developer Network For Yahoo! Search Web Services [en línea] <http://developer.yahoo.com/search/> [Consulta: 13 diciembre 2006]

[14] Google SOAP Search Documentation [en línea] <http://code.google.com/apis/soapsearch/reference.html> [Consulta: 13 diciembre 2006]

[15] Flickr Services API Documentantion [en línea] <http://www.flickr.com/services/api/ > [Consulta: 17 diciembre 2006]

[16] YouTube API Documentantion [en línea] < http://www.youtube.com/dev_docs> [Consulta: 18 diciembre 2006]

[17] Wikipedia: Special Export [en linea] <http://www.mediawiki.org/wiki/Parameters_to_Special:Export> [Consulta: 19 diciembre 2006]

[18] PHP Online Manual [en linea] <http://www.php.net/docs.php> [Consulta: 28 diciembre 2006]

[19] Ajax Patterns Frameworks List [en línea] <http://ajaxpatterns.org/Frameworks> [Consulta: 26 diciembre 2006]

[20] MSDN *Live Search API* [en linea] <http://msdn2.microsoft.com/en-us/library/bb251794.aspx> [Consulta: 12 enero 2007]

[21] W3C: Cascading Stylesheets Homepage [en linea] <http://www.w3.org/Style/CSS/> [Consulta: 8 enero 2007]

[22] Wikipedia: Codec Article [en linea] <http://en.wikipedia.org/wiki/Codec> [Consulta: 10 enero 2007]

[23] Wkipedia: Syndication [en linea] <http://en.wikipedia.org/wiki/Web_syndication> [Consulta: 10 enero 2007]

[24] The Atom Project: RSS and Atom format Comparison [en linea] <http://www.intertwingly.net/wiki/pie/Rss20AndAtom10Compared> [Consulta: 10 enero 2007]

[25] Harvard Law: RSS Specification [en linea] <http://cyber.law.harvard.edu/rss/rss.html> > [Consulta: 10 enero 2007]

[26] Macromedia Flash SWF Format Specification <http://www.adobe.com/licensing/developer/> [Consulta: 10 enero 2007]

[27] Snook.ca - *iTunes search API Around the Corner* [en linea] <http://snook.ca/archives/javascript/itunes_search_a/> [Consulta: 15 enero 2007]

[28] Joshua Bloch, *How to Design a Good API and Why it Matters* [en linea] <http://lcsd05.cs.tamu.edu/slides/keynote.pdf> [Consulta: 15 enero 2007]

[29] Wikipedia: Application Programming Interface Article [en linea] <http://en.wikipedia.org/wiki/Application_programming_interface> [Consulta: 18 enero 2007]