

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Comparación de herramientas para el desarrollo de librerías enfocadas a aplicaciones web¹

Comparison of Tools for the Development of Libraries Intended for Web Applications

Comparaison d'outils pour le développement de librairies dirigés vers applications web

Fabián Hernando López Higuera

Ingeniero de Sistemas y Computación
Universidad Pedagógica y Tecnológica de Colombia.
fabian376@gmail.com

Javier Antonio Ballesteros Ricaurte

Estudiante de Doctorado en Ciencias y Tecnología Informática
Universidad Carlos III de Madrid (España)
Magíster en Ciencias Computacionales
Universidad Autónoma de Bucaramanga y El Instituto
Tecnológico y de Estudios Superiores de Monterrey México
Ingeniero de Sistemas
Universidad de Boyacá.
Escuela de Ingeniería de Sistemas y Computación
Universidad Pedagógica y Tecnológica de Colombia – UPTC.
javier.ballesteros@uptc.edu.co

Tipo de artículo: Investigación científica y tecnológica

Recepción: 2011-05-02

Revisión: 2011-08-18

Aprobación: 2011-08-26

¹ Resultado del proyecto de grado en la modalidad de pasantía: Creación de API a implementar en el sistema de acreditación universitaria bajo Cloud Computing. Escuela de Ingeniería de Sistemas y Computación – Universidad Pedagógica y Tecnológica de Colombia. Inicio junio de 2010, terminación febrero de 2011.

Contenido

1. Introducción
2. Método
 - 2.1 Investigación API de Documentos
 - 2.2 Investigación API de Gráficas
 - 2.3 Investigación API de Validaciones
 - 2.4 Investigación API de Internacionalización
 - 2.5 Investigación API de Multimedia
3. Trabajos Futuros
4. Conclusiones
5. Lista de Referencias

Resumen

Este artículo de investigación presenta el proceso de desarrollo de cinco Applications Programming Interfaces (API): generación de documentos, generación de gráficas, validaciones, internacionalización y multimedia, para aplicaciones web bajo arquitectura Cloud Computing. Para el proceso de desarrollo de cada API se realizaron consultas de herramientas, tecnologías y librerías, comparando sus ventajas y desventajas, teniendo en cuenta que el desarrollo cumpliera con las características establecidas para cada una de éstas, con el fin de contribuir a la funcionalidad en la generación de reportes, gráficas y validaciones de campos en la captura de información, para proyectos Cloud Computing. Dichas aplicaciones optimizan el tiempo para el desarrollo de este tipo de procesos.

Palabras clave

Interfaz de programación de aplicaciones (API), Computación en la nube, Framework.

Abstract

This research article presents the development process of five Application Programming Interfaces or APIs: document generation, graphics generations, validations, internationalization and multimedia, to web applications under Cloud Computing architecture. For the development process of each API we consulted on tools, technologies and libraries, comparing their pros and cons, considering the fact that this development would fulfill the established characteristics for each of them, with the intention of contributing to functionality regarding to report generation, graphics and field validations when capturing information, for Cloud Computing projects. Such applications optimize the time needed for developing this kind of processes.

Keywords

Application Programming Interface, Cloud Computing, Framework

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Résumé

Cet article de recherche présente le processus de développement de cinq Application Programming Interfaces (APIs) : création de documents, création de graphiques, validations, internationalisation et multimédia, pour des applications web sur l'architecture *Cloud Computing* (Informatique en Nuage). Pour le processus de développement de chaque API on a réalisé consultations d'outils, technologies et librairies, en comparant ses avantages et désavantages, en considérant que le développement s'adapte aux caractéristiques établies pour chacun de eux, avec l'intention de contribuer à la fonctionnalité dans la création de reports, graphiques et validations de champs dans la capture de l'information, pour projets *Cloud Computing*. Tels applications optimisent le temps nécessaire pour le développement de ce type de processus.

Mots-clés

Interface de programmation d'applications (API), Informatique en Nuage, *Framework*.

1. Introducción

La empresa Ingenius-Group S.A.S² se encuentra en proceso de desarrollo de proyectos hacia distintos sectores, utilizando el modelo de distribución de software denominado SaaS (Software as a Service), bajo la arquitectura Cloud Computing. Ingenius-Group S.A.S al trabajar bajo arquitectura Cloud Computing necesita crear APIs que sean reutilizables en diversos proyectos de software y que presten soporte a: generación y publicación de documentos en formatos ofimática estándar, generación y publicación de gráficas de datos como: diagrama lineal, diagrama de barras y diagramas de torta; validación de campos de texto comunes como: nombres, direcciones, correos electrónicos, contraseñas; uso de múltiples idiomas en el sistema, y finalmente la importación de recursos multimedia como imágenes o vídeos.

La empresa tiene la necesidad de crear herramientas de software reutilizables que minimicen tiempos y costos en el desarrollo de sus proyectos de software, ya que al implantarlos bajo arquitecturas como Cloud Computing y modelo SaaS implica: calidad, soporte y actualización sin afectación de recursos, seguridad, personalización e integración a un bajo costo. Características que llaman la atención de las empresas y por tanto aumentan la demanda de software que al prescindir de herramientas software reutilizables para sus proyectos malgastarán

² Empresa desarrolladora de software de la ciudad de Tunja (Colombia).

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

tiempo y no obtendrán los resultados esperados.

Para obtener los mejores resultados en la creación de APIs para proyectos bajo arquitectura Cloud Computing con soporte a generación de gráficas de datos, generación de reportes, validaciones, internacionalización y transporte de archivos multimedia en aplicaciones web, fue necesario llevar a cabo un procedimiento que inicia con una etapa de investigación, experimentación, comparación y selección de posibles herramientas o métodos que sirvan de base al desarrollo de las APIs y que permitan ser utilizados bajo tecnologías definidas por la empresa.

Seguidamente se realiza la etapa de diseño, para la cual cada API debe cumplir con características como: tolerancia a modificaciones, facilidad de uso y cumplimiento de las necesidades establecidas. La etapa de desarrollo se lleva a cabo acorde al diseño de las API, cada funcionalidad desarrollada debe ser probada de forma continua para corroborar que estén cien por ciento libre de errores. El resultado de las etapas anteriores es la creación de los respectivos JAR (Java Archive) o empaquetamiento de las clases de cada API, para ser implementados en un proyecto desarrollado por la empresa, llamado Sistema de Acreditación Universitaria.

En el presente artículo se hace especial énfasis en la etapa de investigación, incluyendo los métodos y herramientas utilizadas para el desarrollo de cada API. En la sección dos se explica el proceso desarrollado en la construcción de cada API, el análisis realizado y el resultado obtenido; en la sección tres se habla de trabajos futuros, y finalmente se establecen algunas conclusiones.

2. Método

La etapa de investigación para el desarrollo de las API, fue de importancia para conocer una amplia variedad de tecnologías y métodos que en la actualidad prestan soporte a las funciones requeridas por cada API. Para su selección se tuvo en cuenta que los proyectos desarrollados por la empresa Ingenius-Group S.A.S, estuvieran inmersos en tecnologías que soportan la arquitectura Cloud Computing, como las siguientes:

- Framework de Desarrollo GWT (Google Web Toolkit, 2010).
- Framework de Persistencia Data Nucleus (DataNucleus, 2010).
- Servidor de Hosting APPEngine (Google App Engine, 2010).

Es importante tener claro tres conceptos que son fundamentales para el

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

desarrollo del proyecto: API, Cloud Computing y Framework. En el transcurso del artículo se encontrarán estos términos y su relación con las diferentes herramientas. Veamos:

- a. API: las interfaces de programación de aplicaciones son diseñadas para prestar múltiples funciones y herramientas que ayuden y agilicen los procesos de desarrollo de cualquier tipo de aplicaciones de software, en la actualidad existen APIs que proveen una serie de componentes para construir páginas web, además herramientas para el desarrollo de sitios web completos, para inserción de mapas geoespaciales, para inserción de calendarios, para desarrollo web en Ajax, entre otros.

Los API han revolucionado la manera de desarrollar software, pues anteriormente elaborar cualquier tipo de componente que prestara una utilidad específica como mapas, tablas, gráficos, significaba dolores de cabeza para cualquier programador, pero ahora con la aparición de las APIs es cuestión de encontrar las apropiadas a las necesidades y con su documentación seguir unos breves pasos para incluirlo a los proyectos de software.

- b. CLOUD COMPUTING: la arquitectura Cloud Computing nace a partir de la demanda de software a nivel micro empresarial y educativo, debido al insuficiente capital con que cuentan este tipo de empresas.

El concepto de arquitectura Cloud Computing proviene del concepto de cloud (nube), como un conjunto de elementos virtualizados a través de internet que sigue un modelo de pago por uso, que puede ser reconfigurado dinámicamente para satisfacer las necesidades del cliente, y constituye un modelo de servicio de Tecnologías de la Información basado en la virtualización y la computación distribuida (LOMBARDI, 2010).

- c. FRAMEWORK: un framework permite el desarrollo de software utilizando un conjunto de herramientas integradas en un sistema, es diseñado para brindar modularidad al software, al seguir un procedimiento que abarca desde el modelamiento de la base de datos hasta el diseño de la interfaz de usuario y de esta forma obtener un sistema de software completo y con las características deseadas. Un artículo publicado en el año 2006 por Gun Ho Lee y Junsu Jung titulado "*web framework with Java and XML in multi-tiers for productivity*" que habla sobre frameworks web aclara: "Los frameworks son una tecnología prometedora que provee el diseño

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [http://revistavirtual.ucn.edu.co/], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

e implementación de software centrado en la reducción de costos y tiempo en el desarrollo de sistemas distribuidos, y mejoran la calidad de los sistemas" (LIU & DARLINGTON, 2006).

La tecnología de mayor interés para el desarrollo de los APIs fue el framework de desarrollo GWT, ya que éstas fueron incluidas directamente en las capas de desarrollo del Sistema de Acreditación Universitaria.

A continuación se presenta la investigación realizada para el desarrollo de cada API.

2.1 Investigación API de Documentos

La presente es una investigación de tipo cualitativo sobre librerías que actualmente prestan soporte a generación de reportes en formatos como *docx*, *odt* y *pdf*, pues entre los requisitos principales establecidos para el desarrollo del API de Documentos está la generación de reportes en estos formatos.

Ya que el framework GWT está basado en lenguaje *JAVA*, las librerías investigadas debían prestar soporte a este framework de desarrollo web. La tabla 1 presenta las características principales que fueron evaluadas para cada una de las librerías, teniendo en cuenta su versión actual, el licenciamiento, la documentación para conocer su forma de uso, el soporte a contenidos como tablas o imágenes y el manejo de estilos para la presentación final de cada reporte.

Tabla 1. Comparación de APIs para soporte a generación de Documentos.

Librerías		Características				
		Versión	Licencia	Documentación	Soporte completo de contenido	Manejo de estilos
<i>docx</i>	<i>DOCX4J</i>	2.3.0	Apache 2	BAJA	NO	BAJA
	<i>APACHE POI</i>	3.6	Apache 2	BAJA	NO	BAJA
	<i>JAVADocX</i>	Free	LGPL ³	ALTA	SI	ALTA
<i>odt</i>	<i>JODReports</i>	2.3.0	LGPL	BAJA	SI	BAJA
	<i>ODFDOM</i>	0.8.6	Apache 2	ALTA	SI	ALTA
<i>pdf</i>	<i>iText</i>	5.0.4	AGPL ⁴	ALTA	SI	ALTA

³ Lesser General Public License.

⁴ Affero General Public License.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Para la generación del formato *docx* fueron valoradas las librerías *DOCX4J*, *APACHE POI* y *JAVADocX*. La librería *DOCX4J* (Software Edge Wall, 2010), permite la manipulación de documentos en formato *docx* y *pptx* en JAVA (Jason, 2010), su licencia Apache versión 2 permite el uso de código fuente para desarrollo tanto libre como propietario y requiere de la preservación del copyright (Foundation Apache Software, 2004), utiliza *JAXB* (Java Architecture for XML Binding, 2003) para lectura de archivos XML⁵ y su posterior traducción al correspondiente formato de documento, teniendo en cuenta que los formatos *DOCX* y *PPTX* son creados a partir de múltiples archivos *XML*.

La siguiente librería consultada *APACHE POI*, es utilizada en la manipulación de formatos de archivos basados en el estándar Office Open XML (OOXML)⁶ y Microsoft OLE2⁷, bajo la misma licencia de la librería anterior, esta librería se especializa en la extracción de datos a partir de documentos existentes, característica que para el API de Documentos significa una desventaja ya que contiene poca funcionalidad en la inserción de contenidos como tablas, imágenes, paginación, entre otros, en documentos nuevos utilizando los estándares nombrados.

Existe el formato *docx*, la librería *JAVADocX* que se distribuye en forma libre como *JAVADOCX Free*, pero también se puede pagar por una licencia profesional que contiene complementos o funciones adicionales; teniendo en cuenta la comparación realizada en (JAVADOCX, 2009), se presenta una tabla comparativa de ambas versiones. En la versión Free de *JAVADOCX* la licencia LGPL es menos restrictiva que la licencia GPL (General Public License), pues incorpora todos sus términos y condiciones incluyendo algunos permisos adicionales, el más importante entre ellos es que cualquier biblioteca bajo esta licencia, podrá enlazarse a cualquier programa no GPL libre o privativo (Foundation Free Software, 2007).

En cuanto al formato *odt*, fue investigada la librería *JODReports* (JODReports, 2010), presentada en la tabla 1, la cual se especializa en la creación de documentos y reportes en formato abierto para visualización en *Open Office Writer*. Para la creación de contenidos utiliza XML (*FreeMarker*, 2010), el cual es un generador de plantillas XML utilizado en lenguaje Java, que contiene una sintaxis propia que debe ser estudiada para el uso correcto de la librería *JODReports*.

⁵ XML: Extensible Markup Language.

⁶ OOXML: Estándar de definición de los formatos actuales de la Suite Microsoft Office como: *docx*, *pptx* y *xlsx*.

⁷ OLE2: Estándar de definición de los formatos de la Suite Microsoft Office anteriores al 2007 como: *doc*, *ppt* y *xls*.

“Revista Virtual Universidad Católica del Norte”. No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

La siguiente librería consultada, llamada *ODFDOM* para generación de reportes en formato *odt*, creada sobre el proyecto ODF Toolkit, es una librería que provee la totalidad de funciones necesarias para la escritura, lectura y manipulación de documentos en formato abierto ODF⁸ (Schubert, 2010). Fue escrita bajo lenguaje *JAVA*, se rige bajo un enfoque por capas de acceso para escritura de documentos, como son capa ODF XML y ODF Package. La primera capa *ODF XML* contiene dos APIs funcionales de bajo y alto nivel, el API de bajo nivel llamado ODF DOM API permite el acceso directo a archivos XML utilizando el lenguaje DOM⁹ creado por la comunidad internacional W3C¹⁰.

DOM es utilizado para actualización dinámica de contenido, estilo y estructura de documentos XML o HTML entre otros, mediante un lenguaje de programación como *JAVA* (W3C, 2005), bajo el API ODF DOM cada clase representa un nodo del XML en los métodos provistos por la librería *ODFDOM*, su objetivo es proporcionar al desarrollador una interfaz sencilla para crear elementos del formato ODF mediante su propio XML.

En esta misma capa el API de alto nivel llamado *ODF Document API* es una abstracción del API DOM, pues en ella se oculta cualquier elemento XML y se preocupa por la facilidad de uso al usuario, cada clase de este API representa múltiples nodos XML con sus respectivos atributos, y con sencillas llamadas a métodos se pueden crear componentes completos del formato ODF.

La segunda capa ODF Package Layer que se encarga de acceder a los paquetes internos de ODF como sus metadatos, archivos de manifiesto, firmas digitales de archivo, entre otros, que no es utilizado en el presente proyecto, pero es conveniente nombrar para dar a conocer las capacidades de la librería *ODFDOM*.

Para la generación del formato *pdf* se encontró solamente una librería que soporta lenguaje *JAVA*, llamada *iText* (BVBA, 2010), esta librería tiene buena documentación y ofrece las características necesarias para la generación de reportes en este formato digital.

2.2 Investigación API de Gráficas

En esta investigación fueron comparadas dos librerías que prestan soporte a la generación de gráficas de datos en lenguaje *JAVA*. *GChart* es

⁸ ODF: Open Document Format.

⁹ DOM: Document Object Model.

¹⁰ W3C: World Wide Web Consortium.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

una librería desarrollada por Google en su framework GWT, que es un componente gráfico estándar de GWT, el cual utiliza paneles propios del framework para presentar los gráficos en el navegador del cliente. No requiere librerías adicionales para su funcionamiento.

En cuanto a calidad visual de las gráficas no rectangulares, ésta es deficiente. Para mejorar esta deficiencia puede utilizarse una biblioteca externa de GWT llamada GWT Graphics, especializada en vectorización con SVG (Scalable Vector Graphics) y VML (Vector Markup Language); SVG es un lenguaje gráfico basado en texto, que describe las imágenes con formas vectoriales usado con el fin de proveer alta calidad a las imágenes (Adobe, 2010); VML es un lenguaje basado en XML, que define la codificación de la información vectorial (W3C, 1998); gracias a estos lenguajes la biblioteca GWT Graphics mejora la calidad de visualización de imágenes.

La librería *GChart* provee los tipos de gráficas de datos más utilizados, como: tortas, barras y líneas (Gunther, 2009); contiene suficiente documentación, es distribuida bajo licencia Apache versión 2, así que puede ser utilizada bajo software comercial; una desventaja importante es que no permite exportar las gráficas de datos en formato de imagen, ya que los genera directamente en la pantalla de la aplicación web, característica esencial para los proyectos de la empresa Ingenius-Group S.A.S, siendo estas gráficas de utilidad para añadir a los reportes del sistema.

En cuando a la librería *JFreeChart* es un proyecto fundado en el año 2000, por David Gilbert, depende de la librería *JCommon*; esta librería fue creada bajo el mismo proyecto de *JFreeChart* y es utilizada por la misma para la visualización de información en las gráficas. Para el funcionamiento de ésta, no es necesario tener la última versión de la librería *JCommon*, acepta cualquier versión. Actualmente el proyecto *JFreeChart* (JFree, 2010) es actualizado mediante contribuciones de una comunidad abierta de desarrolladores, entre sus características principales están: buena documentación con soporte a variedad de estilos de gráficas, diseño flexible, soporte a diversos tipos archivos de salida como imágenes y Swing¹¹. Se encuentra bajo licenciamiento libre usando GNU LGPL (Lesser General Public License), que permite su uso bajo software comercial.

En conclusión para la investigación del API de Gráficas de Datos, la

¹¹ Swing: API de Java con soporte de componentes y servicios de interfaz gráfica de usuario en aplicaciones de escritorio o web.

“Revista Virtual Universidad Católica del Norte”. No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [http://revistavirtual.ucn.edu.co/], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

librería seleccionada como base para su desarrollo es JFreeChart, ya que soporta cualquier tipo de gráficas, documentación suficiente y las gráficas pueden ser generadas en múltiples formatos de imagen.

2.3 Investigación API de Validaciones

Para esta investigación se evaluaron tres distintas herramientas, teniendo en cuenta las ventajas que un sistema de validación debe tener. Estas herramientas son:

- GWT-Validation
- Sistema de validaciones de Symphony
- Expresiones Regulares

La librería GWT-Validation, es un proyecto de Google, que tiene soporte único bajo el framework GWT; el proyecto se encuentra aún en desarrollo y su versión es la 1.0, la cual provee características como: validaciones directas a los métodos en cliente y servidor de la aplicación, mediante anotaciones de Java y escritura de validaciones personalizadas.

El sistema de Validaciones del framework Symfony aunque es para aplicaciones web bajo lenguaje PHP, es un sistema robusto basado en el lenguaje YAML (Ain't Another Markup Language), el cual es un lenguaje parecido a XML, pero que contiene una estructura más sencilla para su lectura y escritura (Dulio, 2009), la figura 1 presenta un ejemplo de un archivo de este tipo; este lenguaje es utilizado por el framework Symfony para agregar las restricciones de validación a los campos (Potencier, 2010).

Figura 1. Estructura del lenguaje YAML.

```
default language: es
other languages: [en, it]

default controller:
  web: home
  admin: login

controller alias:
  es:
    acerca: about
    contacto: contact
    busqueda: search
```

Fuente: Dulio. Tutorial de PHP y YAML.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Los sistemas mencionados hacen parte de tecnologías distintas, una desarrollada bajo lenguaje Java y la otra en lenguaje PHP, son tecnologías novedosas, pero que implican el uso de un framework específico; ya que el desarrollo del API de Validaciones del presente proyecto será utilizado en proyectos bajo tecnologías distintas, no serían los más adecuados para su implementación.

Un mecanismo importante en la creación de sistemas de validación son las expresiones regulares, o regex (Goyvaerts, 2009), con las que se logra definir reglas de validación mediante patrones simbólicos. Su uso ayuda además de validar los caracteres propios de una cadena finita, validar longitudes máxima y mínima y tipos de caracteres (numéricos, alfabéticos, simbólicos). Otra propiedad importante de las expresiones regulares es que permiten validar si el contenido de la cadena ingresada es completo. Un ejemplo es el ingreso de un correo electrónico: está constituido de un nombre de usuario, seguido de un símbolo arroba y finalmente un dominio, cada una de las partes que conforman el correo electrónico puede ser validado dentro de la expresión para que sea completo y con la información correcta.

El uso de regex y el diseño de un API con el que se puedan cambiar estas expresiones dependiendo las necesidades, se logra obtener un sistema robusto, cumpliendo con las necesidades de cualquier aplicación web.

Para este proyecto es importante tener un sistema de validación que garantice los mejores resultados y que no permita inconvenientes en la captura de información; por esta razón, y la evaluación realizada, se toma regex para efectuar el proceso de validaciones en los campos de validación.

2.4 Investigación API de Internacionalización

Al igual que el API anterior, se estudiaron librerías que prestan soporte a internacionalización de aplicaciones web para conocer distintas formas de uso y diseñar un API que cumpla con las características deseadas. Entre ellos están:

- Symfony – I18N
- GWT – I18N

El framework *symfony* de PHP, así como soporta validaciones, también soporta *I18N* (*Uso de múltiples idiomas, abreviatura de Internacionalización, donde 18 es el número de letras entre la i y la n*).

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [http://revistavirtual.ucn.edu.co/], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

Para el proceso de traducción de idiomas en la aplicación, *symfony* hace uso de archivos *XML*, los cuales representan archivos de configuración compuestos de cadenas localizables, las cuales se definen como el par clave-valor, en el cual la clave es la identificación, y el valor es la cadena traducida. *Symfony* maneja el formato *XLIFF* basado en *XML* (ver figura 2), que proporciona funcionalidad a la traducción de cadenas localizables, pues contiene características como visualización del estado de traducción de cadenas individuales y proporciona datos como el número de palabras de las oraciones ingresadas, características que permiten que el proceso de internacionalización de aplicaciones sea más sencillo y robusto. Este formato fue creado por Oracle.

Figura 2. Archivos *XLIFF* para internacionalización de aplicaciones con *symfony*.

```

<!-- apps/frontend/i18n/fr/messages.xml -->
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xliiff PUBLIC "-//XLIFF/DTD XLIFF//EN"
"http://www.oasis-open.org/committees/xliiff/documents/xliiff.dtd">
<xliiff version="1.0">
  <file source-language="EN" target-language="fr" datatype="plaintext"
    original="messages" date="2008-12-14T12:11:22"
    product-name="messages">
    <header/>
    <body>
      <trans-unit id="1">
        <source>About Jobeet</source>
        <target>A propos de Jobeet</target>
      </trans-unit>
      <trans-unit id="2">
        <source>Feed</source>
        <target>Fil RSS</target>
      </trans-unit>
      <trans-unit id="3">
        <source>Jobeet API</source>
        <target>API Jobeet</target>
      </trans-unit>
      <trans-unit id="4">
        <source>Become an affiliate</source>
        <target>Devenir un affilié</target>
      </trans-unit>
    </body>
  </file>
</xliiff>

```

Fuente: Google. *Developer's Guide – Internationalization*.

El siguiente API desarrollado por Google en su framework *GWT*, utiliza otro tipo de archivos localizables para la traducción de las aplicaciones, éstos, llamados archivos de propiedades con extensión *".properties"* (MKyong, 2010), son archivos utilizados en lenguaje *Java*, para almacenar información de configuración de cualquier aplicación en este lenguaje. Gracias a la sencillez de su estructura permiten ser leídos y escritos fácilmente, en la figura 3 se presenta un ejemplo de este tipo de archivos, el campo de la izquierda representa la clave y el de la derecha

“Revista Virtual Universidad Católica del Norte”. No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [http://revistavirtual.ucn.edu.co/], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

el valor. En el API de GWT cada clave del archivo de propiedades representa una identificación de la cadena que será traducida, y cada valor, la traducción de esa cadena.

Figura 3. Ejemplo de Archivo de Propiedades de Java

```
UserName=databaseusername
Password=databaseuserpassword

UserName=webapp
Password=frootloops

UserName=bobjones
Password=bluegoat

UserName=megacorp
Password=spooooooooon
```

Fuente: *a Sample properties file* (Wheaton, 2010).

El nombre de cada archivo de propiedades debe acogerse a la norma ISO 639, actualmente en su sexta revisión (ISO, 2010). Gracias a esta norma se estandariza un lenguaje común para la definición del idioma. Un ejemplo de ella es la estandarización del idioma español en dos o tres letras como *es* o *spa*. Así cada archivo de propiedades debe tener un nombre terminado con guión al piso y los caracteres correspondientes al idioma.

El módulo I18N de GWT trabaja de forma estática y de forma dinámica, la diferencia radica en que de forma estática, a medida que se realicen cambios a las cadenas localizables se debe recompilar la aplicación, y de forma dinámica no es necesario hacerlo, ya que estas cadenas son guardadas directamente en el cliente mediante JavaScript.

Una desventaja importante del API de GWT para I18N con respecto al de *Symfony*, radica en que la interfaz creada para leer cada archivo de propiedades debe contener tantos métodos como claves existan en tal archivo, y *symfony* sólo maneja un método que se conecta automáticamente al XML creado para emplear I18N a las aplicaciones.

La etapa de investigación concluyó que de los APIs mencionados, *symfony* permite realizar las mismas funciones que el API de GWT de un modo más simple, así que el funcionamiento del API I18N desarrollado fue similar a éste.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

2.5 Investigación API de Multimedia

En el presente API se realizó una investigación de mecanismos para carga y descarga de cualquier tipo de archivo entre servidor y cliente bajo el framework *GWT*. Para la carga de archivos al servidor, los clientes, mediante la interfaz de usuario de la aplicación web, deben realizar la búsqueda en los directorios de su computadora, para luego realizar la carga del archivo seleccionado, este es el proceso que debe seguir los usuarios del sistema para la carga de cualquier archivo al servidor; para realizar la descarga del archivo, los usuarios del sistema tendrán en su interfaz algún mecanismo para seleccionar los archivos de la aplicación web y descargarlos.

Las utilidades nombradas deben ser construidas por desarrolladores de software, además, se debe realizar la comunicación entre servidor y cliente, primero para que los archivos cargados por el cliente sean recibidos en el servidor y guardados en cualquier medio como base de datos o carpetas temporales, y segundo para que los archivos descargados puedan ser vistos por el cliente.

Este proceso será realizado por el API que internamente tendrá una serie de mecanismos para que el desarrollador simplifique estas tareas. Los mecanismos para la carga de archivos al servidor investigados fueron los siguientes:

- *Servlets - JAVA*: utilizado para la comunicación entre el cliente y servidor en la carga y descarga de archivos.
- *FileUpload - GWT*: componente del framework *GWT* para búsqueda de archivos por el cliente (Google, 2011).
- *FormPanel - GWT*: componente tipo panel para alojar el componente *FileUpload*, es utilizado para lograr interoperabilidad con los servidores, mediante método *GET*¹² u *POST*¹³ (Google, 2010).
- *FileUpload - Apache Commons*: convierte a arreglo de bytes los archivos cargados en el servidor (Apache Software, 2010).

Aunque los componentes mostrados anteriormente pueden ser utilizados con otros propósitos, en cada uno fueron evaluadas las características

¹²GET: Utilizado para recuperar información identificada por un URI (Uniform Resource Identifier) en el navegador web.

¹³POST: Invocación de procesos que generan datos devueltos como respuesta a la petición en el servidor.

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

que son de utilidad para el API de Multimedia. Por precisar, la comunicación con el servidor para la carga de archivos usando el componente *FormPanel-GWT*, se realiza por método *POST*, ya que el método *GET* no soporta este tipo de operaciones. El *API de Multimedia* es construido utilizando estos componentes para la carga de archivos, el desarrollador se encargará solamente de agregar el componente *FormPanel* en la interfaz del cliente, el cual ya se encontrará dentro de una interfaz del *API*, y en el servidor se encarga de manejar los bytes recibidos desde el cliente, para ser almacenados en cualquier medio, como bases de datos o carpetas temporales, teniendo presente que para almacenar bytes de archivo en una base de datos, se utiliza un tipo de datos llamado *BLOB (Binary Large Objetc)* de los cuales existen cuatro tipos según el tamaño de los bytes, como son: *Tinyblob*, *Blob*, *MediumBlob* y *LongBlob*. Para almacenarlos en archivos temporales, los bytes deben ser convertidos al archivo original utilizando la clase *File* de *JAVA*, la cual provee toda la funcionalidad para el manejo de archivos de cualquier tipo en este lenguaje.

Para la descarga de archivos al servidor se utilizarán los siguientes mecanismos:

- Creación de URL: la *URL* debe ser creada teniendo en cuenta los datos del archivo que será descargado, ya que se utiliza el método *GET* para su envío.
- MIME Type: utilizado para que tanto servidor como cliente sepan el tipo de contenido que será descargado (Duamu, 2007).

La creación de la URL se realiza en el cliente según la información del archivo que se desea descargar, al pulsar un botón o link en el cliente se hará la llamada al servlet por método *GET*, el cual envía la información al servidor, allí dependiendo de esta información buscará el archivo en el lugar en que se encuentre almacenado. El API de Multimedia será diseñado con el fin de que el archivo seleccionado por el cliente en el servidor, sea enviado empleando los bytes del archivo, junto con su *MIME Type* para el propósito dicho. De esta forma, se busca que el desarrollador no se preocupe por realizar los *servlets* para la comunicación de carga y descarga de archivos, y que tenga a la mano las funciones necesarias para el transporte de archivos.

3. Trabajos Futuros

Luego de realizar las etapas de investigación presentadas en este artículo, se realiza el diseño y desarrollo de las APIs, en las que se hará

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

uso de los métodos y herramientas investigadas, con el fin de que cumplan con los propósitos establecidos. En los siguientes artículos a desarrollar, se mostrará el diseño de cada API junto con los resultados que se obtienen al ser usados bajo el framework de desarrollo web *GWT*, se demostrará su facilidad de uso y las características principales ofrecidas por cada una.

4. Conclusiones

La investigación de tecnologías recientes en el campo del desarrollo de software, permite conocer herramientas que muchas veces no son consultadas, o puestas a consideración, para prestar soporte a diversas funcionalidades necesarias en el desarrollo de aplicativos web. El presente trabajo dio a conocer herramientas y métodos con los que se simplifican tareas de desarrollo indispensables en cualquier proyecto web, como es el caso de la generación automática de reportes, con los que se presenta contenido de cualquier tipo en múltiples formatos de documentos, según las necesidades del cliente; la generación automática de gráficas de datos, para presentar resultados estadísticos sobre determinada información; la validación de campos de información, indispensable para que la información sea ingresada correctamente por los usuarios del sistema; la internacionalización, esencial para que las aplicaciones puedan ser leídas en cualquier idioma; y el transporte de archivos multimedia, que permite tener una base de información en la aplicación con soportes documentales o audiovisuales.

En el desarrollo de prácticas empresariales, se obtiene experiencia de trabajo, en un ambiente colaborativo, y se toma conciencia de la importancia de estar constantemente actualizado en nuevas tecnologías, para ser profesionales capacitados a sobrellevar todo tipo de retos.

Finalmente las librerías que fueron seleccionadas para el soporte del API de documentos fueron *JAVADOCX*, *ODFDOM* e *iText*, ya que según sus características son muy completas y prestan buen soporte documental.

5. Lista de referencias

Adobe. 2010, Scalable Vector Graphics. Recuperado el 02 de Septiembre de 2010, en: <http://www.adobe.com/svg/>
BVBA, 1T3XT. 2010, iText. Recuperado el 10 de Agosto de 2010, en: <http://itextpdf.com/>

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

DATANUCLEUS. 2010, DataNucleus. Recuperado el 06 de Agosto de 2010, en: <http://www.datanucleus.org/>

DUAMU. 2007, Que son los MIME Types. Recuperado el 13 de Enero de 2010, en: <http://www.duamu.com/re/articulo/1346/id/549/articulos-que-son-los-mime-types.html>

Dulio. 2009, Tutorial de PHP y YAML. Recuperado el 27 de Septiembre de 2010, en: <http://www.cristalab.com/tutoriales/tutorial-de-php-y-yaml-c271l/>

FOUNDATION, Free Software. 2007, GNU Lesser General Public License. Recuperado el 7 de Agosto de 2010, en: <http://www.gnu.org/licenses/lgpl.txt>

FOUNDATION, Apache Software. 2004, Apache License: Version 2.0. Recuperado el 6 de Agosto de 2010, en: <http://www.apache.org/licenses/LICENSE-2.0>

FOUNDATION, APACHE SOFTWARE. 2010, Commons FileUpload. Recuperado el 12 de Enero de 2011, en: <http://commons.apache.org/fileupload/>

FREEMARKER. 2010, Overview. Recuperado el 08 de Agosto de 2010, en: <http://freemarker.sourceforge.net/>

GOOGLE. 2010, Google App Engine. Recuperado el 06 de Agosto de 2010, en: <http://code.google.com/intl/es-ES/appengine/>

GOOGLE. 2010, Google Web Toolkit. Recuperado el 06 de Agosto de 2010, en: <http://code.google.com/intl/es-ES/webtoolkit/>

GOOGLE. Class FileUpload. Recuperado el 12 de Enero de 2011, en: <http://google-web-toolkit.googlecode.com/svn/javadoc/1.5/com/google/gwt/user/client/ui/FileUpload.html>

GOYVAERTS, Jain. 2009, Regular Expression Quick Start. Recuperado el 19 de Octubre de 2010, en: <http://www.regular-expressions.info/>

GUNTHER, John C. 2009, Package com.googlecode.gchart.client. Recuperado el 2 de Septiembre de 2010, en: <http://clientsidegchart.googlecode.com/svn/trunk/javadoc/com/googlecode/gchart/client/package-summary.html#ChartGallery>

IG, W3C DOM. 2005, Document Object Model (DOM). Recuperado el 9 de Agosto de 2010, en: <http://www.w3.org/DOM/>

ISO. 2010, ISO 639-6:2009. Recuperado el 11 de Noviembre de 2010, en: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?cs

"Revista Virtual Universidad Católica del Norte". No. 34, (septiembre-diciembre de 2011, Colombia), acceso: [<http://revistavirtual.ucn.edu.co/>], ISSN 0124-5821 - Indexada Publindex-Colciencias (B), Latindex, EBSCO Information Services, Redalyc, Dialnet, DOAJ, Actualidad Iberoamericana, Índice de Revistas de Educación Superior e Investigación Educativa (IRESIE) de la Universidad Autónoma de México.

number=43380

JAVADOCX. 2009, Choose the JAVADocX version that better suits your needs. Recuperado el 07 de Agosto de 2010, en: <http://www.javadocx.com/features>

JASON. 2010, docx4j v2.3.0 released. Recuperado el 06 de Agosto de 2010, en: <http://dev.plutext.org/blog/category/docx4j/>

JODReports. 2010, JODReports. Recuperado el 08 de Agosto de 2010, en: <http://jodreports.sourceforge.net/>

JFREE. 2010, JFreeChart. Recuperado el 03 de Septiembre de 2010, en: <http://www.jfree.org/jfreechart/>

LOMBARDI, Flavio y PIETRO, Roberto DI. *Secure virtualization for cloud computing*. En: Journal of Network and Computer Applications. June, 2010.p. 1-10. 1 p.

LIU, S. y DARLINGTON, M.J. *A computational framework for retrieval of document fragments based on decomposition schemes in engineering information management*. En: Advanced Engineering Informatics. 2006.p. 401-413. 401 p.

MEHTA, Bhakti. 2003, Java Architecture for XML Binding (JAXB). Recuperado el 06 de Agosto de 2010, en: <http://www.oracle.com/technetwork/articles/javase/index-140168.html#author>

Mkyong. 2010, Java Properties file examples. Recuperado el 27 de Septiembre de 2010, en: <http://www.mkyong.com/java/java-properties-file-examples/>

ORACLE. 2010, Los tipos BLOB y TEXT. Recuperado el 13 de Enero de 2011, en: <http://dev.mysql.com/doc/refman/5.0/es/blob.html>

POTENCIER, Fabien. 2010, The Definitive Guide to Symfony: Chapter 10 - Forms. Recuperado el 27 de Septiembre de 2010, en: http://www.symfony-project.org/book/1_0/10-Forms

SOFTWARE, Edgwall. 2010, docx4j component. Recuperado el 6 de Agosto de 2010, en: <http://dev.plutext.org/trac/docx4j>

SCHUBERT, Svante. 2010, Project Overview. Recuperado el 09 de Agosto de 2010, en: <http://odftoolkit.org/projects/odfdom/pages/ProjectOverview>

W3C. 1998, Vector Markup Language (VML). Recuperado el 02 de Septiembre de 2010, en: <http://www.w3.org/TR/NOTE-VML.html>

Wheaton, Paul. 2010, a Sample properties file. Recuperado el 27 de Septiembre de 2010, en: http://www.javaranch.com/jenny_beta/jennyProps.jsp