ESCUELA POLITÉCNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ESTUDIO COMPARATIVO DE LOS API's DE BÚSQUEDA DE GOOGLE, YAHOO Y BING PARA EL DESARROLLO DE APLICACIONES ANTI PLAGIO DE TEXTOS EN DOCUMENTOS

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR:

JORGE LUIS CÁRDENAS MONAR

SANGOLQUÍ, febrero de 2012

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. Jorge Luis Cárdenas Monar, como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Febrero del 2012

_____	_____
ING. CESAR VILLACIS	ING. MARCO VERGARA
DIRECTOR	COODIRECTOR

DEDICATORIA

Este proyecto le dedico a Dios, que me dio las fuerzas necesarias para poder realizarlo.

A mis padres, Jorge Cárdenas y Mariana Monar, indispensables en mi vida, a quienes respeto y amo con todo mi corazón, gracias por su apoyo moral, económico y sentimental que me ha permitido finalizar esta etapa de mi vida.

JORGE LUIS CÁRDENAS MONAR

AGRADECIMIENTO

Agradezco a Dios por darme la salud, vida y la motivación necesaria para poder seguir adelante a pesar de los obstáculos que se han presentado en mi camino.

A mi padres Jorge Cárdenas y Mariana Monar, quienes con su esfuerzo, dedicación y amor me han enseñado a ser un mejor ser humano, dedicado y perseverante. Gracias por brindarme su apoyo incondicional y estar ahí en cada momento importante de mi vida

A mis hermanos David y Diana quienes con su apoyo incondicional me han permitido obtener más fuerzas para seguir adelante en todo aspecto de mi vida.

A cada uno de los Ingenieros que han dedicado su tiempo, paciencia y conocimiento para hacer realidad este proyecto.

JORGE LUIS CÁRDENAS MONAR

ÍNDICE DE CONTENIDOS

LISTADO DE TABLAS

LISTADO DE FIGURAS

LISTADO DE ANEXOS

RESUMEN

El internet se ha vuelto un recurso indispensable para la búsqueda de información actual e interesante pero al mismo tiempo esta facilidad ha provocado que las personas investiguen menos y copien más; haciendo que el plagio de deberes, trabajos, consultas e incluso tesis sea cada vez más común.

Existen algunas herramientas que se encargan de verificar plagio de textos en documentos pero lastimosamente su costo no se encuentra al alcance de la mayoría de empresas e instituciones, además el grado de dificultad y el tiempo que se requiere para el desarrollo de una aplicación de estas características limita su acceso.

Con la llegada de la Web 2.0 aparecieron los API's públicos que permiten utilizar servicios e infraestructura de terceros para ampliar las posibilidades del desarrollo de todo tipo de aplicaciones (mash-up). En la proliferación de los API's públicos aparecieron los API's de búsqueda de empresas grandes como Google, Yahoo y Bing lo que permitió que se creen sitios como Plagium.com para poder realizar la tarea de identificar plagio en internet de una manera más sencilla.

Este proyecto realiza un estudio comparativo de los API's de búsqueda de Google, Yahoo y Bing para el desarrollo de aplicaciones anti plagio de textos en documentos tomando como criterios de comparación teórica el tiempo de respuesta del API, la precisión de la búsqueda, características adicionales que faciliten la búsqueda, costos, facilidad de uso y otros criterios de comparación basados en la documentación de los API's y para la comparativa práctica seimplementará un prototipo que utilice los tresAPI's de búsquedautilizando para la capa de presentación Windows Forms; con esta información se obtendrá cual es el mejor API para el desarrollo de este tipo de aplicaciones yfinalmente se terminará el desarrollodel prototipo utilizado en el estudio comparativo para que sea completamente funcional utilizando para la capa de presentación ASP.NET MVC 3.

ABSTRACT

The Internet has become an indispensable resource for finding current and interesting information but at the same time this facility have caused people to investigate less and copied more, making the plagiarism of homework, jobs, researches and even Thesis acommoncase.

There are some tools that are responsible for verifying textplagiarism in documents but unfortunately the cost is not accessible to most companies and institutions, and the degree of difficulty and time required to develop an application of these features limit their access.

With the arrival of Web 2.0 public API's appeared to allow the use of third party services and infrastructure to expand the possibilities of development of all types of applications (mash-up). In the proliferation of public API's, Search API's appeared by large companies like Google, Yahoo and Bing that allowto create sites like Plagium.comto perform the task of identifying plagiarism in the Internet in a more simple way.

This project is a comparative study of the search API's of Google, Yahoo and Bing for anti-plagiarism application development of text in documentsusing as theoretical comparison criteria the response time of the API, the precision of the search, additional features that facilitate the search, cost, ease of use and other comparison criteria based in the API's documentation and then for the practice comparison is going to implement a prototype that uses the threesearch API's with Windows Forms for the presentation layer; this information will get which is the best API for the development of these applications and finally, the development of the prototype used in the comparative study will be completed to be fully functional with ASP.NET MVC 3 for the presentation layer.

CAPÍTULO I

1. INTRODUCCIÓN

La aparición de la web 2.0 como blogs, foros, wikis, redes sociales, sitios de e-learning y CMS han generado el incremento del plagio de textos mediante Internet debido a que las personas desconocen que en la gran mayoría de estos sitios web la copia de su contenido total o parcial es ilegal si no se cita al autor o si no se pide autorización al mismo.

Muchas instituciones educativas, empresas y universidades usan este tipo de aplicaciones para poder llevar un mejor control anti plagio de deberes, trabajos, documentos científicos, revistas, entre otros; evitando que estos contengan texto copiado desde algún sitio web de Internet y de esta manera solventar posibles problemas legales y éticos que pueden repercutir en el prestigio de la persona implicada así como tambiéna la imagen de la empresa o institución a la cual representan. Este tipo de aplicaciones se han vuelto indispensables en el ámbito educativo y empresarial en donde la información documental tiene un alto grado de importancia.

Actualmente el uso de Internet para obtener información relevante sobre cualquier tema se ha vuelto común por la facilidad y rapidez con que se la obtiene, generando facilismo en estudiantes y profesionales, además de causar desconocimiento sobre el tema consultado. En la mayoría de ocasiones esto genera problemas legales, ya que el contenido copiado está protegido por leyes y derechos de autor y otro tipo de licencias que no permiten la copia directa de la información expuesta en un sitio.

En ciertos sitios web la información publicada bajo las licencias Creative Commons o CC que inspiradas en las licencias GPL para software permiten el utilizar dicha información, modificarla o mejorarla pero siempre citando al o los autores y cuando no se la utilice con fines comerciales. Las licencias contemplan otras restricciones que están estipuladas en el contenido textual de las mismas; La idea principal es el posibilitar de un modelo legal, para así facilitar la distribución y el uso de contenidos pero muy pocos sitios web implementan este tipo de licencias.

Las características, fortalezas y eficiencia de los API's de búsqueda como Google Custom Search API, Yahoo BOSS API y Bing Search API permiten que los textos de un documento puedan ser comparados con los datos en la Internet y devolver los resultados similares usando la infraestructura de la empresa proveedora del API, estos resultados luego son analizados por un algoritmo que detecta el plagio en el documento porcentualmente. La información que estos API's devuelven usan formatos de datos estandarizados como son JSON[1] y XML[2].

1.1 Planteamiento del problema

1.1.1 Conceptualización del problema

El plagio de textos en documentos ha proliferado de una manera alarmantedebido a la facilidad para obtener información del Internet.Esto ha generado desinterés en los estudiantes y profesionales por la investigación, al mismo tiempo que se comete el delito de plagio de acuerdo a lo establecido en los derechos de autor.

Las aplicaciones anti plagio de textos en documentos utilizancrawlers[3] que deben analizar un alto número de sitios y bases de datos para la obtención de resultados, luego usar un algoritmo de alta complejidad para identificar falsos positivos y comparar los resultados con el texto inicial usando búsquedas semánticas[4]. Hay algunas aplicaciones anti plagio mucho más eficientes que implementan algoritmos propietarios y que llevan mucho tiempo en el desarrollo de sus características; un ejemplo de este tipo de aplicación anti plagio es TurnItIn[5] que presta sus servicios a universidades para la evaluación y verificación de plagio en los deberes realizados por los estudiantes.

El uso de API's de búsqueda como componentes fundamentales de las aplicaciones anti plagio de texto en documentos ha aumentado considerablemente en los últimos años, un ejemplo es la aplicación web llamadaPlagium.com[6] que usa el API de Googley el de Bing, demostrando así que es una alternativa viable para el desarrollo de este tipo de

[1] JSON: JavaScript Object Notation, es un formato ligero para el intercambio de datos
[2] XML:Extensible Markup Language ('lenguaje de marcas extensible'), es un metalenguaje extensible de etiquetas desarrollado por la W3C, es un estándar para el intercambio de información
[3] Crawler: es un programa que inspecciona las páginas web de forma metódica y automatizada
[4] Búsqueda semántica: búsqueda comparativa de caracteres en un texto utilizando algoritmos de diferenciación
[5] TurnItIn: aplicación web para detección de plagio creado por iParadigms, LLC. Institutions
[6] Plagium: aplicación web para detección de plagio en textos utilizando el api de Google y Bing

aplicaciones aportandoal componente que se encarga de obtener los sitios web de internet que posiblemente contengan plagio. Hay que tomar en cuenta que el análisis semántico de texto para verificar plagio así como obtener el contenido del sitio o la página web debe ser implementado por separado. El API de búsqueda no puede realizar estas tareas.

1.1.2 Formulación del problema

En la actualidad no existe ningún estudio comparativo de los API's de búsqueda de Google, Yahoo y Bing para el desarrollo de aplicaciones anti plagio de textos en documentos el cualpermitagenerar un aplicativo de detección de plagio más eficiente, fácil de desarrollar y por ende de menor costo. El estudio permitirá definir el mejor API basado en parámetros comparativos teóricos y prácticos que garanticen que los resultados sean verídicos y útiles para el desarrollo de este tipo de aplicaciones y permitan verificar la factibilidad de usar los API's de búsqueda.

Las aplicaciones anti plagio de textos en documentos tiene componentes complejos y costosos de desarrollar que forman parte de la arquitectura de este tipo de aplicaciones como el crawler que permite conseguir información de sitios web, a un alto costo de procesamiento y ancho de banda al buscar esta información en millones de sitios web; otro componente básico de este tipo de aplicaciones es el análisis semántico para buscar coincidencias. Los API's de búsqueda facilitan el desarrollo del primero de los componentes evitando buscar en toda el internet por sitios que contengan información referente al plagio pero no evita el realizar un crawler que baje el contenido de esta mínima cantidad de sitios filtrados por el API.

Cada uno de estos API's devuelven una respuesta utilizando una o varias notaciones(XML o JSON) que permiten tanto enviar como recibir información a través de cualquier protocolo en este caso HTTP. Esto permite que cualquier sistema sin importar el lenguaje de programación con el que fue desarrollado o el sistema operativo en el cual se ejecuta pueda acceder a dicha información y hacer uso de ella como sea necesario; en este caso para obtener los resultados de los sitios que contengan la información más relevante con respecto al texto plagiado.

Los API's de búsqueda comparten algunas características y funcionalidades comunes que hacen fácil su consumo desde un sistema, como son las tecnologías de notación XML

o JSON, el protocolo que usan HTTP o TCP, tipos de fuentes donde consulta como: imágenes;web; PDF, número de consultas sin costo, característicasespecíficas tales como: tiempos de respuesta; opciones de configuración del API;estado de desarrollo del API; modos de invocación utilizando otras tecnologías como Javascript; porcentaje de certeza de los resultados; etc.

Al no existir ningún estudio comparativo de las características de los API's de búsqueda que permita definir el mejor API para el desarrollo de este tipo de aplicaciones, se desarrollaráun cuadro comparativo teórico en el cual los criterios de comparación están basados en la documentación de cada API y representan las características comunes y más importantes para el desarrollo de este tipo de aplicaciones; ademásun cuadro comparativo práctico en el cual los criterios de comparación están basados en el diseño y las pruebas funcionales de un prototipo de software básico que implemente los tres API's de búsqueda. Se utilizara Scrum[7]como metodología para el desarrollo del estudio comparativo.

Basado en los resultados de estos cuadros comparativos se definirá el mejor API para el desarrollo de aplicaciones anti plagio de textos en documentos y se desarrollaráun prototipo final tomando como base el prototipo utilizado para el estudio comparativo práctico, este prototipo contará con una interfaz web que utilice la tecnología ASP.NET MVC 3 el cual avale los resultados del estudio y que utilice como metodología de desarrollo Scrumpara la planificación, XP[8] para el desarrollo de los componentes internos del sistema y UML[9] para el diseño y modelado del prototipo.

1.1.3 Delimitación espacial

El presente proyecto de tesis contempla un estudio comparativo teórico y práctico de los API's de búsqueda, un prototipo de software básico utilizado en el estudio comparativo práctico y un prototipo de software final que será implementado como una aplicación web y será probado en la Escuela Politécnica del Ejército en el campus ubicado en Sangolqui

[7] Scrum: es una metodología ágil para la gestión y desarrollo de software basada en un proceso iterativo e incremental.
[8] XP: es un proceso ágil de software que pone más énfasis en la adaptabilidad que en la previsibilidad
[9] UML: Lenguaje Unificado de Modelado es el lenguaje de modelado de sistemas de software más utilizado y conocido

para que validelos resultados del estudio y verifique el correcto funcionamiento del aplicativo en la detección de plagio de los documentos subidos por el usuario.

1.1.4 Delimitación temporal

El presente proyecto de tesis tendrá una duración de 6 meses debido a la cantidad de información documental que requiere ser revisada para el desarrollo del estudio comparativo y por la complejidad del desarrollo de los prototipos que además de validar la comparativa prácticapermitiránser utilizados en una situación real.

1.2 Objetivos

1.2.1 Objetivo General

- Realizar un estudio comparativo de los API's de búsqueda de Google, Yahoo y Bing para el desarrollo de aplicaciones anti plagio de textos en documentos con un prototipo de softwarepara validarlo.

1.2.2 Objetivos Específicos

- Analizar la arquitectura, funcionamiento y los fundamentos teóricos aplicados al desarrollo de aplicaciones anti plagio de textos en documentos.
- Estudiar los conceptos teóricos acerca de XML, JSON, el protocolo TCPy HTTP para consumir API's de búsqueda.
- Investigar las principales características y funcionalidades de los API's de búsqueda de Google, Yahoo y Bing.
- Definir los criterios comparativos teóricos basados en la documentación de los API's de búsqueda y los criterios comparativos prácticos basados en el diseño y las pruebas de funcionalidad del prototipo básico de software.
- Realizar una comparativa teórica y prácticabasados en los criterios comparativos respectivamente definidos.
- Determinar el API más adecuado para su uso en el diseño e implementación de un prototipo de software final de una aplicación anti plagio de textos en documentos basados en los resultados del estudio.

- Aplicar la metodología Scrum, XP con UML más el patrón de diseño MVC[10], para desarrollar los prototipos de software básico y finalde una aplicación anti plagio de textos en documentos.
- Utilizar la tecnología ASP.NET MVC 3 con el lenguaje de programación C#, para desarrollar el prototipo de software final de una aplicación anti plagio de textos en documentos.

1.3 Justificación

La mayoría de instituciones, universidades y empresas manejan todo tipo de documentos de carácter privado que pueden de alguna forma contener texto plagiado desde Internet y no ser detectado ya que no cuentan con los medios tecnológicos que les ayuden en la tarea de encontrarlo y los programas anti plagio desarrollados por terceros son inaccesibles debido a su alto costo, lo que conlleva a la impunidad de quienes realizan esta actividad ilícita.

La aparición de los API's de búsqueda de empresas tan grandes como Yahoo, Google y Bing (Microsoft) ha permitido a los desarrolladores usar todo el poder de un motor de búsqueda para encontrar textos dentro de sitios web de manera más rápida y confiable, esto ha permitido que el desarrollo de aplicaciones anti plagio de textos en documentos más sencillas, eficientes y baratas haya incrementado considerablemente en los últimos años, generando la incógnita de cuál es el mejor API para el desarrollo de este tipo de aplicaciones. Por este motivo es de vital importancia realizar un estudio comparativo de los API's de búsqueda que permita definircuáles el mejor API para el desarrollo de estas aplicaciones.

El estudio a realizar tendrá como resultado la determinación del mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos más eficientes, de menor complejidad y costo que sus predecesoras en base a criterioscomparativos teóricos y prácticos donde loscriterios comparativos teóricos están basados en la documentación de los API's de búsqueday los criterios comparativos

[10] MVC:es un patrón de diseño que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (Modelo – Vista – Controlador)

prácticos están basados en el diseño y pruebas funcionales del prototipo de software básico.

La implementación de un prototipo de software final utilizando el mejor API permitirá verificar que los resultados del estudio son correctos y que es completamente viable el desarrollo de este tipo de aplicaciones ya que los resultados que la aplicación devuelvaserán confiables y útiles para la tarea a realizar. Además es necesario probar la teoría del estudio en un aplicativo funcional que permita observar de manera más clara las potencialidades y bondades que el uso de los API`s brindan a este tipo de aplicaciones. El prototipo se desarrollará con un enfoqué educativo y será probado en la universidad para verificar su correcto funcionamiento.

1.4 Alcance

La presente investigación se enfoca en el estudio de las aplicaciones anti plagio de textos en documentos; su funcionamiento a nivel de usuario final y las técnicas que se usan para obtener los resultados adecuados.

Se va a realizar un estudio de los API's de búsqueda de Google, Yahoo y Bing, sus características principales, funcionalidades, costos y opciones avanzadas para búsquedas de texto y documentos en sitios web.

El análisis comparativo contemplará dos partes; uno teórico y otro práctico. El cuadro comparativo teórico establecerá sus criterios de comparación basados en la documentación de los API's mientras que el cuadro comparativo práctico establecerá sus criterios de comparación en base al diseño y pruebas funcionales de un prototipo de software básico.

No se realizará el estudio de otros API's que no sean Google Custom Search API, Yahoo Boss API y Bing Search API.

Se implementará un prototipo de software básico y unofinalde una aplicación anti plagio de textos en documentos; el prototipo básico implementara el consumo de los tres API's, la lectura de los documentos ingresados por el usuario y mostrará los resultados debidamente estructurados; el prototipo finalimplementará las mismas funciones que el básico a excepción de que consumirá solo el mejor API de búsqueda definido por el

estudio y utilizará una base de datos para manejo de usuarios y el almacenamiento de documentos ingresados por los mismos.

Tantoel prototipobásico como el prototipo finalsolo leeráncomodatos de entrada el contenido de archivos de texto plano (txt y rtf) y documentos de Word 2010 (doc y docx) de tamaño no superior a los 2MB, no leerán el contenido de archivos PDF o Excel y como datos comparativos de Internetleeránsolositios web en HTML. El contenido de los documentos de entrada pueden estar escritos en cualquier idioma.

Se desarrollará para el prototipo de software básico una aplicación de escritorio con tecnología Windows Formsmientras que para el prototipo de software finalse desarrollará una aplicación Web con tecnología ASP.NET MVC 3; ambas aplicaciones utilizarán el lenguaje C# sobre la plataforma Windows bajo el framework .NET 4.0 utilizando el IDE Visual Studio 2010 Express. Los prototipos no se portarán a otro lenguaje o sistema operativo. Además, no se va a desarrollar un API de búsqueda.

Se aplicará la metodología SCRUM tanto para la planificacióndelos prototipos de software como para la planificación para el desarrollo del estudio comparativo de los API's de búsqueda.

Se utilizará un modelo matemático lineal para la definición de los criterios comparativos teóricos y prácticos enfocados en la funcionalidad del prototipo

Se utilizará la metodología XP para el desarrollo de los componentes de los prototipos de software básico y final.

Se utilizará UML para el diseño y modeladodelos prototipos de software básico y final.

Para la implementación delos prototipos y las pruebas del mismo, se las realizará en la máquina de desarrollo y no se publicará en Internet.

1.5 Hipótesis de Trabajo

El API de búsqueda de Google es el mejor API para el desarrollo de aplicaciones anti plagio de textos en documentos.

1.6 Metodología

Para el desarrollo de la presente investigación se cumplirá con lo siguiente:

- Recopilación sobre las bases, características y novedades de la Web 2.0
- Obtención de un marco teórico solido sobre lo que es un API, los servicios RESTful y su importancia en la evolución de los API's de búsqueda así como también los métodos HTTP que usan estos servicios para su funcionamiento y las definiciones de los protocolos TCP y HTTP.
- Estudio de los fundamentos teóricos de la notación JSON y del lenguaje de marcado XML empleado en el envío y recepción de peticiones a los API's de búsqueda.
- Investigación de los fundamentos teóricos básicos de la tecnología ASP.NET MVC 3 con el view engine Razor con la cual se desarrollará el prototipo de software final.
- Obtención de un marco teórico sólido sobre lo que es y cómo funciona la autenticación OAuth y la utilidad Diff.
- Investigación teórica sobre los API's de búsqueda y su evolución.
- Obtención de sugerencias mediante foros tecnológicos en la Internet relacionadas con experiencias en el uso de los diferentes API's de búsqueda para el desarrollo de aplicaciones Mash-Up.
- Estudio sobre la historia y evolución de las aplicaciones anti plagio de textos en documentos.
- Obtención de un marco teórico solido sobre la metodología SCRUM, XP y el lenguaje de modelado UM para diseño y modelado.
- Recopilación de información relacionada con los API's de búsqueda de Google, Yahoo y Bing a través de la investigación en medios escritos como libros, revistas y medios informáticos como Internet con lo cual gestionaremos el conocimiento teórico.

- Obtención de un marco teórico sólido, fundamentado por el campo de acción de los API's de búsqueda para clasificar y analizar la información acorde a los contenidos de la presente investigación.
- Aplicación de la metodología Scrum para la planificación del estudio comparativo sobre los API's de búsqueda de Google, Yahoo y Bing como también para el desarrollo delos prototipos de software básico y final de una aplicación anti plagio de textos en documentos utilizando además XP para el desarrollo de los componentes delos prototipos y UML para el modelado de estos. Los Sprints tendrán un tiempo máximo establecido en 15 días.
- Definición de los criterios de comparación teórica de los API's de búsqueda de Google, Yahoo y Bing basados en la documentación de cada uno de los API's
- Obtención de resultados del estudio comparativo teórico basado en los criterios de comparación teóricos definidos anteriormente.
- Definición de los criterios de comparación práctica de los API's de búsqueda de Google Yahoo y Bing basados en el diseño y pruebas funcionales de un prototipo de software básico de una aplicación anti plagio de textos en documentos.
- Obtención de resultados para el estudio comparativo práctico de las pruebas funcionales sobre el prototipo de software básico y los criterios de comparación prácticos definidos anteriormente.
- Definición del mejor API de búsqueda basado en los resultados obtenidos por los análisis teóricos y prácticos que será el caso de estudio para el desarrollo del prototipo de software final de una aplicación anti plagio de textos en documentos.
- Análisis de ideas y conciliación de criterios en términos de la estructura conceptual que sirve de marco teórico, para llegar a formulaciones precisas para el desarrollo del prototipo de software de la investigación basado en el mejor API de búsqueda para el propósito estipulado.
- Obtención de conclusiones y recomendaciones tanto prácticas como teóricas para establecer los niveles de rendimiento y calidad de resultados de los API's de búsqueda investigados.
- Generación de la documentación técnica, manuales de usuario, pruebas e implantación del prototipo de software final.

Entregables

Tabla 1.1: Entregables del proyecto

Nombre	Descripción
Definición del Product Backlog	Requisitos generales del estudio comparativo de los API's de búsqueda y desarrollo del prototipo de software básico y final de una aplicación anti plagio de textos en documentos.
Sprint Backlog	Requisitos específicos y calendario para el estudio comparativo y el desarrollo del prototipo de software.
Sprint	Para el estudio comparativo teórico: • Definición de criterios de comparación de los API's. • Adquisición de información basada en experiencias de terceros. • Entrega de un cuadro de características relevantes basados en los criterios definidos en base a la documentación de los API's. Para el estudio comparativo práctico: • Diseño del prototipo básico con UML, los diagramas a utilizar serán: Diagrama de casos de uso y documentación de cada caso de uso, diagramas de secuencia y diagrama de clases. • Codificación de los componentes del prototipo básico con el alcance definido por el sprint, utilizando XP en el proceso. • Pruebas de los componentes del sprint. • Entrega de un cuadro de características comparativas basados en las pruebas del usuario. Para el prototipo final: • Diseño del prototipo final con UML, los diagramas a utilizar serán: Diagramas de casos de uso y documentación de cada caso de uso, diagramas de secuencia, diagrama de clases, diagrama de datos, diagrama de componentes y diagrama de despliegue. • Codificación de los componentes del prototipo final

	con el alcance definido por el sprint, utilizando XP en el proceso. • Pruebas de los componentes del sprint con el usuario. • Entrega del módulo funcional del prototipo al usuario con las características definidas por el alcance del sprint. • Manual de usuario y de estándares de codificación
Cierre de los Sprint	Para el estudio comparativo teórico y práctico: • Definición del mejor API según los resultados de los cuadros comparativos teóricos y prácticos. Para el prototipo final: • Integración de cada uno de los módulos del prototipo final • Pruebas del prototipo final completo con el usuario.

CAPÍTULO II

2. MARCO TEÓRICO

2.1 La Web 2.0

2.1.1 Definición

La Web 2.0 se define como la nueva generación de sitios web basados en la creación de contenidos de manera colaborativa por los propios usuarios del sitio. En la Web 2.0 los que consume la información se convierten en generadores de la información que ellos mismo consumen. Es la evolución de las aplicaciones web tradicionales hacia aplicaciones orientadas al usuario final. Son aplicaciones colaborativas y servicios que intentan reemplazar las aplicaciones de escritorio. (Ver Figura 2.1)

Algunos ejemplos de la Web 2.0 son las redes sociales, los API's web, las aplicaciones Web (Google Apps), e-learning, servicios de alojamiento de videos (YouTube), wikis, blogs, mashups[11]y folcsonomías[12].

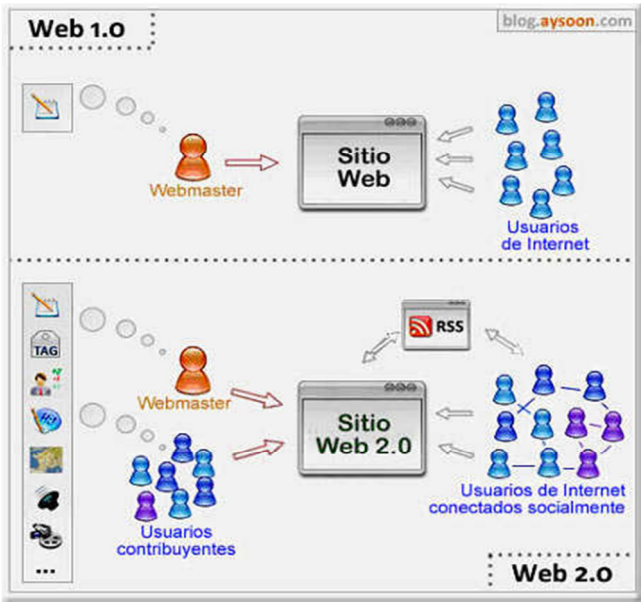

Figura 2.1: La Web 1.0 Vs Web 2.0[13]

[11] Mashups: aplicaciones que utilizan componentes de terceros dentro de las mismas, ejemplo: Google Maps

[12] Folcsonomía: es un tipo de clasificación colaborativa basado en etiquetas simples

[13] Imagen Web 1.0 vs 2.0 publicada por Aysoon: http://blog.cozic.fr/le-web20-illustre-en-une-seule-image

2.1.2 Aplicaciones

Algunos de los tipos de aplicaciones más destacados que se les puede catalogar como parte de la Web 2.0 son:

Blogs: Un blog es un sitio web personal en el que su autor puede escribir de manera cronológica artículos, noticias, etc., que incluyan texto, imágenes, videos y enlaces a otros sitios, y además permite a los lectores poder escribir sus comentarios acerca de cada uno de los artículos (entradas/post) que ha realizado el autor.

Wikis: Una wiki es un sitio web colaborativo, organizado mediante una estructura de páginas referenciadas en un menú lateral, donde varias personas elaboran y modifican contenidos de manera asíncrona. Suelen mantener un archivo histórico de las versiones anteriores. Hay varios servidores de wiki gratuitos como Wikipedia

Entornos para compartir recursos: estos entornos nos permiten almacenar todo tipo de recursos en Internet, compartirlos y visualizarlos cuando nos convenga desde Internet. Constituyen una inmensa fuente de recursos y lugares donde publicar materiales para su difusión mundial.

Documentos:se puede subir nuestros documentos y compartirlos, embebiéndolos en un Blog o Wiki, en un disco duro virtual como SkyDrive o DropBox o enviándolos por correo.

Videos: Al igual que los documentos, se pueden "embeber" un video tomado de algún repositorio que lo permita, tal como YouTube.

Presentaciones: permite subir presentación hechas en PowerPoint a una aplicación web para compartir su contenido como por ejemplo SlideShare.

Fotos: se puede compartir fotos en muchos tipos de aplicaciones web 2.0 sobretodo en redes sociales como Facebook y servicios de almacenamiento de fotografías como Picasa

Plataformas educativas: con la web 2.0 este tipo de aplicaciones ayudo a establecer a la educación a distancia como una forma viable de educación, Moodle es un ejemplo claro de este tipo de aplicación y también es la más utilizada.

Redes Sociales: sin lugar a duda son las aplicaciones web 2.0 más famosas y utilizadas, permiten compartir links, imágenes, videos, documentos, lugares turísticos, etc. entre amigos y familiares, un ejemplo es Facebook, Hi5, Tagged entre otros.

Algunos ejemplos de aplicaciones web 1.0 vs web 2.0 se muestran en la Tabla 2.1

Tabla 2.1: Aplicaciones Web 1.0 vs 2.0[14]

Web 1.0	Web 2.0
DoubleClick	AdSense
Ofoto	Flickr
Terratv	YouTube
Akamai	BitTorrent
mp3.com	Napster
Enciclopedia Británica	Wikipedia
webs personales	Blogging
Evite	upcoming.org y EVDB
Especulación de nombres de dominios	Optimización de los motores de búsqueda
páginas vistas	coste por clic
screen scraping	servicios web
Publicación	Participación
sistema de gestión de contenidos	Wiki
Hotmail	Facebook
directorios (taxonomía)	etiquetas (folcsonomía)
Stickiness	Redifusión

[14] Web 2.0: http://es.wikipedia.org/wiki/Web_2.0

2.1.3 Mapa de la Web 2.0

Este gráfico diseñado por la empresa internality muestra el desarrollo de las aplicaciones consideradas como Web 2.0 (Ver Figura 2.2):

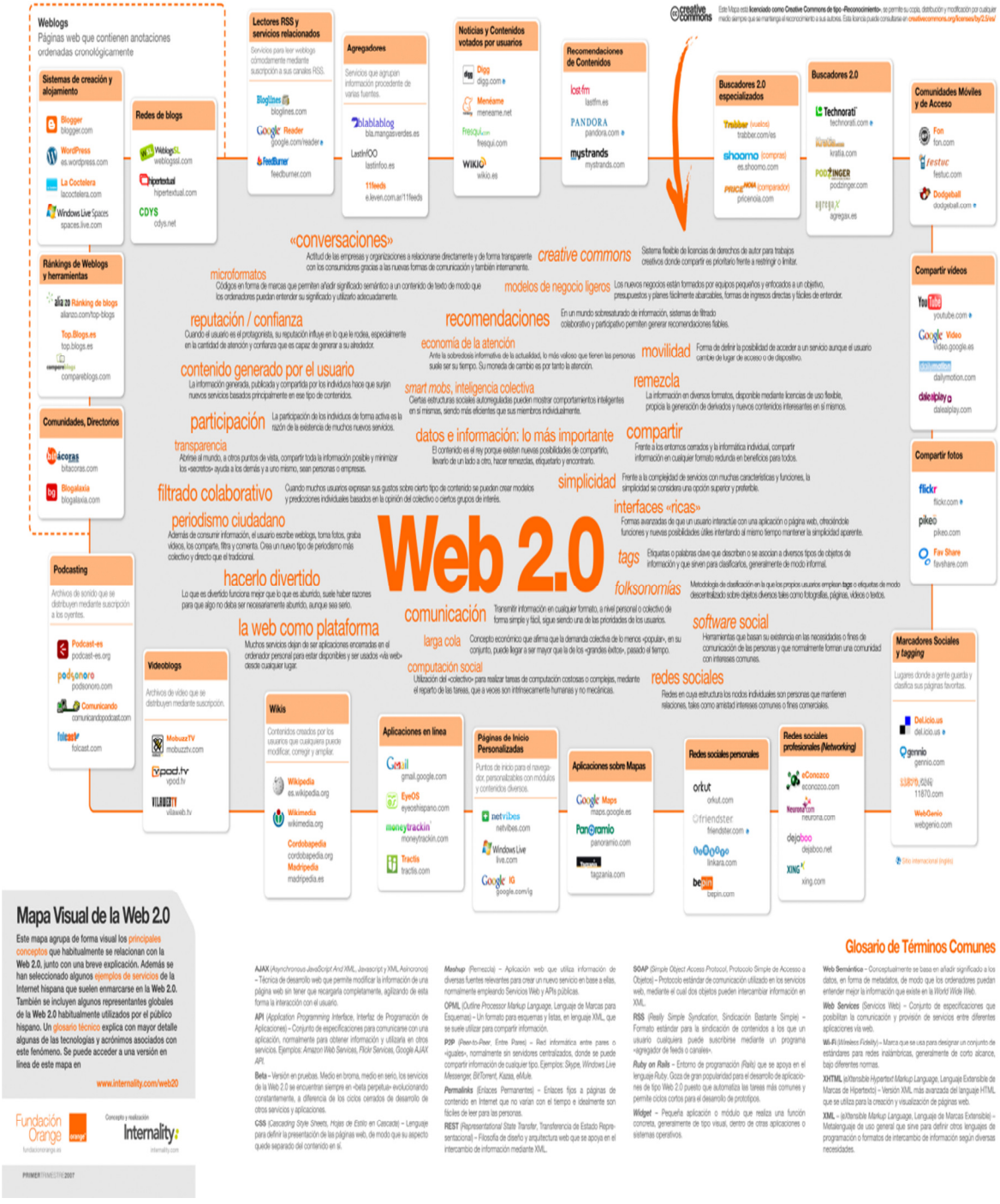

Figura 2.2: Aplicaciones que están consideradas como Web 2.0[15]

[15] Imagen Aplicaciones Web 2.0 creada por internality: http://internality.com/web20/

2.1.4 Características

La funcionalidad de la Web 2.0 se basa en la arquitectura existente de servidores web pero con más énfasis en el software y tecnologías que apoyan al desarrollo de las aplicaciones web 2.0.Se puede decir que una web está construida usando tecnologías de la Web 2.0 si se utiliza alguna de estas técnicas (Ver Figura 2.3):

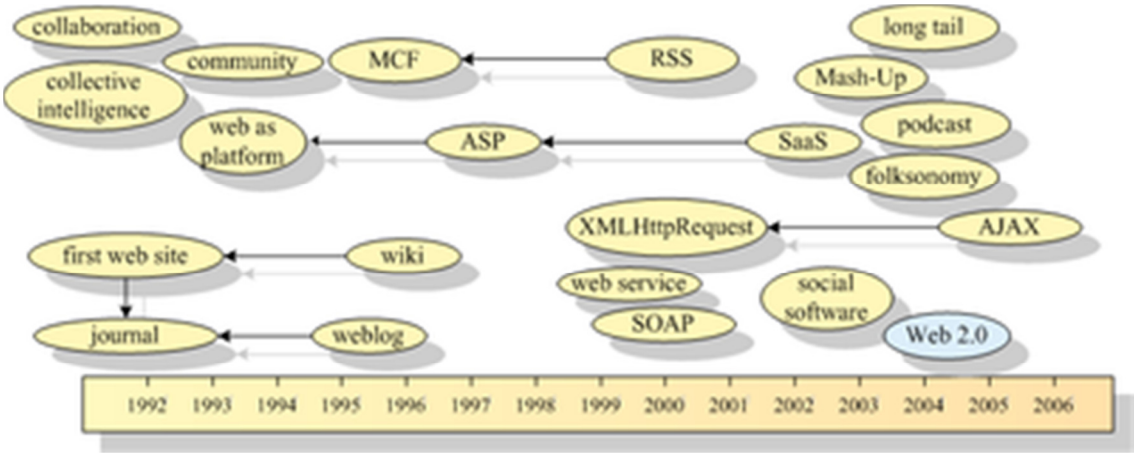

Figura 2.3: Tecnologías que se utilizan en una aplicación Web 2.0[16]

Técnicas:

- CSS, marcado XHTML y Microformatos
- Técnicas de aplicaciones ricas no intrusivas (como AJAX)
- Tecnologías RIA: Silverlight, Adobe Flex, Lazlo
- XUL (XML User Interface Language)
- Redifusión/Agregación de datos en RSS/ATOM
- URLs sencillas con significado semántico
- Soporte para postear en un blog
- JCC (Javascript Client Communication)y APIs REST oXML
- JSON (Javascript Object Notation)
- Dar control total a los usuarios en el manejo de su información.
- Proveer APIs o XML para que las aplicaciones puedan ser manipuladas por otros
- Mashups (aplicación web híbrida)

[16] Técnicas para la Web 2.0: http://es.wikipedia.org/wiki/Web_2.0

2.2 API, servicios RESTFul y el protocolo HTTP y TCP

2.2.1 API(Application Programming Interface)

2.2.1.1 Definición

Una API es una interfaz de programación de aplicaciones (del inglés: Application Programming Interface). Es el conjunto de llamadas a bibliotecas o librerías que ofrecen acceso a ciertas funcionalidades de un componente de software y permite abstraer esa lógica funcional a los desarrolladores que utilizan ese API, además permite la comunicación entre componentes de software.

Algunas de las API's desarrolladas por los constructores de software permiten el acceso a características de bajo nivel o propietarias, detallando solamente el cómo se debe llamar a cada rutina y qué funcionalidad brinda, sin especificar como ésta rutina fue implementada. Los API's son utilizados por los programadores para poder desarrollar software reutilizando código que ya está debidamente probado.

2.2.1.2 Tipo de API's

Un API puedeser:

- **General,** el conjunto completo de una API que se incluye en las bibliotecas de un lenguaje de programación, por ejemplo, Biblioteca de plantillas estándar de C + + o de Java
- **Específico,** pretende dar respuesta a un problema específico, por ejemplo, API de Google Maps o API de Java para servicios web XML.
- **Dependiente del lenguaje,** lo que significa que sólo está disponible mediante el uso de la sintaxis y los elementos de un lenguaje en particular, que hace que el API sea más fácil de usar.
- **Independiente del lenguaje,** escrito para que pueda ser llamado desde varios lenguajes de programación. Característica deseable para una API orientada a servicios que no está vinculado a un proceso o sistema y se puede proporcionar como llamadas a procedimientos remotos o servicios web. Por ejemplo, un sitio web que permite a los usuarios revisar los restaurantes locales es capaz de agregar una capa a sus comentarios sobre los mapas tomados de Google Maps, ya que Google Maps tiene una API

que facilita esta funcionalidad. El API de Google Mapscontrola la información que un sitio de terceros puede utilizar y cómo puede usarla.

2.2.1.3 Funcionalidad

Una de las principales funcionalidades de un API es el proveer de funciones de uso general a los desarrolladores como el dibujar un botón en la pantalla o conectarse a una base de datos y de esta forma evitarles el tener que crear desde cero este tipo de funcionalidad.

2.2.1.4 Web API

Cuando se utiliza en el contexto del desarrollo web, una API es típicamente un conjunto definido de mensajessolicitados a través de solicitudes HTTP, junto con una definición de la estructura de los mensajes de respuesta, que es por lo general en un lenguaje de marcado extensible (XML) o Javascript Object Notation (JSON). Mientras que el "Web API" es un sinónimo de servicio web.

Los Web API's permiten la combinación de múltiples servicios en nuevas aplicaciones esto se conoce también como mashups.

En la web, las API's son publicadas por sitios para brindar la posibilidad de realizar alguna acción o acceder a alguna característica o contenido que el sitio provee. Algunas de las implementaciones de API's web más conocidos son:

- Google Search (Para realizar búsquedas en la web de forma pragmática)
- Flickr (Permite compartir y gestionar fotos)
- Del.icio.us (Permite almacenar marcadores a sitios como los exploradores)
- Twitter (Aplicación que permite publicar mensajes cortos del estado, humor o actividad del usuario)
- Facebook (Red social que permite compartir fotos, enlaces, videos y posee una gran cantidad de juegos en línea)
- Amazon (Permite realizar compras por internet sobretodo de libros)
- Google Maps (Mapa del mundo entero con mucho detalle y capas de relieve, calles, tráfico y satelital)

- Google Custom Search API (API de Google para crear motores de búsqueda personalizados)
- Yahoo Boss Search API (API de Yahoo para crear motores de búsqueda personalizados)
- Bing Search API (API de Bing para crear motores de búsqueda personalizados)

2.2.1.5 El uso de las API's para compartir contenido

La práctica de publicación de API's ha permitido a las comunidades web crear una arquitectura abierta para el intercambio de contenidos y datos entre las comunidades y aplicaciones, ejemplos:

- Se pueden compartir fotos desde sitios como Flickr [17]y Photobucket [18]a redes sociales como Facebook y MySpace.
- Se puede integrar contenido, por ejemplo, incorporar una presentación de SlideShare[19] en un perfil de LinkedIn[20].
- Se puede publicar contenido dinámicamente. Compartir comentarios en vivo desde Twitter con una cuenta de Facebook.
- La información del usuario puede ser compartida desde las comunidades web a aplicaciones externas, ofreciendo nuevas funcionalidades a la comunidad web que comparte sus datos de usuario a través de una API abierta. Uno de los mejores ejemplos de esto es la plataforma de aplicaciones de Facebook.

[17] Flickr: aplicación web que permite compartir fotos
[18] Photobucket: aplicación web que permite compartir fotos
[19] SlideShare: aplicación web que permite subir presentación de PowerPoint
[20] LinkedIn: red social para establecer relaciones laborales

2.2.2 Servicios RESTful

2.2.2.1 Definición

"*La Transferencia de Estado Representacional (Representational State Transfer) o REST es una técnica de arquitectura software para sistemas hipermedia distribuidos como la Web*"[21], fue ganando amplia adopción en toda la web como una alternativa más simple a SOAP y a los servicios web basados en WSDL (Web Services Description Language).

2.2.2.2 Características[22]

REST define un conjunto de principios arquitectónicos con los cuales se diseñan servicios web enfocándose en los recursos del sistema, incluyendo cómo se accede a dichos recursos y cómo se transfieren por HTTP hacia clientes que utilizan varios lenguajes.

Una implementación de un servicio web REST sigue cuatro principios de diseño fundamentales:

- Utiliza los métodos HTTP de manera explícita, ejemplo: HTTP GET, HTTP POST, HTTP PUT y HTTP DELETE
- No mantiene estado
- Expone URIs con forma de directorios que tengan significado semántico, ejemplo: www.pagina.com/recurso/idrecurso
- Transfiere XML, Javascript Object Notation (JSON), o ambos al cliente, ejemplo:
    - XML <dato>valor</dato>
    - JSON { dato: valor }

2.2.2.2.1 REST utiliza los métodos HTTP de manera explícita

REST hace que los desarrolladores usen los métodos HTTP explícitamente lo que permite establecer una asociación uno-a-uno entre las operaciones CRUD y los métodos HTTP. De acuerdo a esta asociación:

[21] REST: http://es.wikipedia.org/wiki/Representational_State_Transfer
[22] Características de REST: http://www.dosideas.com/noticias/java/314-introduccion-a-los-servicios-web-restful.html

- POST para crear un recurso en el servidor
- GET para obtener un recurso
- PUT para cambiar el estado de un recurso o actualizarlo
- DELETE para eliminar un recurso

Ejemplo de actualización de un usuario

Antes:

GET /actualizarusuario?nombre=Zim&nuevoNombre=Dib HTTP/1.1

Después:

```
PUT /usuarios/Zim HTTP/1.1
Host: miservidor
Content'Type: application/xml

<usuario>
<nombre>Dib</nombre>
</usuario>
```

Se aconseja usar sustantivos en vez de verbos en las URIs de los servicios web REST, ya que los verbos están claramente definidos por el protocolo HTTP: POST, GET, PUT y DELETE, y para que los cliente puedan ser explícitos en los métodos que invocan, siendo así estos servicios web tampoco deberían definir más verbos en sus procedimientos remotos, ejemplo: **/agregarusuario**y **/actualizarusuario**.

2.2.2.2.2 REST no mantiene estado

Los clientes de los servicios web REST deben enviar peticiones completas e independientes ya que los servidores no mantendrán el estado de las peticiones localmente

Una petición completa e independiente permite que el servidor no tenga que almacenar ningún tipo de información sobre el estado de una petición. La aplicación o

el cliente que consuma un servicio web REST deben incluir en el encabezado y el cuerpo de la solicitud HTTP todos los datos necesarios para que el servidor pueda generar la respuesta.

2.2.2.2.3 Servicios con estado vs sin estado

La Figura 2.4 nos muestra un **servicio con estado**, dondeuna aplicación realiza peticiones para obtener la imagen siguiente, asumiendo que el servicio mantiene información sobre la última imagen que pidió el cliente. En un diseño con estado, el servicio incrementa y almacena en algún lugar una variable **imagenAnterior** para poder responder a las peticiones siguientes.

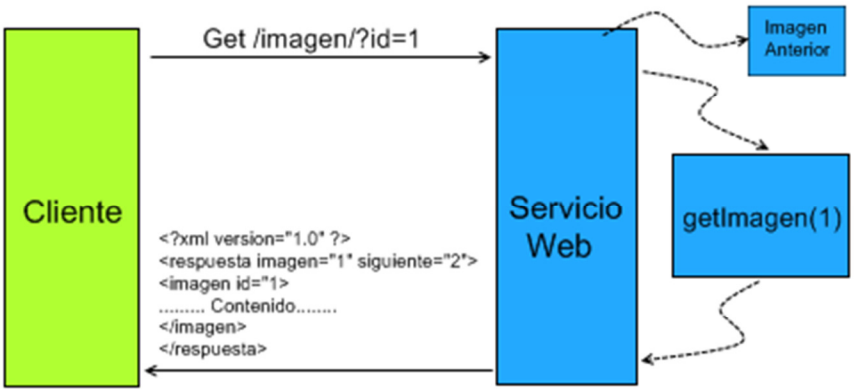

Figura 2.4: Servicio con estado

Los servicios con estado tienden a volverse complicados ya que requieren bastante análisis y diseño desde el inicio para que los datos se almacenen de manera eficiente.

Por otro lado, los servicios sin estado son más simples de diseñar, escribir y distribuir en varios servidores. Un servicio sin estado ofrece mejor rendimiento, deslinda la responsabilidad de mantener el estado al cliente de la aplicación.

En un servicio web REST, el servidor es responsable de generar las respuestas mientras que el cliente debe mantener el estado de la aplicación por su cuenta. Por ejemplo, en el mismo ejemplo de una petición de imágenes, el cliente debería incluir el

número de imagen a recuperar en vez de pedir "la siguiente", tal como se muestra en la Figura 2.5:

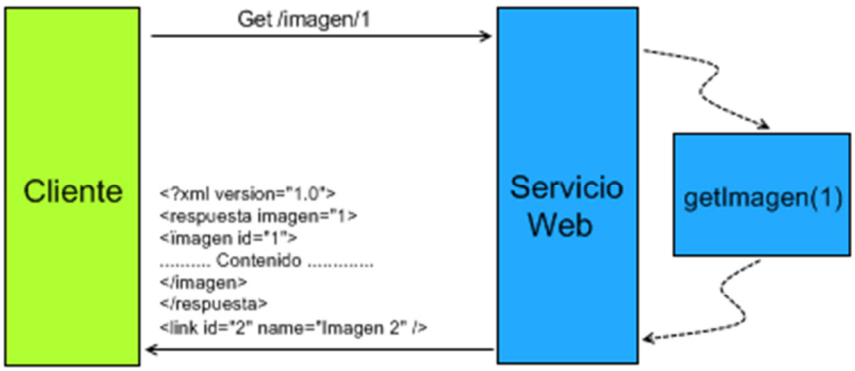

Figura 2.5: Servicio sin estado

"Un servicio web sin estado genera una respuesta que enlaza a la siguiente página del conjunto y le permite al cliente hacer todo lo que necesita para almacenar la página actual."[23]

2.2.2.2.3.1 Responsabilidad del servidor

- Las respuestas que devuelve un servicio REST deben mantener enlaces embebidos a recursos del mismo conjunto o jerárquicos (padre, hijo) para de esta manera se mantengan conectados.
- Los servicios REST generan respuestas indicando si los recursos devueltos son susceptibles de establecerse en cache o no, para esto utiliza los atributos Cache-Control y Last-Modified de la cabecera en la respuesta HTTP para indicarlo.

2.2.2.2.3.2 Responsabilidades del cliente de la aplicación

- Verifica si el encabezado de la respuesta recibida del servicio REST contiene el atributo Cache-Control para determinar si el recurso se puede almacenar en cache o no y también lee el atributoLast-Modified y envía al servidor la fecha en otro atributo If-Modified-Since para preguntarle al servidor si el recurso cambió desde entonces. Si el recurso no cambio el

[23] Servicios sin estado: http://www.dosideas.com/noticias/java/314-introduccion-a-los-servicios-web-restful.html

servidor devuelve una respuesta HTTP 304 o no modificado evitando tener que pedir nuevamente el recurso al servidor.

- Envía peticiones completas e independientes al servicio REST. Esto implica que el cliente hace uso completo de los encabezados HTTP e incluye los datos necesarios en el cuerpo de la petición para que el servidor lo procese

2.2.2.2.4 REST expone URIs con forma de directorios

Las URI de los servicios web REST deben tener significado semántico, ser intuitivas para los clientes. Las URI deben necesitar de poca o ninguna explicación o referencia para que un desarrollador pueda entenderque está consumiendo.

La forma más utilizada para poder llegar a este nivel de usabilidad es definir URIs como una estructura de directorios. Es decir jerárquica, con una única ruta raíz, y que se abran ramas a través de los "subdirectorios" para exponer las áreas principales del servicio

http://www.elservicio.org/problema/temas/{tema} http://www. elservicio.org/ problema/{año}/{mes}/{día}/{tema}

Algunas otras guías generales para crear URIs para un servicio web REST son:

- Ocultar la tecnología usada en el servidor que aparecería como extensión de archivos (.jsf, .jsp, .php, .asp), con esto se podría portar de tecnología sin necesidad de cambiar las URI.
- Mantener todo en minúsculas.
- Sustituir los espacios con guiones o guiones bajos (uno u otro).
- En vez de usar un 404 Not Found, devolver una página o un recurso predeterminado como respuesta.

Las URI deberían ser estáticas, no se debe cambiar el nombre de las URIs para que el cliente pueda crear favoritos o bookmarks al recurso.

REST transfiere XML, JSON, o ambos

"La representación de un recurso en general refleja el estado actual del mismo y sus atributos al momento en que el cliente de la aplicación realiza la petición."[24]

Al representar un recurso en un formato específico se debe tomar en cuenta las relaciones que tiene con otros recursos (recurso conectado), además que la representación sea simple y legible por humanos. Ejemplo:

```
<problemafecha="{fecha}" tema="{tema}">
<comentario>{comentario}</comentario>
<respuestas>
<respuesta de="
        gaz@mail.com" href="/discusión/temas/{tema}/gaz"/>
    <respuesta de="
        gir@mail.com" href="/discusión/temas/{tema}/gir"/>
</respuestas>
</discusión>
```

Se debe construir los servicios REST de manera que usen el atributo HTTP ACCEPT del encabezado, estableciendo el valor de este campo con un tipo MIME[25]. De esta manera, los clientes pueden pedir por un contenido en particular que se pueda representar y analizar de mejor manera, Algunos de los tipos MIME más usados para los servicios web REST son los que se muestran en la Tabla 2.2:

Tabla 2.2: Tipos MIME[26]

MIME-Type	Content-Type
JSON	application/json
XML	application/xml
XHTML	application/xhtml+xml

[24] REST con XML y JSON: http://www.dosideas.com/noticias/java/introduccion-servicios-web-restful.htm
[25] MIME: (Multipurpose Internet Mail Extensions) son una serie de convenciones o especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.)
[26] Aplicaciones de REST: http://www.dosideas.com/noticias/java/314-introduccion-a-los-servicios-web-restful.html

Muchos servicios REST implementan en sus URIs el tipo MIME que el cliente espera recibir, ej.

```
http://www.elservicio.org/problemas/temas/{tema}?format=json
http://www.elservicio.org/problemas/temas/{tema}.xml
```

2.2.2.3 Aplicaciones

Algunos ejemplos de servicios REST y su utilización en algunos tipos de aplicativos y tecnologías son:

- Los blogs,basados en su mayor parte en REST dado que implica descargar ficheros XML (en formato RSS o Atom) que contienen listas de enlaces a otros recursos.
- Amazon.com ofrece su interfaz para desarrolladores tanto en formato REST como en formato SOAP (siendo la versión REST la que recibe mayor tráfico).
- eBay ofrece una interfaz REST para desarrolladores.
- Yahoo, Facebook y Twitterofrece algunosAPI'sen REST para desarrolladores.
- Google ofrece algunosAPI's en REST para desarrolladores.
- El mecanismo de enrutamiento de Ruby on Rails soporta aplicaciones REST utilizando el patrón de diseño MVC.
- Microsoft tiene su implementación en ADO.NET Data Services Framework (anteriormente conocido como "Astoria").
- Implementación REST para Java: RestLet.

2.2.3 Protocolo HTTP

2.2.3.1 Definición[27]

Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo básico utilizado en la web. Es un protocolo queutiliza el esquema petición-respuesta entre un cliente y un servidor. Al cliente se lo conoce como "user agent" (agente del usuario), a la respuesta que devuelve el servidor se la llama recurso y se lo identifica mediante un URL o localizador uniforme de recursos. Los recursos pueden ser archivos, una consulta a una base de datos, la traducción automática de un documento, imágenes, etc.

2.2.3.2 Características[28]

HTTP es un protocolo sin estado, pero las aplicaciones web requieren frecuéntenme mantener estado por lo cual se utilizan técnicas como cookies[29]

La comunicación entre el navegador y el servidor se lleva a cabo en dos etapas:

- El cliente realiza una solicitud HTTP
- El servidor procesa la solicitud y después envía una respuesta HTTP

2.2.3.2.1 Solicitud HTTP

Una solicitud HTTP es un conjunto de líneas que el cliente (navegador) envía al servidor, incluye:

- **Una línea de solicitud:** La línea está formada por tres elementos que deben estar separados por un espacio:
  - El método (GET, PUT, POST, DELETE)
  - La dirección URL
  - La versión del protocolo utilizada por el cliente (por lo general, HTTP/1.1)
- **Los campos del encabezado de solicitud:** es un conjunto de líneas opcionales que permite agregar información adicional a la petición como:

[27] Definición de HTTP: http://www.mitecnologico.com/Main/ProtocolosNivelAplicacion
[28] Características de HTTP: http://es.kioskea.net/contents/internet/http.php3
[29] Cookie: Información que un servidor puede almacenar en la máquina del cliente

sistema operativo o navegador. Cada una de estas líneas maneja un formato de clave: valor.

- **El cuerpo de la solicitud:**conjunto de líneas que deben estar separadas de las anteriores por una línea en blanco, generalmente contienen los datos de la petición.

Una solicitud HTTP posee entonces lasiguiente sintaxis:

```
MÉTODO VERSIÓN URL
ENCABEZADO: Valor
. . . ENCABEZADO: Valor
Línea en blanco
CUERPO DE LA SOLICITUD
```

Un ejemplo de una solicitud HTTP sería entonces:

```
GET http://es.kioskea.net HTTP/1.0
Accept : Text/html
If-Modified-Since : Saturday, 15-January-2000 14:37:11 GMT
User-Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)

Datos……..
```

2.2.3.2.2 Respuesta HTTP

Una respuesta HTTP es un conjunto de líneas que el servidor envía al navegador e incluye:

- **Una línea de estado:** La línea está formada por tres elementos que deben estar separados por un espacio:
  - La versión del protocolo utilizada
  - El código de estado de la petición
  - El significado del código

- **Los campos del encabezado de respuesta:**es un conjunto de líneas opcionales que permite agregar información adicional sobre la respuesta o el servidor. Cada una de estas líneas maneja un formato de clave: valor.
- **El cuerpo de la respuesta:** contiene el documento solicitado.

Una respuesta HTTP posee entonces lasiguiente sintaxis:

```
VERSIÓN-HTTP CÓDIGO EXPLICACIÓN
ENCABEZADO: Valor
. . . ENCABEZADO: Valor
Línea en blanco
CUERPO DE LA RESPUESTA
```

Un ejemplo de una respuesta HTTP sería entonces:

```
HTTP/1.0 200 OK
Date: Sat, 15 Jan 2000 14:37:12 GMT
Server : Microsoft-IIS/2.0
Content-Type : text/HTML
Content-Length : 1245
Last-Modified : Fri, 14 Jan 2000 08:25:13 GMT

Datos…….
```

2.2.3.3 Métodos de petición

HTTP define 8 métodos (verbos) que indica la acción que el cliente desea que se realice sobre el recurso identificado.

- HEAD: solicita una respuesta idéntica a una petición GET pero sin el cuerpo de la respuesta. Es útil para recuperar y verificar solo la meta información escrita en los encabezados de la respuesta.
- GET: solicita un recurso especificado con cabeceras y cuerpo en la respuesta. Por seguridad no debería ser usado por aplicaciones como

método de envió de datos ya que muestra el URI con los datos de consulta utilizados para identificar el recurso.

- POST: los datos se incluyen en el cuerpo de la petición y son procesados por el servidor para el recurso indicado.Permite actualizar un recurso o en los servicios REST utilizar para la creación de nuevos recursos.
- PUT: sube un recurso especificado (archivo) al servidor. En un servicio REST permite actualizar un recurso.
- DELETE: borra el recurso especificado.
- TRACE: este método solicita al servidor que envíe de vuelta un mensaje de respuesta en cuyo cuerpo esténtodos los datos que reciba del mensaje de solicitud. Se utiliza con fines de comprobación y diagnóstico.
- OPTIONS: devuelve los métodos HTTP que el servidor soporta para un URL específico.
- CONNECT: establece una conexión mediante sockets entre el cliente y el servidor web

2.2.3.4 Encabezados

En la Tabla 2.3 se pueden ver los encabezados de petición

Tabla 2.3: Encabezados de petición[30]

Nombre del encabezado	Descripción
Accept	Tipo de contenido aceptado por el navegador (por ejemplo, *text/plain*).
Authorization	Identificación del navegador en el servidor
Content-Length	Tamaño del cuerpo de la solicitud
Content-Type	Tipo de contenido del cuerpo de la solicitud (por ejemplo, *text/html*).
Date	Fecha en que comienza la transferencia de datos
User-Agent	Cadena con información sobre el cliente, por ejemplo, el nombre, la versión del navegador y el sistema operativo

En la Tabla 2.4 se pueden ver los encabezados de respuesta

[30] Encabezados de petición HTTP: http://es.kioskea.net/contents/internet/http.php3

Tabla 2.4: Encabezados de respuesta[31]

Nombre del encabezado	Descripción
Content-Encoding	Tipo de codificación para el cuerpo de la respuesta
Content-Length	Tamaño del cuerpo de la respuesta
Content-Type	Tipo de contenido del cuerpo de la respuesta (por ejemplo, *text/plain*).
Date	Fecha en que comienza la transferencia de datos
Expires	Fecha límite de uso de los datos
Server	Características del servidor que envió la respuesta

2.2.3.5 Códigos de respuesta HTTP

En la Tabla 2.5 se puede ver los códigos de respuesta.

Tabla 2.5: Códigos HTTP[32]

N°	Tipo	Descripción
100, 111	Mensajes	Conexión rechazada
200	Operación exitosa	OK
201-203		Información no oficial
204		Sin Contenido
205		Contenido para recargar
206		Contenido parcial
301	Redirección	Mudado permanentemente
302		Encontrado
305		Utilizó un proxy
307		Redirección temporal
400	Error por parte del cliente	Solicitud incorrecta
403		Prohibido
404		No encontrado
410		Ya no disponible
500	Error del servidor	Error interno
502		Pasarela incorrecta
503		Servicio nodisponible

[31] Encabezados de respuesta HTTP: http://es.kioskea.net/contents/internet/http.php3
[32] Códigos de respuesta HTTP: http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol

2.2.4 Protocolo TCP

2.2.4.1 Definición

Protocolo de Control de Transmisión o TCP, es uno de los protocolos fundamentales en Internet, además es uno de los principales protocolos de la capa de transporte del modelo TCP/IP.

TCP da soporte a muchas de las aplicaciones más populares de Internet (navegadores, intercambio de ficheros, clientes ftp, etc.) y protocolos de aplicación HTTP, SMTP, SSH y FTP.

2.2.4.2 Características[33]

Las principales características del protocolo TCP son las siguientes:

- Es un protocolo orientado a conexión
- Permite controlar de extremo a extremo el estado de la transmisión
- Garantiza la entrega de datos a su destino sin errores
- Permite recibir los datos en el mismo orden en que fueron trasmitidos
- Proporciona un mecanismo para distinguir aplicaciones dentro de una misma computadora utilizando el concepto de puerto
- Permite el monitoreo del flujo de los datos y así evitar la saturación de la red.
- Permite que los datos se formen en segmentos de longitud variada para "entregarlos" al protocolo IP.
- Permite multiplexar los datos, es decir, que la información que viene de diferentes fuentes (por ejemplo, aplicaciones) se transmitan a través de la misma línea simultáneamente.
- Permite comenzar y finalizar la comunicación amablemente.
- Es full-duplex: permite comunicación de dos vías entre los sistemas finales
- Posee verificación de errores utilizando una técnica de checksum (suma de verificación)que verifica que los paquetes no estén corruptos
- Permite la recuperación de paquetes a través de ACK o acuso de recibo

[33] Características de TCP: http://es.kioskea.net/contents/internet/tcp.php3

2.2.4.2.1 La función de multiplexión

TCP posibilitamultiplexar/demultiplexar; es decir transmitir datos de diversas aplicaciones en la misma línea utilizando para esto el concepto de puertos. Para poder realizar estafunción se utilizan sockets que es la unión de una IP con un puerto (Generalmente vinculado a un tipo de aplicación).

2.2.4.3 Puertos TCP

TCP usa el concepto de número de puerto para identificar a las aplicaciones que envían y reciben paquetes. Un número de puerto contiene 16 bits sin signo, con lo que existen 65536 puertos posibles.

Los puertos son clasificados en tres categorías:

- **Bien conocidos:** asignados por IANA (Internet Assigned Numbers Authority), van del 0 al 1023, utilizados generalmente por el sistema operativo, ej.: FTP (21), SSH (22), Telnet (23), SMTP (25) y HTTP (80)
- **Registrados:** empleados por aplicaciones del usuario temporalmente o por servicios registrados por terceros, van del 1024 al 49151
- **Dinámicos/privados:** pueden ser utilizados por aplicaciones del usuario, no tienen significado fuera de la conexión TCP en la que fueron usados, van del 49152 al 65535

2.3 XML (Lenguaje Extensible de Marcado) y JSON (Javascript Object Notation)

2.3.1 XML (Lenguaje Extensible de Marcado)

2.3.1.1 Definición

XML, por sus siglas en inglés eXtensible Markup Language ('lenguaje de marcas extensible'), es un metalenguaje[34]extensible basado en etiquetas, es una simplificación y adaptación del SGML [35]y permite definir la gramática de lenguajes específicos. XML no es solo un lenguaje en particular sino que permite definir otros lenguajes para distintas necesidades, por ejemplo: XHTML, SVG, MathML.

XML no fue concebido para su uso en Internet, sino que se propuso como un estándar para el intercambio de información estructurada entre varias plataformas. XML se puede utilizar tanto en hojas de cálculo (Excel), editores de texto (Word), base de datos (SQL Server, Sybase) entre muchas otras.

XML es una tecnología utilizada ampliamente para permitir compatibilidad entre sistemas y poder compartir información de una manera sencilla, confiable y fácil de entender.XML se complementa de otras tecnologías para poder cubrir mayor espectro de necesidades. En los servicios REST, XML es uno de los formatos en los que el servicio devuelva la respuesta.

2.3.1.2 Estructura de un documento XML

Para que se pueda representar información de manera estructurada las partes que la componen como las partes que componen a su vez esas partes, deben estar bien definidas. A estas partes se las conoce como elementos que en XML se las llama etiquetas.

Las etiquetas son un pedazo de información con un sentido claro y definido que tienen la forma de <nombre>, donde nombre es el nombre del elemento que se está señalando.

[34] Metalenguaje: lenguaje que se usa para hablar sobre otro lenguaje
[35] SGML: Standard Generalized Markup Language, consiste en un sistema para la organización y etiquetado de documentos

Ejemplo de la estructura de un documento XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">
<Edit_Mensaje>
<Mensaje>
<Remitente>
<Nombre>Nombre del remitente</Nombre>
</Remitente>
<Destinatario>
<Nombre>Nombre del destinatario</Nombre>
</Destinatario>
</Mensaje>
</Edit_Mensaje>
```

Este es el ejemplo de código del DTD del documento «Edit_Mensaje.dtd»:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Este es el DTD de Edit_Mensaje -->

<!ELEMENT Mensaje (Remitente, Destinatario)*>
<!ELEMENT Remitente (Nombre)>
<!ELEMENT Nombre (#PCDATA)>
<!ELEMENT Destinatario (Nombre)>
<!ELEMENT Nombre (#PCDATA)>
```

2.3.1.3 Documentos XML bien formados y control de errores

Los documentos bien formados son aquellos que cumplan con todas las condiciones que define el formato y por lo tanto puede ser analizado por cualquier analizador sintáctico (parser) para verificar que cumpla con el estándar.

- Los documentos deben tener una estructura jerárquica con respecto a las etiquetas que delimitan sus elementos. Las etiquetas deben estar correctamente anidadas. Todos los elementos deben estar correctamente cerrados.

- Los documentos XMLsolo pueden tener un elemento inicial.
- Los atributos en XML siempre deben estar encerrados entre comillas simples o dobles.
- XML diferencia entre mayúsculas y minúsculas. Los espacios, tabuladores, retornos de carro, saltos de línea son tratados de forma diferente en el marcado XML.
- Se debe asignar nombres a los diferentes tipos de elementos y entidades. En XML los nombres tienen alguna característica en común.
- Las etiquetas, referencias de entidad y declaraciones se las denomina marcas, las cuales son parte del documento que un procesador XML espera entender. El resto del documento entre marcas son los datos que envía el usuario y es entendible por las personas.

2.3.1.3.1 Validez

Cada lenguaje definido con XML debe especificar cuál es exactamente la relación que debe verificarse entre los elementos presentes en el documento.

Esta relación entre elementos se especifica en un documento externo llamado DTD (Document Type Definition) o como XSchema.

2.3.1.4 Partes de un documento XML

Un documento XML está formado por los siguientes elementos:

2.3.1.4.1 Prólogo

Un documento puede empezar con unas líneas que describan que versión de XML se está utilizando, el tipo de documento, entre otras. El prólogo es opcional y contiene:

- **Una declaración XML:** sentencia que identifica al documento como un documento XML.
- **Una declaración de tipo de documento:** enlaza el documento XML con su respectivo DTD.
- **Uno o más comentarios:**permite escribir comentarios sobre el contenido del documento XML o la acción que este realiza.

2.3.1.4.2 Cuerpo

El cuerpo de un documento XML no es opcional y solo debe contener un elemento raíz. Necesita de datos para su buen funcionamiento.

2.3.1.4.3 Elementos

Los elementos XML pueden tener como contenido más elementos, atributos, caracteres o ambos, o bien ser elementos vacíos.

2.3.1.4.4 Atributos

Los elementos pueden tener atributos, que permiten agregar propiedades a los elementos de un documento. El valor de los atributos debe ir entre comillas, ejemplo:

```
<CiudadNombre="Quito" >Ciudad luz de américa…</Ciudad>
```

2.3.1.4.5 Entidades predefinidas

Entidades que permiten representar caracteres especiales para que un procesador XML no las intérprete como marcado XML, ejemplo:

```
entidad predefinida: & carácter: &.
```

2.3.1.4.6 Secciones CDATA

Específica datos utilizando cualquier carácter sin que se interprete como marcado XML, ejemplo:

```
<![CDATA[ contenido especial: áéíóúñ#@&]] >
```

2.3.1.4.7 Comentarios

Comentarios que son ignorados por el procesador XML, ejemplo:

```
<!--- Esto es un comentario --->
<!-- Otro comentario -->
```

2.3.1.4.8 Espacios de nombres

Permiten separar semánticamente los elementos que forman un documento XML.

2.3.1.5 Document Type Definition

Define los tipos de elementos, atributos y entidades permitidas en el documento XML. Los documentos XML que cumplen su DTD son considerados como documentos válidos.

2.3.1.6 XML Schemas (XSD)

Similar a un DTD. Define los elementos que va a contener un documento XML, su organización, que atributos y de qué tipo de dato pueden ser sus elementos.

2.3.1.6.1 Ventajas de los Schemas frente a los DTD

- Utilizan sintaxis de XML, a diferencia de los DTD.
- Pueden especificar los tipos de datos.
- Son extensibles.

2.3.1.7 Tecnologías de XML[36]

Entre las tecnologías XML disponibles se pueden destacar:

- **XSL:** Lenguaje Extensible de Hojas de Estilo.
- **XPath:** Lenguaje de Rutas XML.
- **XLink:** Lenguaje de Enlace XML.
- **XPointer:** Lenguaje de Direccionamiento XML.
- **XQL:** Lenguaje de Consulta XML.

2.3.1.8 Ventajas de XML

Algunas ventajas son:

- **Es extensible:** es posible extender un documento XML con la adición de nuevas etiquetas.
- **El analizador es un componente estándar,**esto evita que se creen analizadores específicos para cada versión del lenguaje XML.

[36] Tecnologías de XML: http://www.w3c.es/divulgacion/guiasbreves/tecnologiasxml

- **Terceros pueden usar un documento creado en XML,**ya que un documento XML es sencillo de entender y fácil de procesarlo, esto mejora la compatibilidad y comunicaciones entre aplicaciones sin importan la plataforma o el origen de datos empleado.

2.3.2 JSON (Javascript Object Notation)

2.3.2.1 Definición

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) formato ligero de intercambio de datos que se caracteriza por ser simple de ser leída y escrita por humanos y simple de interpretar y generar por las maquinas.JSON no requiere el uso de XML

2.3.2.2 Estructura

JSON está constituido por:

- Un diccionario o tabla hash (colección de pares de nombre: valor).
- Una arreglo de valores ordenada (lista de elementos).

Virtualmente todos los lenguajes de programación soportan estos tipos de dato.En JSON, se representan de estas formas[37]:

2.3.2.2.1 Objeto

Es un conjunto sin orden de pares nombre: valor; comienza con una llave de apertura y termine con una llave de cierre. Cada nombre es seguido por dos puntos junto con su valor y cada par nombre: valor está separado por una coma. Ver Figura 2.6

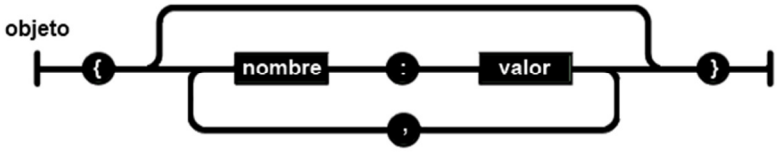

Figura 2.6: Definición de objeto en JSON

[37] Imagen Cadena de caracteres en JSON: http://json.org/json-es.html

2.3.2.2.2 Arreglo

Colección de valores que comienza un con corchete izquierdo y termina con un corchete derecho. Los valores se separan por una coma. Ver Figura 2.7

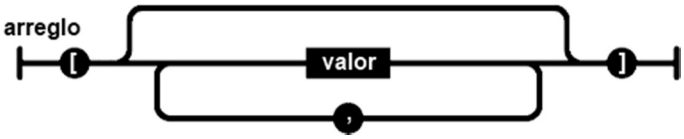

Figura 2.7: Definición de arreglo en JSON

2.3.2.2.3 Valor

Puede ser cualquier tipo de dato como una cadena de caracteres entre comillas dobles, un número, un valor booleano, un valor nulo, un objeto o un arreglo. Ver Figura 2.8

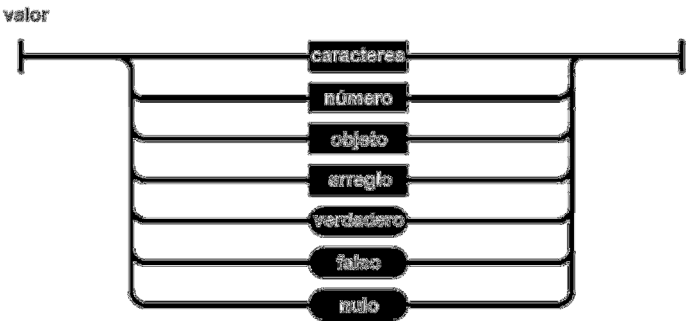

Figura 2.8: Definición de valor en JSON

2.3.2.2.4 Una cadena de caracteres:

El contenido de la cadena de caracteres debe estarencerrado entre comillas dobles y utilizar la barra invertida para especificar caracteres especiales como saltos de línea, espacios tabulados entre otros. Ver Figura 2.9.

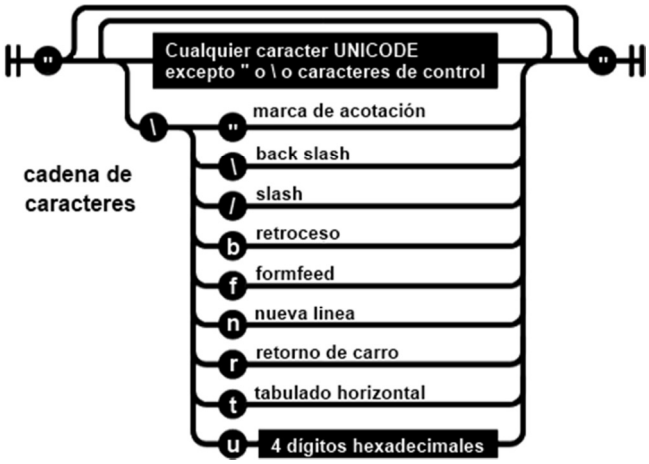

Figura 2.9: Definición de una cadena de caracteres en JSON

Un número se especifica de manera similar que el lenguaje C o Java pero no acepta formatos octales y hexadecimales. Ver Figura 2.10

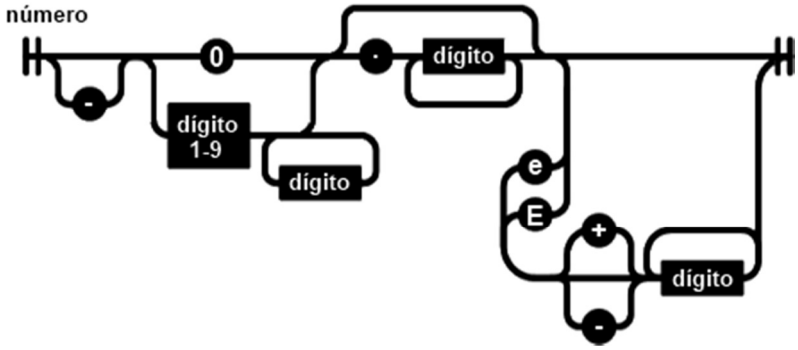

Figura 2.10: Definición de un número en JSON

2.3.2.3 Ejemplos de JSON

A continuación se muestra un ejemplo simple de definición de una persona utilizando JSON y XML.

JSON:

```
{"persona": {
 "nombre": "Jorge",   "apellido": "Cardenas",
 "educacion": {     "primaria": [
    {"nombre": "Borja 3", "ciudad": "Quito"},
    {" nombre ": "Maristas", "ciudad": "Quito" }     ]   } }}
```

XML:

```
<personanombre="Jorge" apellido="Cardenas">
<educacion>
<primaria nombre="Borja 3" ciudad="Quito" />
<primaria nombre="Maristas" ciudad="Quito" />
</educacion></persona >
```

JSONrepresenta mejor la estructura de los datos, requiere menos codificación y procesamiento.

2.3.2.4 Comparación con XML y otros lenguajes de marcado

XML ofrece mayor cantidad de herramientas para su utilización así como también mayor soporte pero JSON tiene muchos analizadores en el lado del servidor en la mayoría de entornosy lenguajes.

Tanto XML como JSON no poseen un mecanismo para representar grandes objetos binarios.

JSON puede ser muy compacto y eficiente si se usa de manera correcta.

YAML [38]es un súper conjunto de JSON que trata de superar algunas de las limitaciones de éste. Es considerado como ligero a pesar de ser significativamente más complejo que JSON. El lenguaje de programación Ruby utiliza YAML como el formato de serialización por defecto.

[38] YAML: Ain't Another Markup Language. es un formato de serialización de datos legible por humanos inspirado en lenguajes como XML, C, Python y Perl.

2.4 ASP.NET MVC 3 y Razor Engine, estructura, características

2.4.1 ASP.NET MVC 3

2.4.1.1 Definición

ASP.NET MVC 3 se basa en ASP.NET MVC 1 y 2. Es un marco de trabajo de aplicaciones web que implementa el patrón modelo-vista-controlador sobre la base de ASP.NET. El código fuente de ASP.NET MVC fue publicado bajo la licenciapública de Microsoft (MS-PL).

2.4.1.2 Características

Permite a los desarrolladores de software construir una aplicación web como una composición de tres funciones Modelo, Vista y Controlador

ASP.NET MVC permite probar fácilmente cada componente de la arquitectura de forma independiente.

No utiliza el modelo por defecto de devolución de datos (postback) de ASP.NET, las interacciones se dirigen a los controladores mediante el mecanismo de enrutamiento de ASP.NET haciendo que las vistaspuedan ser asignadas a URLs REST amigables

2.4.1.3 Motores de Vista (View Engines)

En versiones anteriores a ASP.NET MVC 3 el motor de vista predeterminado era el de Web Forms, que utiliza páginas .aspx normales para realizar el diseño de la vista, sin embargo, en la versión 3 del marco de trabajo se utiliza por defecto el motor de vista Razor Engine que utiliza páginas con extensión .cshtml.

ASP.NET MVC permite otros motores de vista publicados por la comunidad web como NDjango, Spark, StringTemplate, NakedObjectsMVC, entre otros.[39]

[39] ASP.NET MVC: http://en.wikipedia.org/wiki/ASP.NET_MVC_Framework

2.4.1.4 Patrón MVC (Modelo – Vista Controlador)[40]

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

2.4.1.4.1 Descripción del patrón

- **Modelo:**puede ser la representación de una tabla de una base de datos, ademáscontiene la lógica de negocio.
- **Vista:**es la página HTML y el código que provee de datos dinámicos a la página basados en un modelo.
- **Controlador:**es el encargado de la comunicación entre la vista y el modelo a través de eventos que la vista ejecuta.

El flujo que sigue el patrón MVC generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario (Vista)
- El controlador recibe la notificación de la acción que solicito el usuario. El controlador procesa la petición a través de un manejador de eventos como un handler o callback.
- El controlador accede al modelo, realizando la acción solicitada por el usuario.
- El controlador luego delega a la vista la tarea de desplegar la interfaz de usuario basada en los datos representados por el modelo donde se reflejan los cambios realizados al mismo.
- La interfaz de usuario espera que el usuario interactúe nuevamente para comenzar el ciclo nuevamente.

[40] MVC: http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

2.4.1.5 Nuevas características en ASP.NET MVC 3

2.4.1.5.1 Nuevos atributos de validación

- **CompareAttribute**

Permite comparar los valores de dos propiedades distintasdel modelo,ejemplo, la propiedad**ComparePassword**debe ser igual que la propiedad **Password**para que el modelo pueda ser validado.

```
public class Student {
   [Required]
   public string Password { get; set; }
[Required, Compare("Password")]
   public string ComparePassword { get; set; }
}
```

2.4.1.5.2 Nueva propiedad "ViewBag"

ViewBag es una propiedad similar a ViewData la cual permite pasar datos a la vista pero con una sintaxis más sencilla ya que es una propiedad dinámica, ejemplo,

En vez de escribirViewData["Mensaje"]="texto", se puede escribir ViewBag.Mensaje="texto".

2.4.1.5.3 Nuevos tipos de "ActionResult"

Estos son:

- **HttpNotFoundResult**:Devuelve un código HTTP 404 al cliente.
- **RedirectResult**:Retorna una redirección temporal (código de estado HTTP 302) o una redirección permanente (código de estado HTTP 301), dependiendo del parámetro booleano del ActionResult.
- **RedirectPermanent**, **RedirectToRoutePermanent**, y**RedirectToActionPermanent**:Estos métodos devuelven una

instancia de RedirectResult con la propiedad Permanent establecida en true.

- **HttpStatusCodeResult**. Devuelve un código HTTP especificado por el usuario.

2.4.1.5.4 Mejoras en Javascripty Ajax

En MVC 3 se usa por defecto un esquema de Javascript discreto (Unobtrusive). El Javascript discreto evita inyectar Javascript en línea dentro del HTML. Este hace que el HTML sea de menor tamaño.

Los helpers de validación usan el plugin jQueryValidate por defecto.

2.4.1.5.5 Validación del lado del cliente habilitado por defecto

En MVC 3 la validación del lado del cliente está habilitada por defecto

Para que la validación del cliente funcione correctamente, se necesita referenciar de manera apropiada en el sitio a jQuery y las librerías de validación de jQuery.

2.4.1.5.6 Soporte de JSON Binding

ASP.NET MVC 3 incorpora soporte de JSON binding quehabilita a los métodos de acción del controlador recibir datos codificados en JSON y enlazarlos con el modelo en los parámetros que tiene el método de la acción.

2.4.1.5.7 Interfaces de validación

La interfaz IValidatableObject permite realizar la validación a nivel del modelo, y permite devolver errores de validación que son específicos al estado de todo el modelo o entre dos propiedades dentro del modelo.

La interfaz IClientValidatable habilita a ASP.NET MVC el descubrir en tiempo de ejecución si el validador tiene soporte de validación del lado del cliente.

2.4.2 Razor Engine

2.4.2.1 Definición

Razor en una sintaxis de ASP.NET MVC utilizada para crear páginas Web dinámicas con los lenguajes de programación C# o Visual Basic.

La sintaxis Razor está basado en el lenguaje de programación C#. Utiliza el caracter @ para un bloque de código y no requiere que se cierre explícitamente dicho bloque de código.

Razor Engine es el motor de vista utilizado por defecto en ASP.NET MVC 3 que utiliza la sintaxis Razor para la creación de vistas.

2.4.2.2 Características

Algunas nuevas características Razor incluyen lo siguiente:

- Sintaxis @model para especificar el tipo pasado a la vista.
- Sintaxis de comentarios @* *@.
- La habilidad de especificar características por defecto al sitio entero.
- El método Html.Raw permite mostrar texto HTML sin codificación
- Soporte de código compartido entre múltiples vistas (_viewstart.cshtml o archivos _viewstart.vbhtml).
- Razor también incluye helpers, como los siguientes:
    - Chart. Crea un Chart, ofreciendo las mismas funcionalidades que el control de chart in ASP.NET 4.
    - WebGrid. Crea un data grid, completo con paginación y funcionalidad de ordenamiento
    - Crypto. Usa algoritmos de hashing para crear de manera apropiada contraseñas.
    - WebImage. Renderiza una imagen
    - WebMail. Envía un mensaje de correo

2.4.2.3 Ventajas

- Sintaxis optimizada para la generación de HTML
- El diseño reduce el número de caracteres necesarios
- La sintaxis de Razor es limpia y concisa.
- Razor es fácil de aprender porque está basado en lenguajes como C# y Visual Basic.
- Visual Studio da soporte de IntelliSense y coloreado para la sintaxis Razor
- Las vistas Razor permiten ser probadas sin necesidad de que la aplicación corra sobre un servidor web.

2.4.2.4 Sintaxis Razor vs .ASPX

Las Figuras 2.11 y 2.12 demuestran que la sintaxis Razor es más sencilla y menos intrusiva que la sintaxis .ASPX que utiliza los tags <%@ Page%>.

```
<h1>Code Nugget Example with .ASPX file</h1>

<h3>
    Hello <%=name %>, the year is <%= DateTime.Now.Year %>
</h3>

<p>
    Checkout <a href="/Products/Details/<%=productId %>">this product</a>
</p>
```

Figura 2.11: Sintaxis .ASPX[41]

```
<h1>Razor Example</h1>

<h3>
    Hello @name, the year is @DateTime.Now.Year
</h3>

<p>
    Checkout <a href="/Products/Details/@productId">this product</a>
</p>
```

Figura 2.12: Sintaxis Razor[42]

[41] Sintaxis ASPX: http://weblogs.asp.net/scottgu/archive/2010/07/02/introducing-razor.aspx

2.5 Protocolo OAuth

2.5.1 Definición

"Protocolo abierto, propuesto por Blaine Cook y Chris Messina, que permite la autorización segura de un API de una manera estándar y simple para aplicaciones de escritorio, móviles, y web."[43]

2.5.2 Historia[44]

Blaine Cook encargado del desarrollo de OpenID para Twitter en el 2006 junto con Chris Messina y Larry Halff de Ma.gnolia se reunieron con David Recordon para discutir el uso de OpenID con las API's de Twitter y Ma.gnolia para delegar la autenticación llegando a la conclusión de que no existía ningún estándar abierto para delegar acceso a las API's.

En abril de 2007 se creó el grupo de discusión de OAuth, y se empezó a escribir el primer borrador del protocolo abierto que con ayuda de DeWitt Clinton de Google terminó el borrador inicial de la especificación en julio de 2007. El borrador definitivo Oauth Core 1.0 se publicó el 3 de octubre de 2007.

"El protocolo OAuth 1.0 fue publicado como RFC 5849 y un informativo RFC, en abril de 2010".

2.5.3 Fallo de seguridad

En el 2009 se descubrió un fallo de seguridad en el protocolo OAuth en su versión 1.0 que afectaba al flujo de autorización del protocolo, para solventarlo se publicó la versión 1.0a.

2.5.4 Usos

OAuth se lo utiliza como un mecanismo de autorización para consumir de manera segura cualquier tipo de recursos comofeeds RSS/ATOM.

---

[42] Sintaxis Razor: http://weblogs.asp.net/scottgu/archive/2010/07/02/introducing-razor.aspx
[43] Definición OAuth: http://es.wikipedia.org/wiki/OAuth
[44] Historia de OAuth: http://es.wikipedia.org/wiki/OAuth

2.5.5 OAuth en Yahoo BOSS API

Para poder consumir el API de búsqueda de Yahoo, el cliente debe autenticarse utilizando OAuth como mecanismo de autenticación. Para ello se requieren los siguientes pasos:

- Registrarse en el API de Yahoo para obtener el valor del consumer secret y consumer key.
- Establecer el valor de nonce con un númerorandómico.
- Establecer el valor de timespan con el tiempo en milisegundos de la resta de la fecha actual con el 01-01-1970.
- Establecer el valor de signature utilizando los valores de consumer key, consumer secret, el url al que se desea tener acceso, el valor del timespan, valor de nonce y el método de encriptación HMAC-SHA1.
- Colocar los valores generados en los pasos anteriores en la cabecera de autorización de la petición con el formato requerido.

```
OAuth oauth_version="1.0", oauth_nonce="nonce",
oauth_timestamp="timestamp", oauth_consumer_key="consumerKey",
oauth_signature_method="HMAC-SHA1", oauth_signature="signature"
```

Figura 2.13: OAuth en Yahoo BOSS API[45]

[45]http://developer.yahoo.com/oauth/guide/oauth-make-request.html

2.6 Diff

2.6.1 Definición

Es una utilidad para generar las diferencias entre dos o más archivos o los cambios realizados entre una versión y otra del mismo archivo.El resultado de las diferencias encontradas se conoce como diff.

2.6.2 Algoritmo

"La operación de diff se basa en resolver el problema Longest Common Subsequence (LCS)."[46]

En el problema LCS, se tienen dos secuencias de elementos:

```
a b c d f g h j q z
a b c d e f g i j k r x y z
```

Seintenta encontrar la secuencia más larga de elementos que están presentes en ambas secuencias en el mismo orden que pueden ser obtenidas eliminando algunos elementos de la primera secuencia y otros de la segunda. En este caso es:

```
a b c d f g j z
```

De la subsecuencia común más larga solo hay un pequeño paso para conseguir un resultado del tipo diff, si un elementoestá ausente en la subsecuencia pero se encuentra en la original, se debió eliminar (-). Si está ausente en la subsecuencia pero presente en la segunda secuencia, debe haberse agregado (+)

```
e  h i  q  k r x y
+  - +  -  + + + +
```

[46] Algoritmo Diff: http://en.wikipedia.org/wiki/Diff

2.6.3 Uso[47]

En Unix se invoca desde la línea de comando enviando como parámetros los nombres de los dos archivos o directorios a comparar en el orden: original –nuevo. El resultado del comando mostrará los cambios requeridos para hacer que el archivo original se convierta en el nuevo archivo.

Si original y nuevo son directorios, el comando diff se ejecutará para cada archivo que exista en ambos directorios.

En el comando de Unix las líneas comunes a los dos archivos no se muestran, mientras que las líneas que se movieron aparecen como añadidas en el nuevo lugar y como eliminadas en el antiguo lugar.

2.6.4 Variantes

Las modificaciones que se implementaron en la utilidad diff son:

- El poder comparar archivos palabra por palabra
- El poder distinguir si una línea movida en el archivo original tiene el mismo contenido que en el archivo nuevo y si es así no lo considera como línea modificada.
- Soporte a mayor cantidad de formatos como imágenes, documentos de Word, Excel, entre otros.
- Mejoras en el algoritmo base[48]

[47] Uso: http://es.wikipedia.org/wiki/Diff
[48] Variantes del comando Diff: http://en.wikipedia.org/wiki/Diff

2.7 LosAPI's de búsqueda su historia y evolución

2.7.1 Definición

Los API's de búsqueda aparecieron en el año 2002 con el auge de la web colaborativa o 2.0 Estos API's pertenecen al grupo de los Web API que permite a desarrolladores acceder a los servicios de búsqueda de la empresa publicadora del API desde sus propias aplicaciones utilizando los estándares como SOAP, WSDL, XML, JSON, ATOM entre otros

2.7.2 Historia y evolución

Las empresas Google y Yahoo publicaron las primeras versiones de sus API's de búsqueda Google SOAP Search API y Yahoo Search API respectivamente. Cada uno utilizaba distintos estándares para manejar las peticiones a sus servicios web; en el caso de Google utilizaba SOAP y WSDL (XML) para la petición y respuesta del API mientras que Yahoo manejaba REST (JSON) por ser más ligero; ambos API's eran de uso gratuito hasta un número restringido de peticiones de alrededor de 5000.

Después de algunos años Google decidió abandonar el Google SOAP Search API y público uno nueva llamado Google Search AJAX API que fue una biblioteca Javascript que conjuntamente con XML generaban las peticiones y respuestas del API mientras tanto Yahoo renovó su API de búsqueda al cual lo nombró Yahoo Boss API que de igual manera utilizaba REST agregándole más funcionalidades como compatibilidad para micro formatos y más restricciones de uso en la versión no comercial.

Después de más de cinco años de retraso Microsoft con su buscador por defecto Bing desarrollo su API llamado Bing Search API con muy pocas funcionalidades con respecto a su competencia pero con la diferencia de que su utilización era completamente gratuita.

Alrededor del periodo 2009 – 2010 las tres empresas Google, Yahoo y Microsoft empezaron una competencia sería por los API's de búsqueda así Google abandonó nuevamente el Google Search AJAX API y creó un nuevo API llamado Google Custom Search API que tiene un enfoque más comercial restringiendo a 100 peticiones diarias gratuitas pero con muchas más características que sus antecesores

con la posibilidad de consumirlo utilizando XML, JSON o ATOM y muchas características personalizables del motor de búsqueda como buscar en ciertos sitios primero, especificar qué tipo de objetos buscar, micro formatos, etc. En el caso de Yahoo mejoró su versión de Yahoo Boss API con un enfoque igual más comercial sin permitir peticiones gratuitas al API con muchas características personalizables y la opción de buscar en otros aplicativos propios de Yahoo como Yahoo Groups, News, etc. Microsoft en cambio optó por mantener el enfoque gratuito en su API sin restricción de peticiones por el momento y con muchas más opciones configurables para consultar el API utilizando XML o JSON para enviar y recibir peticiones pero lastimosamente no tan completo como su competencia.

2.7.3 Características[49]

La mayoría de los API's de búsqueda actuales poseen estos servicios:

- Búsqueda de audio.
- Análisis de contenido.
- Búsqueda de imágenes.
- Búsqueda local.
- Búsquedas de noticias.
- Búsquedas de video.
- Búsqueda de blogs
- Spelling Suggestion.

Para acceder a los servicios que presta cualquiera de los API's de búsqueda se requiere un API KEY que es la manera habitual de identificar a un usuario con el API y solo se lo puede obtener registrándose en el sitio del proveedor del API sin o con costo en el caso de Yahoo y Google.

[49] Características actuales de un API de búsqueda: http://es.wikipedia.org/wiki/Google_SOAP_Search_API

2.7.4 OpenSearch

"Es un conjunto de tecnologías que permiten publicar los resultados de una búsqueda en un formato adecuado para la sindicación y agregación. Es una forma para que las páginas web y los motores de búsqueda publiquen sus resultados de forma accesible y estándar."[50]

2.7.4.1 Historia

La versión de OpenSearch1.0 fue publicada en marzo del 2005 y los borradores de la versión 1.1 fueron publicados a finales del 2005, fue desarrollado por A9 filial de Amazon.com.

2.7.4.2 Utilización

OpenSearch es ampliamente utilizado en el API de Google Custom Search y el API de Google Custom Search JSON/ATOM que permiten respectivamente crear motores de búsqueda personalizados y consumir estos motores de búsqueda recibiendo los datos en formato JSON o ATOM.

En el API de búsqueda de Yahoo llamado Yahoo BOSS API también se utiliza ampliamente el conjunto de tecnologías OpenSearch y en el API de búsqueda de Bing utiliza solo la interfaz RSS de OpenSearch

2.7.4.3 Componentes de OpenSearch[51]

OpenSearch consiste en:

- Archivos descriptivos OpenSearch.
- OpenSearch Query Syntax.
- OpenSearch RSS.
- OpenSearch Aggregators.
- OpenSearch "Auto-discovery".

[50] Definición de OpenSearch: http://en.wikipedia.org/wiki/OpenSearch
[51] Componentes de OpenSearch: http://en.wikipedia.org/wiki/OpenSearch

2.8 Aplicaciones anti plagio de textos en documentos historia y evolución

2.8.1 Definición

El plagio es la acción de copiar obras ajenas, dándolas como propias. Desde el punto de vista legal, es una infracción del derecho de autor sobre una obra artística o intelectual de cualquier tipo.

Un sistema de detección de plagio es una aplicación que se encargan de analizar un documento y verificar la originalidad del contenido contra fuentes de Internet o contra una base de datos documental.

2.8.2 Historia y evolución

Las aplicaciones anti plagio de textos en documentos aparecieron alrededor del año 2000 como una posible solución para evitar el plagio de documentos entregados por los estudiantes de colegios y universidades.

Una de las primeras aplicaciones anti plagio en aparecer se llama TurnItIn una aplicación web que es utilizada en muchas universidades y colegios a nivel mundial, esta aplicación se encarga de la gestión de deberes de los estudiantes revisando cada uno de los trabajos entregados con el motor de detección de plagio y mostrando los resultados tanto al profesor como el estudiante.

Con el tiempo empezaron a aparecer más aplicaciones anti plagio de textos en documentos en línea y de escritorio destacadas como muestra la Tabla 2.6.

Tabla 2.6: Aplicaciones de detección de plagio[52]

Gratuitas	Comerciales
Chimpsky	Attributor
CopyTracker	Copyscape
eTBLAST	Grammarly
The Plagiarism Checker	Iparadigms: Ithenticate,Turnitin
Plagium	Plagiarismdetect

[52] Aplicaciones de detección plagio: http://en.wikipedia.org/wiki/Plagiarism_detection

Todas estas aplicaciones han mejorado sus características de detección de plagio al pasar de los años, mejorando sus tiempos de respuesta, aumentando el tamaño de sus bases documentales entre otras funcionalidades.

En el apogeo de la Web 2.0 el uso de motores de búsqueda y sus API's han sido utilizados para buscar determinadas palabras clave o frases clave de un documento de sospecha en Internet. Este método puede ser muy eficaz cuando se usa en fragmentos pequeños y específicos.

2.8.3 Requisitos generales de diseño de sistemas de detección de plagio

Los requisitos mostrados en la Tabla 2.7 están orientadosa documentos de texto.

Tabla 2.7: Requisitos generales de diseño de sistemas anti plagio[53]

Factor	Descripción y alternativas
Ámbito de la búsqueda	En el internet, utilizando los motores de búsqueda / bases de datos institucionales / Local, un sistema de base de datos específica.
Análisis en tiempo	Tiempo de espera entre el momento de presentar un documento y el momento en que los resultados estén disponibles.
Capacidad de Documentos	Número de documentos que el sistema puede procesar por unidad de tiempo.
Compruebe la intensidad	¿Con qué frecuencia y qué tipos de fragmentos de un documento (párrafos, oraciones, palabra) consulta el sistema en recursos externos, tales como los motores de búsqueda?.
Comparativa tipo de algoritmo	Los algoritmos que definen la forma en que el sistema compara unos documentos contra otros.
Precisión y Recall	Número de documentos marcados correctamente como plagiados en comparación con el número total de documentos marcados, y el número total de documentos que fueron realmente plagiados. Precisión alta significa que se encontraron algunos falsos positivos, y Recall alta significa que se quedaron sin detectar algunos falsos negativos.

[53] Requisitos de una sistema de detección plagio: http://en.wikipedia.org/wiki/Plagiarism_detection

La gran mayoría de aplicaciones de detección de plagio utiliza bases de datos documentales muy grandes que van creciendo con los documentos adicionados por los usuarios del sistema. Esta característica es considerada como una violación de los derechos de autor del estudiante.

2.8.4 Plagio en código fuente de software

El plagio de código fuente software es muy común pero para poder verificar la existencia del mismo se requiere de herramientas distintas a las empleadas para la detección de plagio en documentos de texto.

Algo que distingue el plagio de código fuente del de texto es que en la mayoría de tareas de programación los requisitos del programa son muy específicos y esto complica el encontrar código que se ajuste exactamente a lo que se pidiópor lo que la mayoría de los estudiantes optan por copiar de sus compañeros.

Según Roy y Cordy, los algoritmos de detección de similitud en código fuente pueden ser clasificadosen base a:[54]

- Strings.
- Tokens.
- Parse Trees.
- Program Dependency Graphs(PDGS).
- Métricas.
- Enfoques híbridos.

[54] Plagio en código fuente de software: http://en.wikipedia.org/wiki/Plagiarism_detection

2.9 Metodología SCRUM, XP con UML

2.9.1 SCRUM

2.9.1.1 Definición

Scrum es una metodología ágil para la gestión y desarrollo de software que se basa en un proceso iterativo e incremental.

Scrum es un proceso basado en la aplicación de un conjunto de mejores prácticas para el trabajo colaborativo. Estas prácticas están basadas en un estudio de la manera de trabajar con equipos altamente productivos.

2.9.1.2 Fundamentos de Scrum

Scrum se basa en:

- El desarrollo incremental e iterativo de pequeños bloques de los requisitos del proyecto con fechas de entrega cortas.
- Priorización de los requisitos del proyecto según el valor y el coste que define el cliente para cada iteración.
- El control del proyecto al final de cada iteración para que el cliente pueda tomar decisiones en base a resultados obtenidos en ese momento. El equipo de trabajo se sincroniza diariamente.
- Se otorga al equipo la autoridad necesaria para cumplir con los requisitos que se comprometieron a entregar en la iteración.
- Fuerte colaboración y comunicación entre el equipo de trabajo y el cliente.

2.9.1.3 Aplicaciones

Scrum se puede utilizar en todo tipo de proyectos pero es una excelente metodología para gestionar proyectos complejos, que requieren obtener resultados rápido, pero los requisitos cambian fácilmente provocando que la creatividad e innovación sean pilares fundamentales para el éxito del proyecto.

Scrum es útil también cuando un proyecto esta fuera de control por costes, tiempos de entrega de versiones largas o por motivos de alta rotación del personal debido a moral baja en el equipo de trabajo.

2.9.1.4 Características

Scrum está basado en un conjunto de mejores prácticas y roles definidos que se deben utilizar a todo lo largo del proceso de desarrollo de un proyecto de software.

Scrum permite que durante la duración del proyecto el cliente cambie los requerimientos, generando desafíos impredecibles al equipo de trabajo que deben ser solventados innovando y siendo muy creativos.

2.9.1.5 Roles de Scrum

Scrum define dos roles bien definidos en dos grupos: cerdos y gallinas. Se los llama así porque el grupo de roles gallina se los toma en cuenta en el desarrollo del proyecto pero no pueden distorsionar el mismo.

2.9.1.5.1 Roles “Cerdo”

- **Product Owner:** es el cliente, el dueño del producto encargado de verificar que el trabajo del equipo de trabajo se esté cumpliendo correctamente.
- **ScrumMaster (o Facilitador):** no es el líder del equipo. Se asegura de que el proceso Scrum mejores prácticas y reglas se cumplan a cabalidad.
- **ScrumTeam o Equipo:** quien debe entregar el producto funcional. Por lo general es un equipo de hasta 9 personas con distintas habilidades necesarias para cumplir con el trabajo (diseñador, desarrollador, etc).

2.9.1.5.2 Roles "Gallina".

- **Usuarios:** destinatario final del producto a desarrollar
- **Stakeholders (Clientes, Proveedores, Inversores):** son quienes financian el proyecto y los principales beneficiados con el proyecto.
- **Managers:** encargados de establecer el ambiente para el desarrollo del producto.

2.9.1.6 Reuniones en Scrum

2.9.1.6.1 Daily Scrum

Cada día dentro de un sprint, se realiza una reunión para averiguar el estado del proyecto.

2.9.1.6.2 Scrum de Scrum

Después del "Daily Scrum", reuniones enfocadas en discutir el trabajo de cada uno de los equipos de una área específica.

2.9.1.6.3 Reunión de planificación del sprint (Sprint Planning Meeting)

Se realice esta reunión al inicio de cada sprint (entre 15 o 30 días) en donde se selecciona el trabajo que se completará durante el actual sprint

Al final del ciclo Sprint, dos reuniones se llevaran a cabo: la "Reunión de Revisión del Sprint" y la "Retrospectiva del Sprint"

2.9.1.6.4 Reunión de Revisión del Sprint (Sprint Review Meeting)

Al final del ciclo Sprint, se realiza esta reunión encargada de mostrar el trabajo que se completó y el que no

2.9.1.6.5 Retrospectiva del Sprint (Sprint Retrospective)

De igual manera la final de cada sprint, se lleva a cabo esta reunión que se encarga de retroalimentarse del sprint recién superado para entrar en un proceso de mejora continua basado en la experiencia obtenida.

2.9.1.7 Sprint

Se define como un periodo entre 15 a 30 días en la que el equipo crea un prototipo de software utilizable que cumpla con los requisitos definidos en el sprint. Los requisitos que forman parte de cada sprint vienen del Product Backlog, Los elementos del Product Backlog que forman parte del sprint son determinados en la reunión de Sprint Planning donde el Product Owner (Cliente) especifica los requisitos del Product Backlog que desea ver completados al final del sprint y los da a conocer al equipo de trabajo; el equipo entonces determina que cantidad del trabajo estipulado

pueden comprometerse a cumplir. Una vez determinado el trabajo a realizar, los requisitos definidos en el sprint no pueden cambiar.

2.9.1.8 Documentos

2.9.1.8.1 Product Backlog

Es un documento que contiene los requerimientos más genéricos definidos por el cliente y están priorizados en orden de su valor de retorno de inversión o ROI.

2.9.1.8.2 Sprint Backlog

Es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint.

2.9.1.8.3 Burn down

Es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint.

2.9.1.9 Beneficios

Los principales beneficios que proporciona Scrum son:

- Entrega mensual (o quincenal) de resultados mostrando un prototipo funcional de los requerimientos estipulados en el sprint
- Flexibilidad de adaptarse q nuevos requerimientos del cliente
- Mayor control de los riesgos del proyecto al ser entregas incrementales
- Productividad y calidad.
- Alto grado de comunicación entre el cliente y el equipo de desarrollo.
- Equipo motivado.

2.9.1.10 El proceso

En Scrum un proyecto se ejecuta en varios Sprints que contiene un cierto número de requisitos tomados del Product Backlog y estos duran entre dos a cuatro semanas. Cada sprint finaliza con un resultado funcional del proyecto (prototipo) y por cada nueva iteración o sprint se van aumentando funcionalidades hasta llegar al producto final. Ver figura 2.14

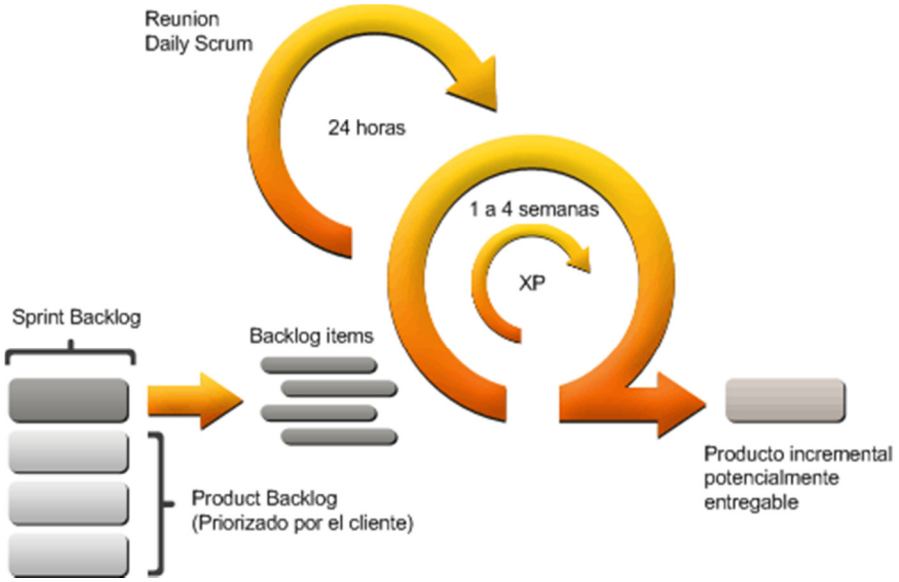

Figura 2.14: Ciclo de Scrum

2.9.2 XP o programación extrema

2.9.2.1 Definición

La programación extrema o extreme programming (XP) es una metodología ágil de desarrollo de software formulado por Kent Beck basado en conjunto de buenas prácticas y valores que definen como llevar el proceso de desarrollo. Pone mucho énfasis en la adaptabilidad del software antes que en la previsibilidad. Es una metodología incremental e iterativo al igual que Scrum pero más enfocada como se debe desarrollar el software (Fase de desarrollo).

2.9.2.2 Valores de XP

Los cinco valores se detallan a continuación:

- **Comunicación:** alto grado de comunicación entre el cliente y el equipo de desarrollo.
- **Simplicidad:**código sencillo, fácil de mantener, auto descriptivo, bien documentado.
- **Retroalimentación:** de cada iteración el cliente y el equipo llegan a conclusiones que afectan directamente a la otra iteración.
- **Valentía:**asumir los problemas y afróntalos lo más pronto posible. Tener la sencillez de aceptar equivocaciones y presentar soluciones.
- **Respeto:**el equipo debe trabajar como uno, sin hacer decisiones repentinas de manera independiente o menospreciar a alguien del equipo.

2.9.2.3 Características

Las características fundamentales del método son:

- **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- **Pruebas unitarias** continuas, frecuentemente repetidas y automatizadas.
- **Programación en parejas:**porqueel código puede ser revisado y discutido mientras se escribe.
- **Frecuente integración del equipo de programación con el cliente o usuario**. Un representante del cliente debería trabajar junto al equipo de desarrollo.
- **Corrección de todos los errores**: antes de añadir nueva funcionalidad
- **Hacer entregas frecuentes.**
- **Refactorización del código:**rescribir ciertas partes del código que no sean legibles o su facilidad para mantenerlo este comprometido.
- **Propiedad del código compartida:**cada parea trabaja en distintos módulos del proyecto haciendo que el código sea de dominio colectivo.
- **Simplicidad en el código:**se debe codificar de la manera más simple posible.

2.9.2.4 Actividades en XP

Las actividades son las siguientes:

- **Codificación:**la parte más importante de XP.
- **Pruebas:**se debe probar continuamente el código generado.
- **Escuchar:**escuchar los requisitos del cliente acerca del sistema a crear.
- **Diseño:**crear una estructura del diseño para evitar problemas.

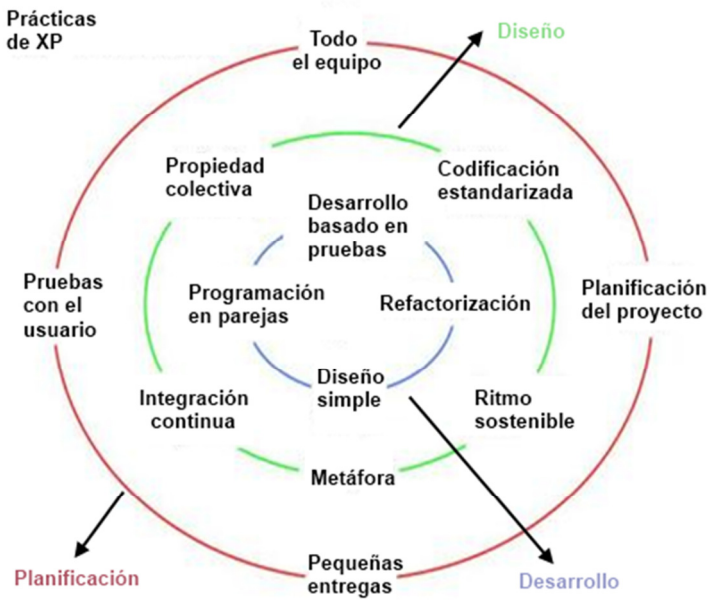

Figura 2.15: Ciclo de XP

2.9.2.5 Fases de XP[55]

Existen diversas prácticas inherentes al desarrollo de software.

2.9.2.5.1 Planificación.

XP plantea la planificación como un permanente dialogo entre el cliente y las personas responsables de crear el proyecto.

2.9.2.5.1.1 Pequeñas versiones.

Se debe entregar versiones funcionales del aplicativo cada 2 o 4 semanas Cada versión debe de ser tan pequeña como fuera posible, conteniendo solo los requisitosdefinidos como más importantes por el cliente.

2.9.2.5.2 Diseño

2.9.2.5.2.1 Metáfora.

Una metáfora es una forma sencilla de explicar lo que el sistema realiza y permite que cualquier persona entienda el objetivo del aplicativo

[55] Fases de XP: http://www.willydev.net/descargas/prev/ExplicaXP.pdf

2.9.2.5.2.2 Diseño sencillo.

El diseño debe ser lo más simple posible. Se puede utilizar UML para modelar el sistema.

2.9.2.5.3 Desarrollo.

2.9.2.5.3.1 Recodificación.

Se debe intentar siempre hacer que programa sea más simple y eficiente sin por ello perder funcionalidad, a este proceso se ledenomina refactorizar (refactoring).

2.9.2.5.3.2 Programación por parejas.

El código es escritor por dos personas frente a una computadora, mientras el uno codifica el otro tiene el papel de verificar si el código es correcto.

2.9.2.5.3.3 Propiedad colectiva.

Significa que ningún miembro del equipo es dueño de parte del código y que todos los miembros del equipo deben saber sobre todo el código del aplicativo.

2.9.2.5.3.4 Integración continúa.

Al final del día se debe integrar los cambios realizados por el equipo y realizar pruebas sobre el sistema ya integrado.

2.9.2.5.3.5 40 Horas semanales.

XP especifica que nunca se deben pasar de 2 semanas seguidas realizando horas extras.

2.9.2.5.3.6 Cliente In-situ.

El cliente debe estar junto al equipo de desarrollo para poder establecer una comunicación directa y permita solventar dudas del equipo lo más pronto posible.

2.9.2.5.3.7 Estándares de codificación.

Se debe establecer un manual que contenga los estándares de codificaciónque se van a utilizar a lo largo del proyecto y que todos los miembros del equipo deben conocer y aplicar.

2.9.2.5.4 Pruebas

No debe existir ninguna característica en el programa que no haya sido probada, losprogramadores escriben pruebas para chequear el correcto funcionamiento del programa, losclientes realizan pruebas funcionales. El resultado un programa más seguro que conformepasa el tiempo es capaz de aceptar nuevos cambios.

2.9.2.6 Ventajas y desventajas

Ventajas:

- Programación mejor organizada.
- Menor ocurrencia de errores.
- Motivación del programador.

Desventajas:

- Recomendado para proyectos a corto plazo.
- Altos costos en caso de fallar

2.9.2.7 Beneficios

Algunos son:

- El cliente tiene el control sobre lo que se está desarrollando y se va a desarrollar basado en las prioridades que el mismo establece.
- Se hacen pruebas continuas durante el transcurso de todo el proyecto.
- XP se utiliza mejor en ambiente donde los requerimientos cambia rápidamente.

2.9.3 UML Lenguaje Unificado de Modelado

2.9.3.1 Definición

Lenguaje gráfico de modelado de sistemas de software respaldado por la OMG [56] que permite visualizar, especificar, construir y documentar un sistema. Ofrece un estándar para poder describir el plano de un aplicativo, incluyendo procesos de negocio, funciones del aplicativo además de aspectos concretos como expresiones de lenguajes de programación, base de datos entre otros.

2.9.3.2 Aplicaciones

UMLpermite visualiza los planos de un sistema de manera estándar, incluyendo elementos como:

- Actividades
- Actores
- Procesos de negocio
- Esquemas de base de datos
- Componentes lógicos
- Instrucciones del lenguaje de programación
- Componentes reusables de software

2.9.3.3 Modelado

Un modelo UML es aquel que contiene documentación además de los diagramas y el conjunto de diagramas del sistema. Un diagrama en cambio es una representación gráfica parcial del modelo del sistema.

Los diagramas UML pueden representan dos visiones distintas del modelo del sistema:

- Vista estática (o estructural): toma en cuenta solo la estructura estática del Sistema.
- Vista dinámica (o de comportamiento): toma en cuenta el comportamiento dinámico del Sistema.

[56] OMG (Object Management Group): organización sin ánimo de lucro dedicada al establecimiento de estándares de tecnologías orientadas a objetos, tales como UML, XML, CORBA.

2.9.3.4 Diagramas de UML

UML 2.2 define 14 tipos de diagramas divide en 2 categorías. 7 diagramas que forman parte de la vista estructural del sistema, y los otros 7 la vista dinámica del sistema. Estos diagramas pueden ser categorizados jerárquicamente como se muestra en la Figura 2.16:

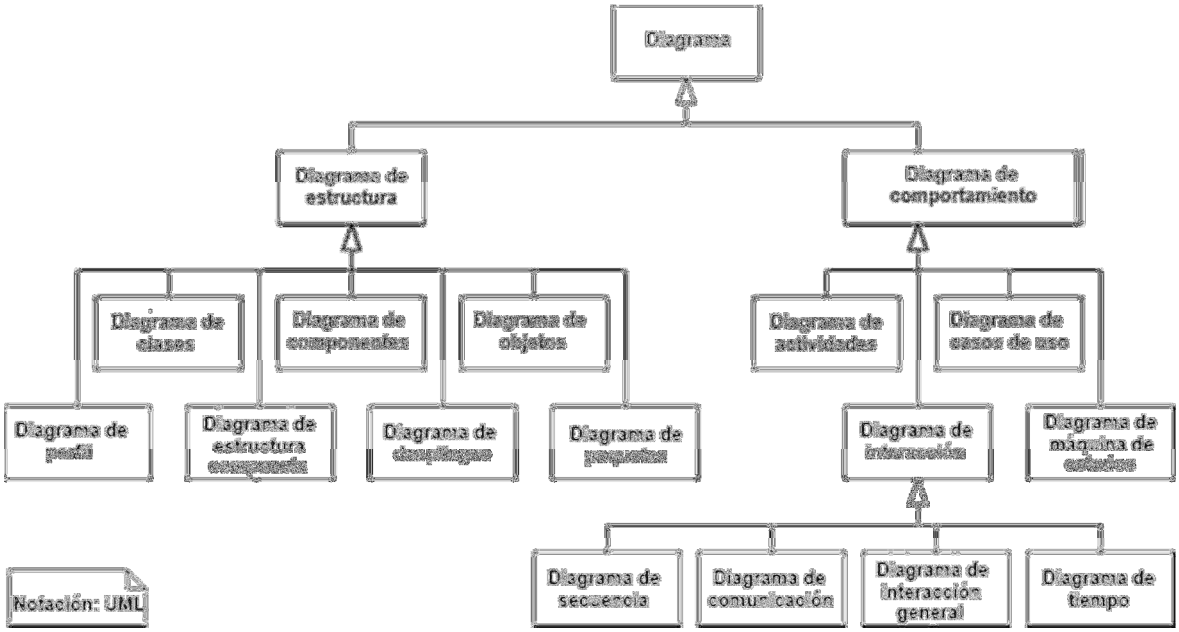

Figura 2.16: Diagramas UML[57]

2.9.3.4.1 Diagramas estructurales[58]

Los diagramas estructurales tratan de representan la parte estática de un sistema de software y son usados extensamente para la documentación de arquitectura del sistema.

- Diagrama de clases.
- Diagrama de componentes.
- Diagrama de estructura compuesta.
- Diagrama de despliegue.
- Diagrama de objetos.
- Diagrama de paquetes.
- Diagrama de perfiles

[57] Imagen Diagramas UML: http://en.wikipedia.org/wiki/Unified_Modeling_Language
[58] Diagramas estructurales: http://en.wikipedia.org/wiki/Unified_Modeling_Language

2.9.3.4.2 Diagramas de comportamiento[59]

Estos diagramas tratan de representan la parte dinámica de un sistema, su comportamiento, lo que debe suceder en el sistema. Son ampliamente utilizados para describir cómo funciona un sistema de software.

- Diagrama de actividades.
- Diagrama de estados finitos.
- Diagramas de casos de uso.

2.9.3.4.3 Los diagramas de interacción[60]

Un subconjunto de los diagramas de comportamiento, que tratan de representar el flujo de control de los datos en el sistema modelado

- Diagrama de Comunicación.
- Diagrama general de interacción.
- Diagrama de secuencia.
- Diagrama de tiempo.

[59] Diagramas de comportamiento: http://en.wikipedia.org/wiki/Unified_Modeling_Language
[60] Diagramas de interacción: http://en.wikipedia.org/wiki/Unified_Modeling_Language

2.9.4 SCRUMP y XP con UML

2.9.4.1 Definición

La unión de estos tres componentes: SCRUM como metodología de manejo de proyectos, XP como metodología de desarrollo de software y UML como estándar de lenguaje de modelado ha permitido resolver en muchas empresas problemas de costos, calidad deficiente y tiempos de entrega difíciles de cumplir.Siempre y cuando se logre implementar de la manera adecuada.

Estas metodologías por lo general se utilizaban por separado pero lastimosamente ninguna de las dos es suficientemente completa como para aplicarla directamente en un proyecto de desarrollo de software real.

UML se utiliza sobretodo en XP como una de las buenas prácticas para el diseño del aplicativo a desarrollar.

2.9.4.2 Características

Scrum y XP por separado representan metodologías útiles pero no muy completas para un ciclo de desarrollo de software por lo cual al combinarlas se utilizan cada una de las fortalezas de estos para mejorar esta realidad.

En el caso de SCRUM es excelente para la planificación y administración del proyecto y la comunicación transparente entre el equipo de trabajo y el cliente pero no para el diseño, mejores prácticas para codificar, ni el cómo probar el aplicativo a desarrollar.

En cambio en XP no existe una planificación definida lo que puede llevar al caos en proyectos grandes pero en cambio es muy bueno en establecer buenas prácticas para diseño, codificación y pruebas del aplicativo a desarrollar

Basándonos en lo escrito anteriormente se podría concluir que SCRUM aporta con la planificación y administración basadas en los siguientes componentes:

- Producto Backlog
- Sprint Backlog
- Sprint

- Daily Scrum
- Producto incremental

Mientras que XP aporta con las buenas prácticas para el desarrollo del aplicativo como:

- Diseño simple
- Pruebas
- Refactorización
- Programación en parejas
- Propiedad colectiva
- Integración continua
- Estándares para codificación

2.9.4.3 Ventajas

XP aporta a Scrum lo indicado en la Figura 2.17.

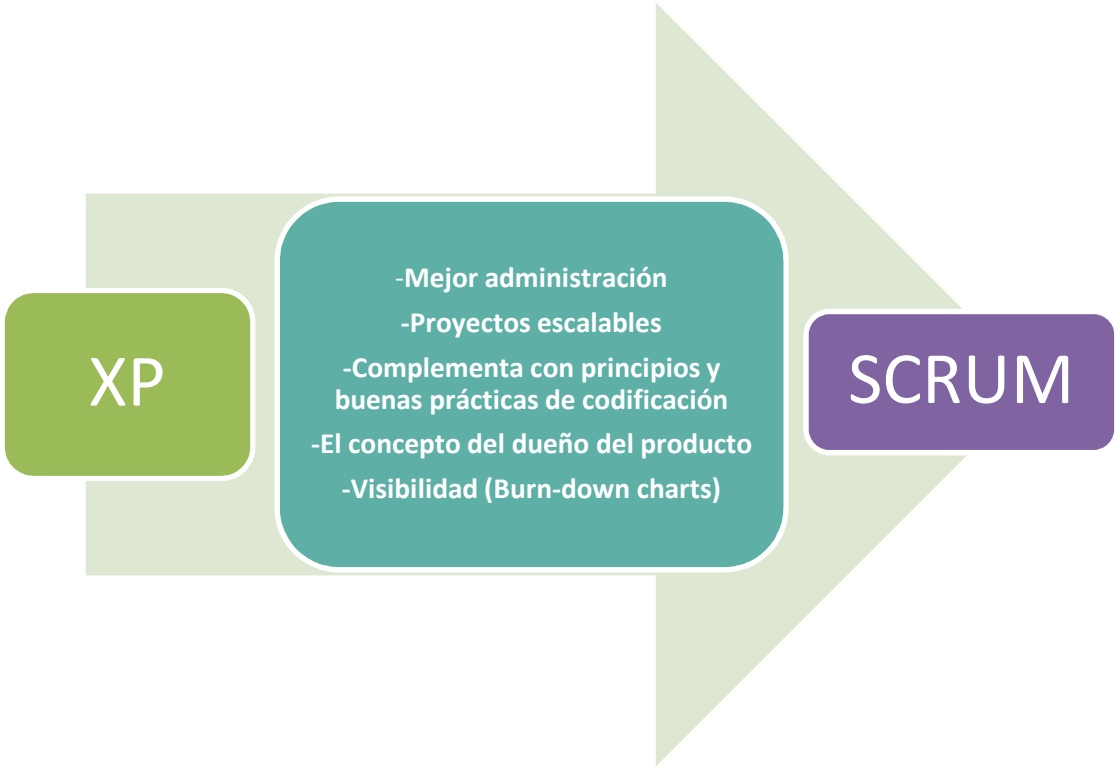

Figura 2.17: Aporte de XP a SCRUM

Scrum aporta a XP lo indicado en la Figura 2.18.

Figura 2.18: Aporte de SCRUM a XP

2.10 El API de búsqueda de Google (Google Custom Search API)

2.10.1 Definición

Es un API desarrollado por Google en el 2006 que permite crear un motor de búsqueda personalizada adaptado a las necesidades del usuario.

2.10.2 Que es la búsqueda personalizada

Con el motor de búsqueda personalizado de Google, se puede crear un motor de búsqueda para un sitio o sitios web. Se puedes ajustar la clasificación, el diseño y el funcionamiento de los resultados de búsqueda.

Se puede crear un motor de búsqueda que se centre en un tema específico, identificando los sitios en los quedebe buscar, a cuales debe dar prioridad y cualesignorar.

2.10.3 Creación de un motor de búsqueda[61]

Para crear un motor de búsqueda personalizado, se realizan las siguientes acciones:

- Utilizar una cuenta de Google para acceder al asistente de la Búsqueda personalizada.
- Rellenar información básica sobre el motor de búsqueda, incluido el nombre, la descripción, el idioma del motor de búsqueda y opcionalmente las palabras clave del motor de búsqueda.
- En la sección **¿Qué quiere buscar?**, indicar a la Búsqueda personalizada la amplitud de la cobertura de la búsqueda.
- En la sección **Seleccione algunos sitios**, indicar a la Búsqueda personalizada en qué sitios web o páginas web debe buscar.
- Seleccionar la edición estándar gratuita.
- Leer los Términos de servicio y, si se está de acuerdo con ellos, seleccionar la casilla de verificación correspondiente.
- Click en el botón **Siguiente**
- Seleccionar la casilla de verificación **Enviar un mensaje de correo electrónico de confirmación**.

[61] Creación de un motor de búsqueda: http://code.google.com/intl/es/apis/customsearch/docs/start.html

- Clic en **Finalizar**.

2.10.4 Panel de control del motor de búsqueda

Una vez que se haya creado un motor de búsqueda personalizado, se puede utilizar el panel de control para modificarlo utilizando alguna de las siguientes fichas:

- **Información básica:**permite definir información pública y temas generales para el motor de búsqueda.
- **Sitios web:**permite indicar a la Búsqueda personalizada de Google en que sitios buscar.
- **Indexación:**ayuda a mejorar la amplitud de la cobertura y la actualización de los resultados de búsqueda del motor de búsqueda personalizado
- Más datos: permite clasificar sitios por temas
- **Diseño:** permite cambiar el diseño del cuadro de búsqueda y personalizar el estilo de la página de resultados
- **Código:** permite copiar el código e insertarlo en páginas web
- **Colaboración:** permite invitar a colaboradores para que ayuden a ajustar el motor de búsqueda
- **Obtener ingresos:**vincular a una cuenta de Google AdSense
- **Cuenta empresarial:**permite crear motores de búsqueda que no incluyan anuncios
- **Opciones avanzadas:** permite modificar el formato en el que se reciben los resultados
- **Vista previa:** prueba los cambios realizados en el motor de búsqueda

2.10.5 JSON/ATOM Custom Search API

El JSON/Atom Custom Search API permite desarrollar sitios web y programas que recuperen y muestren los resultados del API de búsqueda de Google Custom Search programáticamente. Selo utiliza mediante peticiones RESTful, que devuelven la respuesta en formato JSON o Atom. Este API requiere un clave para su uso. Este se puede obtener desde la consola de API's de Google. El API proporciona 100 consultas diarias de forma

gratuita; pasado el límite tiene un costo de 5 dólares por 1000 hasta 10000 consultas al día[62]. Actualmente el API se encuentra en la versión 1 y en fase Labs (Beta)

2.10.5.1 Requisitos previos

Los requisitos previo la utilización del API son:

- Tener una cuenta de Google
- Crear un motor de búsqueda personalizado
- Obtener una clave para el API (API Key)

Una vez obtenido el API Key se debe ir al API's Console y luego al panel de API Access; en Simple API Access se encuentra el Key, este se coloca en el url para cada petición al API sin ser codificado.

2.10.5.2 Características[63]

El API se basa principalmente en dos protocolos:

- **Google WebSearch Protocolo (XML):** JSON / Atom Custom Search API proporciona un subconjunto de la funcionalidad proporcionada por la API XML, pero devuelve los datos en JSON o Atom.
- **Especificación OpenSearch 1.1:** Esta API utiliza la especificación OpenSearch para describir el motor de búsqueda y proporcionar datos sobre los resultados. La especificación OpenSearch está en XML.

2.10.5.3 Precios

Cuota gratuita: el uso es gratuito para todos los usuarios, hasta 100 consultas por día.

Uso pagado: cualquier uso más allá de la cuota de uso libre no funcionará si no se ha inscrito para facturación. Una vez habilitadala facturación, se seguirá recibiendo 100 consultas gratuitas por día. Sin embargo, se cobrará todas las solicitudes adicionales por un valor de $5 por cada 1.000 consultas hasta 10.000 consultas por día.

[62] Costos del API de Google: https://code.google.com/intl/es-ES/apis/customsearch/v1/overview.html#Pricing
[63] Características del API de Google: http://code.google.com/intl/es-ES/apis/customsearch/v1/getting_started.html

2.10.5.4 Operaciones del API

Solo hay un método en JSON/Atom Custom Search API que se puede invocar como se muestran en la Tabla 2.8

Tabla 2.8: Operaciones permitidas en el API de Google[64]

Operación	Descripción	Mapeo REST HTTP
search	Devuelve los resultados de la búsqueda solicitada desde el motor de búsqueda personalizado especificado.	GET

2.10.5.5 Modelo de datos del API[65]

El resultado de una consulta de búsqueda al API es un objeto JSON o Atom, que incluye tres tipos de datos:

- Metadatos que describen la búsqueda solicitada
- Metadatos que describen el motor de búsqueda personalizada
- Resultados de la búsqueda

El modelo de datos se basa en la especificación OpenSearch 1.1. El API define tres propiedades y dos roles de consulta personalizada además de las especificadas en el estándar, estas son:

Propiedades personalizadas

- cx: el identificador del motor de búsqueda personalizada.
- cref: una URL que apunta a la definición de un motor de búsqueda personalizada.
- safe: una descripción del nivel de búsqueda segura para el filtrado de los resultados obtenidos (filtrado pornográfico).

Funciones personalizadasde consulta

- nextPage: una función que indica que la consulta se puede utilizar para acceder a la siguiente página de los resultados, si los hubiere.

[64] Operaciones del API de Google: http://code.google.com/intl/es-ES/apis/customsearch/v1/getting_started.html
[65] Modelo de datos del API de Google: http://code.google.com/intl/es-ES/apis/customsearch/v1/getting_started.html

- previousPage: Un función que indica que la consulta se puede utilizar para acceder a la página anterior de los resultados, si los hubiere.

2.10.5.6 Formato de Datos

El API puede devolver resultados en dos formatos JSON o ATOM. JSON es el formato de datos por defecto, para obtener resultados en formato Atom se coloca el parámetro de consulta alt=atom.

2.10.5.7 Estilo de llamadas[66]

Hay dos formas de invocar el API:

- Utilizando directamente REST, ejemplo:

```
GET
https://www.googleapis.com/customsearch/v1?key=KEY&cx=017576662512468
239146:omuauf_lfve&q=lectures
```

- Utilizando REST desde JavaScript (no es necesario código de servidor), ejemplo:

```
<script
src="https://www.googleapis.com/customsearch/v1?key=KEY&cx=017576662512468239
146:omuauf_lfve&q=cars&callback=hndlr">
```

2.10.5.8 Trabajando con los resultados de búsqueda

Para obtener los resultados de búsqueda del API se envía una solicitud HTTP GET a su URI. El URI tiene el siguiente formato:https://www.googleapis.com/customsearch/v1?**parameters**

Se requieren tres parámetros de consulta con cada solicitud de búsqueda:

- **API Key:** identificala aplicación y al desarrollador.

[66] Estilo de llamadas al API de Google: http://code.google.com/intl/es-ES/apis/customsearch/v1/getting_started.html

- **Identificador del Motor de búsqueda personalizado:**se especifica el motor de búsqueda personalizado que se va a utilizar el API en los parámetros **cx** o **cref**donde **cx**se utiliza para un motor de búsqueda creado con la página de búsqueda personalizada de Google y **cref** para un motor de búsqueda, con vínculos personalizados.Si ambos se especifican, **cx** se utiliza.
- **Consulta de búsqueda**: se coloca la consulta en el parámetro**q**.

El resto de parámetros de la consulta son opcionales, ejemplo:

GET
https://www.googleapis.com/customsearch/v1?**key**=KEY&**cx**=013036536707430787589:_pqjad5hr1a&**q**=flowers&**alt**=json

Si la solicitud tiene éxito, el servidor responde con un código HTTP200 y los datos de respuesta. (Ver Anexo 6)

2.10.5.9 Parámetros de consulta

Los parámetros de consulta que puede utilizar con el API JSON / Atom Custom Search se resumen en esta sección. Todos los valores de los parámetros deben ser codificados.

2.10.5.9.1 Parámetros de consulta estándar

Los parámetros de consulta que se pueden utilizar con el API JSON / Atom Custom Search se muestran en la tabla 2.9.

2.10.5.9.1.1 Consideraciones

- El parámetro del API Key es necesario con cada solicitud, a menos que se utilice como mecanismo de autenticación OAuth 2.0. Si es el caso se debe enviar un token de autorización con cada solicitudpara marcalas como (autenticadas).

Todos los parámetros son opcionales, excepto cuando se indique. La Tabla 2.9 los muestra

Tabla 2.9: Parámetros de consulta estándar del API de Google[67]

Parámetro	Significado	Notas
`alt`	Formato de datos de la respuesta	Valores validos: json, atom Valor por defecto: json
`callback`	Function Callback	Nombre de la función Javascript de callback que maneja la respuesta.
`fields`	Selector especificando un subconjunto de campos incluidos en la respuesta	Respuestas parciales
`key`	API Key (requerido)	Requerido a menos que se provea de un token OAuth 2.0
`access_token`	Token OAuth 2.0 para el usuario actual	El token de autorización para el mecanismo de autenticación OAuth 2.0
`prettyPrint`	Devuelve la respuesta con indentaciones y saltos de linea	Devuelve la respuesta en formato legible por humanos si está en true Valor por defecto en false
`userIp`	Dirección IP del sitio donde se originó la petición	Permite reforzar cuotas limite por usuario

2.10.5.9.2 Parámetros específicos de consulta

Parámetros específicos que utiliza únicamente el API JSON / Atom Custom Search, estos se muestran en la Tabla 2.10.

Tabla 2.10: Parámetros de consulta específicos del API de Google[68]

Parámetro	Significado	Notas
cref	El url de un motor de	El url de un motor de búsqueda

[67] Parámetros de consulta opcionales del API de Google: http://code.google.com/intl/es-ES/apis/customsearch/v1/using_rest.html
[68] Parámetros de consulta específicos del API de Google: http://code.google.com/intl/es-ES/apis/customsearch/v1/using_rest.html

	búsqueda personalizado enlazado	personalizado enlazado a utilizar en esta petición
cx	El id de un motor de búsqueda personalizado para el ámbito de esta consulta	El id de un motor de búsqueda personalizado a utilizar en esta petición
lr	La restricción del lenguaje para los resultados de la búsqueda	Se puede restringir los documentos escritos en un lenguaje en particular
num	Número de resultados de la búsqueda a devolver	Se puede especificar cuantos resultados debe devolver la búsqueda actual Se permite valores entre 1 a 10 inclusive Si num no es usado, el valor por defecto es 10
q	Consulta	La expresión de búsqueda, para especificar un tipo de documento se puede utilizar q=Google+**filetype%3Atipo**, donde tipo puede ser: pdf, doc, ppt, rtf, swf, ps, xls, txt, entre otros
safe	Nivel de seguridad de la búsqueda	Se puede especificar el nivel de seguridad de la búsqueda en tres niveles: high, médium, off Valor por defecto es off
start	El índice del primer resultado a devolver	Se puede establecer el índice de comienzo de la primer resultado devuelto por la búsqueda Los valores permitidos son entre 1 y 101 – num Valor por defecto es 1
filter	Controla si se activa o no el filtro de contenido duplicado	filter=0 – Desactiva el filtro de contenido duplicado filter=1 - Activa el filtro de contenido duplicado

cr	Restricción de país	Restringe la búsqueda de resultados a documentos originados en una país en particular, ejemplo: countryEC - Ecuador
gl	Geolocalización del usuario final	El valor del parámetro gl es dos letras del código del país, permite aumentar los resultados relevantes según el país de origen que coincida con el parámetro, ejemplo: ec - Ecuador

Algunos de los lenguajes permitidos para el parámetro lr son los mostrados en la Tabla 2.11

Tabla 2.11: Lenguajes que pueden ser utilizados por el API[69]

Lenguaje	Valor del parámetro lr
Ingles	lang_en
Español	lang_es
Aleman	lang_de
Portugues	lang_pt
Italiano	lang_it
Frances	lang_fr

2.10.5.9.3 Campos de respuesta

Los campos de respuesta más utilizados en el API de Google Custom Search API JSON/Atom son los que se muestran en la Tabla 2.12.

Tabla 2.12: Lenguajes que pueden ser utilizados por el API[70]

Parámetro	Significado
items.link	El url completo al cual apunta el resultado de la búsqueda
items.snippet	El resumen del resultado de la búsqueda

[69] Lenguaje permitidos por el API de Google: http://code.google.com/intl/en/apis/customsearch/docs/xml_results.html#languageCollections
[70] Campos de respuesta del API de Google: http://code.google.com/intl/es-AR/apis/customsearch/v1/reference.html

2.11 El API de búsqueda de Yahoo (Yahoo Search Boss API)

2.11.1 Definición

BOSS (Build your Own Search Service) es una plataforma de servicios de datos y búsqueda que permite crear motores de búsqueda personalizados. BOSS permite utilizar la infraestructura de búsqueda de Yahoo! a terceros para construir productos de búsqueda usando ideas propias, datos, tecnologías, etc.

BOSS es un servicio que proporciona acceso mediante un servicio web RESTful a la web, imágenes y noticias, blogs, entre otrosa un costo basado en el uso. El servicio también permite acceder labúsqueda de publicidad, requiere autorización de Yahoo.

2.11.2 Características[71]

Algunas de las características de BOSS son:

- API RESTful para la web, imágenes, noticias y blogs con soporte para XML y JSON
- Número ilimitado de consultas por día
- No hay restricciones en la presentación.
- Autenticación OAuth
- Está permitido re-ordenamiento de resultados
- Simple uso con precio basado en el tipo de consulta y el uso
- Está permitido la mezcla de contenido propietario y de búsqueda de Yahoo!
- Acceso a un conjunto de datos Limitedweb
- Requiere atribución de la marca Yahoo!
- Seguridad de autenticación del usuario
- Publicidad Yahoo! Search
- Extensivo soporte y documentación

[71] Características de BOSS: http://developer.yahoo.com/search/boss/

2.11.3 Precios

La Tabla 2.13 muestra los precios por número de peticiones del API de búsqueda Yahoo BOSS API

Tabla 2.13: Costos por tipo de fuentes de búsqueda del API de Yahoo[72]

Tipo de Consulta	Definición	Precio/1000 consultas (USD)	Precio por 1000 consultas en unidades BOSS	Máximo # Resultados/ Consulta
Full Web	Solo resultados de búsqueda en la web	0.80	8	50
LimitedWeb	Resultados de un índice limitado y lento de actualización	0.40	4	50
Images Only	Solo imágenes	0.30	3	35
News Only	Solo noticias	0.10	1	50
Spelling Only	Solo ortografía	0.10	1	1
Blogs Search (beta)	Búsqueda en blogs (beta)	0.10	1	20

Cada unidad BOSS equivale a 125 consultas a un precio de 0.10 centavos si se utiliza el tipo de consulta Full Web

2.11.4 Modelo de autorización con OAuth[73]

Se utiliza OAuth para validar y acceder al API de BOSS.

OAuth permiteun acceso seguro a la web, imágenes y contenido de noticias de Yahoo! que están almacenados en los servidores de Yahoo! utilizando el ID de aplicación BOSS y la clave del API de Yahoo!para comprobar sus privilegios autorizados de acceso y permitir la correcta facturación.

[72] Precios Yahoo BOSS API: http://info.yahoo.com/legal/us/yahoo/search/bosspricing/details.html
[73] Autorización OAuth en BOSS API: http://developer.yahoo.com/search/boss/boss_api_guide/oauth_model.html

La autenticación para consultas a BOSS requiere información OAuth en la cabecera HTTP o por medio de parámetros en una petición GET. Existen seis elementos que se requieren para la autorización:

- oauth_version = 1.0 - El estándar de OAuth soportado por BOSS.
- oauth_timestamp = - La marca de tiempo se expresa en el número de segundos desde Enero 1, 1970 00:00:00 GMT.
- oauth_nonce - es una cadena aleatoria, única generada por todas las solicitudes de una marca de tiempo específico. Esto ayuda a verificar que la solicitud nunca se ha hecho antes y ayuda a prevenir los ataques de repetición cuando las peticiones se realizan a través de un canal no seguro
- oauth_consumer_key = - obtenidos a partir de YDN durante el proceso de registro del proyecto de BOSS. Esto es único para el desarrollador.
- oauth_signature_method = HMAC-SHA1 - (algoritmo específico utilizado para llamadas a OAuth de BOSS).
- oauth_signature - puede ser generada por una biblioteca de OAuth.

2.11.5 Llamadas a BOSS API utilizando el método GET

Hay dos tipos de llamadas al API que pueden ser utilizados, el primero es un servicio múltiple con palabras clave y el otro es un servicio múltiple con múltiples palabras clave. Ej.

- http://yboss.yahooapis.com/ysearch/{**service**,*}?q={**keywords**}
- http://yboss.yahooapis.com/ysearch/{**service**,*}?service1.q={**keywords**}&service2.q={**keywords**}

2.11.5.1 Servicios

En la tabla 2.14 se muestran los diferentes servicios que Yahoo BOSS API soporta:

Tabla 2.14: Tipos de fuentes de búsqueda que soporta el API de Yahoo[74]

Service	Descripción
web	Resultados de índices de búsqueda web con url, título y un resumen de datos
limitedweb	Resultados web limitados
images	Búsqueda de imágenes en Yahoo y Flickr
news	Búsqueda de noticias de los últimos 30 días
blogs	Búsqueda en blogs (beta)
ads	Publicidad

2.11.5.2 Reglas básicas

Para los resultados de web, limitedweb y news se devuelven 50 resultados por consulta.

Toda consulta debe codificar la URL para caracteres reservados y parámetros antes de enviar la llamada al API BOSS.

Ejemplos:

Buscando basquet a través de los datos web:

- http://yboss.yahooapis.com/ysearch/web?**q**=basquet

Buscando los servicios web, images, news con una consulta distinta

- http://yboss.yahooapis.com/ysearch/**web,images**?**web.q**=basquet&**images.q**= deporte

Buscando en servicios web, images, news con diferentes consultas y diferentes restricciones de sitios

[74] Servicios de búsqueda que soporta BOSS API: http://developer.yahoo.com/search/boss/boss_api_guide/v2_api_spec.html

- http://yboss.yahooapis.com/ysearch/**limitedweb,news**?**limitedweb.q**=google **&limitedweb.sites**=yahoo.com,google.com&**news.q**=deporte&**news.sites**=news.yahoo.com

Buscando en múltiples servicios con palabras clave

- http://yboss.yahooapis.com/ysearch/**limitedweb,image**?**q**=google&**format**=xml&count=2

Buscando en múltiples servicios con múltiples palabras clave

- http://yboss.yahooapis.com/ysearch/**news,limitedweb**?**news.q**=google&**limitedweb.q**=microsoft+google&**format**=xml&**count**=2

2.11.6 Servicio web BOSS

La sintaxis para el servicio web es la siguiente:

- http://yboss.yahooapis.com/ysearch/web?**q**={**keywords**}

Ejemplo de la respuesta: (Ver Anexo 9)

- http://yboss.yahooapis.com/ysearch/web?**q**=**yahoo**&**format**=**xml**

2.11.6.1 Argumentos opcionales de consulta

Estos se muestran en la Tabla 2.15.

Tabla 2.15: Argumentos opcionales que soporta el API de Yahoo[75]

Nombre	Descripción
filter	Filtra contenido. Ej.Filter=[-porn]
type	Específica el tipo de documento a devolver, se puede colocar una lista de tipos separados por comas; Los tipos son: html, text, pdf, xl, msword, ppt, msoffice, nonhtml. Se utiliza el signo – para ignorar un tipo
view=language	Identifica el lenguaje del documento a consultar. Los valores esperados son: ca,zh-hans, zh-hant, cs, da, nl, en ,fi, fr, he, hu, it, id, ja, ko, ms, no, pt, ro, ru, es, sv, tl, th, tr, vi y unknow

[75] Argumentos opcionales de consulta del API BOSS: http://developer.yahoo.com/search/boss/boss_api_guide/webv2_service.html

abstract	Devuelve un pequeño resume de alrededor de 300 caracteres sobre el contenido del sitio
title	Devuelve el título que fue parametrizadas en la consulta, ejemplo: q=presidente&title=obama
url	Devuelve el url que fue parametrizadas en la consulta, ejemplo: q=presidente&url=obama
style	style=raw permite devolver el resumen libre de HTML
format	Especifica el formato en el que devuelve los resultados json o xml
count	Especifica el número de resultados que se va a obtener de la consulta. Max 50 para todos excepto blogs 20 e imágenes 35
sites	Restringe los resultados de la búsqueda de BOSS a los sitios predefinidos por sites. Se puede utilizar el signo – para excluir sitios

2.11.6.2 Argumentos requeridos por el servicio web

Los argumentos del servicio requeridos web se muestran en la Tabla 2.16.

Tabla 2.16: Argumentos del servicio web que soporta el API de Yahoo

Argumento	Opcional	Descripción
service	no	Servicios web, limitedweb, images, news y blogs
market	no	Especifica que región (país) utilizar en la consulta por servicio
callback	no	Solo para formato json. Devuelve la respuesta de la consulta envuelto en el nombre de la función pasada

2.11.6.3 Campos de la respuesta

Estos se muestran en la tabla 2.17.

Tabla 2.17: Campos de respuesta del API de Yahoo[76]

Nombre	Descripción
count	Indica el número de resultados a mostrar por página.

[76] Campos de respuesta de Yahoo BOSS API: http://developer.yahoo.com/search/boss/boss_api_guide/webv2_response.html

start	El primer resultado numérico a mostrar
totalresults	El número de resultados en la página de búsqueda actual
abstract	Devuelve un pequeño resumen del contenido del sitio con las palabras clave sobresaltadas
title	Devuelve el título del resultado. Con las palabras clave sobresaltadas
url	Devuelve el url del resultado
clickurl	Devuelve un url navegacional
dispurl	Devuelve el url que empata con el documento del resultado de la consulta. Solo sirve para propósitos visuales
date	Devuelve la fecha de la última vez cuando paso por el crawler de Yahoo en el formato YYYY/MM/DD
language	Muestra el lenguaje con el que se realizó la búsqueda de resultados en el documento

2.11.6.4 Regiones y lenguajes soportados

Según el tipo de servicio se definen los lenguajes y regiones soportadas:

2.11.6.4.1 Web

El país que solo soporta el servicio web es el que se muestra en la Tabla 2.18

Tabla 2.18: Países que soportan solo fuente web del API de Yahoo[77]

País/Lenguaje	Mercado
Greece-Greek	el-gr

2.11.6.4.2 Web y News:

Los paísesque soportan fuentes web y noticias se muestran en la Tabla 2.19

Tabla 2.19: Países que soportan fuentes web y news del API de Yahoo[78]

País/Lenguaje	Mercado
Argentina-Spanish	es-at

[77] Solo servicio WEB del API BOSS:http://developer.yahoo.com/search/boss/boss_api_guide/supp_regions_lang.html

[78] Servicio Web y News: http://developer.yahoo.com/search/boss/boss_api_guide/supp_regions_lang.html

Austria – German	de-at
Australia – English	en-au
Brazil – Portuguese	pt-br
Canada - English	en-ca
Canada - French	fr-ca
Chile – Spanish	es-cl
Columbia - Spanish	es-co
Czech Republic	cs-cz
Denmark – Danish	da-dk
Finland – Finnish	fi-fi
France - French	fr-fr
Germany - German	de-de
Hong Kong* – Traditional Chinese	zh-hant-hk
Hungary - Hungarian	hu-hu
United Kingdom - English	en-gb
United States - English	en-us
United States - Spanish	es-us
Vietnam – Vietnamese	vi-vn

2.11.6.4.3 Blogs (Beta):

Los países que soportan la fuente de blogs se muestran en la Tabla 2.20

Tabla 2.20: Países que soportan fuentes blogs del API de Yahoo[79]

País/Lenguaje	Mercado
Hong Kong* – Traditional Chinese	zh-hant-hk
Korea	ko-kr
Spain – Spanish	es-es
Taiwan – Traditional Chinese	zh-hant-tw
Vietnam - Vietnamese	vi-vn

[79] Soporte para Blogs en el API BOSS:
http://developer.yahoo.com/search/boss/boss_api_guide/supp_regions_lang.html

2.12 El API de búsqueda de Bing (Bing Search API)

2.12.1 Definición

Es un API de búsqueda desarrollado por Microsoft que utiliza el motor de Bing para realizar las consultas. Actualmente en la versión 2.0, este posee características básicas como soporte de tipo de fuentes (servicios en Yahoo), datos fuertemente tipados, acceso ilimitado de tráfico y no tiene costo alguno. La versión 2.0 reemplaza al Bing API retirado 1.1b

2.12.2 Características

La versión 2 del API de Bing tiene las siguientes características:

- Conexión HTTP que puede proveer de resultados en los formatos XML o JSON
- Soporte de SOAP
- Habilidad para monetizar las aplicaciones de terceros con publicidad
- Completamente compatible con la interfaz RSS de OpenSearch que permite acceder a mucha de la información de las fuentes del API

2.12.3 Utilización del API

Se requiere una serie de pasos para poder realizar esta tarea, estos son:

- **Obtener un AppID:** el parámetro AppID es un valor que permite validar al API que la petición fue realizada por desarrollador registrado
- **Determinar los valores para los parámetros requeridos de la petición al API:** A parte del parámetro AppID se requieren dos parámetros más Query y Sources. Query es el texto a buscar con el API y Sources es uno o más valores separados por el signo + que indican el tipo de fuentes a utilizar en la petición; los tipos de fuentes se muestran en la Tabla 2.21:

Tabla 2.21: Tipos de fuentes de búsqueda que soporta el API de Bing[80]

Tipo de Fuente	Descripción	Ejemplo de consulta del parámetro
Web	Busca por contenido web	Sushi
Image	Buscar por imágenes en la web	Sushi
News	Busca noticias	Sushi
InstantAnswer	Busca en Encarta online	X*5 = 7 Convert 5 feet to meters
Spell	Busca en el diccionario de Encarta por sugerencia de ortografía	Cofee
PhoneBook	Busca entradas de una agenda telefónica	Sushi in miami
RelatedSearch	Devuelve cadenas de texto de consulta lo más similar a las realizadas	{inari sushi; sushi restaurant; California roll}
Ad	Devuelve publicidad incorporado en los resultados	Sushi
Translation	Traduce un término o un pequeño bloque de texto de un lenguaje a otro	Espero
Video	Busca por videos en la web	David Guetta

- **Trabajar los parámetros opcionales de la petición al API:** a parte de los tres parámetros requeridos se puede utilizar una serie de parámetros opcionales que permite refinar la petición. Cada tipo de fuente tiene sus parámetros opcionales, para el tipo de fuente web los parámetros son los indicados en la Tabla 2.22.

Tabla 2.22: Parámetros opcionales que soporta el API de Bing[81]

Nombre	Descripción
Market	Por defeco intenta determinar el parámetro basado en el IP de la petición, cookies y otros elementos. Representa el lenguaje y la

[80] Tipos de fuentes Bing API: http://www.bing.com/developers/s/API%20Basics.pdf
[81] Parámetros opcionales de petición del tipo de fuente web Bing API: http://msdn.microsoft.com/en-us/library/dd250845.aspx

<table>
<tr><td></td><td>región para la petición, ejemplo:

<table><tr><th>Market</th><th>Lenguaje – Ciudad/Region</th></tr><tr><td>en-US</td><td>English – United States</td></tr><tr><td>es-ES</td><td>Spanish – España</td></tr><tr><td>es-MX</td><td>Spanish – Mexico</td></tr><tr><td>es-XL</td><td>Spanish – Latin America</td></tr></table></td></tr>
<tr><td>Version</td><td>La versión del API de Bing a la que la petición va a ser enviada, ejemplo: 2.0, 2.1</td></tr>
<tr><td>Adult</td><td>El valor por defecto depende del market al que apunta, permite filtrado pornográfico en las búsquedas</td></tr>
<tr><td>Latitude</td><td>El componente de latitud de la ubicación de una petición, los valores validos se encuentran en el rango de -90 a 90 decimales</td></tr>
<tr><td>Longitude</td><td>El componente de longitud de la ubicación de una petición, los valores validos se encuentran en el rango de -180 a 180 decimales</td></tr>
<tr><td>Options</td><td>Opciones específicas para cualquier tipo de fuente.Permite establecer las opciones de DisableLocationDetection y EnableHighlighting</td></tr>
<tr><td>Count</td><td>Especifica el número de resultados obtenidos por petición al API entre los valores 1 a 50</td></tr>
<tr><td>Offset</td><td>Especifica el offset requerido, desde 0 para el punto de inicio del conjunto de resultados a ser devueltos por el API, los valores están entre 0 a 1000</td></tr>
<tr><td>FileType</td><td>Especifica el tipo de archivos que va a devolver el API
<table><tr><th>Web FileType</th><th>Valor</th></tr><tr><td>Microsoft Word Document</td><td>DOC</td></tr><tr><td>Autodesk Drawing File</td><td>DWF</td></tr><tr><td>Hypertext Markup Language</td><td>HTM</td></tr><tr><td>Hypertext Markup Language File</td><td>HTML</td></tr><tr><td>Adobe Acrobat Portable Document</td><td>PDF</td></tr><tr><td>Microsoft PowerPoint Presentaton</td><td>PPT</td></tr><tr><td>Microsoft Rich Text Format Document</td><td>RTF</td></tr></table></td></tr>
</table>

	Generic Text	TEXT	
	Generic Text File	TXT	
	Microsoft Excel Workbook	XLS	
WebRequest Options	Opciones específicas para el tipo de fuente web. Permite establecer las opciones de DisableHostCollapsing y DisableQueryAlterations		

- **Seleccionar un protocolo**[82]**:**Bing soporta los siguientes protocolos JSON, XML y SOAP.
    - **JSON:** es ideal para aplicaciones AJAX. La interfaz JSON puede devolver respuestas en 3 formatos diferentes: Raw, Callback y Function
    - **XML:** para trabajar con aplicaciones Silverlight/Flash o cualquier tipo de aplicación que soporte XML pero no soporte SOAP. La única limitante es la longitud máxima de la URL
    - **SOAP:** provee tipos fuertemente tipados, un modelo de objetos request/response y es capaz de enviar peticiones complejas sin limitación de la longitud máxima de la URL. Apropiado para aplicaciones desarrollados por lenguajes de nivel alto C#, Java o VB

- **Enviar la petición:**seinvoca un HTTP GET al URL correcto. Hay dos URIs, uno para resultados en XML y otro para JSON. Estos son http://api.search.live.net/xml.aspx y http://api.search.live.net/json.aspx, respectivamente.

- **Trabajar con los resultados:** el API devuelve los resultados que contienen la información requerida basada en los parámetros enviados al API, ejemplos:
    - Utilizando un solo tipo de fuente con total, offset y count:
        - http://api.search.live.net/xml.aspx?**Appid**=<AppID>**&query**=sushi**&sources=web&web.count**=40**&web.offset**=41.

[82] Protocolos soportados por Bing API: http://msdn.microsoft.com/en-us/library/dd250909.aspx

Figura 2.19: Ejemplo un solo tipo de fuente[83]

- Utilizando múltiples tipos de fuente
  - http://api.search.live.net/xml.aspx?**Appid**=<AppID>&**query**=sushi&**sources**=**web**+**image**.

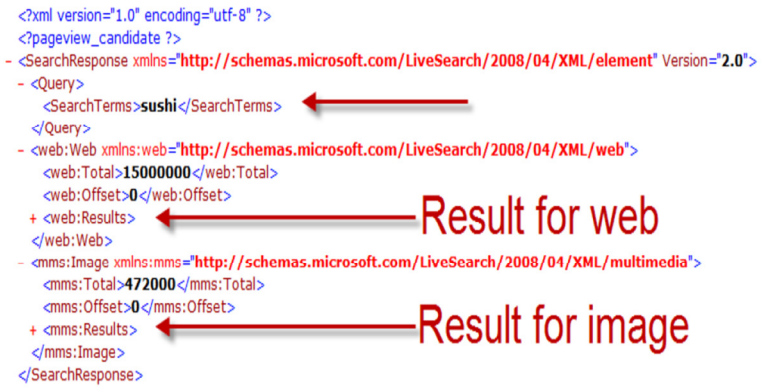

Figura 2.20: Ejemplo varios tipos de fuente[84]

[83] Imagen ejemplo solo tipo de fuente Bing API: http://www.bing.com/developers/s/API%20Basics.pdf

- **Estructura de los resultados del API en un tipo de fuente web:**la estructura está definida como muestra la Tabla 2.23.

Tabla 2.23: Estructura de los resultados que devuelve la fuente web del API de Bing[85]

Nombre	Descripción
Version	La versión del API de Bing q devolvió la respuesta
Query.SearchTerms	Consulta original de la petición
Url	Devuelve el URL completo del resultado como cadena de texto
CacheUrl	Devuelve el URL a la página en cache de Bing del resultado
DateTime	Devuelve un objeto DateTime que contiene el año, mes, día, hora, minuto y segundo de la última actualización o crawl
DeepLinks	Devuelve un arreglo de objetos DeepLink, que contiene propiedades que devuelven el título y el URL de un DeepLink. Un DeepLink es un link dentro de un sitio web
Description	Devuelve el texto de la descripción de los resultados
DisplayUrl	Devuelve la cadena de texto del URL a mostrar al usuario
SearchTags	Devuelve un arreglo de WebSearchTag devueltos por la petición. La clase contiene propiedades que indican el nombre y valor de un tag de búsqueda devuelto por la petición del tipo de fuente web
Title	Devuelve la cadena de texto del título para la petición web

  - **Ejemplo de petición:**(Ver Anexo 7)
  - **Ejemplo de respuesta:**(Ver Anexo 8)

[84] Imagen ejemplo múltiples tipos de fuente Bing API: http://www.bing.com/developers/s/API%20Basics.pdf
[85] Campos de resultados del tipo de fuente web Bing API: http://msdn.microsoft.com/en-us/library/dd250845.aspx

2.13 Ventajas y desventajas del uso de API's de búsqueda como parte fundamental de un aplicativo anti plagio de textos en documentos

Las ventajas y desventajas de los del uso de los API's de búsqueda para el desarrollo de este tipo de aplicativos se muestran en la Tabla 2.24

Tabla 2.24: Ventajas y desventajas del uso de los API's de búsqueda como parte de una aplicación anti plagio

Ventajas	Desventajas
Los tiempos de respuesta para obtener resultados son muy cortos	Los resúmenes q devuelven los API son muy limitados como para buscar coincidencias de plagio
Devuelven resultados con menos falsos positivos	Algunas API's implementaron costos basados en el número de peticiones encareciendo el desarrollo del aplicativo
Los API's devuelven sus resultados en varios formatos como son JSON, XML o SOAP, que facilitan el consumo de los mismos utilizando cualquier sistema operativo, tecnología o lenguaje de programación.	La cantidad de caracteres que se puede utilizar en una consulta es limitado por lo que solo se puede utilizar párrafos y no documentos completos para la búsqueda
Permite verificar plagio no solo en sitios web sino en todo tipo de formato que soporte el API	Requiere la implementación de un crawler que verifique los sitios devueltos por el API en búsqueda de coincidencias
Permite devolver hasta 50 resultados relevantes por cada consulta	Sin conexión a internet el aplicativo no funcionaría
Aplican estándares de búsqueda y autenticación como OpenSearch y OAuth que facilitan la implementación para el consumo del API	Número de peticiones diarias limitadas en ciertos API.

CAPÍTULO III

3. ANÁLISIS COMPARATIVO DE LOS API`S DE BÚSQUEDA DE GOOGLE, YAHOO y BING

3.1 Definición del Product Backlog para el estudio comparativo teórico y práctica de los API'S de búsqueda

3.1.1 Herramienta para administración de proyectos ágiles

Para la administración del proyecto con SCRUM y todos los artefactos que esta metodología propone; se está utilizando la herramienta VersionOne en el link: https://www15.v1host.com/Team70

3.1.2 Product Backlog

El Product Backlog está definido más detalladamente en la herramienta VersionOne.

Tabla 3.1: Product Backlog para el estudio comparativo teórico

Título	Dueño	Prioridad	Esfuerzo
Definición de los criterios de comparación teórica de los API's de búsqueda basados en la documentación técnica de cada API	Jorge	Alta	50,00
Definición de los criterios de comparación teórica fundamentada en experiencia de terceros en el uso de los API's de búsqueda investigados	Jorge	Alta	50,00
Generación de un cuadro comparativo ponderado de las características de los API's de búsqueda basados en los criterios de comparación teóricos definidos anteriormente	Jorge	Alta	50,00
Definición del Product Backlog para el desarrollo de un prototipo de software básico de una aplicación anti plagio de textos en documentos	Jorge	Alta	120,00
Definición del mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos basados en el estudio comparativo	Jorge	Alta	100,00

teórico y práctico			
Actualización del Product Backlog para el desarrollo del prototipo de software de una aplicación anti plagio de textos en documentos implementación final	Jorge	Alta	300,00
Ejecución de Sprints para el prototipo final de software de una aplicación anti plagio de textos en documentos	Jorge	Alta	300,00

3.2 Sprint Backlog, requisitos específicos y calendario para el estudio comparativo y el desarrollo del prototipo de software básico

3.2.1 Sprint Backlog

El Sprint Backlogestádefinido más detalladamente en la herramienta VersionOne.

Tabla 3.2: Sprint Backlog para el estudio comparativo teórico y el desarrollo del prototipo básico

# Sprint	Título	Dueño	# de historias	Esfuerzo
1	Definición y utilización de los criterios de comparación teórica de los API'S de búsqueda	Jorge	3,00	150,00
2	Definición y utilización de los criterios de comparación práctica en un prototipo de software básico de una aplicación anti plagio de textos en documentos	Jorge	1,00	120,00
3	Resultados del estudio comparativo de los API's de búsqueda basados en los criterios de comparación teóricos y prácticos	Jorge	1,00	100,00
4	Diseño e implementación del prototipo de software final de una aplicación anti plagio de textos en documentos	Jorge	2,00	600,00

Tabla 3.3: Product Backlog para el estudio comparativo teórico y el desarrollo del prototipo básico con prioridad, dueño, esfuerzo y número de Sprint

Título	Dueño	Prioridad	Esfuerzo	# Sprint
Definición de los criterios de comparación teórica de los API's de búsqueda basados en la documentación técnica de cada API	Jorge	Alta	50,00	1
Definición de los criterios de comparación teórica fundamentada en experiencia de terceros en el uso de los API's de búsqueda investigados	Jorge	Alta	50,00	1
Generación de un cuadro comparativo ponderado de las características de los API's de búsqueda basados en los criterios de comparación teóricos definidos anteriormente	Jorge	Alta	50,00	1
Definición del Product Backlog para el desarrollo de un prototipo de software básico de una aplicación anti plagio de textos en documentos	Jorge	Alta	120,00	2
Definición del mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos basados en el estudio comparativo teórico y práctico	Jorge	Alta	100,00	3
Actualización del Product Backlog para el desarrollo del prototipo de software de una aplicación anti plagio de textos en documentos implementación final	Jorge	Alta	300,00	4
Ejecución de Sprints para el prototipo final de software de una aplicación anti plagio de textos en documentos	Jorge	Alta	300,00	4

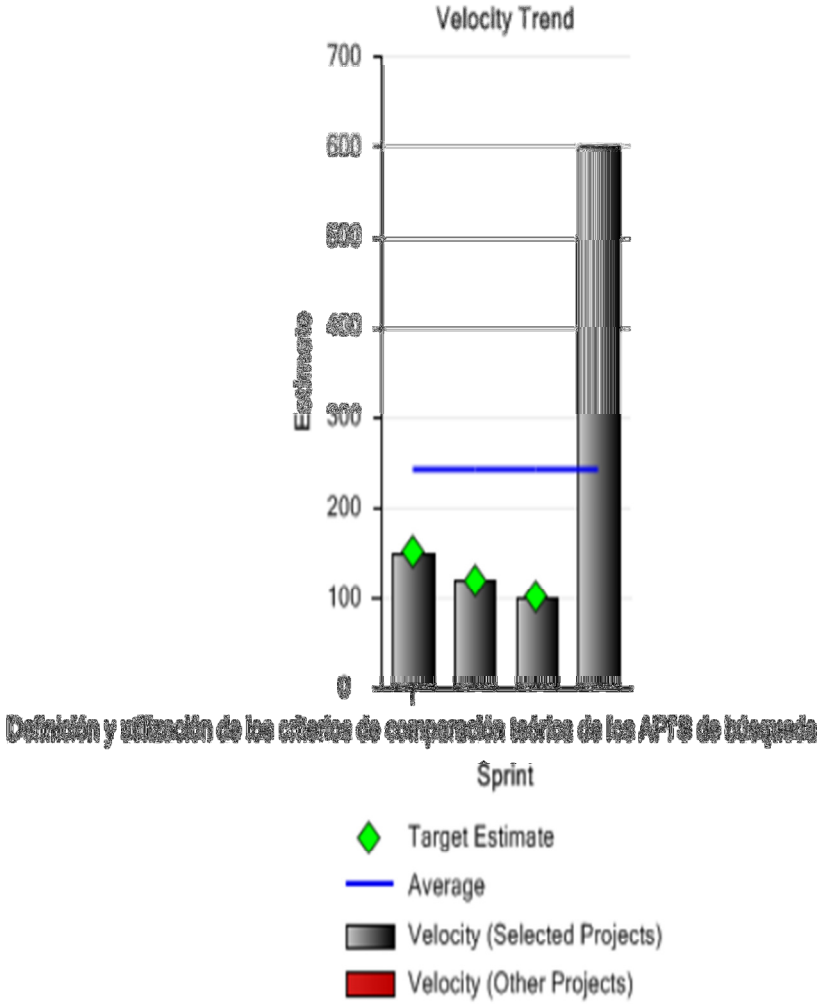

Figura 3.1: Sprint Backlog del estudio

3.3 Sprint de definición y utilización de los criterios de comparación teórica de los API'S de búsqueda

3.3.1 Definición de los criterios de comparación teórica de los API's de búsqueda basados en la documentación técnica de cada API

Basados en la documentación técnica de cada uno de los API's de búsqueda de Google, Yahoo y Bing, se identificaron las criterios comparativos teóricos comunes mostrados en la Tabla 3.4los cuales son de relevancia para el desarrollo de aplicaciones anti plagio de textos en documentos.

Tabla 3.4: Criterios de comparación teórica basados en la documentación de los API's

Criterio	Descripción
Tipos de documentos de búsqueda	Si el API permite especificar tipos de documentos, cuántos y cuáles de estos tienen más importancia en la búsqueda de plagio.
Resumen extenso	Si el resumen devuelto por el API tiene una extensión considerable para tomarlo en cuenta y buscar coincidencia; la extensión se basa en número de caracteres (300 o más).
Fecha de última actualización del sitio	Si el API devuelve junto con cada uno de los resultados la última fecha de actualización del sitio o la última vez que paso el crawler por el mismo.
Madurez del API	Grado de madurez del API utilizado, tomando como referencia el número de versiones y si es una versión estable o en fase de desarrollo.
Tipos de formatos de respuesta	Cantidad de formatos de respuesta que ofrece el API y cuantos son relevantes, ejemplo: JSON, XML, ATOM.
Tipos de fuentes de búsqueda	Cantidad de tipos de fuentes de búsqueda del API de donde se puede obtener resultados relevantes de plagio, por ejemplo: web, imágenes, noticias, blogs.
Número máximo de resultados por página	Cantidad de documentos que devuelve el API por página.
Filtrado de pornografía	Si el API tiene la opción de filtrar las búsquedas por contenido pornográfico.
Lenguajes que soporta	Cantidad de lenguajes que soporta el API para las búsquedas de plagio.

Implementación de OpenSearch	Si el API implementa una o varias de las tecnologías que plantea OpenSearch.
Autorización OAuth	Si el API tiene la opción de autenticarse en el API a través de OAuth.
Sitios preferenciales de búsqueda	Si el API permite especificar por cuales sitios buscar primero.
Restringir sitios de búsqueda	Si el API permite especificar cuáles sitios puede incluir en la búsqueda y cuáles no.
Resumen libre de HTML	Si el API permite devolver el resumen de cada uno de los resultados en texto plano sin HTML.
Filtrado de duplicados	Si el API tiene la opción de filtrar los resultados duplicados en la búsqueda actual.
Costo de las peticiones	Costo por utilización del API, basada en un promedio entre 1.000, 10.000 y 100.000 peticiones diarias (Costo 1.000 peticiones + Costo 10.000 peticiones + Costo 100.000 peticiones) / 3.
Configuración del API	Facilidades para la configuración del API, si posee interfaz web.
Monetización	Si el API permite generar ingresos mediante publicidad.
Tipos de fuentes de búsqueda simultáneos	Si el API tiene la opción de buscar en algunos tipos de fuentes de búsqueda al mismo tiempo.

3.3.2 Definición de los criterios de comparaciónteórica fundamentada en experiencia de terceros en el uso de los API's de búsqueda investigados

Basados en experiencia de terceros en la utilización de cada uno de los API's de búsqueda de Google, Yahoo y Bing, se identificaron las siguientes criterios comparativos

teóricos comunes mostrados en la Tabla 3.5 los cuales son de relevancia para el desarrollo de aplicaciones anti plagio de textos en documentos[86]

Tabla 3.5: Criterios de comparación teórica basados en experiencia de terceros

Criterio	Descripción
Número aproximado de aplicaciones utilizando el API	Cantidad aproximada de aplicaciones de terceros que consumen el API y este forma parte fundamental del diseño del aplicativo.
Número de aplicaciones anti plagio utilizando el API o directamente el motor de búsqueda	Cantidad aproximada de aplicaciones anti plagio de textos o documentos que consumen el API o utilizan el motor de búsqueda directamente y que este forme parte fundamental del diseño del aplicativo.

3.3.3 Generación de un cuadro comparativo ponderado de las características de los API's de búsqueda basados en los criterios de comparación teóricos definidos anteriormente

3.3.3.1 Ponderación de los criterios de comparación teórica

La ponderación de los criterios de comparación teórica están basados en la relevancia que tiene cada uno de ellos en el ámbito del desarrollo de aplicaciones anti plagio de textos en documentos. Los rangos de valoresse muestran en la Tabla 3.6.

Tabla 3.6: Ponderación para los criterios de comparación teórica

Ponderación/Valor	Descripción
0-10	Poco relevante
11-20	Relevante
21-30	Muy relevante

Al final se realiza una regla del tres para obtener la ponderación total del API sobre 100 puntos. La ponderación por criterio son los mostrados en la Tabla 3.7.

[86] Criterios de comparación teórica: http://www.programmableweb.com/apis

Tabla 3.7: Criterios de comparación teórica con sus ponderaciones

Criterio	Ponderación
Tipos de documentos de búsqueda	30
Resumen extenso	30
Fecha de última actualización del sitio	20
Madurez del API	20
Tipos de formatos de respuesta	20
Tipos de fuentes de búsqueda	30
Número máximo de resultados por página	20
Filtrado de pornografía	20
Lenguajes que soporta	30
Implementación de OpenSearch	20
Autorización OAuth	20
Sitios preferenciales de búsqueda	30
Restringir sitios de búsqueda	20
Resumen libre de HTML	30
Filtrado de duplicados	30
Costo de las peticiones	30
Configuración del API	10
Monetización	10
Tipos de fuentes de búsqueda simultáneos	20
Número aproximado de aplicaciones utilizando el API	10
Número de aplicaciones anti plagio utilizando el API o directamente el motor de búsqueda	20
Total	**470**
Total API / 100	**(# * 100) / 470**

El peso define la utilidad que la característica de un criterio brinda al emplearlo en el desarrollo de aplicaciones anti plagio de textos en documentos tiene los valores mostrados en la Tabla 3.8.

Tabla 3.8: Pesos para el detalle de criterios específicos

Peso/Valor	Descripción
0	Nada útil
1	Poco útil
2	Útil
3 o +	Muy útil

La Tabla 3.9 muestra los valores ponderados de cada uno de los criterios. En algunos casos se especifica las características de un criterio al que se le asigna un peso para que la evaluación de los criterios sea más detallada.

Tabla 3.9: Detalle de medición en pesos y ponderaciones para los criterios de comparación teórica

<table>
<tr><th>Criterio</th><th>Ponderación</th></tr>
<tr><td>Tipos de documentos de búsqueda</td><td>Los pesos son los siguientes:
<table><tr><th>Característica</th><th>Peso</th></tr><tr><td>No soporta</td><td>0</td></tr><tr><td>Opción del API</td><td>5</td></tr><tr><td>Soporta Word</td><td>3</td></tr><tr><td>Soporta Excel</td><td>2</td></tr><tr><td>Soporta Ppt</td><td>3</td></tr><tr><td>Soporta Html</td><td>3</td></tr><tr><td>Soporta Htm</td><td>3</td></tr><tr><td>Soporta Text</td><td>2</td></tr><tr><td>Soporta Rtf</td><td>2</td></tr><tr><td>Soporta Pdf</td><td>3</td></tr><tr><td>Soporta Txt</td><td>2</td></tr><tr><td>Más tipos</td><td>2</td></tr><tr><td>Total</td><td>30</td></tr></table></td></tr>
<tr><td>Resumen extenso</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Si soporta> 300</td><td>30</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>caracteres</td><td></td></tr>
<tr><td>Soporta<= 300 caracteres</td><td>10</td></tr>
<tr><td>No soporta</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>Fecha de última actualización del sitio</td><td>Las ponderaciones son las siguientes:

<table>
<tr><th>Opción</th><th>Ponderación</th></tr>
<tr><td>Si soporta</td><td>20</td></tr>
<tr><td>No soporta</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>Madurez del API</td><td>Las ponderaciones son las siguientes:

<table>
<tr><th>Opción</th><th>Ponderación</th></tr>
<tr><td>Beta o Labs</td><td>10</td></tr>
<tr><td>Estable</td><td>20</td></tr>
</table>
</td></tr>
<tr><td>Tipos de formatos de respuesta</td><td>Los pesos son los siguientes:

<table>
<tr><th>Característica</th><th>Peso</th></tr>
<tr><td>Xml</td><td>10</td></tr>
<tr><td>Json</td><td>10</td></tr>
<tr><td>Total</td><td>20</td></tr>
</table>
</td></tr>
<tr><td>Tipos de fuentes de búsqueda</td><td>Los pesos son los siguientes:

<table>
<tr><th>Característica</th><th>Peso</th></tr>
<tr><td>No soporta</td><td>0</td></tr>
<tr><td>Opción del API</td><td>5</td></tr>
<tr><td>Soporta Web</td><td>5</td></tr>
<tr><td>Soporta Imagen</td><td>5</td></tr>
<tr><td>Soporta Noticias</td><td>5</td></tr>
<tr><td>Soporta Blogs</td><td>5</td></tr>
<tr><td>Más tipos</td><td>5</td></tr>
<tr><td>Total</td><td>30</td></tr>
</table>
</td></tr>
<tr><td>Número máximo de resultados por página</td><td>Las ponderaciones son las siguientes:

<table>
<tr><th>Opción</th><th>Ponderación</th></tr>
<tr><td>Hasta 20 resultados</td><td>10</td></tr>
<tr><td>Hasta 50 resultados</td><td>20</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Filtrado de pornografía</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Si soporta</td><td>20</td></tr><tr><td>No soporta</td><td>0</td></tr></table></td></tr>
<tr><td>Lenguajes que soporta</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td><= 30 lenguajes incluido español</td><td>20</td></tr><tr><td>> 30 lenguajes incluido español</td><td>30</td></tr></table></td></tr>
<tr><td>Implementación de OpenSearch</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Si soporta</td><td>20</td></tr><tr><td>No soporta</td><td>0</td></tr></table></td></tr>
<tr><td>Autorización OAuth</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Si soporta</td><td>20</td></tr><tr><td>No soporta</td><td>0</td></tr></table></td></tr>
<tr><td>Sitios preferenciales de búsqueda</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Si soporta</td><td>30</td></tr><tr><td>No soporta</td><td>0</td></tr></table></td></tr>
<tr><td>Restringir sitios de búsqueda</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Si soporta</td><td>20</td></tr><tr><td>No soporta</td><td>0</td></tr></table></td></tr>
<tr><td>Resumen libre de HTML</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Si soporta</td><td>30</td></tr><tr><td>No soporta</td><td>0</td></tr></table></td></tr>
</table>

<table>
<tr><td>Filtrado de duplicados</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Si soporta</td><td>30</td></tr><tr><td>No soporta</td><td>0</td></tr></table></td></tr>
<tr><td>Costo de las peticiones</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>0$</td><td>30</td></tr><tr><td><25$</td><td>20</td></tr><tr><td>>=25$</td><td>0</td></tr></table></td></tr>
<tr><td>Configuración del API</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Si soporta</td><td>10</td></tr><tr><td>No soporta</td><td>0</td></tr></table></td></tr>
<tr><td>Monetización</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Si soporta</td><td>10</td></tr><tr><td>No soporta</td><td>0</td></tr></table></td></tr>
<tr><td>Tipos de fuentes de búsqueda simultáneos</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Si soporta</td><td>20</td></tr><tr><td>No soporta</td><td>0</td></tr></table></td></tr>
<tr><td>Número aproximado de aplicaciones utilizando el API</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td><35</td><td>5</td></tr><tr><td>>=35</td><td>10</td></tr></table></td></tr>
<tr><td>Número de aplicaciones anti plagio utilizando el API o directamente el motor de búsqueda</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td><5</td><td>10</td></tr><tr><td>>=5</td><td>20</td></tr></table></td></tr>
</table>

3.3.3.2 Cuadro comparativo teórico

Se generó el siguiente cuadro comparativode características (Tabla 3.10) basado en la documentación de cada API's y experiencia de terceros en el uso de los API's

Tabla 3.10: Información de cada API en base a los criterios de comparación teórica

<table>
<tr><th rowspan="2">Criterio</th><th colspan="6">API</th></tr>
<tr><th colspan="2">Google</th><th colspan="2">Yahoo</th><th colspan="2">Bing</th></tr>
<tr><td>Tipos de documentos de búsqueda</td><td colspan="2">Si, dentro del Query.
Soporta 35 tipos.
Relevantes:
<table><tr><th>Tipo</th><th>Permite</th></tr><tr><td>Word</td><td>Si</td></tr><tr><td>Excel</td><td>Si</td></tr><tr><td>Ppt</td><td>Si</td></tr><tr><td>Pdf</td><td>Si</td></tr><tr><td>Html</td><td>Si</td></tr><tr><td>Htm</td><td>Si</td></tr><tr><td>Rtf</td><td>Si</td></tr><tr><td>Txt</td><td>Si</td></tr><tr><td>Text</td><td>Si</td></tr></table></td><td colspan="2">Si, opción del API.
Soporta 9 tipos.
Relevantes:
<table><tr><th>Tipo</th><th>Permite</th></tr><tr><td>Word</td><td>Si</td></tr><tr><td>Excel</td><td>Si</td></tr><tr><td>Ppt</td><td>Si</td></tr><tr><td>Pdf</td><td>Si</td></tr><tr><td>Html</td><td>Si</td></tr><tr><td>Htm</td><td>Si</td></tr><tr><td>Rtf</td><td>Si</td></tr><tr><td>Txt</td><td>Si</td></tr><tr><td>Text</td><td>Si</td></tr></table></td><td colspan="2">Si, opción del API.
Soporta 10 tipos.
Relevantes:
<table><tr><th>Tipo</th><th>Permite</th></tr><tr><td>Word</td><td>Si</td></tr><tr><td>Excel</td><td>Si</td></tr><tr><td>Ppt</td><td>Si</td></tr><tr><td>Pdf</td><td>Si</td></tr><tr><td>Html</td><td>Si</td></tr><tr><td>Htm</td><td>Si</td></tr><tr><td>Rtf</td><td>Si</td></tr><tr><td>Txt</td><td>Si</td></tr><tr><td>Text</td><td>Si</td></tr></table></td></tr>
<tr><td>Resumen extenso</td><td colspan="2">No
Hasta 300caracteres</td><td colspan="2">Si
Más de 300 caracteres</td><td colspan="2">No
Hasta 300 caracteres</td></tr>
<tr><td>Fecha de última actualización del sitio</td><td colspan="2">No</td><td colspan="2">Si</td><td colspan="2">Si</td></tr>
<tr><td>Madurez del API</td><td colspan="2">V1.0 Labs (Beta)</td><td colspan="2">V2.0 Final</td><td colspan="2">V2.0 Final</td></tr>
<tr><td>Tipos de formatos de respuesta</td><td colspan="2">Soporta 2 formatos.
Relevantes:
<table><tr><th>Tipo</th><th>Permite</th></tr><tr><td>Xml</td><td>Si. Atom</td></tr></table></td><td colspan="2">Soporta 2 formatos.
Relevantes:
<table><tr><th>Tipo</th><th>Permite</th></tr><tr><td>Xml</td><td>Si</td></tr></table></td><td colspan="2">Soporta 2 formatos.
Relevantes:
<table><tr><th>Tipo</th><th>Permite</th></tr><tr><td>Xml</td><td>Si</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td>Json</td><td>Si</td></tr></table></td><td><table><tr><td>Json</td><td>Si</td></tr></table></td><td><table><tr><td>Json</td><td>Si</td></tr></table></td></tr>
<tr><td>Tipos de fuentes de búsqueda</td><td>Soporta 1 tipos de fuentes.
Relevantes:<table><tr><th>Tipo</th><th>Permite</th></tr><tr><td>Web</td><td>Si</td></tr><tr><td>Imagen</td><td>No</td></tr><tr><td>Noticias</td><td>No</td></tr><tr><td>Blogs</td><td>No</td></tr></table></td><td>Soporta 6 tipos de fuentes.
Relevantes:<table><tr><th>Tipo</th><th>Permite</th></tr><tr><td>Web</td><td>Si</td></tr><tr><td>Imagen</td><td>Si</td></tr><tr><td>Noticias</td><td>Si</td></tr><tr><td>Blogs</td><td>Si</td></tr></table></td><td>Soporta 10 tipos de fuentes.
Relevantes:<table><tr><th>Tipo</th><th>Permite</th></tr><tr><td>Web</td><td>Si</td></tr><tr><td>Imagen</td><td>Si</td></tr><tr><td>Noticias</td><td>Si</td></tr><tr><td>Blogs</td><td>No</td></tr></table></td></tr>
<tr><td>Número máximo de resultados por página</td><td>Máximo 10</td><td>Máximo 50, 20 para Blogs y 35 para Imagen</td><td>Máximo 50</td></tr>
<tr><td>Filtrado de pornografía</td><td>Si</td><td>Si</td><td>Si</td></tr>
<tr><td>Lenguajes que soporta</td><td>Soporta 34 lenguajes incluido español</td><td>Soporta 27 lenguajes incluido español</td><td>Soporta 32 lenguajes incluido español</td></tr>
<tr><td>Implementación de OpenSearch</td><td>Si, autenticación</td><td>Si, autenticación</td><td>Si, interfaz RSS</td></tr>
<tr><td>Autorización oAuth</td><td>Si</td><td>Si</td><td>No</td></tr>
<tr><td>Sitios preferenciales de búsqueda</td><td>Si</td><td>No</td><td>No</td></tr>
<tr><td>Restringir sitios de búsqueda</td><td>Si</td><td>Si</td><td>No</td></tr>
<tr><td>Resumen libre de HTML</td><td>Si siempre</td><td>Si, como opción del API</td><td>Sí, siempre</td></tr>
<tr><td>Filtrado de duplicados</td><td>Si</td><td>No</td><td>No</td></tr>
<tr><td>Costo de las</td><td>5 dólares por 1000, 5</td><td>Solo web:</td><td>Sin costo</td></tr>
</table>

peticiones	dólares 10000 y 50 dólares por 10000 peticiones diarias (100 gratuitas). Costo total: 20 dólares	0.80dólarespor 1000 peticiones, 8 dólares por 10000 y 80 dólares por 100000 diarias (0.80 por 1000 peticiones). Costo total: 29.6	
Configuración del API	Si Portal Web	Si Portal Web	Si Portal Web
Monetización	Si	Si	Si
Tipos de fuentes de búsqueda simultáneos	No	Si	Si
Número aproximado de aplicaciones utilizando el API[87]	36	53	34
Número de aplicaciones anti plagio utilizando el API o directamente el motor de búsqueda	6más conocidas: DupliChecker PlagiarismChecker SearchEngineReports ArticleChecker Plagiarized Plagium.com	5 más conocidas: DupliChecker PlagiarismChecker ArticleChecker Plagiarized Reprint Writers' tool	4 más conocidas: Plagium.com DupliChecker Plagiarized SearchEngineReports

[87] Aplicaciones utilizando el API: http://www.programmableweb.com/apis

3.3.3.3 Cuadro comparativo teórico ponderado

Se generó el siguiente cuadro comparativo teórico ponderado (Tabla 3.11) de características basado en las ponderaciones definidas en el punto 3.3.3.1 de este documento.

Tabla 3.11: Cuadro comparativo teórico resultante de cada API basados en el análisis de la documentación de los API's y los criterios de comparación teórica

Criterio	API		
	Google	Yahoo	Bing
Tipos de documentos de búsqueda	25/30	28/30	30/30
Resumen extenso	10/30	30/30	10/30
Fecha de última actualización del sitio	0/20	20/20	20/20
Madurez del API	10/20	20/20	20/20
Tipos de formatos de respuesta	20/20	20/20	20/20
Tipos de fuentes de búsqueda	5/30	30/30	25/30
Número máximo de resultados por página	10/20	20/20	20/20
Filtrado de pornografía	20/20	20/20	20/20
Lenguajes que soporta	30/30	20/30	30/30
Implementación de OpenSearch	20/20	20/20	20/20
Autorización OAuth	20/20	20/20	0/20
Sitios preferenciales de búsqueda	30/30	0/30	0/30
Restringir sitios de búsqueda	20/20	20/20	0/20
Resumen libre de HTML	30/30	30/30	30/30
Filtrado de duplicados	30/30	0/30	0/30
Costo de las peticiones	20/30	0/30	30/30
Configuración del API	10/10	10/10	10/10
Monetización	10/10	10/10	10/10
Tipos de fuentes de búsqueda simultáneos	0/20	20/20	20/20
Número aproximado de aplicaciones utilizando el API	10/10	10/10	5/10

Número de aplicaciones anti plagio utilizando el API o directamente el motor de búsqueda	20/20	20/20	10/20
Total	**350/470**	**368/470**	**330/470**
Total API / 100	**74,47**	**78,30**	**70,21**

Los resultados obtenidos demuestran que el mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos en el ámbito teórico es el API de Yahoo (Yahoo BOSS API) con un resultado de 78,30luego le sigue el API de Google (Google Custom Search API Json/Atom) con un resultado de 74,47yen último lugar el API de Bing (Bing API) con 70,21.

Yahoo BOSS API es el mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos en el ámbito teórico debido a la gran cantidad de opciones de personalización que provee el API como son: los tipos de fuentes o servicios, la posibilidad de buscar simultáneamente en varios de estos servicios y la última fecha en la que se analizó el sitio devuelto por el API.

3.4 Sprint de definición y utilización de los criterios de comparación práctica en un prototipo de software básico de una aplicación anti plagio de textos en documentos

3.4.1 Definición del Product Backlog para el desarrollo de un prototipo de software básico de unaaplicaciónanti plagio de textos en documentos

3.4.1.1 Herramienta para administración de proyectos ágiles

Para la administración del proyecto con SCRUM y todos los artefactos que esta metodología propone; se está utilizando la herramienta VersionOne en el link: https://www16.v1host.com/DGAMC/

3.4.1.2 Product Backlog

El Product Backlog está definido más detalladamente en la herramienta VersionOne:

Tabla 3.12: Product Backlog para el estudio comparativo práctico

Título	Dueño	Prioridad	Esfuerzo
Diseño de la arquitectura del prototipo de software básico utilizando UML	Jorge	Alta	300,00
Definición de los criterios de comparación práctica basados en el diseño del prototipo de software.	Jorge	Alta	150,00
Consumo de los API's de búsqueda	Jorge	Alta	200,00
Filtro de datos genérico	Jorge	Alta	200,00
Algoritmo de búsqueda de plagio aplicado a documentos encontrados con los API's	Jorge	Alta	200,00
Implementación de la interfaz de usuario básica en Windows Forms	Jorge	Alta	100,00
Pruebas	Jorge	Alta	200,00
Generación de un cuadro comparativo ponderado de las características de los API's de búsqueda basados en los criterios de comparación prácticos y las pruebas realizadas del prototipo	Jorge	Alta	300,00

3.4.1.3 Sprint Backlog, requisitos específicos y calendario del prototipo de software básico

3.4.1.3.1 Sprint Backlog

El Sprint Backlog está definido más detalladamente en la herramienta VersionOne.

Tabla 3.13: Sprint Backlog para el estudio comparativo práctico basados en el desarrollo del prototipo básico

# Sprint	Titulo	Dueño	# historias	Esfuerzo
1	Diseño, desarrollo y pruebas del prototipo de software básico de una aplicación anti plagio de textos en documentos	Jorge	7,00	1350,00
2	Desarrollo de cuadros comparativos ponderados basados en los criterios comparativos prácticos definidos	Jorge	1	300

Tabla 3.14: Product Backlog para el estudio comparativo práctico basados en el desarrollo del prototipo básico con prioridad, dueño, esfuerzo y número de sprint

Título	Dueño	Prioridad	Esfuerzo	# Sprint
Diseño de la arquitectura del prototipo de software básico utilizando	Jorge	Alta	300,00	1
Definición de los criterios de comparación práctica basados en el diseño del prototipo de software.	Jorge	Alta	150,00	1
Consumo de los API's de búsqueda	Jorge	Alta	200,00	1
Filtro de datos genérico	Jorge	Alta	200,00	1
Algoritmo de búsqueda de plagio aplicado a documentos encontrados con los API's	Jorge	Alta	200,00	1
Implementación de la interfaz de usuario básica en Windows Forms	Jorge	Alta	100,00	1
Pruebas	Jorge	Alta	200,00	1
Generación de un cuadro comparativo ponderado de las características de los API's de búsqueda basados en los criterios de comparación prácticos y las pruebas realizadas del prototipo	Jorge	Alta	300,00	2

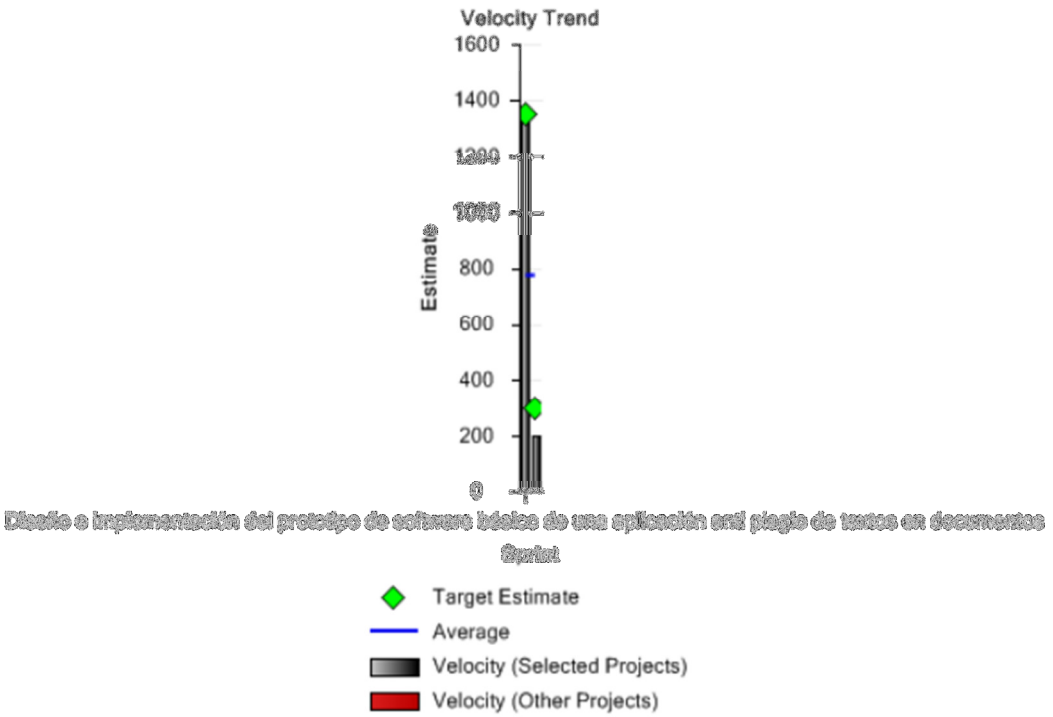

Figura 3.2: SprintBacklog Prototipo Básico

3.4.1.4 Sprint de diseño, desarrollo y pruebas del prototipo de software básico de una aplicación anti plagio de textos en documentos

3.4.1.4.1 Diseño de la arquitectura del prototipo de software básico utilizando UML

Los diagramas de casos de uso, secuencia y de clases fueron diseñados utilizando la herramienta Power Designer versión 16. Estos diagramas se encuentran en el archivo MODELOS_TESIS_PLAGIO.prjy en el documento de la especificación de requisitos con la norma IEEE830 con el nombre RequerimientosV2.docx dentro del CD de contenidos

3.4.1.4.2 Definición de los criterios de comparación práctica basados en el diseño del prototipo de software

Basados en el diseño del prototipo básico de softwarey pruebas de funcionalidad sobre el mismo, se identificaron los siguientes criterios comparativos prácticos comunes mostrados en la Tabla 3.15 que son de relevancia para el definir el mejor API para el desarrollo de aplicaciones anti plagio de textos en documentos.

Tabla 3.15: Criterios de comparación práctica basados en el desarrollo del prototipo

Criterio	Descripción
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos en la primera ejecución	El tiempo promedio en milisegundos de espera en el API para obtener los 5 resultados de cada una de las consultas especificadas en la primera ejecución (2 párrafos). Tiempo promedio de los 2 resultados obtenidos: Suma de tiempos de espera / 2.
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución	El tiempo promedio en milisegundos de espera en el API para obtener los 5 resultados de cada una de las consultas especificadas desde la segunda ejecución iterando 3 veces seguidas (2 párrafos). Tiempo promedio de los 6 resultados obtenidos: Suma de tiempos de espera / 6.
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos en la primera ejecución	El tiempo promedio en milisegundos de espera en el API para obtener los 5 resultados de cada una de las consultas especificadas en la primera ejecución (5 párrafos). Tiempo promedio de los 5 resultados obtenidos: Suma de tiempos de espera / 5.
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución	El tiempo promedio en milisegundos de espera en el API para obtener los 5 resultados de cada una de las consultas especificadas desde la segunda ejecución iterando 3 veces seguidas (5 párrafos). Tiempo promedio de los 15 resultados obtenidos: Suma de tiempos de espera / 15.
Tiempo necesario para el desarrollo del consumo del API de búsqueda	Tiempo en horas necesitadas para desarrollar el consumo del API de

	búsqueda.
Similitud de los sitios consultados mediante el API vs los de la web del proveedor del API	Cantidad de sitios iguales que devuelve el API con respecto a los sitios que devuelve la web del proveedor del API en la primera página.
Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 2 párrafos	Tiempo en milisegundos de espera en obtener los 5 resultados de cada uno de los dos párrafos del documento.
Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 5 párrafos	Tiempo en milisegundos de espera en obtener los 5 resultados de cada uno de los cinco párrafos del documento.

La ponderación de los criterios de comparación práctica se muestran en la Tabla 3.16 y están basados en la relevancia que tiene cada uno de ellos en el ámbito del desarrollo de aplicaciones anti plagio de textos en documentos tomando en cuenta el rendimiento del aplicativo y el tiempo de desarrollo necesario para implementar la parte del prototipo que consume el API de búsqueda

Tabla 3.16: Ponderación para los criterios de comparación práctica

Ponderación/Valor	Descripción
0-10	Poco relevante
11-20	Relevante
21-30	Muy relevante

Al final se realiza una regla del tres para obtener la ponderación total del API sobre 100 puntos. La ponderación por criterio se muestra en la Tabla 3.17.

Tabla 3.17: Criterios de comparación práctica con sus ponderaciones

Criterio	Ponderación
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos en la primera ejecución	30
Tiempo promedio de espera en el API de búsqueda utilizando un	30

documento de 2 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución	
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos en la primera ejecución	30
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución	30
Similitud de los sitios consultados mediante el API vs los de la web del proveedor del API	30
Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 2 párrafos	20
Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 5 párrafos	20
Total	**190**
Total API / 100	**(# * 100) / 190**

El siguiente cuadro (Tabla 3.18) muestra los valores ponderados de cada uno de los criterios.

Tabla 3.18: Detalle de medición en ponderaciones para los criterios de comparación práctica

<table>
<tr><th>Criterio</th><th>Ponderación</th></tr>
<tr><td>Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos en la primera ejecución</td><td>Las ponderaciones son las siguientes:
<table>
<tr><th>Opción</th><th>Ponderación</th></tr>
<tr><td>Tiempo promedio <= 300 ms</td><td>30</td></tr>
<tr><td>Tiempo promedio >300ms <= 600ms</td><td>20</td></tr>
<tr><td>Tiempo promedio > 600ms</td><td>10</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Tiempo promedio <= 300 ms</td><td>30</td></tr><tr><td>Tiempo promedio > 300ms <= 600ms</td><td>20</td></tr><tr><td>Tiempo promedio > 600ms</td><td>10</td></tr></table></td></tr>
<tr><td>Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos en la primera ejecución</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Tiempo promedio <= 600 ms</td><td>30</td></tr><tr><td>Tiempo promedio> 600ms <= 1200ms</td><td>20</td></tr><tr><td>Tiempo promedio > 1200ms</td><td>10</td></tr></table></td></tr>
<tr><td>Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>Tiempo promedio <= 600 ms</td><td>30</td></tr><tr><td>Tiempo promedio > 600ms <= 1200ms</td><td>20</td></tr><tr><td>Tiempo promedio > 1200ms</td><td>10</td></tr></table></td></tr>
<tr><td>Similitud de los sitios consultados mediante el API vs los de la web del proveedor del API</td><td>Las ponderaciones son las siguientes:
<table><tr><th>Opción</th><th>Ponderación</th></tr><tr><td>>= 3 sitios</td><td>30</td></tr><tr><td>>= 1<= 2 sitios</td><td>15</td></tr><tr><td>0 sitios</td><td>0</td></tr></table></td></tr>
<tr><td>Tiempo de espera para obtener los</td><td>Las ponderaciones son las siguientes:</td></tr>
</table>

<table>
<tr><td>resultados del análisis de plagio en un documento con 2 párrafos</td><td><table>
<tr><th>Opción</th><th>Ponderación</th></tr>
<tr><td>Tiempo promedio <= 2 s</td><td>20</td></tr>
<tr><td>Tiempo promedio > 2 s <= 3 s</td><td>10</td></tr>
<tr><td>Tiempo promedio > 3 s</td><td>0</td></tr>
</table></td></tr>
<tr><td>Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 5 párrafos</td><td>Las ponderaciones son las siguientes:<table>
<tr><th>Opción</th><th>Ponderación</th></tr>
<tr><td>Tiempo promedio <= 7 s</td><td>20</td></tr>
<tr><td>Tiempo promedio > 7 s <= 14 s</td><td>10</td></tr>
<tr><td>Tiempo promedio > 14 s</td><td>0</td></tr>
</table></td></tr>
</table>

3.4.1.4.3 Consumo de los API's de búsqueda

Capa de datos encargada de realizar las conexiones, consumo y deserialización de los resultados obtenidos por los API de búsqueda utilizados.

El user story de esta etapa de desarrollo es la mostrada por la Tabla 3.19.

Tabla 3.19: User Stories para la capa de consumo de API's de búsqueda

User Story	Prioridad
Como aplicación. Necesito conectarme al API de Yahoo utilizando oAuth	Alta
Como aplicación. Necesito conectarme al API de Google y Bing utilizando el API Key del proveedor	Alta
Como aplicación. Necesito deserializar en objetos el resultado que devuelve el API ya sea que este en formato XML o JSON	Alta
Como aplicación. Necesito obtener las páginas en texto plano de cada uno de los sitios web que devolvió la consulta al API. Deben ser guardadas en el servidor como almacenamiento temporal	Alta
Como aplicación. Necesito generar un algoritmo de comparación de textos	Alta

Como aplicación. Necesito crear una fábrica para los API's de búsqueda consultados.	Alta

3.4.1.4.4 Filtro de datos genérico

Capa de abstracción encargada de crear una interfaz genérica entre los diferentes APIs de búsqueda para devolver un resultado único independiente del API utilizado e implementación de un crawler básico encargado de obtener el texto de cada una de los sitios devueltos por el API en cada consulta.

El user story de esta etapa de desarrollo es la mostrada por la Tabla 3.20.

Tabla 3.20: User Stories para la capa de filtro de datos genérico

User Story	Prioridad
Como aplicación. Necesito crear cada uno de los API's utilizando la fábrica de API's de la capa inferior, con los parámetros de lenguaje y tipo de fuente que se van a utilizar	Alta
Como aplicación. Necesito devolver los resultados de cualquiera de los API's de búsqueda utilizando una interfaz genérica que los abstraiga	Alta
Como aplicación. Necesito leer los parámetros de configuración de cada uno de los API's	Alta
Como aplicación. Necesito cargar los sitios web guardados en el servidor	Alta
Como aplicación. Necesito crear un algoritmo que obtenga el porcentaje de plagio de dos textos ingresados utilizando el algoritmo de comparación de textos de la capa de consumo de los API's de búsqueda	Alta
Como aplicación. Necesito obtener las páginas en texto plano de cada uno de los sitios web que devolvió la consulta al API que no estén ya en la aplicación. Estas deben ser guardadas en el servidor como almacenamiento temporal	Alta

3.4.1.4.5 Algoritmo de búsqueda de plagio aplicado a documentos encontrados con los API's

Capa encargada de leer los documentos ingresados por el usuario y verificar el plagio entre el documento y el sitio web devuelto por el API utilizando el algoritmo de comparación de textos para realizar la tarea.

El user story de esta etapa de desarrollo es la mostrada por la Tabla 3.21.

Tabla 3.21: User Stories para la capa de algoritmo de búsqueda de plagio aplicado a documentos encontrados por los API's

User Story	Prioridad
Como aplicación. Necesito leer documentos de tipo doc, docx, txt y rtf de no más de 2MB a la aplicación y guardarlas en el servidor.	Alta
Como aplicación. Necesito devolver los resultados del documento analizado con cualquiera de los API's de búsqueda utilizando una interfaz que los abstraiga	Alta
Como aplicación. Necesito que cada uno de los resultados abstraídos tengan como atributos: el API que se utilizó para el análisis, el tiempo que se demoró en procesar el documento, el porcentaje de plagio de cada uno y el tiempo que se demoró el API en devolver los resultados por cada párrafo del documento	Alta
Como aplicación. Necesito obtener el porcentaje de plagio de cada uno de los párrafos del documento utilizando la interfaz de la capa de filtrado genérico	Alta

3.4.1.4.6 Implementación de la interfaz de usuario básica en Windows Forms

Capa superior encargada de la interfaz gráfica con la que interviene el usuario con la aplicación realizada en Windows Forms.

El user story de esta etapa de desarrollo es la mostrada por la Tabla 3.22.

Tabla 3.22: User Stories para la capa de interfaz de usuario básico en Windows Forms

User Story	Prioridad
Como usuario. Necesito escoger el documento a analizar filtrando los tipos de documentos que soporta el aplicativo	Alta
Como usuario. Necesito seleccionar que API's de búsqueda voy a utilizar para el análisis de plagio	Alta
Como usuario. Necesito ver los resultados del análisis de plagio del documento en un formato sencillo y legible.	Alta
Como aplicación. Necesito validar el ingreso de los datos necesarios para realizar la tarea y mostrar al usuario los errores en caso de que existan.	Alta

Como aplicación. Necesito generar la tarea basada en las preferencias del usuario para analizar plagio en el documento, utilizando la capa de Algoritmo de búsqueda de plagio aplicado a documentos encontrados con el API	Alta

3.4.1.4.7 Pruebas

Capa encargada de realizar las pruebas unitarias de cada uno de los componentes del prototipo de software básico y pruebas de funcionalidad para obtener los resultados necesarios para el desarrollo del cuadro comparativo práctico del prototipo de software básico.

El user story de esta etapa de desarrollo es la mostrada por la Tabla 3.23.

Tabla 3.23: User Stories para las pruebas del prototipo básico

User Story	Prioridad
Como aplicación. Necesito crear pruebas unitarias para la conexión y deserialización de los resultados obtenidos por el API utilizando la fábrica de API's.	Alta
Como aplicación. Necesito crear pruebas unitarias para la descarga de los sitios web en texto plano al servidor.	Alta
Como aplicación. Necesito crear pruebas unitarias para la creación y configuración de parámetros de los API's utilizando la fábrica de API's y los resultados que devuelve la capa de filtro de datos genérico.	Alta
Como aplicación. Necesito crear pruebas unitarias para la carga de documentos de formato doc, docx, rtf o txt al servidor.	Alta
Como aplicación. Necesito crear pruebas unitarias para la creación de la tarea para analizar el plagio en un documento de texto utilizando la capa de Algoritmo de búsqueda de plagio aplicado a documentos encontrados con el API.	Alta
Como Usuario. Necesito probar un flujo completo de ejecución de la aplicación de escritorio: Seleccionar Documento – Seleccionar API's de búsqueda a utilizar – Analizar Documento – Ver Resultados de Análisis de Plagio del Documento Analizado.	Alta

3.4.1.5 Sprint de desarrollo de cuadros comparativos ponderados basados en los criterios comparativos prácticos definidos

3.4.1.5.1 Generación de un cuadro comparativo ponderado de las características de los API's de búsqueda basados en los criterios de comparación prácticos y las pruebas realizadas del prototipo

Se generó el siguiente cuadro comparativo de características (Tabla 3.24) basado en los resultados de las pruebas realizadas en el prototipo utilizando una conexión de ancho de banda de 2.5 Mbps y el tiempo que tomo el desarrollar la implementación del consumo del API; esto incluye conexión, autenticación y deserialización de resultados obtenidos desde el API tanto en XML como JSON.

Tabla 3.24: Información de cada API en base a los criterios de comparación práctica

<table>
<tr><th rowspan="2">Criterio</th><th colspan="3">API</th></tr>
<tr><th>Google</th><th>Yahoo</th><th>Bing</th></tr>
<tr><td>Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos en la primera ejecución</td><td>Párrafos:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.7901408</td></tr><tr><td>2</td><td>00:00:01.0695712</td></tr></table>Promedio:
0.929856</td><td>Párrafos:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.5826721</td></tr><tr><td>2</td><td>00:00:00.5264082</td></tr></table>Promedio:
0.55454015</td><td>Párrafos:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.3256152</td></tr><tr><td>2</td><td>00:00:00.3080781</td></tr></table>Promedio:
0.31684665</td></tr>
<tr><td>Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos</td><td>Párrafo 1:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.6857551</td></tr><tr><td>2</td><td>00:00:00.3714882</td></tr><tr><td>3</td><td>00:00:00.4053012</td></tr></table></td><td>Párrafo 1:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.9441303</td></tr><tr><td>2</td><td>00:00:00.5920898</td></tr><tr><td>3</td><td>00:00:00.5920898</td></tr></table></td><td>Párrafo 1:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.3899430</td></tr><tr><td>2</td><td>00:00:00.3296162</td></tr><tr><td>3</td><td>00:00:00.2587267</td></tr></table></td></tr>
</table>

<table>
<tr>
<td>ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución</td>
<td>Párrafo 2:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.67 50575</td></tr><tr><td>2</td><td>00:00:00.43 64794</td></tr><tr><td>3</td><td>00:00:00.44 16161</td></tr></table>
Promedio:
0.50261625</td>
<td>Párrafo 2:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.91 88962</td></tr><tr><td>2</td><td>00:00:00.57 30837</td></tr><tr><td>3</td><td>00:00:00.57 30837</td></tr></table>
Promedio:
0.69889558</td>
<td>Párrafo 2:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.36 25824</td></tr><tr><td>2</td><td>00:00:00.31 98854</td></tr><tr><td>3</td><td>00:00:00.23 07842</td></tr></table>
Promedio:
0.31525632</td>
</tr>
<tr>
<td>Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos en la primera ejecución</td>
<td>Párrafos:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:01.10 20152</td></tr><tr><td>2</td><td>00:00:01.25 18923</td></tr><tr><td>3</td><td>00:00:01.41 39194</td></tr><tr><td>4</td><td>00:00:00.74 58724</td></tr><tr><td>5</td><td>00:00:00.81 61962</td></tr></table>
Promedio:
1.0659791</td>
<td>Párrafos:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:02.32 77915</td></tr><tr><td>2</td><td>00:00:01.99 34813</td></tr><tr><td>3</td><td>00:00:01.19 78095</td></tr><tr><td>4</td><td>00:00:02.84 13532</td></tr><tr><td>5</td><td>00:00:01.18 96265</td></tr></table>
Promedio:
1.9100124</td>
<td>Párrafos:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.53 34593</td></tr><tr><td>2</td><td>00:00:01.12 55649</td></tr><tr><td>3</td><td>00:00:00.32 57184</td></tr><tr><td>4</td><td>00:00:00.38 45242</td></tr><tr><td>5</td><td>00:00:00.73 36214</td></tr></table>
Promedio:
0.62057764</td>
</tr>
<tr>
<td>Tiempo promedio de espera en el API de búsqueda utilizando un</td>
<td>Párrafo 1:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.30 61454</td></tr><tr><td>2</td><td>00:00:00.31 99443</td></tr></table></td>
<td>Párrafo 1:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.66 68417</td></tr><tr><td>2</td><td>00:00:00.57 61986</td></tr></table></td>
<td>Párrafo 1:
<table><tr><th>#</th><th>Tiempo</th></tr><tr><td>1</td><td>00:00:00.21 59238</td></tr><tr><td>2</td><td>00:00:00.61 26855</td></tr></table></td>
</tr>
</table>

documento de 5 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución	3 | 00:00:00.4738165	3 | 00:00:00.5920078	3 | 00:00:00.4199760
	Párrafo 2: \# | Tiempo 1 | 00:00:00.4927321 2 | 00:00:00.3134329 3 | 00:00:00.2849101	Párrafo 2: \# | Tiempo 1 | 00:00:00.6124973 2 | 00:00:00.5751418 3 | 00:00:01.9932104	Párrafo 2: \# | Tiempo 1 | 00:00:00.6872062 2 | 00:00:00.2217245 3 | 00:00:00.2811813
	Párrafo 3: \# | Tiempo 1 | 00:00:00.8381683 2 | 00:00:00.4855918 3 | 00:00:00.4555771	Párrafo 3: \# | Tiempo 1 | 00:00:01.1929010 2 | 00:00:01.2948658 3 | 00:00:01.3185244	Párrafo 3: \# | Tiempo 1 | 00:00:00.5247507 2 | 00:00:00.4644014 3 | 00:00:00.2251594
	Párrafo 4: \# | Tiempo 1 | 00:00:00.5683249 2 | 00:00:00.5706996 3 | 00:00:00.3101865	Párrafo 4: \# | Tiempo 1 | 00:00:01.4363373 2 | 00:00:02.0419525 3 | 00:00:01.2917953	Párrafo 4: \# | Tiempo 1 | 00:00:00.4515697 2 | 00:00:00.3150400 3 | 00:00:00.4094704
	Párrafo 5: \# | Tiempo 1 | 00:00:00.90	Párrafo 5: \# | Tiempo 1 | 00:00:02.02	Párrafo 5: \# | Tiempo 1 | 00:00:00.34

	<table><tr><td></td><td>18565</td></tr><tr><td>2</td><td>00:00:00.7607695</td></tr><tr><td>3</td><td>00:00:00.6764006</td></tr></table> Promedio: **0.51723707**	<table><tr><td></td><td>00163</td></tr><tr><td>2</td><td>00:00:01.5341560</td></tr><tr><td>3</td><td>00:00:00.6904994</td></tr></table> Promedio: **1.18912971**	<table><tr><td></td><td>73671</td></tr><tr><td>2</td><td>00:00:00.4322193</td></tr><tr><td>3</td><td>00:00:00.6136305</td></tr></table> Promedio: **0.41482039**
Similitud de los sitios consultados mediante el API vs los de la web del proveedor del API	4 Sitios similares a la web del proveedor Google	1 Sitio similar a la web del proveedor Yahoo España	0 Sitios similares a la web del proveedor Bing
Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 2 párrafos	00:00:03.1846506	00:00:01.8437099	00:00:02.7675855
Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 5	00:00:06.8579393	00:00:04.0138035	00:00:32.6741539

párrafos			

Se generó el siguiente cuadro comparativo práctico ponderado de características (Tabla 3.25) basado en las ponderaciones definidas en el punto 3.4.1.4.2 de este documento.

Tabla 3.25: Cuadro comparativo práctico resultante de cada API basados en el análisis de la ejecución del prototipo básico de una aplicación anti plagio de textos en documentos y los criterios de comparación práctica

Criterio	API		
	Google	Yahoo	Bing
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos en la primera ejecución	10/30	20/30	20/30
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución	20/30	10/30	20/30
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos en la primera ejecución	20/30	10/30	20/30
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución	30/30	20/30	30/30
Similitud de los sitios consultados mediante el API vs los de la web del proveedor del API	30/30	15/30	0/30
Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 2 párrafos	0/20	20/20	10/20
Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 5 párrafos	20/20	10/20	0/20
Total	**130/190**	**105/190**	**100/190**
Total API / 100	**68,42**	**55,26**	**52,63**

Los resultados obtenidos demuestran que el mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos en el ámbito práctico es el API de Google (Google Custom SearchAPI Json/Atom) con un resultado de 68,42 luego le sigue el API de Yahoo(Yahoo BOSS API) con 55,26y al final el API de Bing (BingAPI) con 52,63.

Google Custom Search Json/Atom API es el mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos en el ámbito práctico principalmente por sus tiempos de respuesta en el API y porque los sitios devueltos por el mismo son los más aproximados al documento original.

3.5 Sprint de resultados del estudio comparativo

3.5.1 Definición del mejor API de búsqueda para el desarrollo de aplicación anti plagio de textos en documento basado en el estudio práctico y teórico

Basados en los estudios comparativos prácticos y teóricos de los API's de búsqueda de Google, Yahoo y Bing se definió el mejor API para el desarrollo de un aplicativo anti plagio de textos en documentos.

Tabla 3.26: Peso en porcentaje del estudio comparativo teórico y práctico

Estudio	Peso en porcentaje
Comparativo teórico	20%
Comparativo práctico	80%
Total	**100%**

El estudio comparativo práctico tiene más peso que el teórico ya que este demuestra objetivamente el funcionamiento del API de búsqueda analizado para este tipo de aplicaciones. Esta ponderación está basada en la teoría de Pareto. Los resultados obtenidos en los estudios teóricos y prácticos para los API de búsqueda de Google, Yahoo y Bing son los mostrados en la Tabla 3.27.

Tabla 3.27: Resultados del estudio comparativo teórico y práctico

Estudio/API 100%	Google	Yahoo	Bing

Comparativo teórico	74,47	78,30	70,21
Comparativo práctico	68,42	55,26	52,63

Los resultados tomando en cuenta el peso porcentual utilizando la fórmula $Y = \frac{X*20}{100}$ y la fórmula $Y = \frac{X*80}{100}$ son los mostrados en la Tabla 3.28.

Tabla 3.28: Resultados en porcentaje del estudio comparativo teórico y práctico basados en el peso establecido para cada estudio

Estudio/API 100%	Google	Yahoo	Bing
Comparativo teórico	14,89/20	15,66/20	14,04/20
Comparativo práctico	54,73/80	44,20/80	42,10/80
Total	**69,62/100**	**59,86/100**	**56,14/100**

Los resultados obtenidos demuestran que el mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos basados en el análisis práctico como teórico es el **API de Google (Google Custom Search API Json/Atom) con un resultado de 69,62** luego le sigue el API de Yahoo (Yahoo BOSS API) con 59,86 y al final el API de Bing (Bing API) con 56,14

3.6 Sprint dediseño e implementación del prototipo de software final de una aplicación anti plagio de textos en documentos

3.6.1 Definición del Product Backlog para el desarrollo del prototipo de software final de una aplicación anti plagio de textos en documentos

3.6.1.1 Herramienta para administración de proyectos ágiles

Para la administración del proyecto con SCRUM y todos los artefactos que esta metodología propone; se está utilizando la herramienta VersionOne en el link: https://www16.v1host.com/DGAMC/

3.6.1.2 Product Backlog

El Product Backlog está definido más detalladamente en la herramienta VersionOne:

Tabla 3.29: Product Backlog para el desarrollo del prototipo final

Título	Dueño	Prioridad	Esfuerzo
Diseño de la arquitectura del prototipo de software final utilizando UML	Jorge	Alta	300,00
Administración de usuarios	Jorge	Alta	200,00
Repositorio de documentos	Jorge	Alta	200,00
Implementación gráfica web con ASP.NET MVC 3	Jorge	Alta	200,00
Pruebas prototipo final	Jorge	Alta	200,00

3.6.1.3 Sprint Backlog, requisitos específicos y calendario del prototipo de software final

3.6.1.3.1 Sprint Backlog

El Sprint Backlog está definido más detalladamente en la herramienta VersionOne.

Tabla 3.30: Sprint Backlog para el desarrollo del prototipo final

# Sprint	Titulo	Dueño	# de historias	Esfuerzo
3	Diseño, desarrollo y pruebas del prototipo de software final de una aplicación anti plagio de textos en documentos	Jorge	5,00	1100,00

Tabla 3.31: Product Backlog para el desarrollo del prototipo final con prioridad, dueño, esfuerzo y número de sprint

Título	Dueño	Prioridad	Esfuerzo	# Sprint
Diseño de la arquitectura del prototipo de software final utilizando UML	Jorge	Alta	300,00	3
Administración de usuarios	Jorge	Alta	200,00	3
Repositorio de documentos	Jorge	Alta	200,00	3
Implementación gráfica web con ASP.NET MVC 3	Jorge	Alta	200,00	3
Pruebas prototipo final	Jorge	Alta	200,00	3

3.6.1.4 Sprint de diseño, desarrollo y pruebas del prototipo de software final de una aplicación anti plagio de textos en documentos

3.6.1.4.1 Diseño de la arquitectura del prototipo de software final utilizando UML

Los diagramas de casos de uso, secuencia, clases, datos, componentes y despliegue fueron diseñados utilizando la herramienta Power Designer versión 16. Estos diagramas se encuentran en el archivo MODELOS_TESIS_PLAGIO_FINAL.prj y en el documento de la especificación de requisitos con la norma IEEE 830 con el nombre RequerimientosV2Final.docx dentro del CD de contenidos

3.6.1.4.2 Administración de usuarios

Encargada de realizar las conexiones a la base de datos y la lógica de administración de usuario.

El user story de esta etapa de desarrollo es la mostrada en la Tabla 3.32.

Tabla 3.32: User Stories para laadministración de usuarios

User Story	Prioridad
Como aplicación. Necesito conectarme a la base de datos SQL Server	Alta
Como aplicación. Necesito tener entidades que reflejen las propiedades de las tablas de la base de datos y las relaciones entre ellas	Alta
Como aplicación. Necesito una interfaz de acceso genérica para las operaciones CRUD a la base	Alta
Como aplicación. Necesito tener modelos tipo POCO que reflejen las propiedades y relaciones de las entidades creadas	Alta
Como aplicación. Necesito saber cuáles son los campos clave de cada uno de los modelos	Alta
Como aplicación. Necesito que los modelos implementen las operaciones de insertar, actualizar, eliminar, obtener uno y obtener todos	Alta
Como aplicación. Necesito que el modelo de Usuario puede realizar el proceso de login	Alta
Como aplicación. Necesito saber que roles tiene el usuario	Alta

3.6.1.4.3 Repositorio de documentos

Encargada de subir los documentos del usuario, analizarlos y verificar el plagio estado de los documentos.

El user story de esta etapa de desarrollo es la mostrada en la Tabla 3.33.

Tabla 3.33: User Stories para el repositorio de documentos

User Story	Prioridad
Como aplicación. Necesito saber en qué estado está cierto documento subido por el usuario.	Alta
Como aplicación. Necesito devolver los resultados del documento analizado con el API de búsqueda utilizando una interfaz que los abstraiga	Alta
Como aplicación. Necesito almacenar la información de los documentos en la base de datos por usuario	Alta
Como aplicación. Necesito almacenar la información del análisis de los documentos en la base de datos por usuario	Alta
Como aplicación. Necesito almacenar los documentos en el servidor por usuario	Alta

3.6.1.4.4 Implementación gráfica web ASP.NET MVC 3

Capa superior encargada de la interfaz gráfica con la que interviene el usuario con la aplicación realizada en ASP.NET MVC 3.

El user story de esta etapa de desarrollo es la mostrada en la Tabla 3.34.

Tabla 3.34: User Stories para la capa de interfaz de usuario final en ASP.NET MVC 3

User Story	Prioridad
Como aplicación. Necesito poder mostrar los menús de la aplicación según el rol del usuario	Alta
Como aplicación. Necesito poder restringir acciones que puede ejecutar el usuario en la aplicación según el rol del mismo	Alta
Como Administrador. Necesito poder eliminar usuarios que no tengan documentos subidos al sistema	Alta
Como Usuario o Administrador. Necesito poder modificar mis datos de la	Alta

cuenta	
Como Usuario. Necesito poder subir documentos al aplicativo	Alta
Como Usuario. Necesito poder consultar mis documentos subidos	Alta
Como Usuario. Necesito poder analizar plagio en mis documentos subidos	Alta
Como Usuario. Necesito poder consultar mis documentos analizados	Alta
Como Usuario. Necesito poder consultar los resultados del análisis de plagio de mis documentos	Alta

3.6.1.4.5 Pruebas

Capa encargada de realizar las pruebas unitarias de cada uno de los componentes del prototipo de software final y pruebas de funcionalidad para verificar el correcto funcionamiento del prototipo.

El user story de esta etapa de desarrollo es la mostrada en la Tabla 3.35.

Tabla 3.35: User Stories para las pruebas del prototipo final

User Story	Prioridad
Como aplicación. Necesito crear una prueba unitaria para la conexión a la base de datos	Alta
Como aplicación. Necesito una prueba unitaria para la subida de documentos por el usuario al servidor	Alta
Como aplicación. Necesito una prueba unitaria para la modificación de las cuentas de los usuarios	Alta
Como aplicación. Necesito una prueba unitaria para la eliminación de usuarios que no tengan documentos subidos al sistema	Alta
Como aplicación. Necesito una prueba unitaria para mostrar los documentos subidos por el usuario	Alta
Como aplicación. Necesito una prueba unitaria para mostrar los documentos analizados por el usuario	Alta
Como aplicación. Necesito una prueba unitaria ejecutar el proceso de análisis de plagio de los documentos subidos por el usuario	Alta
Como aplicación. Necesito una prueba unitaria para mostrar los resultados del análisis de plagio de los documentos analizados por el usuario	Alta

Como Usuario. Necesito probar un flujo completo de ejecución de la aplicación web: Login – Modificar Datos Cuenta– Subir Documento – Ver Documento Subido – Analizar Documento – Ver Documento Analizado – Ver Resultados de Análisis de Plagio del Documento Analizado	Alta
Como Administrador. Necesito probar un flujo completo de ejecución de la aplicación web: Login – Modificar Datos Cuenta– Eliminar Usuario sin Documentos	Alta

CAPÍTULO IV

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Las aplicaciones anti plagio de textos en documentos tienen una arquitectura definida en 2 componentes elementales: Fuente de comparación ya se

mediante la implementación de un Crawler para obtener fuentes de internet o una base de datos documental; y el analizador de similitudes en textos.

- XML y JSON son los formatos más utilizados en servicios web REST y en los API's de búsqueda, por ser sencillos de leer y escribir además de utilizar poco ancho de banda.
- Cada API de búsqueda tiene sus particularidades por ejemplo: Yahoo y Bing pueden buscar en distintas fuentes o servicios como web, imágenes, blogs noticias; mientras que Google es más preciso en los resultados que devuelve y posee un filtro para quitar duplicados.
- El cuadro comparativo creado para el estudio se basó en la documentación técnica de los API's y las pruebas en un prototipo básico de software, los criterios de comparación tanto práctico como teórico fueron definidos en base al aporte del criterio al desarrollo de aplicaciones anti plagio de textos en documentos.
- Los API's de búsqueda simplifican la tarea de buscar fuentes de plagio en sitios de internet.
- El API de Google es el mejor API para el desarrollo de aplicaciones anti plagio de textos en documentos en el ámbito práctico principalmente por sus tiempos de respuesta y porque los sitios devueltos por el API son muy aproximados a la fuente real.
- El API de Yahoo es el mejor API para el desarrollo de aplicaciones anti plagio de textos en documentos en el ámbito teórico ya que tiene más opciones de personalización para la búsqueda, como también más tipos de fuentes en donde buscar. Imágenes, Blogs, entre otros.
- El API de Bing a pesar de ser el que menores tiempos de respuesta genera y el único API gratuito, este devuelve resultados poco o nada relacionados a la fuente real, haciendo que el algoritmo de diff se demore más.
- El mejor API para el desarrollo de aplicaciones anti plagio de textos en documentos basados en el estudio comparativo teórico y práctico es el de Google llamado Google Custom Search API Json/Atom.
- El algoritmo diff tiene incidencia directa en el rendimiento del aplicativo porque es el encargado de verificar el plagio entre el documento original y las fuentes de comparación.

- La metodología SCRUM con XP y UML permitió generar un prototipo sencillo, eficiente, debidamente probado y bien documentado, además de que se cumplió con las fechas estipuladas del cronograma.
- El uso de diagramas UML con SCRUM y XP no es un requisito indispensable si el aplicativo es pequeño.
- ASP.NET MVC 3 es una tecnología sencilla de utilizar que utiliza un modelo de programación muy distinto a ASP.NET. Permite realizar pruebas unitarias fácilmente debido al patrón que usa.

4.2 Recomendaciones

- Utilizar un API de búsqueda para que un Crawler no busque en toda la Internet por fuentes de comparación.
- Usar JSON como formato para consumir los API's de búsqueda ahorra más ancho de banda que XML y es más sencillo de analizar[88].

[88]JSON ocupa menos ancho de banda que XML: http://ignoranto.info/articulos/ventajas-de-json-sobre-xml-en-el-desarrollo-web/,http://stackoverflow.com/questions/3075265/differences-advantages-of-using-json-over-xml-or-vice-versa

- Buscar otras alternativas de API's de búsqueda para verificar si cumplen con los requisitos para el desarrollo de este tipo de aplicaciones.
- Definir más criterios de comparación teórica y práctica basados en experiencia de terceros u otros estudios realizados sobre el tema.
- Implementar un algoritmo diff más eficiente para mejorar el rendimiento de la aplicación.
- Implementar una versión de detección de plagio de imágenes utilizando el tipo de fuente Imagen de los API de búsqueda de Yahoo o Bing
- No utilizar el API de Google para uso en un ambiente de producciónhasta que este se encuentre en una versión estable. Actualmente es una versión Labs.
- Considerar un costo por la utilización del API de Bing si se planteara utilizar en un ambiente de producción, ya que esta versión será gratuita por un periodo de tiempo.
- Utilizar solo los diagramas UML necesarios cuando se utiliza la metodología Scrum con XP.
- Usar una herramienta CASE para la gestión de la metodología Scrum para poder tener un mayor control del proyecto.
- Utilizar ASP.NET MVC en todo tipo de aplicaciones web grandes o pequeñas por su facilidad de uso, extensa documentación y por ser de código abierto.

BIBLIOGRAFÍA

LIBROS:

- RESTful Web Services - Web services for the real world por Leonard Richardson y Sam Ruby
- Microsoft Visual Studio 2010: A Beginner's Guide (A Beginners Guide) por Joseph
- Learning XML por Erik T. Ray
- HTTP Pocket Reference: Hypertext Transfer Protocol [Paperback] por Clinton Wong

REFERENCIAS WEB:

- Página oficial del API de búsqueda de Bing, [En línea], 11-01-2012, http://www.bing.com/developers
- Gráfico metodología SCRUM, [En línea], 11-01-2012, http://www.clubdesarrolladores.com/articulos/mostrar/63-metodologia-scrum
- Página oficial del API de búsqueda de Google Custom Search, [En línea], 11-01-2012,http://code.google.com/intl/es-ES/apis/customsearch/docs/dev_guide.html
- Página oficial del API de búsqueda de Yahoo BOSS API, [En línea], 11-01-2012,http://developer.yahoo.com/search/
- Página oficial de Visual Studio 2010 Express Edition, [En línea], 11-01-2012,http://www.microsoft.com/express/Downloads/
- LibraryThing JSON Books API, [En línea], 11-01-2012,http://www.librarything.com/wiki/index.php/JSON_Books_API_:_Standard_library
- UML Case Diagramming Guidelines, [En línea], 11-01-2012,http://www.agilemodeling.com/style/useCaseDiagram.htm
- ASP.NET MVC 3, [En línea],http://www.asp.net/mvc/mvc3
- Sintaxis Razor para ASP.NET MVC, [En línea], 11-01-2012,http://www.asp.net/webmatrix/tutorials/2-introduction-to-asp-net-web-programming-using-the-razor-syntax
- Estándar OAuth, [En línea], 11-01-2012,http://en.wikipedia.org/wiki/OAuth
- Utilidad Diff, [En línea], 11-01-2012,http://en.wikipedia.org/wiki/Diff
- Utilidad Diff3, [En línea], 11-01-2012,http://en.wikipedia.org/wiki/Diff3
- Metodología XP, [En línea], 11-01-2012,http://www.esp.uem.es/jccortizo/xp.pdf
- Metodología de Programación Extrema historia y definición, [En línea], 11-01-2012, http://wikis.uca.es/wikiCE/index.php/Extreme_programming
- Código de la aplicación PLAGIARISM diseñado en JAVA, [En línea], 11-01-2012,http://open.jira.com/source/browse/PLAGIARISM/trunk/plagiarism-core/src/test/java/com/google/code/antiplagiarism/persistence/hib_impl/TestFieldDatabaseValueDaoHibernateImpl.java?r1=511&r2=512

- Comparación de herramientas de detección de plagio, [En línea], 11-01-2012,http://www.cs.uu.nl/research/techreps/repo/CS-2010/2010-015.pdf
- Complemento Visual NUnit, [En línea], 11-01-2012,http://visualstudiogallery.msdn.microsoft.com/c8164c71-0836-4471-80ce-633383031099
- Como utilizar Visual NUnit en VS2010, [En línea], 11-01-2012,http://stackoverflow.com/questions/4881282/nunit-with-vs2010-and-mvc3
- Como utilizar NUnit en VS2010, [En línea], 11-01-2012,http://www.marthijnvandenheuvel.com/2010/06/09/using-nunit-in-visual-studio-2010/
- Utilizar NUnit en VS2010, [En línea], 11-01-2012,http://msmvps.com/blogs/kevinmcneish/archive/2010/04/21/using-nunit-in-visual-studio-2010.aspx
- Utilizar NUnit, [En línea], 11-01-2012,http://www.elguille.info/colabora/puntoNET/giovannyf_NUnit.htm
- Información del sitio original de NUnit, [En línea], 11-01-2012,http://www.nunit.org/index.php?p=quickStart&r=2.2.10
- Foro de programación paralela en .NET, [En línea],http://blogs.msdn.com/b/pfxteam/
- Cuando utilizar Parallel.ForEach y cuando PLinq, [En línea], 11-01-2012,http://blogs.msdn.com/b/pfxteam/archive/2010/04/21/9997559.aspx
- Parallel.For utiliza o no una tarea por iteración, [En línea], 11-01-2012,http://blogs.msdn.com/b/pfxteam/archive/2009/05/26/9641563.aspx
- Parallel.ForEach para el marco de trabajo.NET 3.0, [En línea], 11-01-2012,http://www.codeproject.com/KB/dotnet/PoorMansParallelForEach.aspx
- Ejemplos de números de Fibonacci utilizando Parallel.ForEach, [En línea], 11-01-2012,http://blog.magenic.com/blogs/jefff/archive/2008/03/18/Parallel.ForEach-Sample_3A00_-Fibonacci-Numbers.aspx
- Hilos en C#, programación en paralelo, [En línea], 11-01-2012,http://www.albahari.com/threading/part5.aspx

- ParallelLoopState y Parallel.ForEach, [En línea], 11-01-2012,http://odetocode.com/Blogs/scott/archive/2011/10/10/parallelloopstate-and-parallel-foreach.aspx
- Un Crawler simple con sockets, [En línea], 11-01-2012,http://www.codeproject.com/KB/IP/Crawler.aspx
- ASP.NET MVC enlace del modelo, [En línea], 11-01-2012,http://blog.gfader.com/2010/05/aspnet-mvc-model-binding-to-list-or-how.html
- ASP.NET MVC enlace del modelo a una lista, [En línea], 11-01-2012,http://kristofmattei.be/2009/10/19/asp-net-mvc-model-binding-to-a-list/
- ASP.NET MVC atributos de manejo de errores y salida de cache, [En línea], 11-01-2012,http://www.pnpguidance.net/post/ASPNETMVCFrameworkPreview4HandleErrorAuthorizeOutputCacheActionFilterAttributes.aspx
- Autenticación de usuarios con autenticación de formularios en ASP.NET MVC, [En línea], 11-01-2012,http://www.asp.net/mvc/tutorials/authenticating-users-with-forms-authentication-cs
- Autenticación de usuarios en ASP.NET MVC con roles personalizados, [En línea], 11-01-2012,http://forums.asp.net/p/1428467/3192831.aspx
- Código fuente de un atributo personalizados de autorización , [En línea], 11-01-2012,https://github.com/machine/msmvc/blob/20453ecf4a87a3be9e0ce2f3ef26e1fd6c89a2d7/Mvc/AuthorizeAttribute.cs
- Página de descargas del proyecto Yahoo Finance con autenticación OAuth, [En línea], 11-01-2012,http://code.google.com/p/yahoo-finance-managed/downloads/list
- Como utilizar OAuth en C#, [En línea], 11-01-2012,http://blog.stevienova.com/2008/04/19/oauth-getting-started-with-oauth-in-c-net/
- Autenticarse con Google OAuth, [En línea], 11-01-2012,http://heartofangel.com/tutorial-authenticating-with-google-oauth/

- Como utilizar la autenticación OAuth para aplicaciones de Twitter en Windows, [En línea], 11-01-2012,http://www.byteblocks.com/post/2010/06/21/Using-OAuth-Authentication-For-Twitter-Windows-Applications.aspx
- Construyendo un página básica de devuelta de llamada para OAuth, [En línea], 11-01-2012,http://msdn.microsoft.com/en-us/library/ff752581.aspx
- Tipos de fuente Web en Bing Search API, [En línea], 11-01-2012,http://msdn.microsoft.com/en-us/library/dd250845.aspx
- Protocolos soportados por Bing Search API, [En línea], 11-01-2012,http://msdn.microsoft.com/en-us/library/dd250909.aspx
- Documentación del API de búsqueda de Bing versión 2, [En línea], 11-01-2012,http://msdn.microsoft.com/en-us/library/dd251056.aspx
- Introducción del API de búsqueda de Bing versión 2, [En línea], 11-01-2012,http://msdn.microsoft.com/en-us/library/dd900818.aspx
- Tipo de fuente Translation en Bing Search API, [En línea], 11-01-2012,http://msdn.microsoft.com/en-us/library/dd877872.aspx
- Tipo de fuente Video en Bing Search API, [En línea], 11-01-2012,http://msdn.microsoft.com/en-us/library/dd560940.aspx
- Implementación JSON para el tipo de fuente Translation en Bing Search API, [En línea], 11-01-2012,http://msdn.microsoft.com/en-us/library/dd877892.aspx
- Implementación JSON para el tipo de fuente Web en Bing Search API, [En línea],11-01-2012,http://msdn.microsoft.com/en-us/library/dd250861.aspx
- Página principal de desarrolladores del Bing Search API, [En línea], 11-01-2012,http://www.bing.com/toolbox/bingdeveloper/
- Especificación de Yahoo BOSS API , [En línea], 11-01-2012,http://developer.yahoo.com/search/boss/boss_api_guide/v2_api_spec.html
- Ejemplos de peticiones al Yahoo BOSS API, [En línea], 11-01-2012,http://developer.yahoo.com/search/boss/boss_api_guide/v2_examples.html

- Tipos de servicio Web en Yahoo BOSS API, [En línea], 11-01-2012,http://developer.yahoo.com/search/boss/boss_api_guide/webv2_service.html
- Campos de respuesta del tipo de servicio Web en Yahoo BOSS API, [En línea], 11-01-2012,http://developer.yahoo.com/search/boss/boss_api_guide/webv2_response.html
- Regiones y lenguajes recomendados en Yahoo BOSS API, [En línea], 11-01-2012,http://developer.yahoo.com/search/boss/boss_api_guide/webv2_lang_region.html
- Peticiones al API de Yahoo BOSS con múltiples servicios y palabras clave, [En línea], 11-01-2012,http://developer.yahoo.com/search/boss/boss_api_guide/v2multi_srv_kw.html
- Sintaxis para realizar peticiones a alguno de los servicios del Yahoo BOSS API, [En línea], 11-01-2012,http://developer.yahoo.com/search/boss/boss_api_guide/msk_syntax2.html
- Argumentos universales del API de Yahoo BOSS, [En línea], 11-01-2012,http://developer.yahoo.com/search/boss/boss_api_guide/v2_univer_api_args.html
- Precios del API de Yahoo BOSS API basados en el tipo de servicio y número de peticiones, [En línea], 11-01-2012,http://developer.yahoo.com/search/boss/#pricing
- Aplicación web para manejo de proyectos de software utilizando , [En línea], metodologías agiles, [En línea], 11-01-2012,http://www.agilefant.org/wiki/display/AEF/Features
- Aplicación web beta para manejo de proyectos de software utilizando Scrum, [En línea], 11-01-2012,http://pangoscrum.com/en-US/sign_up/new
- Aplicación web gratuita en java para manejo de proyectos de software utilizando Scrum, [En línea], 11-01-2012,http://www.icescrum.org/en/download-en/

- Product Backlog de Scrum, [En línea], 11-01-2012,http://www.navegapolis.net/content/view/622/59/
- Aprendiendo Scrum. El Product Backlog, [En línea], 11-01-2012,http://www.mountaingoatsoftware.com/scrum/product-backlog
- Aplicación web de pago para manejo de proyectos de software utilizando Scrum, [En línea], 11-01-2012,http://www.scrumdo.com/plans
- Aplicación de escritorio gratuita para manejo de proyectos de software utilizando Scrum o XP, [En línea], 11-01-2012,http://sprintometer.com/
- Aplicación web de pago para manejo de proyectos de software utilizando Scrum, [En línea], 11-01-2012,http://www.bananascrum.com/home
- Lista de aplicaciones web y de escritorio open source para manejo de proyectos de software utilizando Scrum, [En línea], 11-01-2012,http://www.opensourcescrum.com/
- Aplicación de escritorio gratuita para manejo de proyectos de software utilizando Scrum, [En línea], 11-01-2012,http://www.scrum-factory.com/en-us/Default.aspx
- Aplicación web gratuita en java para manejo de proyectos de software utilizando Scrum, [En línea], 11-01-2012,http://agileexpress.sourceforge.net/screenshots.html
- 10 mejores aplicaciones gratuitas para el manejo de proyectos de software utilizando la metodología Scrum, [En línea], 11-01-2012,http://knowscrum.com/10-best-free-scrum-tools/
- Plantilla de un Product Backlog de Scrum, [En línea], 11-01-2012,http://agile101.net/2009/07/29/scrum-product-backlog-template-with-priority-overview/
- Plantillas y ejemplos de un Product Backlog de Scrum, [En línea], 11-01-2012,http://www.brighthub.com/office/project-management/articles/65535.aspx
- Aplicaciones mashups que utilizas en API de Yahoo BOSS , [En línea], 11-01-2012,http://www.programmableweb.com/api/yahoo-boss/mashups
- Construir un sitios de búsqueda utilizando el API de búsqueda de MSN, [En línea], 11-01-2012,http://www.fiftyfoureleven.com/weblog/web-development/programming-and-scripts/apis/msn-search-api

- Utilizar el API de búsqueda de MSN en PHP, [En línea], 11-01-2012,http://www.dyeager.org/post/2009/01/msn-live-search-api-v20-php
- Descripción del API de Yahoo BOSS, [En línea], 11-01-2012,http://www.programmableweb.com/api/yahoo-boss
- Descripción del Bing Search API, [En línea], 11-01-2012,http://www.programmableweb.com/api/microsoft-bing
- Descripción del Google Custom Search API, [En línea], 11-01-2012,http://www.programmableweb.com/api/google-custom-search
- Aplicación web de detección de plagio de texto, [En línea], 11-01-2012,http://www.plagiarismchecker.com/
- Aplicación web de detección de plagio de texto utilizando el API de búsqueda de Google y Bing, [En línea], 11-01-2012,http://www.plagium.com/
- Aplicación web de detección de plagio de texto, [En línea], 11-01-2012,http://searchenginereports.net/articlecheck.aspx
- Aplicación web de detección de plagio de texto, [En línea], 11-01-2012,http://www.articlechecker.com/
- Aplicación web de detección de plagio de texto utilizando el API de búsqueda de Google, Bing, Yahoo y otros, [En línea], 11-01-2012,http://www.plagiarized.org/
- Aplicación web de detección de plagio de texto utilizando el API de búsqueda de Google, Bing y Yahoo, [En línea], 11-01-2012,http://www.duplichecker.com/
- Aplicación de pago para crear diagramas de UML, [En línea], 11-01-2012,http://astah.net/editions/uml
- Aplicación gratuita para crear diagramas de UML, [En línea], 11-01-2012,http://www.softwareideas.net/
- Aplicación web de detección de plagio de texto utilizando el API de búsqueda de Yahoo, [En línea], 11-01-2012,http://www.reprintwriters.com/copyright-checker/check.php
- Aplicación open source para crear diagramas de UML, [En línea], 11-01-2012,http://staruml.sourceforge.net/en/
- Código fuente en PHP de consumo de APIs de Google en formato XML y JSON, [En línea], 11-01-2012,https://gist.github.com/721637

- Manual básico del Bing Search API, [En línea], 11-01-2012,http://www.bing.com/developers/s/API%20Basics.pdf
- Ejemplo de una petición al Bing Search API en XML, [En línea], 11-01-2012,http://api.search.live.net/xml.aspx?Appid=0CCE5FC6E5C59C7986F50BD4CBF9D92512D5D8EE&query=futbol&sources=web
- Implementación XML para el tipo de fuente Web en Bing Search API, [En línea], 11-01-2012,http://msdn.microsoft.com/en-us/library/dd250894.aspx
- Información básica del Google Custom Search API JSON/ATOM, [En línea], 11-01-2012,http://code.google.com/intl/es-AR/apis/customsearch/v1/overview.html
- Empezando con Google Custom Search API, [En línea], 11-01-2012,http://code.google.com/intl/es-AR/apis/customsearch/v1/getting_started.html
- Consola de API´s de Google, [En línea], 11-01-2012,https://code.google.com/apis/console/?api=customsearch&pli=1#project:268269739027:access
- Usando REST para invocar el Google Custom Search API, [En línea], 11-01-2012,http://code.google.com/intl/es-AR/apis/customsearch/v1/using_rest.html
- Administración de Google Custom Search API, [En línea], 11-01-2012,http://www.google.com/cse/manage/all
- Referencia XML de Google Custom Search API, [En línea], 11-01-2012,http://www.google.com/cse/docs/resultsxml.html
- Tipos de archivos que Google puede indexar, [En línea], 11-01-2012,http://www.google.com/support/webmasters/bin/answer.py?hl=en&answer=35287
- Referencia XML de Google Custom Search API, parámetros de búsqueda avanzados, [En línea], 11-01-2012,http://www.google.com/cse/docs/resultsxml.html#Advanced_Search_Query_Parameters
- Costos de Google Custom Search API , [En línea], 11-01-2012,https://code.google.com/intl/es-AR/apis/customsearch/v1/overview.html#Pricing

- Proyectos de desarrollador de YAHOO, [En línea], 11-01-2012,http://developer.apps.yahoo.com/projects/1kj9Eg70
- Como escribir un Crawler en C#, [En línea], 11-01-2012,http://www.thecodinghumanist.com/Content/HowToWriteAWebCrawlerInCSharp.aspx
- Leer y escribir en ficheros de texto en .NET, [En línea], 11-01-2012,http://www.elguille.info/NET/dotnet/leer_escribir_ficheros_texto.htm
- Sitio que transforma HTML en texto plano, [En línea], 11-01-2012,http://cgi.w3.org/cgi-bin/html2txt
- Página principal del proyecto open source Html Agility Pack, [En línea], 11-01-2012,http://htmlagilitypack.codeplex.com/
- Como convertir HTML a texto en C#, [En línea], 11-01-2012,http://stackoverflow.com/questions/731649/how-can-i-convert-html-to-text-in-c
- Aprendiendo posicionamiento CSS, [En línea], 11-01-2012,http://www.barelyfitz.com/screencast/html-training/css/positioning/
- Subir ficheros al servidor ASP.NET MVC, [En línea], 11-01-2012,http://geeks.ms/blogs/etomas/archive/2010/09/08/subir-ficheros-al-servidor-en-asp-net-mvc.aspx
- Un genérico y reusable algoritmo Diff en C#, [En línea], 11-01-2012,http://www.codeproject.com/KB/recipes/c__diff_algorithm.aspx
- Un genérico y reusable algoritmo Diff en C#, parte II, [En línea], 11-01-2012,http://www.codeproject.com/KB/recipes/diffengine.aspx
- Algoritmo de diferencia An O (ND) en C#, [En línea], 11-01-2012,http://www.mathertel.de/Diff/DiffDoku.aspx
- Página principal del proyecto open source google-diff-match-patch, [En línea], 11-01-2012,http://code.google.com/p/google-diff-match-patch/
- Página principal del proyecto open source DiffPlex, [En línea], 11-01-2012,http://diffplex.codeplex.com/wikipage?title=library&referringTitle=Home
- Definición de expresiones regulares, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/Expresi%C3%B3n_regular

- jQuery Ajax uploader plugin, [En línea], 11-01-2012,http://blog.stevensanderson.com/2008/11/24/jquery-ajax-uploader-plugin-with-progress-bar/
- Subida de archivos con Ajax, [En línea], 11-01-2012,http://valums.com/ajax-upload/
- Subida de archivos en MVC3 con Valumns Ajax File Upload, [En línea], 11-01-2012,http://stackoverflow.com/questions/4884920/mvc3-valums-ajax-file-upload
- jQuery File Upload en ASP.NET MVC sin flash, [En línea], 11-01-2012,http://aspzone.com/tech/jquery-file-upload-in-asp-net-mvc-without-using-flash/
- Uso de jQuery Ajax-upload con MVC3, [En línea], 11-01-2012,https://kiwidev.wordpress.com/2011/01/18/using-jquery-ajax-upload-with-mvc3/
- Subida de archivos en Razor con Uploadify jQuery plugin, [En línea], 11-01-2012,http://midnightprogrammer.net/post/Upload-Files-In-RAZOR-With-jQuery-Uploadify-Plugin.aspx
- Documentación de Uploadify, [En línea], 11-01-2012,http://www.uploadify.com/documentation/
- Evento onComplete de Uploadify, [En línea], 11-01-2012,http://www.uploadify.com/documentation/events/oncomplete-2/
- .NET 4.0 y el System.Threading.Tasks, [En línea], 11-01-2012,http://www.codethinked.com/net-40-and-systemthreadingtasks
- Usando un HashTable dentro de un Parallel.ForEach, [En línea], 11-01-2012,http://stackoverflow.com/questions/1657874/using-a-hashtable-inside-a-parallel-foreach
- Librería de tareas en paralelo, [En línea], 11-01-2012,http://msdn.microsoft.com/en-us/library/dd460717.aspx
- Paralelismo en .NET, [En línea], 11-01-2012,http://reedcopsey.com/2010/01/22/parallelism-in-net-part-4-imperative-data-parallelism-aggregation/

- Procesamiento paralelo, [En línea], 11-01-2012,http://stackoverflow.com/questions/6319682/parallel-process-a-intensive-io-function
- Clase ConcurrentDictionary(Of TKey, TValue) , [En línea], 11-01-2012,http://msdn.microsoft.com/en-us/library/dd287191.aspx
- Sitio web que contiene una lista de los sitios Web 2.0 más conocidos, [En línea], 11-01-2012,http://weba20.com/
- Web 2.0 Summit, [En línea], 11-01-2012,http://www.web2summit.com/web2011
- Mapa visual de la Web 2.0, [En línea], 11-01-2012,http://internality.com/web20/
- Introducción a los servicios web RESTful, [En línea], 11-01-2012,http://www.dosideas.com/noticias/java/314-introduccion-a-los-servicios-web-restful.html
- Definición de REST, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/Representational_State_Transfer
- Definición de HTTP, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- Protocolo HTTP, [En línea],http://es.kioskea.net/contents/internet/http.php3
- Definición de TCP, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/Transmission_Control_Protocol
- Protocolo TCP, [En línea],http://es.kioskea.net/contents/internet/tcp.php3
- Historia de XML, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/Extensible_Markup_Language#Historia
- Guía breve de tecnologías XML, [En línea], 11-01-2012,http://www.w3c.es/divulgacion/guiasbreves/tecnologiasxml
- Introducción a XML, [En línea], 11-01-2012,http://www.desarrolloweb.com/manuales/18/
- Definición de JSON, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/JSON
- Introducción a JSON, [En línea], 11-01-2012,http://json.org/json-es.html
- Nuevas características de ASP.NET MVC 3, [En línea], 11-01-2012,http://www.asp.net/mvc/mvc3#BM_Other_New_Features

- Introducción a ASP.NET MVC 3, [En línea], 11-01-2012,http://www.asp.net/mvc/tutorials/getting-started-with-mvc3-part1-cs
- Marco de trabajo de ASP.NET MVC, [En línea], 11-01-2012,http://en.wikipedia.org/wiki/ASP.NET_MVC_Framework
- Modelo vista controlador, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/Modelo_Vista_Controlador
- Introducción a ASP.NET MVC, [En línea], 11-01-2012,http://www.desarrolloweb.com/articulos/introduccion-asp-net-mvc-dotnet.html
- Video de MVC 3 Razor engine , [En línea], 11-01-2012,http://www.asp.net/mvc/videos/mvc-3-%E2%80%93-razor-view-engine
- Tutorial de Razor engine, [En línea], 11-01-2012,http://buildstarted.com/2011/03/07/razor-engine-v2-x/
- Introducción a Razor engine, [En línea], 11-01-2012,http://weblogs.asp.net/scottgu/archive/2010/07/02/introducing-razor.aspx
- Microsoft ASP.NET Razor view engine, [En línea], 11-01-2012,http://en.wikipedia.org/wiki/Microsoft_ASP.NET_Razor_View_Engine
- ASP.NET MVC comparación de motores de vistas, [En línea], 11-01-2012,http://stackoverflow.com/questions/1451319/asp-net-mvc-view-engine-comparison
- Anuncio de ASP.NET MVC 3 RC, [En línea], 11-01-2012,http://weblogs.asp.net/scottgu/archive/2010/11/09/announcing-the-asp-net-mvc-3-release-candidate.aspx
- Página principal del proyecto open source Razor engine, [En línea], 11-01-2012,http://razorengine.codeplex.com/
- Definición del Google SOAP Search API, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/Google_SOAP_Search_API
- Definición del estándar OpenSearch, [En línea], 11-01-2012,http://en.wikipedia.org/wiki/OpenSearch
- Definición del Google Search AJAX API, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/Google_search_ajax_api

- Aplicación web de detección de plagio en documentos, [En línea], 11-01-2012,http://www.compilatio.net/es/
- Aplicación más utilizada de detección de plagio en documentos, [En línea], 11-01-2012,https://www.turnitin.com/static/index.php
- Definición de detección de plagio, [En línea], 11-01-2012,http://en.wikipedia.org/wiki/Plagiarism_detection
- Que es plagio, [En línea], 11-01-2012,http://en.wikipedia.org/wiki/Plagiarism
- Definición de plagio, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/Plagio
- Que es Scrum, [En línea], 11-01-2012,http://www.proyectosagiles.org/que-es-scrum
- Beneficios de Scrum, [En línea], 11-01-2012,http://www.proyectosagiles.org/beneficios-de-scrum
- Fundamentos de Scrum, [En línea], 11-01-2012,http://www.proyectosagiles.org/fundamentos-de-scrum
- Requisitos para utilizar Scrum, [En línea], 11-01-2012,http://www.proyectosagiles.org/requisitos-de-scrum
- Definición de Scrum, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/Scrum#cite_note-1
- Definición de XP, [En línea], 11-01-2012,http://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema
- Explicación de XP, [En línea], 11-01-2012,http://www.willydev.net/descargas/prev/ExplicaXP.pdf
- Introducción a la programación extrema (XP) , [En línea], 11-01-2012,http://www.cristalab.com/blog/introduccion-a-la-programacion-extrema-c44013l/
- Definición de UML, [En línea], 11-01-2012,http://en.wikipedia.org/wiki/Unified_Modeling_Language
- Ingeniería de software y UML, [En línea], 11-01-2012,http://www.monografias.com/trabajos5/insof/insof.shtml
- Elementos de UML, [En línea], 11-01-2012,http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html

- XP podría ser embebido dentro de Scrum, [En línea], 11-01-2012,http://es.scribd.com/doc/36188311/sps81
- Scrum y XP desde las trincheras en Español, [En línea], 11-01-2012,http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf
- Introducción de Google Custom Search API, [En línea], 11-01-2012,http://code.google.com/intl/es-EC/apis/customsearch/docs/dev_guide.html
- Guía de Yahoo BOSS API, [En línea], 11-01-2012,http://developer.yahoo.com/search/boss/boss_api_guide/index.html

GLOSARIO DE TÉRMINOS

- Web 2.0: web colaborativa donde el contenido de los sitios es generado por los usuarios
- API: conjunto de llamadas a librerías o bibliotecas que abstraen la funcionalidad para ser utilizada por terceros
- Mash-up: aplicaciones que implementan API's de terceros dentro de las mismas
- ASP.NET MVC 3: tecnología de Microsoft basada en el marco de trabajo ASP.NET que implementa el patrón de diseño MVC
- MVC: patrón de diseño modelo, vista y controlador
- Bing: motor de búsqueda de Microsoft
- CMS: sistema de administración de contenidos
- CC: licencias aplicadas a sitios web que permiten utilizar, modificar y mejorar el contenido publicado en el mismo
- GPL: licencia general pública
- JSON: formato ligero para el intercambio de datos, fácil de leer por humanos
- XML: es un metalenguaje extensible basado en etiquetas
- Crawler: programa que inspecciona las páginas web de una forma metódica y automatizada
- TurnItIn: aplicación para la detección de plagio
- Búsqueda semántica: búsqueda comparativa de caracteres en un texto utilizando algoritmos de diferenciación
- HTTP: protocolo de aplicación utilizado ampliamente en internet. Utiliza la arquitectura cliente-servidor
- TCP: protocolo orientado a conexión encargado de asegurar la calidad de la conexión
- PDF: formato de documento portátil
- SCRUM: metodología ágil de desarrollo, que utiliza un proceso iterativo y está enfocado a la planificación de proyectos
- XP: metodología ágil de desarrollo, que utiliza un proceso iterativo y está enfocado en desarrollo del software
- HTML: lenguaje de marcado de hipertextos

- .NET: marco de trabajo de Microsoft para el desarrollo de aplicaciones
- UML: lenguaje gráfico que permite modelar, diseñar y documentar sistemas
- RESTful: servicio web orientado a recursos, sin estado que utiliza el protocolo HTTP
- OAuth: protocolo abierto para la autenticación segura a un API de una manera estándar y sencilla
- Diff: utilidad de Unix para obtener las diferencias entre dos archivos o carpetas
- Product Backlog: especificación de requisitos generales
- Sprint Backlog:
- Sprint: periodo de tiempo de entre 15 a 30 días que al culminar entrega un prototipo funcional de software
- CSS: hojas de estilo en cascada utilizados para el diseño de páginas web
- XHTML: HTML basado en XML
- RIA: aplicaciones con interfaces ricas
- AJAX: JavaScript asincrónico y XML
- JavaScript: permite la creación de scripts del lado del cliente
- URL: localizador uniforme de recursos
- RSS: formato XML para compartir contenido en la web
- ATOM: formato de redifusión basado en XML
- C++: lenguaje de programación de alto nivel con acceso directo al hardware
- Java: lenguaje de programación de alto nivel de código libre
- REST: transferencia de estado representacional
- CRUD: operaciones básicas realizadas a una base de datos: crear, leer, actualizar y eliminar
- WSDL: lenguaje de descripción de servicios web
- Cache: almacenamiento temporal
- SOAP: protocolo de acceso a objetos simples
- Windows Forms: tecnología de Microsoft para el desarrollo de aplicaciones de escritorio
- IDE: entorno integrado de desarrollo
- HTTP POST: crea un recurso en el servidor
- HTTP PUT: cambia el estado de un recurso

- HTTP GET: obtiene un recurso
- HTTP DELETE: elimina un recurso
- MIME: extensiones de correo de internet multipropósito
- SGML: lenguaje de marcado estándar generalizado
- SVG: gráficos vectoriales escalables
- Metalenguaje: lenguaje que se usa para hablar sobre otro lenguaje
- MathML: lenguaje matemático de marcado
- DTD: definición de tipos del documento
- XSD: esquema XML
- YAML: formato de intercambio de datos legibles por humanos
- Ruby: lenguaje de programación dinámico open source
- MS-PL: licencia permisiva de Microsoft
- ASP.NET: tecnología Microsoft para el desarrollo de aplicaciones web
- Postback: ida y vuelta al servidor
- OpenSearch: conjunto de tecnologías que permiten publicar resultados de búsqueda de forma estandarizada
- HMAC-SHA1: algoritmo de criptografía utilizado en el protocolo OAuth
- LCS: subsecuencia común más larga

BIOGRAFÍA

Nacido en Quito el 28 de diciembre de 1987.

Estudios de primaria en la escuela Borja 3 y finalizada en la escuela Marista en agosto del 2000.

Estudio de secundaria en el colegio Marista finalizado en agosto del2005 obteniendo el título de Bachiller en la especialidad de físico matemático.

Curso Cisco CCNA realizado en el CTT de la ESPE finalizado el mes de diciembre del 2008.

Curso Linux LPI realizado en el CTT de la ESPE finalizado el mes de marzo del 2009.

ANEXOS

ANEXO 1: Estándares de codificación para el lenguaje C#

Estándares Codificación.docx

ANEXO 2: Especificación de requisitos del prototipo básico de software

RequerimientosV2.docx

ANEXO 3: Especificación de Requisitos del prototipo final de software

RequerimientosV2Final.docx

ANEXO 4: Manual de usuario del prototipo final de software

Manual de Usuario del Aplicativo DeathPlagiarism.docx

ANEXO 5: Archivos de resultados del estudio comparativo práctico

Resultados API

ANEXO 6: Documentación técnica del aplicativo DeathPlagiarism

Documentación DeathPlagiarism

ANEXO 7: Ejemplo de respuesta JSON exitosa del API de Google Custom Search API JSON/Atom

```
200 OK
{
"kind": "customsearch#search",
"url": {
"type": "application/json",
"template":
```

```
      "https://www.googleapis.com/customsearch/v1?q\u003d{searchTerms}&num\u003d{
count?}&start\u003d{startIndex?}&hr\u003d{language?}&safe\u003d{safe?}&cx\u003d{
cx?}&cref\u003d{cref?}&sort\u003d{sort?}&alt\u003djson"
     },
     "queries": {
      "nextPage": [
       {
        "title": "Google Custom Search - flowers",
        "totalResults": 10300000,
        "searchTerms": "flowers",
        "count": 10,
        "startIndex": 11,
        "inputEncoding": "utf8",
        "outputEncoding": "utf8",
        "cx": "013036536707430787589:_pqjad5hr1a"
       }
      ],
      "request": [
       {
        "title": "Google Custom Search - flowers",
        "totalResults": 10300000,
        "searchTerms": "flowers",
        "count": 10,
        "startIndex": 1,
        "inputEncoding": "utf8",
        "outputEncoding": "utf8",
        "cx": "013036536707430787589:_pqjad5hr1a"
       }
      ]
     },
     "context": {
      "title": "Custom Search"
     },
```

```
"items": [
{
 "kind": "customsearch#result",
 "title": "Flower - Wikipedia, the free encyclopedia",
 "htmlTitle": "\u003cb\u003eFlower\u003c/b\u003e - Wikipedia, the free encyclopedia",
 "link": "http://en.wikipedia.org/wiki/Flower",
 "displayLink": "en.wikipedia.org",
 "snippet": "A flower, sometimes known as a bloom or blossom, is the reproductive structure found in flowering plants (plants of the division Magnoliophyta, ...",
 "htmlSnippet": "A \u003cb\u003eflower\u003c/b\u003e, sometimes known as a bloom or blossom, is the reproductive structure \u003cbr\u003e found in flowering plants (plants of the division Magnoliophyta, \u003cb\u003e...\u003c/b\u003e",
 "pagemap": {
  "RTO": [
   {
    "format": "image",
    "group_impression_tag": "prbx_kr_rto_term_enc",
    "Opt::max_rank_top": "0",
    "Opt::threshold_override": "3",
    "Opt::disallow_same_domain": "1",
    "Output::title": "\u003cb\u003eFlower\u003c/b\u003e",
    "Output::want_title_on_right": "true",
    "Output::num_lines1": "3",
    "Output::text1": "24 강으로분류한린네였다. 그후꽃은식물분류학상.",
    "Output::gray1b": "- 위키백과",
    "Output::no_clip1b": "true",
    "UrlOutput::url2": "http://en.wikipedia.org/wiki/Flower",
    "Output::link2": "위키백과 (영문)",
    "Output::text2b": "  ",
    "UrlOutput::url2c": "http://ko.wikipedia.org/wiki/꽃",
```

```
          "Output::link2c": "위키백과",
          "result_group_header": "백과사전",
          "Output::image_url":
"http://www.gstatic.com/richsnippets/b/fcb6ee50e488743f.jpg",
      "image_size": "80x80",
      "Output::inline_image_width": "80",
          "Output::inline_image_height": "80",
      "Output::image_border": "1"
          }
      ]
        }
       },
      ]
      }
```


ANEXO 8: Ejemplo de petición al API de búsqueda de Bing


```
http://api.bing.net/json.aspx?AppId=Insert your AppId here&Query=msdn
blogs&Sources=Web&Version=2.0&Market=en-
us&Adult=Moderate&Options=EnableHighlighting&Web.Count=10&Web.Offset=0&We
b.Options=DisableHostCollapsing+DisableQueryAlterations&JsonType=callback&JsonCa
llback=SearchCompleted
```

ANEXO 9: Ejemplo de respuesta JSON exitosa delAPI de búsqueda de Bing

{ "SearchResponse":{ "Version":"2.0", "Query":{ "SearchTerms":"msdn blogs" }, "Web":{ "Total":1490, "Offset":0, "Results":[{ "Title":"Monads", "Description":"Monad 1 2 Manifesto . Jeffrey P. Snover. Aug 8, 2002. Version 1.2. What is Monad? Monad is the next generation platform for administrative automation.", "Url":"http:\/\/blogs.msdn.com\/powershell\/attachment\/1910567.ashx", "DisplayUrl":"http:\/\/blogs.msdn.com\/powershell\/attachment\/1910567.ashx", "DateTime":"2008-10-29T05:27:14Z" }, { "Title":"The Trouble with Threat Modeling", "Description":"Larry Osterman lays out some other reasons threat modeling is hard in a blog post: http:\/\/□blogs□.□msdn□.com\/larryosterman\/archive\/2007\/08\/30\/threat-modeling-once-again.aspx", "Url":"http:\/\/blogs.msdn.com\/sdl\/attachment\/7702305.ashx", "DisplayUrl":"http:\/\/blogs.msdn.com\/sdl\/attachment\/7702305.ashx", "DateTime":"2008-10-29T03:25:32Z" } , { "Title":"Mobile Messaging with Exchange Server and Windows Mobile 5.0-based ...", "Description":"Security for Windows Mobile Messaging in the Enterprise . Date: February 2007. Applies to: Windows Mobile 5.0 with MSFP Windows Mobile 6 Exchange Server 2003 SP2", "Url":"http:\/\/blogs.msdn.com\/jasonlan\/attachment\/1873662.ashx", "DisplayUrl":"http:\/\/blogs.msdn.com\/jasonlan\/attachment\/1873662.ashx", "DateTime":"2008-10-23T09:52:02Z" }] } }/*pageview_candidate*/}

ANEXO 10: Ejemplo de respuesta XML exitosa delAPI de búsqueda de Yahoo

<bossresponse responsecode="200">
<web start="0" count="50" totalresults="362000">
<results>
<result>
<date>2011/02/01</date>
<clickurl>http://www.barackobama.com/</clickurl>
<url>http://www.barackobama.com/</url>
<dispurl type="default">www.barackobama.com</dispurl>
<title type="default">Organizing for America | BarackObama.com</title>
<abstract type="default">WhiteHouse.gov is the official web site for the White ...</abstract></result>
<result>
<date>2011/02/01</date>
<clickurl>http://www.whitehouse.gov/</clickurl>
<url>http://www.whitehouse.gov/</url>
<dispurl type="default">www.whitehouse.gov</dispurl>
<title type="default">The White House, President Barack Obama</title>
<abstract type="default">WhiteHouse.gov is the official web site for the White ...</abstract></result>
<result>
<date>2011/01/31</date>
<clickurl>http://en.wikipedia.org/wiki/Barack_Obama</clickurl>
<url>http://en.wikipedia.org/wiki/Barack_Obama</url>
<dispurl type="default">en.wikipedia.org/wiki/Barack</dispurl>
<title type="default">Barack Obama - Wikipedia, the free encyclopedia</title>
<abstract type="default">Barack Hussein Obama II (born August 4, 1961) is the 44th and current President of the United States. He is the first African American to hold the office. Obama previously ...</abstract></result>