



Universidad de las Ciencias Informáticas
Facultad 2



**Representación de datos bibliográficos mediante
técnicas de visualización**

**Trabajo de Diploma para optar por el título
de Ingeniero en Ciencias Informáticas**

Autoras: Ivelisse Montero Jimenez
Yeneisy Barroso Marquez

Tutor: Ing. Vladimir Milián Núñez

La Habana, Junio de 2015
“Año 57 de la Revolución”



"...el éxito de una visualización se basa en el conocimiento profundo y el cuidado de la sustancia, así como en la calidad, relevancia e integridad del contenido"

John W. Tukey



Declaración de autoría

Declaramos que somos las únicas autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autoras

Yeneisy Barroso Marquez

Ivelisse Montero Jimenez

Tutor

Ing. Vladimir Milián Núñez



Datos de contacto

Datos del Autor:

Yeneisy Barroso Marquez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo: ymarquez@estudiantes.uci.cu

Datos del Autor:

Ivelisse Montero Jimenez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo: ijimenez@estudiantes.uci.cu

Datos del Tutor:

Ing. Vladimir Milián Núñez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo: vmilian@uci.cu



Agradecimientos

De Ivelisse

A mis padres, por todos estos años y cada segundo de sus vidas que me han dedicado sin pedir nada a cambio, todo se lo debo a ustedes. Gracias por confiar en mí en todo momento, por apoyarme en los momentos más difíciles y sobre todo por comprenderme siempre. Toda la vida les estaré agradecida por darme su amor, por sus consejos y sacrificio para cumplir el sueño de hacerme ingeniera.

Yeny, gracias por ser mi amiga, la mejor compañera de tesis y hasta mi dolor de cabeza en estos últimos tiempos.

A mi tutor Vladimir gracias por su apoyo y comprensión en todo momento.

Pablo gracias por la gran ayuda que nos brindaste y la paciencia que tuviste con nosotras.

A todas las amistades que hice en estos 5 años, a mi grupo y en especial a Cary por haber sido como una hermana para mí.

A mi novio Victor por haberme aguantado todos estos meses, llenos de tensión y nervios, gracias por brindarme tu apoyo incondicional en todo momento, por preocuparte por mí y por tu cariño.

A los profesores que de una forma u otra colaboraron con la elaboración de esta tesis y a aquellos que contribuyeron a mi formación durante los 5 años en la universidad.

De Yeneisy

A mis padres Marta e Idelfonso, a quienes les debo la vida y viviré eternamente agradecida por inculcarme siempre sus sabios consejos y experiencias, por confiar en mí incluso primero que yo, por su amor y sacrificio para que yo pudiera hacer realidad este sueño de hacerme ingeniera como ellos.

A mis familiares, abuelos, hermanos, primos, tías y tíos en general, que me apoyaron durante toda mi carrera.

A mi novio por los momentos a mi lado en los que estuvo dándome fuerzas, siendo mis ojos cuando no conseguía ver con claridad, siendo tolerante ante mis desesperos, siendo mi punto de apoyo para alcanzar mi equilibrio emocional.

A mi compañera de tesis Ive, por permitirme compartir este gran momento con alguien.

A mi amiga, Cary por el apoyo desmedido en los momentos alegres y los de dolor.

A mis compañeros de fiestas y estudio, con los cuales compartí los mejores años de mi vida.

Para todas aquellas personas que estuvieron presentes cuando más los necesitaba, para darme ánimo y por enseñarme a entender lo importante que es perseguir un sueño y luchar por él.



Dedicatoria

De Ivelisse

A mis padres por tanto sacrificio y la ilusión de ver a su hijita convertirse en ingeniera.

De Yeneisy

A mi mamá y papá, porque son los eternos culpables de hacer de mí una mejor persona, porque son el mejor regalo que me ha dado la vida, mi mayor alegría, mi amuleto, mi gran virtud, son la mayor muestra de amor, sacrificio y bondad que hayan guardado para mí. Ustedes han sido mi fuente inspiradora, son lo que más amo en esta vida.

A mi abuela mimada que ha hecho de mí una mejor nieta cada día.

A mis hermanos, y primas para los cuales me gustaría ser ejemplo, exhortándolos a continuar superándose para que sean hombres y mujeres de bien, el día de mañana.



Resumen

En la medida en que aumenta el cúmulo de datos que debe ser procesado y comprendido por las personas, cobra mayor importancia el empleo de técnicas que permitan explorar y analizar la información. Para una mejor comprensión de la información que se requiere buscar en las distintas fuentes de investigación, se suelen emplear formas convencionales para la representación de datos, tales como: gráficos de línea, histogramas y otros. La mayoría de las herramientas actuales que presentan los datos bibliográficos de forma textual, no permiten comunicar detalles a los espectadores que no sean posibles ser descubiertos por métodos comunes. Las técnicas no convencionales como: grafos, imágenes o 3D permiten a las personas interactuar directamente con los datos y brindan una mejor interpretación de la información al permitir la exploración visual de los datos que se generan.

El presente trabajo de diploma ofrece una solución web que permita la representación gráfica de datos bibliográficos de publicaciones científicas mediante técnicas de visualización. Para ello se llevó a cabo el estudio de las técnicas, algoritmos y métodos de visualización existentes para la representación de la información. Una vez culminado el estudio, se obtuvo como resultados un sistema que brinde información detallada de los datos bibliográficos de documentos y los autores, una representación de un grafo egocéntrico dirigido centrado en documento y una representación de un grafo egocéntrico no dirigido centrado en autor.

La solución web está elaborada principalmente a partir del uso del lenguaje de programación del lado del servidor Python 2.7, la utilización del marco de desarrollo Django 1.4 y el empleo de la librería Igraph para el dibujo de los grafos. Para guiar el proceso de desarrollo se utiliza la metodología Programación Extrema (XP).

Palabras Clave: datos bibliográficos, representación de la información, representación gráfica, técnicas de visualización.



Índice de contenido

Introducción.....	1
Capítulo 1 Fundamentación Teórica	5
1.1 Introducción	5
1.2 Documentación relacionada con el dominio del problema	5
1.3 Análisis de soluciones similares a la aplicación a desarrollar.....	10
1.4 Presentación de la Información.....	11
1.5 Proceso de desarrollo del software	23
1.6 Metodología de desarrollo de software	24
1.7 Lenguaje de programación y tecnologías	25
1.8 Interfaz de programación de aplicaciones.....	27
1.9 Marco de trabajo.....	28
1.10 Lenguaje de Modelado	29
1.11 Herramientas de desarrollo.....	30
Conclusiones parciales del capítulo.....	31
Capítulo 2 Propuesta del Sistema.....	32
2.1 Introducción	32
2.2 Propuesta de solución de la aplicación web.....	32
2.3 Modelo del Dominio	34
2.4 Captura de requisitos.....	36
2.5 Etapa de Planificación	40
2.6 Etapa de Diseño	46
Conclusiones parciales del capítulo.....	55
Capítulo 3 Implementación y Prueba	56
3.1 Introducción	56
3.2 Etapa de Implementación	56
3.3 Prototipos de interfaz de usuario	61
3.4 Pruebas del Sistema.....	62
Conclusiones parciales del capítulo.....	66
Conclusiones Generales	67
Recomendaciones.....	68
Referencias Bibliográficas	69



Índice de ilustraciones

Ilustración 1: Arquitectura de un buscador.....	8
Ilustración 2: Presentación visual de noticias por la Aplicación Newsmap	16
Ilustración 3: Tag Cloud, tomado de la librería Dalton State (Cool Apps & Web Tools)	16
Ilustración 4: A la izquierda el grafo dirigido y a la derecha el grafo no dirigido (Zafra, 2013)	18
Ilustración 5: A la izquierda la red sociocéntrica y a la derecha la red egocéntrica	18
Ilustración 6: A) Gráficos de nodos y B) después de realizar un paso en el segundo gráfico.....	20
Ilustración 7: Grafo no dirigido aplicando el algoritmo (Smith, 2010)	20
Ilustración 8: Equilibrio en gráficos de nodos (Kumar, 2008)	21
Ilustración 9: Aplicando al grafo de la izquierda el método KK.....	22
Ilustración 10: Algoritmo de Kamada-Kawai tomado de (Csárdi, 2012)	23
Ilustración 11: Algoritmo de Kamada-Kawai (Igraph, 2012)	23
Ilustración 12: Propuesta de solución Representación de Bibliografías Científicas.....	32
Ilustración 13: Grafo egocéntrico no dirigido centrado en un autor	33
Ilustración 14: Funcionamiento de RBC	34
Ilustración 15: Proceso de acceso a la gestión de documentos y usuarios en la administración .	34
Ilustración 16: Diagrama de Modelo de Dominio	35
Ilustración 17: Arquitectura Cliente-Servidor.....	46
Ilustración 18: Patrón de Arquitectura Modelo-Plantilla-Vista.....	47
Ilustración 19: Contenido del paquete models	48
Ilustración 20: Contenido de las vistas dentro de la carpeta templates.....	49
Ilustración 21: Contenido del paquete views.....	49
Ilustración 22: Diagrama de componentes del sistema.....	51
Ilustración 23: Ejemplo del formulario UserCreationForm para validar un nuevo usuario	52
Ilustración 24: Ejemplo de la clase Person	52
Ilustración 25: Ejemplo de la clase Organization	53
Ilustración 26: Ejemplo de la clase Reference	53
Ilustración 27: Ejemplo de utilización del patrón Decorador.....	54
Ilustración 28: Ejemplo de utilización del patrón Observer.....	54
Ilustración 29: Diagrama de paquetes de la aplicación web RBC	58
Ilustración 30: Diagrama de clases persistentes.....	59
Ilustración 31: Modelo físico de la base de datos	60
Ilustración 32: Vista detallada del documento seleccionado	61



Ilustración 33: Grafo dirigido mostrando las relaciones entre documentos	62
Ilustración 34: Resultado de las pruebas realizadas	64
Ilustración 35: No conformidades detectadas durante las 4 iteraciones.....	65



Índice de tablas

Tabla 1: Descripción de las personas relacionadas con el sistema.....	36
Tabla 2: Definición de los requisitos funcionales.....	37
Tabla 3: Mostrar de forma gráfica la relación entre autores	40
Tabla 4: Estimación de esfuerzo por HU.....	41
Tabla 5: Plan de duración de las iteraciones.....	44
Tabla 6: Planificación de entregas de las iteraciones.....	46
Tabla 7: Mostrar de forma gráfica la relación entre autores	54
Tabla 8: Tarea #1 Búsqueda de datos bibliográficos	56
Tabla 9: Prueba de aceptación #19	64



Introducción

En la medida en que las Tecnologías de la Información y las Comunicaciones (TIC) han contribuido con el aumento del volumen y la diversidad de los datos que se manejan en las organizaciones, la investigación y la visualización de los datos, se vuelve cada vez más compleja. La necesidad que presentan los internautas de mantenerse informados sobre diversos temas de interés, ha conllevado al surgimiento de numerosos campos de investigación enfocados en la búsqueda de herramientas de análisis que permitan representar gran cúmulo de información, facilitando su rápida interpretación.

Los Sistemas de Gestión de Información (SGI) están concebidos para facilitar el acceso al conocimiento y apoyar en la toma correcta de decisiones. La repercusión que ha tenido su evolución en el procesamiento y representación de los datos, ha dado lugar al empleo de herramientas que permitan la búsqueda de la variada literatura científica que se encuentra dispersa en la web. Entre las herramientas de búsqueda de información científica más utilizadas por los usuarios, se ubican a: las bibliotecas digitales, base de datos bibliográficos, repositorios y buscadores académicos.

De las herramientas de búsqueda de información científica, mencionadas con anterioridad, resultan muy conocidas para los investigadores: Google Scholar, Microsoft Academic Search y CiteseerX en el ámbito internacional y en el ámbito nacional: SciELO Cuba y Redcien. Los sistemas anteriores, se caracterizan por presentar la información textual de datos bibliográficos¹ de libros, tesis, ponencias, artículos y revistas científicas, entre otros. La gran mayoría no cuentan con técnicas de visualización como imágenes, diagramas o grafos, que ayuden a interpretar grandes volúmenes de datos y relaciones entre ellos a la vez.

En Cuba, los centros de investigación no pueden acceder a gran parte de los repositorios internacionales, debido que son privativos y es necesario pagar por el servicio que ofrecen. Los jóvenes investigadores que desean informarse sobre la bibliografía más reciente que se encuentra en el área de las Ciencias Básicas y la Cibernética, no disponen de una vía de acceso que permita obtener datos de calidad, navegar por la información, obtener la de mayor impacto y la relacionada con ella. Los hechos expuestos con anterioridad pueden interrumpir el curso exitoso de las investigaciones.

¹ Los datos bibliográficos contienen la información del título, palabras claves, resúmenes, nombre de autores, citas bibliográficas, referencias, año de publicación y, a menudo enlaces al texto completo de un documento.



En este contexto, el Ministerio de Ciencia Tecnología y Medio Ambiente (CITMA²), de conjunto con el Instituto de Cibernética, Matemática y Física (ICIMAF³), han decidido impulsar la investigación científica y la innovación en estas áreas, ampliando el conocimiento científico y tecnológico de los profesionales.

El proyecto "Metodología y herramienta de vigilancia tecnológica para la investigación y la innovación" que forma parte del programa de prioridad nacional: Ciencia, Tecnología e Innovación denominado: "Desarrollo de la industria cubana del software" desde junio del 2013, rectorado por el Instituto (ICIMAF), desea mediante el empleo de técnicas de visualización de información, representar la información de los datos bibliográficos, de las publicaciones científicas en el área de las Ciencias Básicas y la Cibernética, respectivamente. El proyecto en cuestión, requiere de un espacio que permita a los jóvenes investigadores elegir el documento o autor de su preferencia, navegar a través de la jerarquía y conocer relaciones entre varios autores y documentos de forma resumida. Al mismo tiempo, muestre los documentos que ha publicado el autor seleccionado en colaboración con otros autores; y las publicaciones y citas de un documento en específico.

Teniendo en cuenta la **problemática** expuesta anteriormente queda formulado el siguiente **problema científico**: El empleo de métodos de representación gráfica adecuados, capaces de comunicar de forma resumida y relacionada la información de datos bibliográficos de publicaciones científicas, es insuficiente.

Para dar solución al problema científico, el **objeto de estudio** de la investigación lo constituyen los Sistemas de Gestión de Información científica.

Se propone como **objetivo general**: Desarrollar una aplicación web para la representación gráfica de datos bibliográficos de publicaciones científicas.

Del objetivo general planteado, se desglosan los siguientes **objetivos específicos**.

- Realizar la fundamentación teórico-metodológica de la investigación permitiendo tener dominio sobre el tema a desarrollar.
- Elaborar el análisis y diseño de la aplicación web para la representación gráfica de datos bibliográficos.
- Implementar una aplicación web para la representación gráfica de datos bibliográficos.
- Validar la solución propuesta a partir de la realización de pruebas a la aplicación web a desarrollar.

² CITMA: sitio oficial, <http://www.medioambiente.cu/>

³ ICIMAF: sitio oficial, <http://www.icimaf.cu/>



El **campo de acción** se encuentra enmarcado en la Representación gráfica de datos bibliográficos. Para dar cumplimiento a los objetivos planteados, se definen las siguientes **tareas de la investigación**:

- ✓ Revisión de la documentación relacionada con los Sistemas de Gestión de Información, así como el análisis de sistemas similares a la aplicación web a desarrollar.
- ✓ Revisión de la documentación relacionada con las técnicas de visualización de información.
- ✓ Selección de la metodología, herramientas de desarrollo, marcos de trabajo y lenguajes de programación a utilizar en el análisis, diseño e implementación de la aplicación web.
- ✓ Definición de las funcionalidades y características principales que tendrá la aplicación web.
- ✓ Diseño e implementación del grafo egocéntrico dirigido centrado en documento y del grafo egocéntrico no dirigido centrado en autor.
- ✓ Implementación de las funcionalidades identificadas por el cliente para obtener una aplicación web que cumpla con los requisitos determinados.
- ✓ Validación de la aplicación web para la representación de datos bibliográficos.

Una vez culminado el trabajo se obtuvieron los siguientes **resultados**:

- ❖ Un sistema que brinde información detallada de los datos bibliográficos de documentos y los autores.
- ❖ Una representación de un grafo egocéntrico dirigido centrado en documento.
- ❖ Una representación de un grafo egocéntrico no dirigido centrado en autor.
- ❖ Un sistema que permita la representación gráfica de datos bibliográficos de publicaciones científicas.

Para el desarrollo de la investigación se emplearon los siguientes métodos científicos:

Métodos teóricos:

- ✓ **Histórico / Lógico:** Se evidencia en el estudio realizado a los Sistemas de Gestión de Información científica existentes a nivel nacional e internacional. De los mismos, se dividieron los aspectos investigados de forma tal, que facilitara su entendimiento, guiando la búsqueda hacia las temáticas más importantes, permitiendo la extracción de los elementos más significativos y el arribo a conclusiones teóricas y prácticas bien definidas.
- ✓ **Analítico / Sintético:** Se evidencia en el análisis del comportamiento de los sistemas, así como sus características y funcionalidades. Además de las características y oportunidades que brindan las técnicas y métodos de visualización de la información.



Métodos empíricos:

- ✓ **Observación:** Se realiza un estudio minucioso de trabajos realizados con anterioridad que han logrado resultados excelentes y que se encuentran estrechamente vinculados con la problemática a la que se hace referencia.
- ✓ **Entrevista:** Permite conocer las necesidades del cliente y obtener los requisitos del sistema a tener en cuenta para la solución de la investigación.

El presente trabajo se encuentra dividido en tres capítulos, estructurados de la siguiente forma:

Capítulo 1: Fundamentación Teórica: En el capítulo se evidencia el estudio preliminar de varias herramientas de búsqueda de información científica existentes a nivel nacional e internacional, basado en como presentan la información. Se fundamenta la elección de las técnicas y métodos de visualización de la información a emplear; además, se especifica y argumenta la metodología, las herramientas y tecnologías a utilizar para el desarrollo de la aplicación web.

Capítulo 2: Propuesta del Sistema: En el capítulo se describe todo lo referente al análisis y diseño de la aplicación web para la representación de datos bibliográficos. Se plantea la propuesta del sistema a desarrollar, se especifican las características que la herramienta debe cumplir, y se describen las funcionalidades a automatizar para darle solución al problema.

Capítulo 3: Implementación y prueba: En el capítulo se describe el proceso de implementación de la herramienta, garantizando su correcto funcionamiento y mediante las pruebas realizadas a la aplicación web se verifica que el producto resultante cumple con los requisitos definidos por el cliente.

Capítulo 1 *Fundamentación Teórica*

1.1 Introducción

El presente capítulo aborda los conceptos fundamentales asociados al dominio de la problemática planteada. Se describen criterios valorativos sobre los Sistemas de Gestión de la Información y las características que poseen las herramientas de búsqueda existentes en la web, tanto a nivel nacional como internacional, con respecto al modo en que presentan la información que contienen. Además se detallan algunas ideas relacionadas con el empleo de técnicas de visualización de información, y se fundamenta la elección de algunas de ellas para la aplicación web a desarrollar. En el capítulo, se define la metodología de desarrollo de software utilizada así como los lenguajes, tecnologías y herramientas que dieron soporte a la planificación, diseño, implementación y validación de la propuesta de solución.

1.2 Documentación relacionada con el dominio del problema

1.2.1 Los Sistemas de Gestión de Información

Los Sistemas de Gestión de Información, en lo adelante (SGI), han ido incorporando diversas prestaciones necesarias, para la manipulación de la información en el ámbito digital. Son un conjunto de procesos por los cuales se controla el ciclo de vida de la información, desde su obtención (por creación o captura), hasta su disposición final (su archivación o eliminación).

Codina comenta que los SGI son *"... el conjunto de los elementos y procesos que intervienen dinámicamente en la explotación de información cognitiva, concebida en el marco de un grupo social concreto y para áreas determinadas, cuyo propósito es facilitarles el acceso al conocimiento y apoyarlos en la toma correcta de decisiones."* (Codina, 1996)

Los SGI constituyen sistemas que: *"ofertan, regulan y gestionan todo tipo de recursos de información, con el objetivo de realizar los procesos de almacenamiento, identificación, transformación, organización, tratamiento y recuperación de la información"*, expresa (Pérez, 2009).

La repercusión que ha tenido la evolución de los SGI en el procesamiento y representación de los datos, ha permitido el desarrollo y empleo de nuevas herramientas que permitan la búsqueda de la información científica que se encuentra dispersa en la web. Las base de datos bibliográficas, bibliotecas digitales, repositorios y buscadores académicos, son ejemplos de herramientas de



búsqueda de información científica más utilizadas por los usuarios debido a la capacidad de almacenar grandes colecciones de datos bibliográficos.

1.2.2 Bases de datos bibliográficas

Las Bases de datos bibliográficas se caracterizan por contener en sus registros, los datos bibliográficos de libros, revistas y otras publicaciones como artículos científicos, actas de conferencias, congresos, capítulos de libros, entre otros.

Las bases de datos anteriores se crean y actualizan de manera continua con el propósito de resolver necesidades de información concretas de la sociedad. Además, son empleadas para acceder a las últimas contribuciones científicas que se han realizado en un área determinada de la ciencia.

Rodríguez-Yunta establece diferentes tipologías⁴ para caracterizar los diversos modelos de bases de datos bibliográficas. (Rodríguez-Yunta, 2001)

Las bases de datos bibliográficas han sido tradicionalmente un pilar fundamental en la formación relacionada con fuentes de información en centros de investigación y universidades.

Una década después los estudios señalados en (Giménez-Toledo, et al., 2012) revelan que los usuarios y profesionales están dejando de lado a los productos especializados para centralizar su interés en las herramientas multidisciplinares que aportan datos de citas e indicadores bibliométricos⁵ derivados de ellas. Las causas de esta aparente convergencia tienen que ver, sin dudas, con la comodidad de las opciones que propician, la transformación del sector de la información científica y los nuevos hábitos de los usuarios finales.

Hoy en día, la evolución de herramientas para la búsqueda de información a través de la Web, tales como: bibliotecas digitales, repositorios y, buscadores que agregan servicios y fuentes de información han construido una enorme biblioteca de contenidos especializados.

1.2.3 Bibliotecas digitales

Se entiende por Bibliotecas digitales a *“una organización virtual que, de modo exhaustivo, reúne, gestiona y conserva a largo plazo contenidos digitales, ofreciendo a sus comunidades de usuarios funcionalidades especializadas, con calidad medible y respetando una política o directrices dadas.”* (Candela, 2007)

Las bibliotecas digitales deben desarrollar tres características generales, como las descritas por (Marcum, 2005).

⁴ La tipología es el análisis y la categorización de tipos, dígame clases, modelos o ejemplos de algo.

⁵ Existen distintos indicadores bibliométricos (producción, circulación, uso de la literatura científica, visibilidad o impacto) que son medidas que proporcionan información sobre los resultados de la actividad científica en cualquiera de sus manifestaciones.



- Ser una colección global de recursos importantes para la investigación, la enseñanza y el aprendizaje.
- Ser de fácil acceso para todo tipo de usuarios, tanto principiantes como expertos.
- Estar gestionada y mantenida por profesionales que se consideren administradores del patrimonio intelectual y cultural.

Bartomeo, expresa que las bibliotecas digitales proponen la interacción en tres ámbitos: el tecnológico, el personal y el informacional. Este último puede ser de acceso abierto o privado. (Bartomeo, 2012)

SciELO⁶, constituye un ejemplo vigente de que la mayoría de las universidades cuentan con un servicio de bibliotecas con el fin de proveer y facilitar información como apoyo en el proceso docente-investigativo.

1.2.4 Repositorios

“Un repositorio, depósito o archivo es un sitio web centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.” (Universidad de Salamanca, 2015)

La colección de un repositorio abarca artículos, trabajos científicos, tesis doctorales y de maestría, revistas temáticas, material docente, y otros documentos, en distintos formatos digitales. Los repositorios más conocidos son los de carácter académico e institucional y tienen por objetivo organizar, archivar, preservar y difundir la producción intelectual resultante de la actividad investigadora de una entidad.

Un repositorio académico o temático es creado por un grupo de investigadores, una institución, entre otros, que reúnen documentos relacionados en áreas científicas y un *“Repositorio Institucional es un archivo electrónico de la producción científica de una institución, almacenada en un formato digital, en el que se permite la búsqueda y la recuperación para su posterior uso nacional o internacional.”* (González, et al., 2007)

1.2.5 Buscadores web

Los buscadores web o motores de búsqueda como se suele encontrar en algunas bibliografías, son aplicaciones informáticas formadas por complejos algoritmos de búsqueda e indexación que permiten realizar búsquedas en la red, al rastrear la web, catalogar, clasificar y organizar la información. Además, brindan el acceso a los usuarios para buscar información sobre temas de su preferencia.

⁶ SciELO: sitio oficial, <http://scielo.sld.cu/scielo.php?lng=es>

Según (Gutiérrez, 2008), “...casi un tercio del tiempo que los usuarios pasan en Internet lo dedican a realizar búsquedas en la Web.”

1.2.5.1 Arquitectura general de un buscador

Los buscadores poseen un sistema denominado spider o robot de búsqueda, un programa que rastrea (descubre) la web automáticamente. Analiza y localiza las páginas web en los servidores de todo el mundo a través de la búsqueda de términos, enlaces, y en general todas las URL⁷ que proporcionan dichas páginas, para que posteriormente el usuario pueda acceder y obtener una información global de la búsqueda realizada.

La arquitectura de un buscador “araña-indexador centralizada, es decir, la araña y el componente de indexación (crawler)⁸ se encuentran unidos.” (Méndez, 2002)

A continuación se puede observar en la **(Ilustración 1)** la arquitectura de un buscador web.



Ilustración 1: Arquitectura de un buscador

Una vez que el robot recupera las páginas que encuentra, el crawler las recibe (recopila) e indexa todo el texto depositándolo en una base de datos. Los buscadores poseen enormes bases de datos que contienen millones de direcciones de información referente a páginas web.

Cuando el usuario escribe una o varias palabras, el buscador consulta su base de datos y muestra las direcciones de las páginas que se relacionan con la palabra. Es evidente, que si una página no está indexada en la base de datos del buscador, jamás la podrá encontrar mediante este. Luego, la información se mostrará en la interfaz (la que utilizan los usuarios cuando acceden a un buscador) para buscar las páginas.

Baeza-Yates; Castillo y Rodríguez expresan que “...desde la perspectiva del usuario existen dos requisitos principales que deben cumplir todos los buscadores: (1) un tiempo corto de respuesta y

⁷ Uniform Resource Locator (Localizador Uniforme de Recurso), que representa una dirección en la Web.

⁸ (Crawlers) es como se llaman en inglés a las arañas web de los buscadores que rastrean las páginas web en busca de información para un correcto posicionamiento de las webs en sus resultados.



(2) *una gran colección de documentos web disponibles en su índice. La calidad de un buscador reside en lo abundante, relevante y actualizada que sea su colección.*" (Baeza-Yates, y otros, 2005)

Los buscadores suelen permitir mediante interfaces sencillas la acotación de las búsquedas, las definidas en (Lamarca, 2013) son:

- Palabras claves
- La frase exacta
- La búsqueda booleana⁹

Una de las desventajas que poseen los buscadores web es que retornan muchos resultados irrelevantes. Además, herramientas de búsqueda diferentes pueden arrojar distintos resultados con una misma consulta. Sin embargo, logran que los usuarios dada su preferencia y necesidad, posean una vía exitosa a la hora de buscar información.

En el grupo de buscadores web se pueden encontrar a los buscadores temáticos, referidos a un campo del conocimiento en específico (ciencias puras o ciencias aplicadas, tecnología, informática, entre otros). (Lamarca, 2013)

Dentro de la tipología de buscadores temáticos se hallan los buscadores académicos que son sistemas de información que sólo registran sitios web vinculados al mundo científico. A continuación se exponen algunos criterios importantes sobre los buscadores académicos o científicos, como también se les conoce.

1.2.5.2 Buscadores académicos

Los buscadores académicos¹⁰, son utilizados para la localización de bibliografías científicas de: artículos de revistas, investigaciones, ponencias presentadas a congresos, tesis, tesinas, patentes, libros, y páginas web de instituciones académicas.

En el trabajo de Giménez-Toledo y Rodríguez-Yunta, expresan que: "*Los grandes protagonistas del cambio en los hábitos de búsqueda bibliográfica son los buscadores.*" (Giménez-Toledo, y otros, 2012), basado en las encuestas realizadas por (OCLC, 2011) que arrojaron resultados devastadores poniendo en evidencia que el uso de los servicios online de las bibliotecas, ocupa una posición cada vez más secundaria para los usuarios norteamericanos.

En Cuba existe un motor de búsqueda que indexa y almacena información, de páginas, revistas, entre otras, en la red nacional, como es el caso de **Orion**¹¹, realizado por la **UCI**¹² desde el 2011. A

⁹ Se trata de un tipo de interrogación que incluye alguno de los operadores lógicos.

¹⁰ Se pueden encontrar en varias literaturas como buscadores científicos o temáticos.

¹¹ Orion: sitio oficial, <http://orion.mes.edu.cu/>

¹² Universidad de las Ciencias Informáticas.



Orion, pueden acceder todas las personas que tienen conexión en el país. Sin embargo, no existe un buscador temático nacional que contenga información bibliográfica de publicaciones científicas en el área de las Ciencias Básicas y la Cibernética respectivamente.

1.3 Análisis de soluciones similares a la aplicación a desarrollar

En el desarrollo de cualquier proyecto de investigación, constituye un paso fundamental el estudio de las principales tendencias relacionadas con la aplicación a desarrollar. En función de lo planteado, se muestra el estudio de algunas de las herramientas de búsquedas utilizadas en el ámbito internacional y nacional.

1.3.1 Buscadores académicos internacionales

En la actualidad existe una extensa lista de buscadores académicos, de los cuales, se pueden mencionar a: Google Scholar, CitesserX, Microsoft Academic y ScienceDirect que son utilizados internacionalmente por los investigadores.

Google Scholar¹³, permite realizar búsquedas en la literatura científica y académica. Desde un solo sitio se pueden realizar búsquedas en un gran número de disciplinas y fuentes como publicaciones periódicas, tesis, libros e informes técnicos, sociedades profesionales, universidades. Permite ordenar los resultados de la búsqueda por relevancia y muestra la cantidad de referencias que están vinculadas a la misma.

Microsoft Academic Search¹⁴ utiliza formas innovadoras de explorar trabajos científicos, conferencias y revistas. Se especializa en mostrar una vista más detallada de la información a los usuarios respecto a los demás buscadores, mediante una gráfica que representa la cantidad de publicaciones que tiene un documento y la fecha en que fueron publicados. Ofrece además de la búsqueda de literatura científica, la visualización de las relaciones entre autores y organizaciones en forma de grafo, permitiendo la comprensión de las relaciones de un modo intuitivo.

ScienceDirect¹⁵ es el buscador científico más amplio que ofrece **Elsevier**¹⁶ especializado en ciencias. Permite realizar búsquedas en literatura científica, noticias, patentes, informes y datos médicos y técnicos.

Las búsquedas en **CitesserX**¹⁷ se centran principalmente en la literatura en ciencias de la computación y de la información, y tiene como principal objetivo mejorar la difusión de la literatura

¹³ Google Académico: sitio oficial, <http://scholar.google.com/>

¹⁴ Microsoft Academic Search: sitio oficial, <http://academic.research.microsoft.com/>

¹⁵ ScienceDirect: sitio oficial, <http://www.sciencedirect.com/>

¹⁶ Es un proveedor líder mundial de soluciones de información que mejoran el rendimiento de la ciencia, la salud y profesionales de la tecnología basada en libros de medicina.

¹⁷ CiteseerX: sitio oficial, <http://citeseer.ist.psu.edu/index>



científica. Fue el primer buscador académico en permitir que los documentos de navegación se generaran automáticamente mediante el empleo de enlaces de citas.

Los buscadores académicos analizados en el ámbito internacional, son muy buenos para localizar bibliografías científicas, debido a la variedad de literatura que contienen en las disímiles áreas y la posibilidad de producir mejores resultados que los buscadores generales. Sin embargo, en ocasiones, es muy engorroso el proceso de investigación para los usuarios que tienen poca experiencia en el uso de las herramientas de búsqueda mencionadas con anterioridad. La extensa lista de resultados que pueden arrojar sobre un criterio en específico hace que los usuarios se pierdan al decidirse por un autor o documento en particular para su investigación, debido al gran volumen y la desorientada información que se genera en Internet.

La variedad de usuarios que pueden acceder a las herramientas de búsqueda de información científica mencionadas, requiere de *“la creación de interfaces flexibles que puedan adaptarse a las preferencias de cada grupo (usuarios expertos, avanzados o neófitos¹⁸.”* (Giménez-Toledo, et al., 2005)

El criterio de Giménez-Toledo y Rodríguez-Yunta se tuvo en cuenta en la aplicación web para la representación de datos bibliográficos, para mostrar interfaces sencillas y amigables que posibiliten a los investigadores el rápido acceso a la bibliografía científica de su interés. Los jóvenes investigadores podrán navegar por la información contenida en la aplicación web, sin un elevado nivel de conocimiento en el trabajo con la herramienta.

1.3.2 Herramientas de búsqueda científica en Cuba

A nivel nacional no existen buscadores temáticos pero si herramientas de búsqueda de información vinculadas a la ciencia y la computación.

Un ejemplo es La Red Cubana de la Ciencia **“Redcien¹⁹”** que ofrece contenidos y servicios de información concebidos como un espacio de colaboración e interacción con instituciones científicas y otros profesionales de Cuba y el resto del mundo. Redcien ofrece la búsqueda por categorías y muestra un pequeño resumen del texto como resultado. Dentro de los servicios que ofrece, se encuentra disponible **La Biblioteca Virtual de la Ciencias en Cuba**, que tiene como objetivo facilitar el acceso y gestión de la información, así como el acceso a conocimientos relacionados con la Ciencia, la Tecnología y la Innovación.

¹⁸ Novatos, principiantes, inexpertos.

¹⁹ Redcien: sitio oficial, <http://www.redciencia.cu/>



SciELO Cuba es una biblioteca electrónica o digital, que incluye una colección seleccionada de revistas científicas cubanas en todas las áreas del conocimiento. La herramienta ha sido desarrollada, por la Red Telemática de Salud en Cuba (**INFOMED**) en colaboración con el Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud de Brasil (**BIREME**). La interfaz proporciona el acceso a la colección de revistas por medio de una lista alfabética de títulos, una relación de materias o una formulación de la búsqueda por palabras, entre otras opciones. Permite además, el acceso al texto completo de los documentos a partir de los índices de autores y de materias, por el título y las palabras clave en el texto del documento.

La existencia de las bibliotecas analizadas en el ámbito nacional, es una muestra del crecimiento y el profundo *proceso de informatización* que se está llevando a cabo dentro de la sociedad.

Las herramientas de búsqueda de información científica analizadas tanto en el ámbito nacional como en el internacional se caracterizan por presentar la información básica de datos bibliográficos (título, autores, resumen, palabras claves) de documentos tales como: artículos y revistas científicas, tesis y tesinas, ponencias, entre otros. Las diferencias entre una herramienta y otra están dadas por la variedad de los campos de estudio a los que van dirigidas; y por los detalles que cada una brinda sobre los datos bibliográficos que contienen.

Los campos de estudio pueden variar desde la medicina, artes y humanidades, ingeniería, física, matemática, ciencias de la computación y ciencias sociales, y muchas otras. Los detalles de los datos bibliográficos que dan a conocer son diversos, la cantidad de documentos que citan a un documento seleccionado, las versiones de ese documento y los artículos que se relacionan con el criterio de búsqueda especificado. Otros detalles que pueden dar a conocer son los últimos artículos que hayan sido publicados, los más visitados y los autores más citados, entre otros.

La mayoría de las herramientas analizadas, permiten limitar las búsquedas avanzadas ya sea mediante la especificación de criterios de acotación por palabras claves, nombre de autores, fecha de publicación u otras. Además, admiten realizar ordenaciones de forma ascendente o descendente por varios criterios (el que predomina en las herramientas analizadas es el ordenamiento de documentos por la fecha de publicación). Sin embargo, aunque son excelentes para la localización de bibliografía científica, presentan gran parte de la información de manera textual, lo que constituye un impedimento a la hora de interpretar grandes cantidades de datos.

Microsoft Academic Search es la única de las herramientas analizadas que comunica de forma resumida y novedosa los detalles pertenecientes a la cantidad de publicaciones y de citas de un autor, así como la cantidad de citas que tiene un documento. Además muestra las relaciones de coautoría de un autor con otros colaboradores y la relación de citas de un documento con otros. Los detalles anteriores serían difíciles de comprender a simple vista, sin la utilización de



métodos de visualización de información que permiten comunicar de forma relacionada y resumida los detalles de los datos bibliográficos que se requiere mostrar.

Luego del análisis realizado a los diferentes Sistemas de Gestión de Información científica, se decide emplear las funcionalidades de búsqueda básica y avanzada para la localización de información de datos bibliográficos en la aplicación web. Se eligió el criterio de búsqueda por palabras claves para especificar la información científica que se desea encontrar. Además, se tomó como referencia la forma en que se presentan los datos de las publicaciones y citas en Microsoft Academic Search, para representar las variaciones de las publicaciones y citas de un documento, y las publicaciones de un autor en un intervalo de fecha determinado, en la aplicación web.

1.4 Presentación de la Información

Arrollo advierte de *“un crecimiento exponencial de la cantidad de información disponible”* y plantea *“la necesidad de profesionales cuyo objetivo fuera organizarla, darle sentido y presentarla de forma coherente, sistemática y comprensible.”* (Arrollo, 2013)

Un problema muy común es que los usuarios deben navegar a través de muchos documentos irrelevantes antes de encontrar el documento de su preferencia o interés. En la actualidad, mostrar la información bibliográfica de forma tal, que permita a los usuarios decidirse por un autor o documento en particular, constituye un reto.

Es tanto el volumen de información que se muestra en forma textual, que se necesita utilizar alguna técnica que permita presentarla de forma resumida. Pero, ¿cómo mostrarle al usuario la presentación de dicha información?

Si los datos son presentados textualmente, la cantidad de estos que pueden ser mostrados están en el rango de algunos cientos de artículos, pero es como buscar una aguja en un pajar cuando se trata de conjuntos de datos que contengan millones. Sin tener la posibilidad de explorar adecuadamente las grandes cantidades de datos que han sido coleccionadas debido a su utilidad potencial, los datos se tornan inservibles. De este modo los datos por sí solos, desprovistos de contexto no brindan información alguna; pero su interrelación y significado permiten al ser humano mantenerse informado sobre un suceso, hecho, materia o fenómeno de la realidad.

La representación de datos en forma gráfica, ayuda a mostrar la información de manera sencilla, permitiendo a los usuarios interpretarla mediante (figuras, imágenes) haciéndola más comprensible y atractiva a la vista.

“Las representaciones gráficas son también importantes porque nos ayudan a comunicar conceptos abstractos como el tiempo, el espacio, las categorías y la jerarquía. También nos permiten



comprender la evolución diacrónica²⁰, la situación, la relación entre elementos y su importancia por comparación.” (Arrollo, 2013)

El objetivo que se persigue es lograr una representación gráfica que brinde al usuario información de su preferencia, siempre con la concientización del área que se quiere investigar, siendo esencial para una mejor comprensión de la información.

1.4.1 Visualización de la información

“La visualización de información es la representación visual de colecciones a gran escala de información no numérica, tales como archivos, bases de datos bibliográficas, redes de relaciones sociales, entre otros.” (Jesus, y otros, 2010)

La visualización es una técnica para comunicar un mensaje a través de imágenes, diagramas, grafos y animaciones, que permite a los usuarios ver, explorar y comprender grandes cantidades de información a la vez.

El empleo de la visualización permite observar lo que no es posible ver directamente, debido, entre otras razones, al gran volumen de los datos o a que estos no tengan una representación gráfica adecuada. El hecho de que el resultado sea una imagen, posibilita una mayor comprensión, claridad y aprovechamiento de este; además ofrece a las personas una interacción directa con los datos.

“La visualización es multidisciplinaria ya que comunica los datos mediante una interfaz, transformándolos en información semántica a través de distintos medios, combinando su funcionalidad y su estética.” (oa.usun, 2012)

Keim plantea que *“la exploración visual es intuitiva, no requiere de complicados conocimientos matemáticos, estadísticos o de otra índole.” (Keim, 2002)*. Otra gran ventaja de la visualización es la inmensa cantidad de conocimiento que puede ser rápidamente interpretado.

“La interpretación, es un proceso individualizado y creativo que permite hacer conexiones entre la información procedente de distintas fuentes para ampliar las ideas propias asignando a las palabras y conceptos un determinado significado y generar nueva información,” hace alusión (Molina, 2011).

Muchas son las áreas donde se aplica la visualización de la información, en (Roldán, 2003) se mencionan: la GEOVisualización, Infografías, Educación mediante Visualización, Visualización Creativa, Visualización Científica, Visualización de Software, entre otras.

²⁰ Hecho o fenómeno que se desarrolla a lo largo del tiempo.



1.4.2 Técnicas de visualización

Las técnicas de visualización permiten comprender o percibir los datos y relaciones entre ellos de manera más fácil. La propagación de éstas, *“ha permitido representar la esencia del conocimiento, al desempeñar un papel fundamental en el proceso de modelización y representación de la información.”* (Quesada, 2005)

“La mayor ventaja de las técnicas de visualización es que permiten una interacción directa con el usuario y provee una retroalimentación inmediata que no alcanzan otros enfoques no visuales” hace alusión (Keim, 2002).

Existen muchas técnicas según (Pullannagari, 2008) libres para representar los datos gráficamente entre las que se encuentran:

- Los diagramas de pastel²¹ y barras gráficas que se utilizan para las comparaciones y determinar las relaciones entre los datos.
- Los mapas, que son una de las formas más eficaces de representar la visualización. Dentro de los mejores ejemplos se encuentra “mapa de Google” con una fotografía de ruta y ubicaciones para percibir más fácil las cosas.
- El 3D es usado para presentar los datos pluridimensionales.
- Las imágenes que usan el color y la intensidad para representar los datos.
- Las redes para las relaciones entre (actores, comunidades, organizaciones).
- La animación, los bloques y muchos otros.

Las formas convencionales que suelen emplearse para representar la información de forma estática son: los gráficos de barras y de líneas, histogramas, tablas, mapas conceptuales²², entre otros. Cualquiera que sea el método elegido, el objetivo es darle sentido a los datos y comunicar detalles a los espectadores. Sin embargo, uno de los problemas detectados que presentan es su baja precisión.

Existen también técnicas no convencionales como grafos, árboles, 3d, imágenes, redes, animaciones que permiten a los usuarios interactuar directamente con los datos, brindando una mejor comprensión de la información al explorar adecuadamente las grandes cantidades de datos que pueden ser generados.

²¹ También conocidos por gráficos de sectores.

²² Permite la presentación de datos estadísticos, además de aclarar, analizar y relacionar conceptos.

Las técnicas de visualización se pueden clasificar basándose en tres criterios indicados por (Keim, 2002; Cortés, 2007), ejemplo de estos criterios lo constituyen: los datos a visualizar, la técnica de visualización y la técnica de interacción y distorsión.

Pullannagari plantea que *“hay ciertas aplicaciones en las que la visualización de datos debe ser interactiva para los usuarios, donde se puedan recorrer los datos mediante la variación de las vistas (zoom de los datos).”* (Pullannagari, 2008)

Un ejemplo de técnica no convencional se observa en la **(Ilustración 2)** la presentación visual de noticias por Newsmap, donde se muestran todos los reportajes organizados por categorías. Estos reportajes indican en un rectángulo grande de un único color y dentro de cada categoría (cada rectángulo en el máximo nivel) los reportajes divididos en subcategorías más pequeñas, distinguiendo el grado de relevancia que posee cada noticia.



Ilustración 2: Presentación visual de noticias por la Aplicación Newsmap

La selección de tags²³ o nube de etiquetas es una representación visual de los datos de texto y pretende servir al usuario como resumen global sobre las materias más populares por palabras claves, categorías, entre otros. Begelman indica que *“... se encuentra dominada por las temáticas de ‘diseño’ y ‘tecnologías’.”* (Begelman, 2006)

La **(Ilustración 3)** muestra un ejemplo de técnica no convencional, donde el tamaño de la fuente de cada palabra indica su relevancia determinada por su popularidad o capacidad representativa.



Ilustración 3: Tag Cloud, tomado de la librería Dalton State (Cool Apps & Web Tools)

²³ Los tags son términos o palabras clave en lenguaje no controlado con los que el usuario describe los recursos.



1.4.3 Visualización de Grafos

La representación de la información mediante grafos brinda la posibilidad de organizar de manera eficiente gran cantidad de datos no homogéneos. *“El dibujado de grafos direcciona el problema de visualizar información estructural o relacional construyendo representaciones visuales geométricas de grafos o redes que son los modelos subyacentes en una gran cantidad de datos abstractos.”* (Martig, y otros, 2000)

Actualmente, un gran número de aplicaciones necesitan representar visiblemente por medio de grafos la información de sus trabajos y *“tiene gran importancia en aplicaciones tales como la Ingeniería de Software, la Visualización de Información, Data Mining, el diseño de Bases de Datos e Interfaces Visuales, la Representación del conocimiento y las Telecomunicaciones entre otros dominios.”* (Martig, et al., 2000)

La visualización a través grafos, ayuda a representar la información de una forma más comprensible y atractiva a la vista, lo que posibilita una mejor interpretación por parte de los usuarios. Donde los nodos pueden ser personas, documentos, grupos, organizaciones, entre otros y las aristas o arcos las relaciones entre los nodos.

Mediante el empleo de grafos, el usuario puede elegir a un escritor específico y navegar completamente sobre la jerarquía y hallazgo de los trabajos realizados por dicho escritor. Todo esto mostrado en una sola página, para que el usuario pueda comprender y ver las relaciones entre ellos, además de la posibilidad que brinda a las personas de una interacción directa con los datos.

1.4.3.1 Teoría de grafos

“La Teoría de Grafos es una rama de las Matemáticas y de las Ciencias de la Computación que estudia las propiedades de los grafos.” (Zafra, 2013)

Un grafo es una estructura de datos $G = (V, A)$, que consiste en un conjunto de nodos o vértices, $V = \{v_1, v_2, \dots, v_n\}$, unidos por enlaces (aristas o arcos), $A = \{a_1, a_2, \dots, a_t\}$ que relacionan un vértice con otro, o con él mismo. Cada arista es en realidad una pareja de vértices, $A = (v_1, v_2)$, definiendo así la adyacencia de los vértices, dos vértices son adyacentes si están unidos por una arista. (Chacón, 2005)

En función de la dirección de las aristas un grafo puede ser:

- **Dirigido:** Un grafo dirigido es aquel cuyas aristas pueden tener una única dirección y se suelen representar de forma gráfica mediante flechas indicando que existe una relación entre los nodos.
- **No dirigido:** Un grafo no dirigido es aquel cuyas aristas no tienen una dirección definida debido a que la relación se produce en ambos sentidos. Por lo tanto, los pares de vértices (v_1, v_2) y (v_2, v_1) representan el mismo arco.

La (**Ilustración 4**), a continuación representa los dos grupos en que se pueden clasificar los grafos.

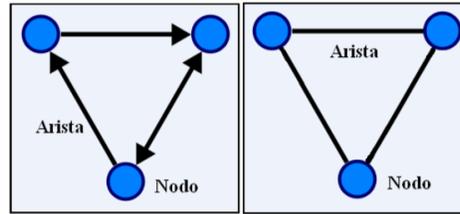


Ilustración 4: A la izquierda el grafo dirigido y a la derecha el grafo no dirigido (Zafra, 2013)

1.4.3.2 Grafo egocéntrico

La representación de un grafo o una red es un conjunto de lazos, todos del mismo tipo, entre una serie de actores. Los actores pueden ser personas, organizaciones, entre otros; y un lazo es un episodio de una relación social, dígase una red social.

Existen dos enfoques diferentes de las redes, por un lado el sociocéntrico y por el otro lado el egocéntrico (Barry, et al., 1988; Molina, 2005).

El análisis de las redes sociocéntricas o redes completas se basa, en un conjunto de actores y las relaciones que ocurren entre ellos. Mientras que en “... las redes egocéntricas, también denominada de influencia o redes personales, parte de las conexiones se trazan a partir del denominado ego (nodos centrales o actor específico) y en aquellos actores con quien se relaciona éste.” (Carrasco, et al., 2006).

A continuación la (**Ilustración 5**) muestra los diferentes enfoques de las redes (Borgatti, y otros, 2003).

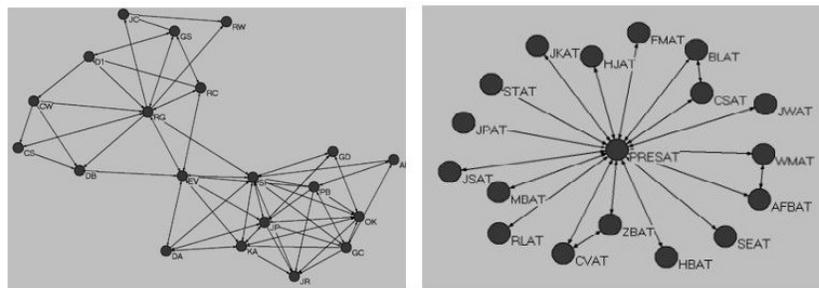


Ilustración 5: A la izquierda la red sociocéntrica y a la derecha la red egocéntrica

Las redes egocéntricas se basan, no en mostrar todo el reflejo del universo de la red, sino en extrapolar la perspectiva reflejada en la visión de los individuos centrales en ese universo y sus relaciones, como un todo. En estas redes los nodos centrales de ese universo, pueden ser personas, grupos u organizaciones y el enfoque facilita una buena imagen de las redes.



1.4.4 Selección de las técnicas de visualización

El estudio de una gran variedad de técnicas de visualización de información que ayudan a representar la esencia del conocimiento de diversas formas, permitió la elección de alguna de estas técnicas para ayudar a comprender mejor la información. Se determinó la elección de las técnicas a representar mediante forma convencional y no convencional para mostrar la información de manera dinámica y atractiva a la vista.

La forma convencional que se seleccionó fue gráfico lineal o poligonal que permiten mostrar en forma de segmento de rectas los datos mediante puntos que precisan (series de tiempo, rango de fecha en que se realizó un evento), entre otros. Estos gráficos tratan de representar de forma inmediata una idea general sobre los principales aspectos de los datos sin proporcionar detalles.

La forma no convencional para representar los datos de forma gráfica seleccionada fue mediante grafos, particularmente los grafos egocéntricos que permiten mostrar las relaciones entre el nodo central que se está analizando y los que se relacionan con él. Se mostrarán las relaciones mediante la dirección de las aristas con un grafo dirigido y un grafo no dirigido, ambos egocéntricos. Estos grafos permitirán representar y explorar todo el espectro visual centrado en ese actor o nodo específico.

1.4.5 Algoritmos de Visualización

Los algoritmos de visualización se emplean para diseñar la forma en que se desea presentar dicha información (en redes, jerarquías, entre otros). Esta forma de representar los datos son una de las técnicas más usadas actualmente y sobre todo en campos de diversas empresas.

Estos algoritmos desempeñan un papel muy importante en la representación de la información, pues comprende tanto la representación estática de la información como la dinámica. La representación estática se realiza por medio de gráficos, grafos y organigramas que permanecen invariables a lo largo del tiempo y la naturaleza dinámica de los algoritmos trata sobre la animación de estos. Este modo de representar la información ayuda a los usuarios a realizar exploraciones para mejorar su visualización y así comprender mejor la información.

Para la representación del grafo egocéntrico elegido como técnica de visualización de información no convencional, se analizaron los algoritmos de visualización propuestos por Kumar, que permitirán representar el grafo de diversas formas. Ejemplo de ellos lo constituyen: Fuerza dirigida basada en el diseño, Trazado ortogonal, Diseño del árbol y Presentación jerárquica. (Kumar, 2008)

1.4.5.1 Algoritmo Force-Directed Layout

Mediante el estudio de los diferentes algoritmos planteados anteriormente se selecciona el algoritmo Fuerza dirigida basada en el diseño conocido mundialmente como **Force-Directed Layout** por sus

excelentes prestaciones en el dibujado de gráficos, mostrando los grafos de una manera estéticamente agradable, con la mayor claridad posible. El algoritmo intenta evitar el cruce de aristas y el solapamiento de los nodos, para mejorar la visibilidad de los componentes del grafo.

Las ilustraciones mostradas a continuación describen como se aplica el algoritmo a los grafos no dirigidos (sin orden entre los nodos) y el resultado de este después de una serie de iteraciones. El algoritmo se ejecuta en pasos hasta que llega al estado de equilibrio.

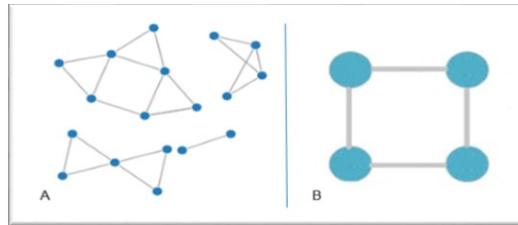


Ilustración 6: A) Gráficos de nodos y B) después de realizar un paso en el segundo gráfico

En la **(Ilustración 7)** se muestra un grafo no dirigido donde se observa la relación entre los nodos, dada la intensidad que este posea. Los nodos que tienen alguna relación están colocados cerca uno del otro y los que no tienen relación están muy alejados. En general, estas clases de algoritmos son muy útiles para el posicionamiento de los nodos.

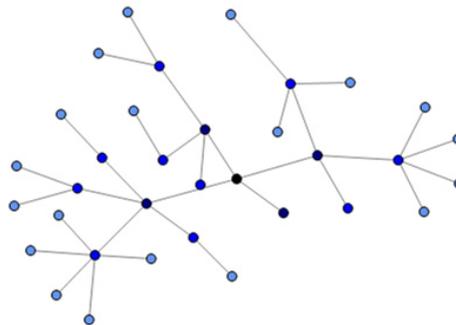


Ilustración 7: Grafo no dirigido aplicando el algoritmo (Smith, 2010)

Los algoritmos de fuerza dirigida tienen como *“propósito general, colocar los nodos del grafo en la disposición tal, que todos los bordes sean del mismo tamaño y la gráfica general tenga menos bordes de cruce (...) donde los nodos están conectados entre sí a través de algún parámetro de la fuerza.”* (Kumar, 2008)

El algoritmo utiliza una ley llamada **Ley de Hooke** la cual posibilita que cada nodo, se encuentre en un estado de equilibrio correspondiente a la relación que exista entre ellos, **(Ilustración 8)**. Se muestran los nodos que no tienen relación de forma separada y los que sí, están uno cerca del otro. (Kumar, 2008)

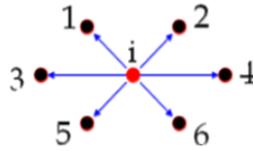


Ilustración 8: Equilibrio en gráficos de nodos (Kumar, 2008)

Ventajas

El algoritmo muestra muy buenos resultados para gráficos con tamaño de (50 a 100 nodos). El diseño gráfico está más simplificado y flexible, de manera que es más fácil de seguir. Además estos gráficos actúan como fuerzas de resorte y se pueden estirar o unir permitiendo realizar exploraciones al usuario para mejorar su visualización.

Desventajas

No es compatible con gráficos de mayor tamaño, más de 100 nodos. Hu comenta que “es prácticamente imposible el empleo de los algoritmos dirigidos por fuerza en el diseño de gráficos muy grandes”, por la existencia de mínimos locales que afectan a la calidad de los resultados obtenidos (Hu, 2006). Otra de las desventajas es que presenta altos tiempos de ejecución, debido a que se utiliza en cada paso hasta que se encuentra en un estado de equilibrio.

1.4.5.2 Métodos de fuerza dirigida

“Los métodos dirigidos por fuerzas se han convertido en la técnica más popular para dibujar grafos generales, y por ende constituyen una de las técnicas principales de graph drawing²⁴. Los principales motivos de esta popularidad son los siguientes: son intuitivos, relativamente sencillos de programar, dan buenos resultados y son muy flexibles.” (Silveira, y otros, 2004)

Estos métodos que están entre los algoritmos más utilizados para el delineado de gráficos, por lo general tratan de satisfacer criterios, y para evaluar este tipo de algoritmo son principalmente criterios estéticos. La simetría, la distribución uniforme de nodos, la longitud uniforme de los enlaces, la reducción del número de cruces de los mismos, entre otros, son esenciales a la hora de elegir por un algoritmo u otro (Abellanas, et al., 2004; Quesada, 2005).

El algoritmo Force-Directed Layout propone varios métodos dirigidos por fuerzas para el trazado de gráficos, en la búsqueda del equilibrio entre los vértices del grafo y las aristas. Siguiendo el esquema propuesto por (Kumar, 2008) existen tres métodos de fuerza dirigida. Ejemplo de ellos lo constituyen: Kamada y Kawai, Fruchterman y Reingold, Davidson y Harel.

²⁴ Dibujo de grafo.

1.4.5.2.1 Método de Kamada-Kawai

Mediante el estudio de los métodos planteados anteriormente se selecciona el método Kamada-Kawai (KK) para representar el algoritmo Force-Directed Layout para la visualización del grafo egocéntrico dirigido y no dirigido. El método se centra en un único criterio estético que implica de cierta manera la longitud de aristas uniforme y los vértices distribuidos uniformemente.

Otro son los motivos que llevan a esta elección tales como: su comportamiento ante los mínimos locales y los buenos tiempos de ejecución.

En general el método *“propone un algoritmo en el que la posición de los vértices no está restringida, y los enlaces son dibujados como líneas rectas. Su propósito es determinar únicamente la posición de los nodos o vértices.”* (Quesada, 2005)

La **(Ilustración 9)** muestra un ejemplo típico del grafo que se obtiene al utilizar el método (Kamada, et al., 1989), donde partiendo del grafo de la izquierda el método KK, obtiene el grafo de la derecha

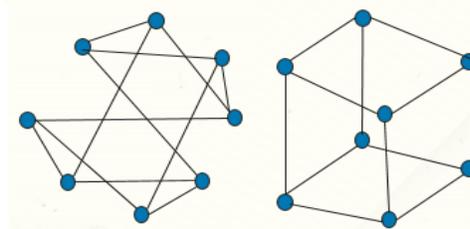


Ilustración 9: Aplicando al grafo de la izquierda el método KK

“Kamada y Kawai proponen un modelo en el que hay un único tipo de fuerzas, que pueden ser vistas como resortes, que conectan entre sí a todos los vértices. Los resortes obedecen a la ley de Hooke (fuerzas lineales) y cada resorte tiene una longitud ideal y una rigidez distinta.” (Silveira, y otros, 2004)

La **(Ilustración 10)** es un ejemplo del método KK usando la API²⁵ Igraph²⁶, donde se puede evidenciar la relación que tienen los nodos en este caso (por los nombres) en un grafo dirigido.

²⁵ *Application Programming Interface (API)*, Interfaz de Programación de Aplicaciones.

²⁶ Es una colección de paquetes de software para la teoría de grafos y análisis de redes.

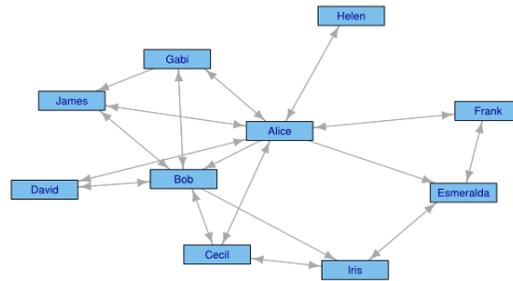


Ilustración 10: Algoritmo de Kamada-Kawai tomado de (Csárdi, 2012)

La (**Ilustración 11**) que se muestra a continuación evidencia la relación entre los nodos en dos gráficos (ambos no dirigidos). Los colores que sobresalen de las aristas muestran una relación fuerte entre estos nodos.

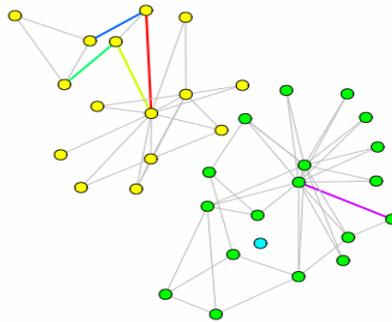


Ilustración 11: Algoritmo de Kamada-Kawai (Igraph, 2012)

“Se trata de un algoritmo sencillo, (...) produce buenos resultados en grafos que contienen cientos de vértices, en un tiempo razonable. Distribuye los nodos en el espacio de tal forma que los más similares tienden a aparecer juntos, mientras que los que lo son menos, se muestran separados. (...) Es posible incorporarle distintas variaciones, (...)” (Quesada, 2005).

1.5 Proceso de desarrollo del software

“Un proceso de desarrollo del software es la descripción de una secuencia de actividades que deben ser seguidas por un equipo de trabajadores para generar un conjunto coherente de productos.” (Drake, 2008)

El objetivo básico del proceso es hacer predecible el trabajo que se requiere:

- Predecir el costo.
- Mantener un nivel de calidad.
- Predecir el tiempo de desarrollo.

No hay constancia de que exista un proceso de desarrollo universal aplicable y efectivo para todos los contextos de proyectos de desarrollo, por tanto, se dice que no es único.



1.6 Metodología de desarrollo de software

Actualmente para el desarrollo de los sistemas informáticos es imprescindible el empleo de las metodologías. El objetivo es contar con un marco de trabajo claramente definido y estandarizado, que permita obtener productos que garanticen la calidad, que cumplan con las expectativas del cliente, se desarrollen en un tiempo estimado y bajo los costos presupuestados.

Sánchez en su libro, define el concepto de metodología de desarrollo como: *“un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevas aplicaciones.”* (Sanchez, 2004)

Las metodologías de desarrollo pueden clasificarse a grandes rasgos en ágiles o ligeras y pesadas o tradicionales. Las tradicionales son aquellas que se centran fundamentalmente en el control del proceso, además son las más efectivas para proyectos de gran tamaño. Mientras que el enfoque ágil da mayor importancia a la capacidad de respuesta a los cambios, se enfatiza en la satisfacción del cliente y promueve el trabajo en equipo. Entre las metodologías ágiles más destacadas hasta el momento se puede nombrar: XP (Extreme Programming), Scrum, Crystal Methodologies, entre otras.

1.6.1 Programación Extrema

XP (por sus siglas en inglés), es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. Se enfoca en el aprendizaje de los desarrolladores y propicia un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, y comunicación fluida entre todos los involucrados, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. *“XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico.”* (Beck, 2000)

El ciclo de vida ideal de XP según Beck, en su libro (Beck, 2000) consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto. Sin embargo Escribano propone cuatro fases, que la constituyen: Planificación, Diseño, Codificación y Pruebas. (Escribano, 2002)

Para guiar el desarrollo de esta aplicación se utilizó la metodología ágil XP porque se centra básicamente en la implementación de las soluciones y no en el soporte documental. Además da respuesta y posible solución a cambios repentinos en las funcionalidades del sistema y concluye un proyecto en el menor tiempo posible. Esta propuesta se debe también al tamaño del grupo de desarrollo, donde todo el trabajo se realiza por una pareja de programadores promoviendo el trabajo en equipo. Logra que cada miembro del equipo incluyendo el cliente de desarrollo esté listo para enfrentar cualquier cambio a medida que avanza el trabajo. El intercambio de opiniones con el



cliente es importante a lo largo de todo el proceso de desarrollo de software con el propósito de alcanzar un producto que satisfaga las necesidades del cliente en el menor tiempo posible y con la calidad requerida.

1.7 Lenguaje de programación y tecnologías

Un lenguaje de programación permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis.

1.7.1 Lenguaje utilizado del lado del servidor

Los lenguajes del lado del servidor son necesarios para desarrollar la lógica del negocio en el servidor, acceder a las bases de datos y procesar la información.

Python

El lenguaje de programación Python se caracteriza por tener una sintaxis extremadamente sencilla y es fácil de aprender. Python es un lenguaje interpretado²⁷ lo que permite ahorrar un tiempo considerable en el desarrollo del programa. Usa tipado dinámico que se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable. Es multiplataforma, lo que significa que no obliga a los programadores a adoptar un estilo particular de programación, permitiendo varios estilos: programación orientada a objetos, programación imperativa y programación funcional.

Debido al soporte que brinda y su integración con otros lenguajes y herramientas, se utiliza para la solución propuesta el intérprete de Python en su versión 2.7 permitiendo el desarrollo de aplicaciones web rápidas y fáciles, además de la integración de numerosas librerías.

NumPy

“Es un módulo de extensión de código abierto para Python que proporciona funciones precompiladas rápidas para las rutinas numéricas. Añade soporte para Python para grandes arreglos y matrices multidimensionales, además de suministrar una gran biblioteca de funciones matemáticas de alto nivel para operar en estas matrices.” (Klein, 2011)

NumPy se basa en dos paquetes de matriz de Python: Uno de ellos es Numérico, para alto rendimiento y la computación numérica; y el otro es Numarray que es una reescritura completa del numérico. *“Una de las principales ventajas de NumPy es su ventaja en el tiempo en comparación con el estándar de Python.” (Klein, 2011)*

²⁷ Lenguaje diseñado para ser ejecutado a través de un intérprete.



“NumPy también puede ser utilizado como un contenedor multi-dimensional eficiente de datos genéricos. Permitiendo su integración sin problemas y rápidamente con una amplia variedad de bases de datos.” (Numpy, 2013)

La librería Numpy cuenta con las funciones definidas en (Tendero, 2011) y para el trabajo se hace uso de las funciones: linspace y digitize que se utilizan para trabajar con los grados de los nodos, digitalizarlos y convertirlos en número.

1.7.2 Lenguajes utilizados del lado del cliente

Las técnicas de desarrollo web y lenguajes del lado del cliente más utilizados, encargados de visualizar la información en el navegador son:

HTML5

Es una nueva versión del lenguaje HTML²⁸, con nuevos elementos, atributos y comportamientos, y un conjunto amplio de tecnologías que permite crear sitios web y diversas aplicaciones de gran alcance (Mozilla Developer Network and individual contributors, 2015).

El nuevo estándar de HTML permite que el formato de la web formado por elementos como cabecera, pie y navegadores se agrupe en nuevas etiquetas que representan cada una de las partes típicas de una página.

CSS 3

CSS3 es la versión más reciente de CSS²⁹ que se usan para definir la presentación de un documento estructurado escrito en HTML o XHTML³⁰. Esta nueva versión permite aumentar y extender las capacidades de un navegador para el soporte de mejores características y mejor presentación de contenido basado en HTML (ACTIV).

El uso de CSS3 proporciona algunas ventajas como:

- Definición de la distancia entre líneas del documento.
- Permite colocar elementos en la página con mayor precisión.

JavaScript

Es un lenguaje interpretado que ha permitido un gran desarrollo en la animación de las páginas web. Su diseño se enfoca específicamente para el desarrollo de aplicaciones cliente-servidor, encargándose de efectuar acciones del lado del cliente. Es muy utilizado para controlar la apariencia y manipular los eventos dentro de la ventana del navegador web, así como para validar datos de

²⁸ *HyperText Markup Language*, Lenguaje de Marcas de Hipertexto.

²⁹ *Cascading Style Sheets*, hojas de estilo en cascada.

³⁰ *eXtensible HyperText Markup Language*, lenguaje de marcado de hipertexto extensible.



entrada en las interfaces de las aplicaciones.

JQuery

Se utilizó JQuery³¹ en su versión 1.8.2 como librería de JavaScript que permite programar páginas dinámicas compatibles con todos los navegadores.

HighCharts

“HighCharts³² es una librería escrita en Javascript que permite la creación de gráficas. La librería ofrece un método fácil e interactivo para insertar gráficas en un sitio web o aplicación web.” (Alvaro, 2013)

Las etiquetas Tooltip (información sobre herramientas) permiten mostrar la descripción de la información de cada punto y serie al mover el mouse sobre el gráfico y con el módulo de exportación activada, los usuarios pueden exportar el gráfico a formato SVG, PNG, JPG, PDF o imprimir el gráfico directamente desde la página web (Hønsi, 2011).

Se utiliza la librería HighCharts para la creación de gráficas lineales o poligonales, haciendo uso del tipo de técnica de visualización de información convencional que permitan mostrar en detalles la información.

1.8 Interfaz de programación de aplicaciones

“Las siglas API provienen del inglés Application Programming Interface (Interfaz de Programación de Aplicaciones). Estas constituyen un conjunto de funciones y procedimientos o métodos, en la programación. Son usadas generalmente en las bibliotecas, denominadas comúnmente librerías para ser utilizadas por otro software como una capa de abstracción. Su propósito principal consiste en brindar un conjunto de funciones de uso general, evitando a los programadores tener que desarrollar todo un software desde el principio”. (Stoughton N, 2005)

IGraph

De código abierto distribuida bajo la licencia GPL³³, *“IGraph es una colección de paquetes de software para la teoría de grafos y análisis de redes. Su núcleo es una biblioteca de software escrito en C / C ++, y tiene interfaces con R y Python GNU.”* (Igraph, 2012).

Además presenta énfasis en la eficiencia, portabilidad y facilidad de uso. La librería permite manipular grafos dirigidos y no dirigidos de miles de vértices y aristas. Utiliza para ello, una variedad de estructuras de datos propias de la librería.

³¹ JQuery: sitio oficial, <http://jquery.com/>

³² HighCharts: sitio oficial, <http://www.highcharts.com/>

³³ *General Public License* (Licencia Pública General). La más extendida de tales licencias es la Licencia Pública General de GNU, o GPL de GNU, para abreviar.



Se utilizará Igraph en el desarrollo de la aplicación web para graficar, debido a que tiene implementado el método Kamada-Kawai estudiado anteriormente.

Cairo

“Cairo es una completa librería gráfica que permite la renderización tanto a pantalla como a impresora de gráficos en 2D. El trabajo con gráficos vectoriales, la integración con GTK+³⁴ y OpenGL³⁵ son características fuertes. Si a esto le sumamos la utilización de un lenguaje como Python, la productividad está asegurada “ (Antón, 2009)

A pesar de que está escrito en C, existen implementaciones en otros lenguajes de programación, incluyendo C++, C#, Java, Python, Perl, Ruby y muchos otros.

“Es un software gratuito y está disponible para ser redistribuido y/o modificado bajo los términos de cualquiera de la Licencia Pública General de GNU (LGPL) versión 2.1 o la Licencia Pública de Mozilla (MPL³⁶) versión 1.” (Cairo, 2014)

Se emplea la librería gráfica Cairo para mostrar los grafos de documentos y autores en la aplicación web.

1.9 Marco de trabajo

Un marco o ambiente de trabajo está orientado a la reutilización de componentes permitiendo el desarrollo rápido de aplicaciones. Los marcos de trabajo a emplear para la implementación del sistema, son Django 1.4 y Twitter Bootstrap 3.0.

Django 1.4

Se hace uso del marco de desarrollo web Django³⁷ en su versión 1.4, de código abierto escrito en Python que permite construir aplicaciones web de forma rápida y con menos código. Django pone énfasis en la reutilización de código, la conectividad, extensibilidad de componentes y un desarrollo rápido.

Dentro de sus características cuenta con:

- Un mapeador objeto-relacional³⁸ (ORM nativo).
- Un sistema incorporado de "vistas genéricas" que ahorra tener que escribir la lógica de ciertas tareas comunes.

³⁴ API GTK+ es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario.

³⁵ OpenGL es un entorno principal de desarrollo, 2D interactiva portátil y aplicaciones de gráficos 3D.

³⁶ *Mozilla Public License* (MPL), una licencia de software libre de código abierto.

³⁷ Django: sitio oficial, <http://django.es/>

³⁸ Técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia (La persistencia es la capacidad que tiene una aplicación de mantener el estado más allá de su ejecución).



- Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.

El núcleo de Django consiste en un mapeo objeto-relacional que media entre los modelos de datos, definidos como clases de Python, y la base de datos relacional; un sistema para procesar peticiones y un despachador de URL basado en expresiones regulares. El ORM de Django brinda una API libre para el acceso dinámico a bases de datos y permite definir modelos directamente dentro del código de una aplicación.

Twitter Bootstrap 3.0

Bootstrap es un marco de trabajo que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. Además, incluye numerosos componentes web que permiten ahorrar esfuerzo y tiempo en el desarrollo de aplicaciones.

Ventajas de su uso:

- Se integra perfectamente con las principales librerías JavaScript.
- Ofrece un diseño sólido usando estándares como CSS3/HTML5.

1.10 Lenguaje de Modelado

El lenguaje de modelado es útil debido a que permiten detectar problemas y proponer soluciones mediante el uso de notaciones gráficas. Facilitan la fluidez en la comunicación de todos los miembros del equipo de desarrollo.

1.10.1 Lenguaje de Modelado Unificado

El lenguaje unificado para la construcción de modelos (UML por sus siglas en inglés) se define como un lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas informáticos. Ocupa un perfil orientado a objetos en el modelado de aplicaciones enfocándose en el diseño de software.

1.10.2 Notación para el Proceso de Modelado del Negocio

Diagrama de Procesos de Negocio (BPMN por sus siglas en inglés) es una notación gráfica que describe los pasos de un proceso de negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. BPMN ocupa un perfil orientado a procesos en el modelado de sistemas, enfocándose en el modelado del negocio.

Ventajas de modelar con BPMN (Bizagi, 2014):

- BPMN crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos.
- Permite modelar los procesos de una manera unificada y estandarizada permitiendo un entendimiento a todas las personas de una organización.



1.11 Herramientas de desarrollo

Para el desarrollo del sistema se utilizarán un conjunto de tecnologías y herramientas que se describen a continuación:

1.11.1 Entorno de desarrollo Integrado

Un entorno de desarrollo integrado (IDE, por sus siglas en inglés), es un programa informático compuesto por un conjunto de herramientas de programación. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

PyCharm

Como entorno de desarrollo integrado se utiliza PyCharm IDE en su versión 4.0 que entre sus principales características se encuentra el autocompletado, el resaltador de sintaxis para código Python, HTML, CSS, JavaScript, así como para las plantillas de Django. PyCharm permite la integración de pruebas unitarias y múltiples opciones para refactorizar el código.

1.11.2 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es una colección de datos relacionados entre sí, estructurados y organizados, además de un conjunto de programas que acceden y gestionan dichos datos.

PostgreSQL 9.2.4

Es un sistema de gestión de base de datos objeto-relacional de código abierto, el cual puede ser ejecutado sobre la mayoría de los sistemas operativos que existen en la actualidad. El sistema es usado para manejar grandes cantidades de información y se basa en el modelo relacional, aunque incorpora conceptos del modelado orientado a objeto. Se destaca por ser robusto y cumplir con los estándares SQL. Es soportado ampliamente por una gran comunidad a nivel mundial (PostgreSQL, 2013).

Se utiliza **PgAdmin** en su versión 1.16.1 como herramienta de código abierto con el propósito general de diseñar, mantener, y administrar las bases de datos de PostgreSQL.

1.11.3 Herramienta Case

La herramienta Case³⁹ de modelado con UML permite aplicar la metodología de análisis y diseño orientado a objeto, además de abstraerse del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios, más fáciles de comprender y modificar.

³⁹ Computer Aided Software Engineering (Ingeniería de Software Asistida por Computadora).



1.11.4 Visual Paradigm for UML 8.0

Para el desarrollo del trabajo se selecciona Visual Paradigm for UML como herramienta Case para el modelado de la propuesta de solución. Es una herramienta de software libre para el modelado visual UML que ayuda a la planificación, análisis y diseño, generación de código fuente y documentación de programas informáticos. Además permite representar todo tipo de diagramas en el ciclo de vida del desarrollo de software.

Algunas de las características que presenta son:

- Disponibilidad en múltiples plataformas.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Modelo y código permanecen sincronizados en todo el ciclo de desarrollo.

Conclusiones parciales del capítulo

En el presente capítulo se realizó un estudio de las principales tendencias relacionadas con la aplicación a desarrollar.

- Se demostró la necesidad de representar la información mediante técnicas de visualización de información en las herramientas de búsqueda tanto a nivel nacional como internacional.
- La revisión de la documentación de trabajos relacionados con las técnicas y como estas representan la información, permitió elegir las diferentes técnicas a utilizar en el desarrollo de la aplicación.
- Posteriormente se realizó el análisis detallado de la metodología, las herramientas y tecnologías a utilizar, para una mejor perspectiva del funcionamiento de la herramienta.

Capítulo 2 *Propuesta del Sistema*

2.1 Introducción

El presente capítulo tiene un enfoque práctico y funcional, cuyo objetivo es presentar el resultado del análisis y diseño de la propuesta de la aplicación web. Además, de especificar los requisitos funcionales y no funcionales que debe cumplir para satisfacer las necesidades del cliente. Se establecen los artefactos requeridos en la planificación definidos por la metodología XP (Extreme Programming).

2.2 Propuesta de solución de la aplicación web

La aplicación web Representación de Bibliografías Científicas (RBC) será capaz de mostrar de forma detallada la información bibliográfica de publicaciones científicas realizadas por autores en el área de las Ciencias Básicas y la Cibernética, con la ayuda de gráficos que permitan representar y comprender mejor la información.

El usuario al interactuar con la aplicación puede introducir un criterio de búsqueda y tendrá la posibilidad de buscar información sobre estas áreas. Esta búsqueda va a estar implementada mediante consultas hechas a una base de datos donde se encontrará almacenada toda la información.



Ilustración 12: Propuesta de solución Representación de Bibliografías Científicas

La herramienta RBC mostrará opciones de búsqueda simple, sin dejar atrás una búsqueda más avanzada donde los usuarios podrán filtrar la búsqueda de acuerdo a los campos que esta ofrece (por la frase exacta y sin las palabras en el documento). Cada documento mostrará sus datos bibliográficos, dígame (título, tipo de documento, cantidad de citas, el autor(es), un breve resumen y el vínculo donde se encontrará ubicado dicho documento. De los autores se mostrará el

nombre, la afiliación a la que pertenece, el correo electrónico, la cantidad de publicaciones y colaboradores, la unidad organizativa y las organizaciones. Además al mostrar los resultados de la búsqueda permitirá ordenar la búsqueda por el título, el tipo de documento y los autores de forma ascendente y descendente.

La aplicación representará mediante un gráfico lineal para los documentos, los reportes de publicaciones y citas de un documento seleccionado en un rango de fechas desde el 2010 al 2015. Además, representará para el autor una gráfica lineal con el reporte de las publicaciones de dicho autor en un rango de fechas del 2012 al 2015. Esta forma de representar la información permitirá a los usuarios interactuar directamente con los datos estadísticos mostrados en la gráfica. También contará con una novedosa forma de representar la información mediante el empleo de grafos egocéntricos. Estos grafos estarán centrados en un documento o autor específico. La relación de cada uno de ellos estará dada por la dirección de las aristas del grafo, mostrando así la relación de un documento con otros mediante un grafo dirigido y la relación de un autor con sus colaboradores mediante un grafo no dirigido. Las aristas para el grafo del documento brindarán información de acuerdo a sus referencias (si lo están citando, las aristas apuntarán hacia el documento seleccionado y si en él se cita algún otro documento las aristas apuntarán hacia los documentos que él cite). Los grafos permitirán navegar por un solo nivel de jerarquía mostrando la relación directa con el documento o autor seleccionado. Además, ofrecerá al usuario una manera más fácil para comprender e interpretar la información que está localizando.

El grafo de documentos brindará a los espectadores detalles que visualicen los documentos más citados en la relación y aquellos que hayan sido publicados en el intervalo de 2010 al 2015. El grafo de autores representará aquellos autores que se relacionen con el autor seleccionado y además tengan publicaciones en el intervalo de 2012 al 2015.

La **(Ilustración 13)** muestra un grafo no dirigido centrado en un autor. El autor está representado por un círculo en azul y los nombres de los co-autores que colaboraron con él, se muestran de color naranja. La arista verde en la relación representa el autor que ha publicado documentos con el autor central en los últimos 3 años. Este diseño permitió simular el grafo de autores a desarrollar.

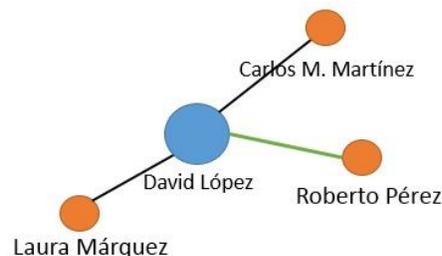


Ilustración 13: Grafo egocéntrico no dirigido centrado en un autor



El cliente (miembro del Instituto ICIMAF) dispone de una base de datos de información bibliográfica científica especializada en el área de la Ciencias Básicas y la Cibernética. La herramienta RBC podrá acceder a la base de datos del cliente donde obtendrá la información científica almacenada para luego mostrarla al usuario. Este hecho permitirá que los usuarios puedan acceder libremente a la herramienta en la búsqueda de información científica en estas áreas, sin la necesidad de buscar en Internet.

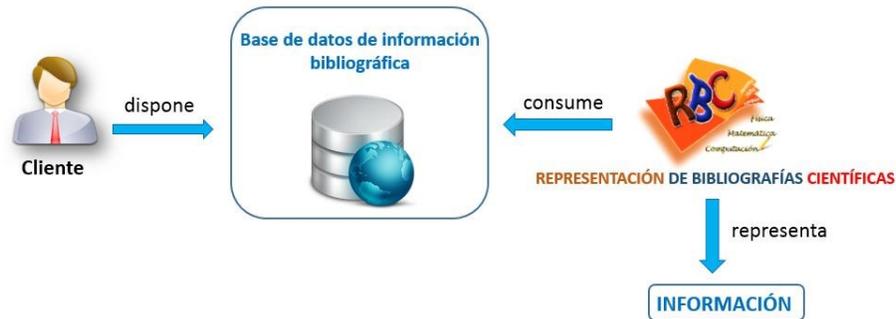


Ilustración 14: Funcionamiento de RBC

La aplicación contará con un espacio donde se podrán gestionar todos los procesos asociados a la gestión de documentos y los usuarios. El cliente primeramente para acceder a la interfaz de administración deberá autenticarse en el sistema y una vez dentro será el responsable de gestionar los documentos y los usuarios autorizados a acceder a la administración. La **(Ilustración 15)** muestra el proceso de acceso a la gestión de documentos y usuarios en la administración.



Ilustración 15: Proceso de acceso a la gestión de documentos y usuarios en la administración

2.3 Modelo del Dominio

El Modelo de Dominio o Modelo Conceptual es una representación visual de los conceptos u objetos que se manejan en el dominio del sistema. Los objetos o conceptos incluidos en el Modelo de Dominio no describen clases u objetos del software; sino entidades o conceptos del mundo real que están asociados al problema en cuestión. Dicho modelo podrá ser utilizado como una base de las abstracciones relevantes en el proceso de construcción del sistema.

2.3.1 Conceptos fundamentales del dominio

A continuación se proporcionará una breve descripción de los conceptos encontrados en el ámbito del problema, con el fin de facilitar una mejor comprensión del Diagrama del Modelo de Dominio.

Usuario: Persona que por medio de un ordenador puede acceder a la herramienta para realizar la búsqueda.

Cliente: Es un usuario del sistema que además, es el responsable de la gestión de documentos y otros usuarios.

Criterio de búsqueda: El usuario podrá introducir un criterio de búsqueda para buscar la información mediante la búsqueda básica y la búsqueda avanzada.

Resultados de la búsqueda: Es un espacio donde se mostrarán los resultados de la búsqueda definida por el usuario.

Administración: Es un espacio ubicado dentro del sistema, donde se podrán gestionar todas las funcionalidades.

Gestionar Documento: Es la funcionalidad donde se podrán gestionar todos los datos referentes a la gestión de documentos.

Gestionar Usuario: Es la funcionalidad donde se podrán gestionar todos los datos referentes a los usuarios autorizados a acceder a la administración.

Base de datos de publicaciones científicas: Es el espacio donde se encontrarán todos los datos referentes a los documentos, los autores y donde se comprobarán los datos de acceso a la administración.

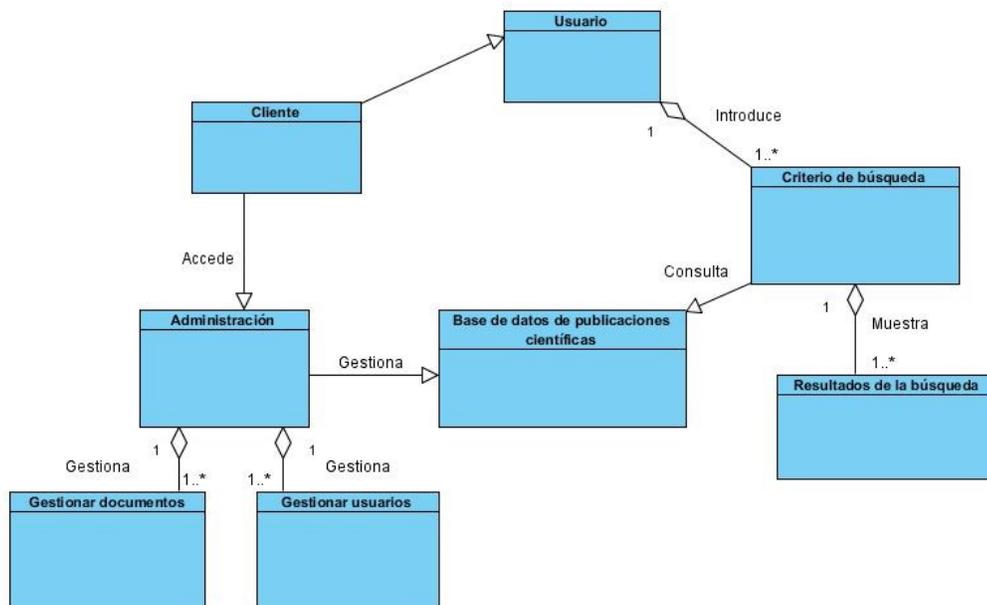


Ilustración 16: Diagrama de Modelo de Dominio



Como se evidencia en el Diagrama de Modelo de Dominio, un usuario tiene la posibilidad de buscar información al introducir un criterio de búsqueda en el sistema. Esta búsqueda va a estar implementada mediante consultas hechas a una base de datos donde se encontrará almacenada toda la información. Luego se mostrará al usuario los resultados de la búsqueda realizada de forma detallada. El cliente que a su vez es un usuario podrá acceder a la administración después de autenticarse correctamente en el sistema y tendrá la responsabilidad de gestionar los documentos y los usuarios que tengan permitido el acceso a la administración.

2.3.2 Personas relacionadas con el sistema a desarrollar

Tabla 1: Descripción de las personas relacionadas con el sistema

Personas	Descripción
Cliente	Es el responsable de gestionar todos los procesos asociados a la gestión de documentos y usuarios que tengan permitido el acceso a la administración del sistema.
Usuario	El usuario podrá realizar la búsqueda en el sistema y observar los resultados de las publicaciones científicas definidas en la búsqueda, además de seleccionar los documentos de su interés.

2.4 Captura de requisitos

La ingeniería de requisitos es la encargada de recoger todos los detalles de los requisitos funcionales y no funcionales obtenidos de un extenso proceso de entendimiento con el cliente y es uno de los procedimientos más importantes para el desarrollo de cualquier software. En esta etapa el principal objetivo es identificar lo que el cliente realmente necesita, de manera que el equipo de desarrollo pueda trabajar en base a estos requerimientos y logre obtener un producto que funcione con la calidad deseada.

2.4.1 Requisitos funcionales del sistema

Los requisitos funcionales de un software definen el comportamiento interno de la aplicación, describen en detalles cada funcionalidad siempre aprobados en mutuo acuerdo con el cliente. Según las necesidades del cliente las funcionalidades definidas fueron elaboradas de acuerdo a la importancia en que cada una necesitaba de la otra para su correcto funcionamiento.

La tabla muestra un total de 48 requisitos funcionales que la aplicación web debe ser capaz de brindar.

Tabla 2: Definición de los requisitos funcionales

Número	Nombre	Prioridad
1	Búsqueda de datos bibliográficos	Alta
Gestionar un documento		
2	Registrar un documento	Media
3	Modificar un documento	Media
4	Eliminar un documento	Media
5	Listar un documento	Media
Gestionar una palabra clave		
6	Registrar una palabra clave	Media
7	Modificar una palabra clave	Media
8	Eliminar una palabra clave	Media
9	Listar una palabra clave	Media
Gestionar una palabra clave de un documento		
10	Registrar una palabra clave de un documento	Media
11	Listar una palabra clave de un documento	Media
Gestionar una persona		
12	Registrar una persona	Media
13	Modificar una persona	Media
14	Eliminar una persona	Media
15	Listar una persona	Media
Gestionar una autoría		
16	Registrar una autoría	Media
17	Listar una autoría	Media
Gestionar una afiliación		
18	Registrar una afiliación	Media
19	Listar una afiliación	Media
Gestionar una organización		
20	Registrar una organización	Media
21	Modificar una organización	Media
22	Eliminar una organización	Media
23	Listar una organización	Media

Gestionar una unidad de organización		
24	Registrar una unidad de organización	Media
25	Modificar una unidad de organización	Media
26	Eliminar una unidad de organización	Media
27	Listar una unidad de organización	Media
Gestionar una pertenencia a		
28	Registrar una pertenencia a	Media
29	Listar una pertenencia a	Media
Gestionar una publicación		
30	Registrar una publicación	Media
31	Modificar una publicación	Media
32	Eliminar una publicación	Media
Gestionar una revista		
33	Registrar una revista	Media
34	Modificar una revista	Media
35	Eliminar una revista	Media
36	Listar una revista	Media
Gestionar una referencia		
37	Registrar una referencia	Media
38	Modificar una referencia	Media
39	Eliminar una referencia	Media
40	Autenticar usuario	Media
Gestionar usuario		
41	Registrar usuario	Baja
42	Eliminar usuario	Baja
43	Listar usuarios	Baja
44	Mostrar en detalle los datos bibliográficos de los autores	Alta
45	Mostrar en detalle los datos bibliográficos de los documentos	Alta
46	Mostrar de forma gráfica la relación entre autores	Alta
47	Mostrar de forma gráfica la relación entre documentos	Alta
48	Mostrar documentos de forma ordenada	Media



2.4.2 Requisitos no funcionales del sistema

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son características que hacen al producto atractivo, usable, rápido o confiable. Estos requerimientos se agrupan en varias categorías y conforman un total de 13 requisitos no funcionales:

❖ Usabilidad

1. La aplicación deberá presentar una interfaz que permita la fácil interpretación por el usuario, además de llegar de manera rápida y efectiva a la información buscada.
2. Posibilitará al usuario sin experiencia una rápida adaptación para operar con el sistema, teniendo en cuenta que la aplicación posee un buen diseño e interfaces intuitivas, que permiten una mejor comprensión por el usuario.

❖ Disponibilidad

1. La aplicación debe estar disponible en todo momento para los usuarios autorizados que necesiten acceder y manejar la información contenida en la misma.

❖ Confiabilidad

1. El sistema deberá mostrar un mensaje de confirmación antes de realizar cualquier acción definitiva o irreversible sobre los datos, como modificación o eliminación.

❖ Seguridad

1. Solo podrá ser accedida para manipular y gestionar la información de documentos y usuarios en la interfaz de administración las personas autorizadas.
2. Las contraseñas podrán cambiarse únicamente por el propio usuario o por el cliente que es el responsable de asignar a estos usuarios el acceso a la administración.

❖ Apariencia o Interfaz

1. La aplicación propuesta contará con una interfaz fácil de usar, sencilla que permita interactuar de forma cómoda a los usuarios.

❖ Hardware

Para el usuario:

1. Requerimientos mínimos 512MB de RAM recomendada o superior.
2. Tarjeta de red para establecer la conexión.

Para el servidor:

1. Computador con procesador Intel Xeon que es el tipo de microprocesador que utilizan los servidores de la Universidad de las Ciencias Informáticas (UCI), 4 GB de memoria RAM, 1 Tb de disco duro.
2. Tarjeta de red para establecer la conexión.

❖ Software



Para el usuario:

1. Sistema Operativo Windows XP, Windows 7 o Windows 8 y Linux con interfaz gráfica y soporte para conectarse a la red.
2. Para la utilización del sistema se requerirá el uso de un navegador web Mozilla Firefox 30 o superior.

Para el servidor:

1. En cuanto a las restricciones de software para el servidor se requiere el gestor de base de datos PostgreSQL 9.2.4, el marco de trabajo Django 1.4, las librerías y dependencias necesarias como Cairo, Igraph, Numpy, ipython entre otras y Sistema Operativo Windows 7, Windows 8 o Linux.

2.5 Etapa de Planificación

La planificación es la primera fase definida en la metodología XP propuesta por (Escribano, 2002). El cliente define a grandes rasgos lo que necesita mediante la redacción de sencillas Historias de Usuarios que son de gran interés para la entrega del producto y posibilita a los programadores estimar los tiempos de desarrollo. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el desarrollo de la herramienta.

2.5.1 Historias de Usuario (HU)

Las HU representan una breve descripción del comportamiento de la herramienta. Son utilizadas para especificar los requisitos funcionales del software desde el punto de vista del cliente, siendo el realizador de una HU por cada característica principal de la aplicación. Es el encargado de asignarles una prioridad, definir la estimación de riesgo del desarrollo y cuánto tiempo conllevará dicha implementación, además servirán para crear el plan estimado de entregas.

Las HU son representadas mediante tablas divididas por diferentes secciones, a continuación se expone una muestra de la HU que presenta prioridad alta en el negocio. Para ver más en los anexos.

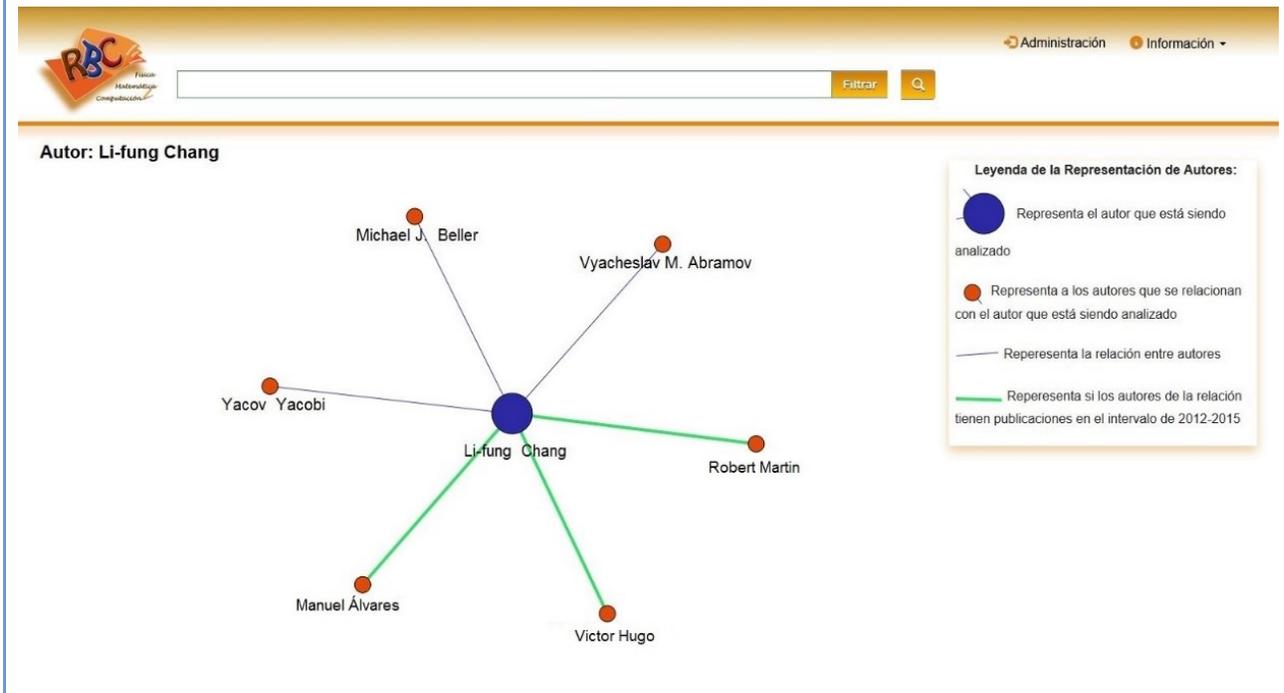
Tabla 3: Mostrar de forma gráfica la relación entre autores

Historia de usuario	
Número: 46	Usuario: cliente
Nombre de Historia de Usuario: Mostrar de forma gráfica la relación entre autores	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 7/5	Iteración Asignada: 4
Programador(es) responsable(s): Yeneisy Barroso e Ivelisse Montero	

Descripción: Inicia cuando el cliente luego de haber definido la búsqueda, y mostrarse en pantalla los resultados de la misma, decide si desea ver la relación de un autor en específico con otros.

Observaciones: Si el autor no tiene co-autores no se mostrará el grafo con las relaciones.

Prototipo de interfaz:



2.5.2 Estimación de esfuerzo por Historias de Usuario

Las estimaciones de esfuerzo asociada a la implementación de las HU, las establecen los programadores, y el cliente decide sobre el ámbito y tiempo de las entregas utilizando como medida el punto. Un punto, equivale a una semana ideal de programación, planificada para 3 o 5 días, donde los miembros del equipo de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. A continuación se muestra la estimación del esfuerzo por cada HU propuesta para el desarrollo de la aplicación.

Tabla 4: Estimación de esfuerzo por HU

No.HU	Historias de Usuario	Puntos de estimación
1	Búsqueda de datos bibliográficos	7/5
2	Registrar un documento	1/5
3	Modificar un documento	1/5
4	Eliminar un documento	1/5
5	Listar un documento	1/5



6	Registrar una palabra clave	1/5
7	Modificar una palabra clave	1/5
8	Eliminar una palabra clave	1/5
9	Listar una palabra clave	1/5
10	Registrar una palabra clave de un documento	1/5
11	Listar una palabra clave de un documento	1/5
12	Registrar una persona	1/5
13	Modificar una persona	1/5
14	Eliminar una persona	1/5
15	Listar una persona	1/5
16	Registrar una autoría	1/5
17	Listar una autoría	1/5
18	Registrar una afiliación	1/5
19	Listar una afiliación	1/5
20	Registrar una organización	1/5
21	Modificar una organización	1/5
22	Eliminar una organización	1/5
23	Listar una organización	1/5
24	Registrar una unidad de organización	1/5
25	Modificar una unidad de organización	1/5
26	Eliminar una unidad de organización	1/5
27	Listar una unidad de organización	1/5
28	Registrar una pertenencia a	1/5
29	Listar una pertenencia a	1/5
30	Registrar una publicación	1/5
31	Modificar una publicación	1/5
32	Eliminar una publicación	1/5
33	Registrar una revista	1/5
34	Modificar una revista	1/5
35	Eliminar una revista	1/5
36	Listar una revista	1/5
37	Registrar una referencia	1/5



38	Modificar una referencia	1/5
39	Eliminar una referencia	1/5
40	Autenticar usuario	1/5
41	Registrar usuario	1/5
42	Eliminar usuario	1/5
43	Listar usuarios	1/5
44	Mostrar en detalle los datos bibliográficos de los autores	2/5
45	Mostrar en detalle los datos bibliográficos de los documentos	2/5
46	Mostrar de forma gráfica la relación entre autores	7/5
47	Mostrar de forma gráfica la relación entre documentos	7/5
48	Mostrar documentos de forma ordenada	1/5

2.5.3 Plan de Iteraciones

Una iteración es un conjunto de períodos de tiempo dentro de un proyecto, en el cual se produce una versión ejecutable del producto. El plan de iteraciones incluye varias iteraciones sobre el sistema antes de ser entregado. Todo el trabajo de la iteración es expresado en tareas de programación. Posteriormente de ser detalladas las HU y estimar el esfuerzo dedicado a la realización de cada una de ellas, se establece una división de cuatro iteraciones.

✓ Iteración 1

En la iteración 1 se llevará a cabo el desarrollo de la HU número 1, donde se muestra la búsqueda de datos bibliográficos, mediante distintas opciones (por búsqueda simple y avanzada). El usuario al ingresar un criterio de búsqueda en la aplicación, se mostrará los resultados obtenidos de dicha búsqueda. Al terminar la iteración esto representará un 10.30 % de la implementación de la aplicación.

✓ Iteración 2

En la iteración 2 se llevará a cabo el desarrollo de las HU del número 2 hasta el número 23, perteneciente al sistema que gestiona cada una de las funcionalidades del sistema. Al terminar la iteración esto representará un 42.65 % de la implementación de la aplicación.

✓ Iteración 3

En la iteración 3 se llevará a cabo el desarrollo de las HU del número 24 hasta el número 43, perteneciente al sistema que gestiona cada una de las funcionalidades del sistema. Al terminar la iteración esto representará un 72.06 % de la implementación de la aplicación.



✓ **Iteración 4**

En la iteración 4 se llevará a cabo el desarrollo de las HU desde el número 44 hasta el número 48 donde se mostrará de forma gráfica la relación entre autores y la relación entre documentos. Además de mostrar una vista más detallada de los resultados obtenidos por documento, así como de un autor en específico. También mostrará de forma ordenada por el título, el tipo de documento y los autores de manera ascendente y descendente. Al terminar la iteración esto representará un 100 % de la implementación de la aplicación.

2.5.4 Plan de duración de las iteraciones

El plan de duración de las iteraciones se encarga de mostrar las HU en el orden en que se implementarán en cada iteración así como la duración estimada de las mismas.

Tabla 5: Plan de duración de las iteraciones

Iteración	Orden de las Historias de Usuario a implementar	Duración total
1	✓ Búsqueda de datos bibliográficos	2 semanas
2	<ul style="list-style-type: none"> ✓ Registrar un documento ✓ Modificar un documento ✓ Eliminar un documento ✓ Listar un documento ✓ Registrar una palabra clave ✓ Modificar una palabra clave ✓ Eliminar una palabra clave ✓ Listar una palabra clave ✓ Registrar una palabra clave de un documento ✓ Listar una palabra clave de un documento ✓ Registrar una persona ✓ Modificar una persona ✓ Eliminar una persona ✓ Listar una persona ✓ Registrar una autoría ✓ Listar una autoría ✓ Registrar una afiliación ✓ Listar una afiliación ✓ Registrar una organización ✓ Modificar una organización 	4 semanas

	<ul style="list-style-type: none"> ✓ Eliminar una organización ✓ Listar una organización 	
3	<ul style="list-style-type: none"> ✓ Registrar una unidad de organización ✓ Modificar una unidad de organización ✓ Eliminar una unidad de organización ✓ Listar una unidad de organización ✓ Registrar una pertenencia a ✓ Listar una pertenencia a ✓ Registrar una publicación ✓ Modificar una publicación ✓ Eliminar una publicación ✓ Registrar una revista ✓ Modificar una revista ✓ Eliminar una revista ✓ Listar una revista ✓ Registrar una referencia ✓ Modificar una referencia ✓ Eliminar una referencia ✓ Autenticar usuario ✓ Registrar usuario ✓ Eliminar usuario ✓ Listar usuarios 	4 semanas
4	<ul style="list-style-type: none"> ✓ Mostrar de forma gráfica la relación entre autores ✓ Mostrar de forma gráfica la relación entre autores ✓ Representación de la relación entre autores ✓ Representación de la relación entre documentos ✓ Mostrar documentos de forma ordenada 	5 semanas

2.5.5 Plan de entregas

Luego de elaborar las Historias de Usuario, se realiza el plan de entrega donde se estima el tiempo de desarrollo de las HU, para marcar cuanto tiempo se demora la implementación de cada una.

Tabla 6: Planificación de entregas de las iteraciones

Herramienta	Final			
	1ra iteración	2da iteración	3ra iteración	4ta iteración
Aplicación web “Representación de Bibliografías Científicas” (RBC)	27-2-2015	26-3-2015	21-4-2015	28-5-2015

2.6 Etapa de Diseño

Se describen las fases de diseño propias de la metodología de desarrollo XP y se detalla la arquitectura del sistema, así como patrones de diseño utilizados en el desarrollo de la aplicación.

2.6.1 Diseño de la arquitectura

Una arquitectura de software “*constituye un modelo comprensible de cómo está estructurado el sistema y cómo trabajan juntos sus componentes.*” (Cervantes, 2010)

Para el desarrollo del sistema se empleará la arquitectura cliente-servidor pues cada usuario podrá acceder a la aplicación web y podrá operar sobre el mismo realizando peticiones al servidor donde se encontrará alojado el sistema. A continuación se muestra en la **(Ilustración 17)** dicha arquitectura para comprender mejor su funcionamiento.

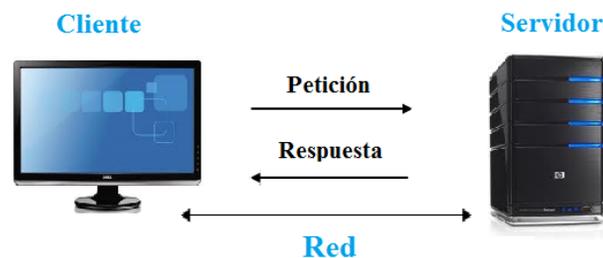


Ilustración 17: Arquitectura Cliente-Servidor

2.6.2 Patrón Arquitectónico

Un patrón arquitectónico de software representa un diseño organizativo estructural fundamental para guiar el desarrollo del software. En la arquitectura Modelo-Vista-Controlador (MVC), el marco de desarrollo Django usa una modificación llamada MTV Model-Template-View que sería Modelo-Plantilla-Vista empleado en la implementación de la solución. El modelo en Django continúa siendo modelo, la vista se llama Plantilla y el controlador se nombra Vista.

- **La Capa Modelo (models):** Se refiere a la capa de acceso a datos. Esta capa contiene todo lo referido a los datos: cómo acceder a ellos, cómo validarlos, qué comportamiento tienen y las relaciones entre ellos.

- **La Capa Plantilla (templates):** Se refiere a la capa de presentación. Esta capa contiene las decisiones relacionadas con la presentación: cómo debería mostrarse la información.
- **La Capa Vista (views):** Esta capa contiene el acceso al modelo y delega en las plantillas apropiadas.

Como parte del funcionamiento del patrón MTV, primeramente el navegador manda una solicitud a la vista (views), la urls que permite controlar el despliegue de las vistas interpreta la solicitud y luego la vista interactúa con el modelo (models) para obtener los datos, después hace una llamada a la plantilla (templates) y la plantilla se encarga de renderizar la respuesta a la solicitud del navegador. Seguidamente se muestra en la **(Ilustración 18)** dicha relación.

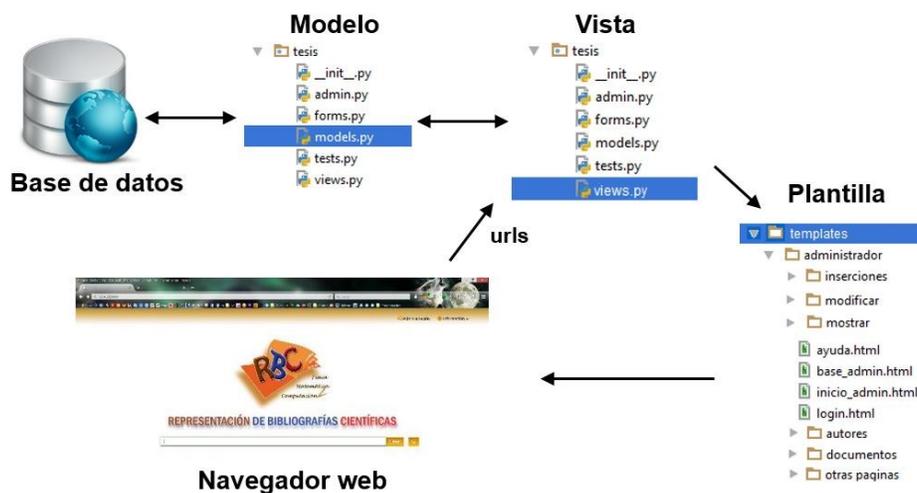


Ilustración 18: Patrón de Arquitectura Modelo-Plantilla-Vista

2.6.2.1 Representación de las capas de la arquitectura MTV

A continuación se describen cada una de las capas de la arquitectura del patrón MTV que se encuentra estructurada de la siguiente forma: Modelo, Plantilla, Vista.

Capa Modelo

“El modelo define los datos almacenados, se encuentra en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros, (...). Todo esto permite indicar y controlar el comportamiento de los datos.” (Montero, 2012)

El modelo representa la lógica del negocio y permite gestionar la información, además de acceder a la capa de almacenamiento de datos actuando como “intermediario”. En el archivo o paquete denominado models.py se almacenan las clases entidades pertenecientes al modelo, las cuales representan un mapeo de las tablas de la base de datos **(Ilustración 19)**.

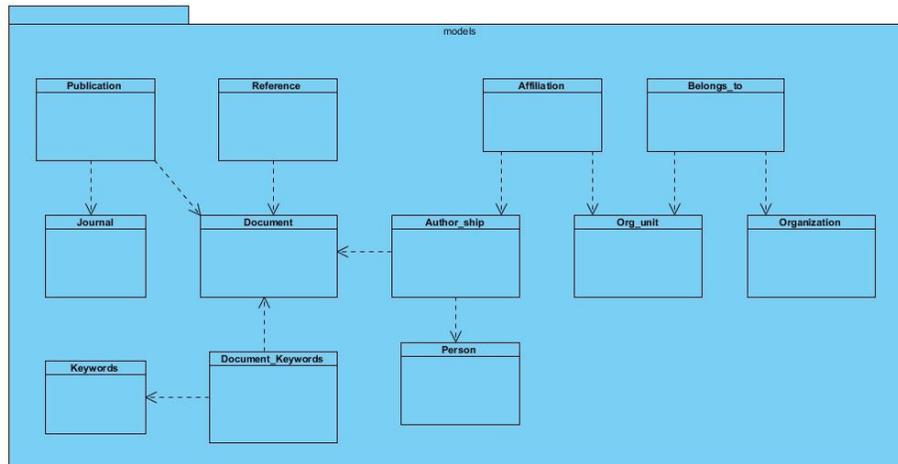


Ilustración 19: Contenido del paquete models

document: clase responsable de registrar, modificar, eliminar y listar los detalles de un documento.

keywords: clase responsable de registrar, modificar, eliminar y listar los detalles de una palabra clave.

document_keywords: clase responsable de registrar, modificar y eliminar los detalles de una palabra clave de un documento.

org_unit: clase responsable de registrar, modificar, eliminar y listar los detalles de las unidades organizativas.

publication: clase responsable de registrar, modificar y eliminar los detalles de una publicación.

reference: clase responsable de registrar, modificar y eliminar los detalles de una referencia.

affiliation: clase responsable de registrar, modificar y eliminar los detalles de una afiliación.

authorship: clase responsable de registrar, modificar y eliminar los detalles de una comunidad de autores.

belongs_to: clase responsable de registrar, modificar y eliminar los detalles que pertenezcan a una unidad organizativa o una organización.

journal: clase responsable de registrar, modificar, eliminar y listar los detalles de una revista.

person: clase responsable de registrar, modificar, eliminar y listar los detalles de un autor.

organization: clase responsable de registrar, modificar, eliminar y listar los detalles de una organización.

Capa Plantilla

La plantilla es la encargada de intercambiar información con el usuario tanto de entrada como salida.

Recibe los datos de la vista y luego los organiza para la presentación de los datos al navegador web.

Las plantillas del sistema se encuentran dentro de la carpeta llamada templates (**Ilustración 20**) y se organizan en carpetas independientes para cada entidad. Son clases que a través de JavaScript, CSS, HTML, y la librería JQuery interactúan con el usuario para obtener y mostrar información.

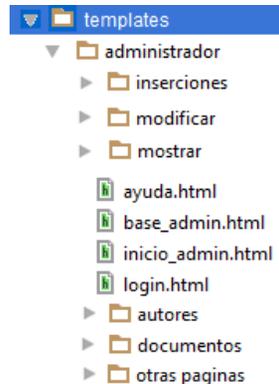


Ilustración 20: Contenido de las vistas dentro de la carpeta templates

Capa Vista

“La vista se presenta en forma de funciones en Python, su propósito es determinar qué datos serán visualizados.” (Montero, 2012)

La vista también se encarga de la validación de datos a través de formularios. Es el intermediario entre la plantilla y el modelo. La vista recibe las peticiones del usuario a través de los métodos que contienen las clases librerías para hacer peticiones de datos a las clases modelos, que tienen el acceso a los datos. Luego al obtenerlos se los envía a través de las librerías a la vista, que una vez que los recibe se los entrega a las plantillas. En el archivo o paquete denominado views.py se almacenan las funciones de las vistas (**Ilustración 21**).

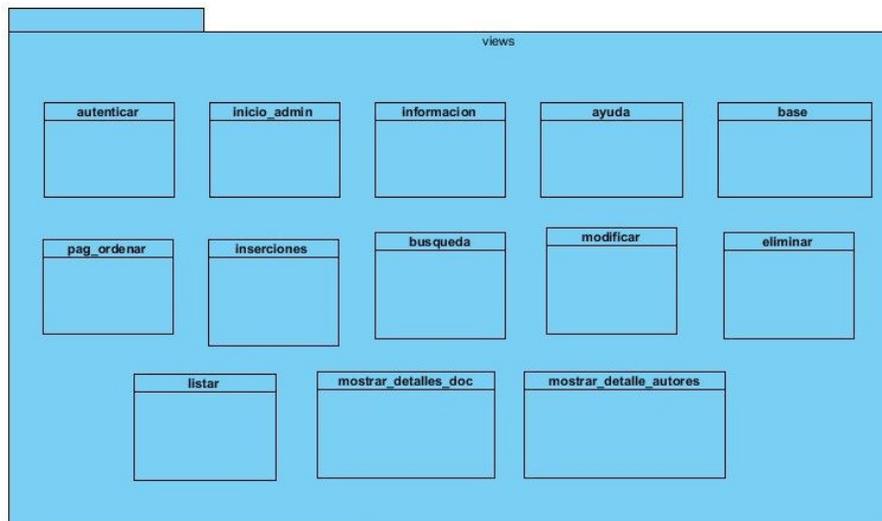


Ilustración 21: Contenido del paquete views



autenticar: en este archivo se encuentra el método para que los usuario se autentiquen y puedan acceder a la página de administración.

inicio_admin: en este archivo se encuentra el método necesario para que se le muestre al usuario la página de administración.

informacion: en este archivo se encuentra el método necesario para que se muestre al usuario la información.

ayuda: en este archivo se encuentra el método necesario para que se muestre al usuario la ayuda.

base: en este archivo se encuentra el método necesario para que se muestre al usuario la página principal.

busqueda: en este archivo se encuentran los métodos necesarios para realizar la búsqueda.

inserciones: en este archivo se encuentran los métodos necesarios para insertar (documentos, autores, palabras clave, organizaciones, unidades organizativas, usuarios, referencias y revistas).

modificar: en este archivo se encuentran los métodos necesarios para la modificación de (documentos, autores, palabras clave, organizaciones, unidades organizativas, referencias y revistas).

eliminar: en este archivo se encuentran los métodos necesarios para realizar la eliminación a (documentos, autores, palabras clave, organizaciones, unidades organizativas, usuarios, referencias y revistas).

listar: en este archivo se encuentran los métodos necesarios para el manejo del listado de (documentos, autores, palabras clave, organizaciones, unidades organizativas, referencias, revistas y usuarios).

mostrar_detalle_doc: en este archivo se encuentran los métodos necesarios para el manejo de la visualización de los documentos.

mostrar_detalle_autores: en este archivo se encuentran los métodos necesarios para el manejo de la visualización de los autores.

pag_ordenar: en este archivo se encuentran los métodos necesarios para el manejo de todos los ordenamientos posibles.

2.6.2.2 Diagrama de componentes

Los diagramas de componentes permiten describir los elementos físicos que integran el sistema y las relaciones que existen entre ellos. *“Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente se refiere a los servicios ofrecidos por otro componente.”* (Roldan, 2009)

Los componentes representan todos los tipos de elementos de software que entran en la producción de aplicaciones informáticas. Pueden ser simples archivos, paquetes, y/o bibliotecas cargadas

dinámicamente. A continuación se expone el diagrama de componentes asociado al sistema implementado donde se utilizó la dependencia <<use>> que conecta entre interfaces y componentes.

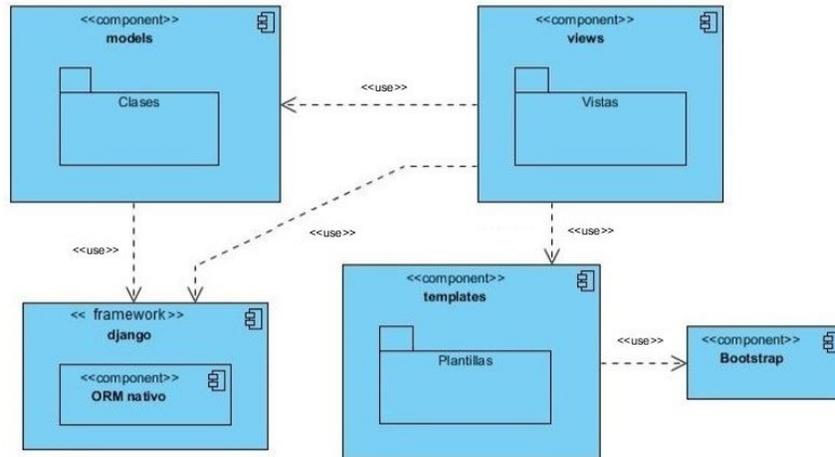


Ilustración 22: Diagrama de componentes del sistema

2.6.3 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. En otras palabras, *“brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.”* (Tedeschi, 2010)

Dentro de los patrones de diseño se encuentran los Patrones Generales de Software para Asignación de Responsabilidades (**GRASP**⁴⁰ por sus siglas en inglés), los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos. Seguidamente se explican los patrones GRASP que fueron utilizados para el desarrollo de la aplicación.

- El patrón Creador *“indica a qué clase se le asigna la responsabilidad de la creación de instancias, puesto que esta posee la información necesaria para la creación del objeto.”* (Grosso, 2011)
 - Ejemplo: Los formularios UserCreationForm y AuthenticationForm son responsable de la creación de objetos.

⁴⁰ GRASP, acrónimo de General Responsibility Assignment Software Patterns.

```
def nuevo_usuario(request):
    if request.method == 'POST':
        form = UserCreationForm(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponseRedirect('/usuario')
    else:
        form = UserCreationForm()
    return render_to_response('administrador/inserciones/usuario.html', {'form': form},
                              context_instance=RequestContext(request))
```

Ilustración 23: Ejemplo del formulario UserCreationForm para validar un nuevo usuario

- El Patrón Experto consiste en asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Su empleo se ve reflejado en la aplicación, mediante la declaración de los atributos y funciones destinadas al trabajo de las clases. Este indica que la responsabilidad de la creación de las tablas de la base de datos, debe recaer sobre la clase que conoce toda la información necesaria para crearlo.
 - Ejemplo: Las clases Document, Person y Reference cuentan con la información necesaria para cumplir cada una de las responsabilidades que le corresponden.

```
class Person(models.Model):
    first_name=models.CharField(max_length=400)
    last_name=models.CharField(max_length=400)
    def __str__(self):
        return ' %s %s' % (self.first_name, self.last_name)
```

Ilustración 24: Ejemplo de la clase Person

- El Patrón Bajo Acoplamiento consiste en asignar una responsabilidad para mantener bajo acoplamiento. *“El acoplamiento es una medida de la fuerza con que una clase está conectada a otras, con las que conoce y con que recurre a ellas.”* (Buono, 2012)
El patrón propone el diseño de clases más independientes, lo que reduce el impacto del cambio y facilita la reutilización en otros sistemas. Permite definir las funcionalidades de cada clase sin que estas presenten tanta dependencia entre ellas, así los cambios realizados en una, no afecta el funcionamiento de la otra. Dado que el marco de desarrollo Django introduce la utilización de vistas genéricas, brinda la posibilidad de reutilizar las funciones definidas en otras operaciones del sistema.
 - Ejemplo: A las clases Organization, Person y Org_unit se le asignan responsabilidades de forma tal que solo se comuniquen con las clases que se encargan de facilitar el proceso de pertenencia a cada una de ellas.

```
class Organization(models.Model):
    name=models.CharField(max_length=150)
    country=models.CharField(max_length=100)
    org_level=models.CharField(max_length=500)
    def __unicode__(self):
        return self.name
```

Ilustración 25: Ejemplo de la clase Organization

➤ El Patrón Alta Cohesión se basa en asignar una responsabilidad de modo que la cohesión siga siendo alta. “La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase” (Buono, 2012).

El patrón propone el diseño de clases con responsabilidades moderadas en su área funcional y que colabore con las otras para llevar a cabo una tarea.

- Ejemplo: A las clases Reference y Publication se le asignan responsabilidades con el objetivo de que trabajen en la misma área de la aplicación y que no tengan mucha complejidad.

```
class Reference(models.Model):
    citing_article=models.ForeignKey(Document,related_name='citing_article')
    cited_article=models.ForeignKey(Document,related_name='cited_article')
    author=models.CharField(max_length=100)
    journal=models.CharField(max_length=100)
    title=models.CharField(max_length=200)
    publication_year=models.IntegerField(max_length=4)
    def __unicode__(self):
        return self.title
```

Ilustración 26: Ejemplo de la clase Reference

Los patrones Banda de los cuatro (**GoF**⁴¹ por sus siglas en inglés) describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Permiten crear grupos de objetos que ayudan a realizar tareas complejas. Facilitan el aprendizaje y la comunicación entre programadores y diseñadores. Estos patrones se clasifican en tres tipos: creacionales, estructurales y de comportamiento (Gamma, et al., 1995).

Los patrones **Estructurales** separan la interfaz de la implementación y se encargan de guiar la forma en que las clases y los objetos se combinan para formar nuevas estructuras más complejas y proporcionar nuevas funcionalidades (Gamma, et al., 1995).

Dentro de los patrones Estructurales se utilizó:

⁴¹ GoF, acrónimo de The Gang of Four.



➤ **Decorator:** Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. Es aplicado a la generación de vistas, la solución que ofrece dicho patrón es la de añadir funcionalidad adicional a las plantillas.

- Ejemplo: La utilización de este patrón se evidencia mediante el archivo *inicio_admin.html* que almacena el código HTML que es común para todas las vistas del sistema.

```
{% extends 'administrador/base_admin.html' %}
```

Ilustración 27: Ejemplo de utilización del patrón Decorador

Los patrones **De comportamiento** plantean la interacción y cooperación entre las clases, estudian las relaciones entre llamadas entre los diferentes objetos y normalmente están ligados con la dimensión temporal. (Gamma, et al., 1995)

Dentro de los patrones De comportamiento se utilizó:

➤ **Observer:** Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan automáticamente todos los objetos que dependen de él.

- Ejemplo: La utilización de este patrón se evidencia mediante las clases Document y Publication que muestran una relación de uno-a-muchos.

```
class Publication(models.Model):
    article=models.ForeignKey(Document)
    journal=models.ForeignKey(Journal)
    volume=models.IntegerField()
    issue=models.CharField(max_length=10)
    publication_date=models.DateField()
    publication_year=models.IntegerField(max_length=4)
    def __unicode__(self):
        return self.publication_year
```

Ilustración 28: Ejemplo de utilización del patrón Observer

2.6.4 Tarjetas CRC (Clase – Responsabilidad – Colaborador)

Las tarjetas CRC son confeccionadas durante la etapa de diseño de la metodología de desarrollo XP. De forma organizada cada tarjeta representa una clase, donde se describen las responsabilidades que tiene y las clases colaboradoras que se relacionan con la misma. A continuación se describe una de las tarjetas CRC de la clase Author_ship. Para ver más en los anexos.

Tabla 7: Mostrar de forma gráfica la relación entre autores

Clase: Author_ship	
Responsabilidad	Colaboración
Mostrar de forma gráfica la relación entre autores.	models



Conclusiones parciales del capítulo

En el presente capítulo se abordaron temas relacionados con el análisis y diseño de la propuesta de solución.

- Los requisitos funcionales y no funcionales de conjunto con las HU y la descripción de las mismas, ayudaron a definir las condiciones y capacidades que deben estar presentes en el sistema para satisfacer las necesidades del cliente.
- A partir de las HU descritas se definieron (4) iteraciones y la estimación del esfuerzo dedicado a la realización de cada una de ellas. El plan de entregas permitió delimitar el ciclo de desarrollo de la aplicación.
- El empleo de la arquitectura Cliente/Servidor, el estilo arquitectónico MTV (Modelo-Plantilla-Vista) y algunos de los patrones de diseño GRASP y GoF permitió organizar el diseño lógico de la solución propuesta.

Capítulo **3** *Implementación y Prueba*

3.1 Introducción

Para el desarrollo de la aplicación es de vital importancia la fase de implementación y prueba del sistema propuesto, para dar cumplimiento a los objetivos planteados. Para alcanzar dicho propósito en el capítulo se describen las tareas de ingeniería generadas por cada Historia de Usuario, las cuales han sido utilizadas como base para la implementación del software y se evalúa la calidad de la aplicación a través de las pruebas de software realizadas a la herramienta.

3.2 Etapa de Implementación

Una vez definidas las Historias de Usuario y concluido el diseño se llega a la etapa de codificación de la solución propuesta. Durante la codificación la programación en parejas tiene muchas ventajas, una de ellas es la constante revisión del código, debido a que existe una persona observando todo lo que el otro programa. De esta manera se crea el código para cada Historia de Usuario en dependencia de lo concebido en el plan de iteraciones y las tareas ingenieriles. A medida que concluya cada iteración, el código desarrollado se integrará con el resto, puesto que esta estrategia de integración continua, ayuda a evitar los problemas de compatibilidad de interfaces, y a descubrir a tiempo los errores.

3.2.1 Desarrollo de las iteraciones

Las tareas de programación o tareas de ingeniería son escritas por el equipo de desarrollo a partir de las Historias de Usuario elaboradas por el cliente, brindando un detalle más profundo para realizar una implementación de las mismas y estimando un tiempo más cercano a la realidad. Cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores. Son numeradas generalmente de forma consecutiva y siguen una a otra de manera continua. Seguidamente se muestra una de las tareas de ingeniería correspondientes a una muestra de las HU de prioridad alta. Se detallan más tareas en los anexos.

Tabla 8: Tarea #1 Búsqueda de datos bibliográficos

Tarea de Ingeniería	
No. Tarea: 46	No. HU: 46
Nombre Tarea: Mostrar de forma gráfica la relación entre autores	



Tipo de Tarea : Desarrollo	Puntos Estimados: 7
Fecha Inicio: 22-4-2015	Fecha Fin: 28-5-2015
Programador(es) responsable(s): Yeneisy Barroso e Ivelisse Montero	
Descripción: Inicia cuando el cliente decide observar los resultados de la búsqueda realizada y desea ver las relaciones del autor elegido junto con otros. Muestra un grafo no dirigido con las relaciones entre los autores, centrado en un autor en específico y los colaboradores.	

3.2.2 Estándares de codificación

Un estándar de codificación comprende todos los aspectos de la generación de código, de tal manera que sea práctico y entendible para todos los programadores. Por lo general, incluye pautas sobre cómo nombrar variables y constantes, dónde ubicar comentarios, entre otras. No detecta los errores existentes, más bien evita la ocurrencia de estos, lo que permite obtener un código de alta calidad.

❖ **Indentación**

En la mayoría de los lenguajes, se considera que indentar correctamente las instrucciones del programa es una buena práctica muy recomendable para hacer más legible el código. Sin embargo, en Python esto no es opcional, dependiendo de la posición en que una instrucción concreta esté colocada, el intérprete realizará una tarea u otra o incluso considerará que la aplicación no está correctamente construida y mostrará un error o ni siquiera la ejecutará. En el caso de Python, la indentación es obligatoria, ya que de ella, dependerá su estructura. La indentación se puede realizar añadiendo espacios o tabulaciones.

❖ **Variables y constantes**

Las variables se pueden mostrar en minúscula con las palabras separadas con guiones bajos, tantos como sea necesario para mejorar la legibilidad. Las constantes deben declararse con todas sus letras en mayúsculas y con guiones bajos separando palabras.

❖ **Comentarios, separadores, líneas, espacios en blanco y márgenes**

Ubicación de comentarios: Se recomienda comentar al inicio de cada clase o función de forma que se especifique el objetivo de la misma así como los parámetros que usa (declarar tipos de datos, y objetivo del parámetro, entre otras). Los comentarios en bloque están al mismo nivel que el código.

Cada línea de un comentario en bloque comienza con un # (numeral).

Líneas en blanco: Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.

Espacios en blanco: Se recomienda usar espacios en blanco entre operadores lógicos y aritméticos para lograr una mayor legibilidad del código. Ejemplo: `lista_doc = list (busqueda_documents)`.

❖ Clases y objetos

El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se conozca el propósito de los mismos. Los nombres de las clases deben comenzar con la primera letra en mayúscula, se debe utilizar la convención CapWords (palabras que comienzan con mayúscula). Ejemplo: `class Author_ship (models.Model)`.

El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula y estará en correspondencia al tipo de dato al que se refiere, en caso de que sea un nombre compuesto, la segunda palabra de igual forma comenzará con minúscula.

Para nombrar las funciones se debe utilizar la convención mixedCase (la primera palabra será con minúscula). Ejemplo: `def autenticar (request)`. Si son funciones que obtienen un dato se emplea el prefijo “get”.

3.2.3 Diagrama de paquetes

El diagrama de paquetes representa la estructura lógica de un sistema y es empleado para organizar el sistema en elementos más pequeños y permitir una mejor interpretación del mismo. Muestran además, cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias o relaciones entre éstas. *“Gráficamente, una dependencia se representa como una línea discontinua dirigida, que incluye a veces una etiqueta.”* (Ruiz, et al., 2011)

En la (Ilustración 29) se muestra el diagrama de paquetes correspondiente a la aplicación web Representación de Bibliografías Científicas, donde se utilizó el estereotipo `<<access>>` que representa el acceso de un paquete a otro.

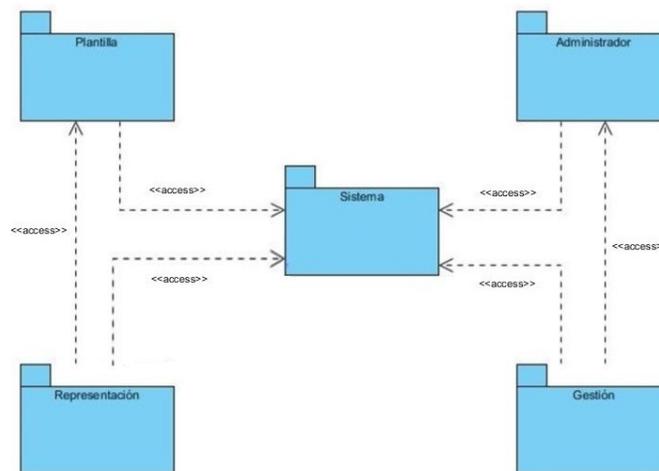


Ilustración 29: Diagrama de paquetes de la aplicación web RBC

Sistema: Este paquete contiene las clases y funcionalidades necesarias de la aplicación web.

Administrador: Este paquete contiene las funcionalidades necesarias que permite la autenticación de un usuario.

Gestión: Este paquete contiene las clases y funcionalidades necesarias que permiten la gestión de documentos, personas, organizaciones, entre otras.

Plantilla: Este paquete contiene las plantillas que permiten mostrar los resultados de la búsqueda, siendo estos la información obtenida y los gráficos.

Representación: Este paquete contiene los métodos necesarios que permiten la representación de gráficos.

3.2.4 Diagrama de clases persistentes

El diagrama de clases persistentes es la base para la creación del modelo físico de la base de datos. Estas clases, son aquellas que presentan durabilidad en el tiempo, más allá de la ejecución. A continuación la (Ilustración 30) muestra el diagrama de clases persistentes perteneciente a la aplicación web RBC.

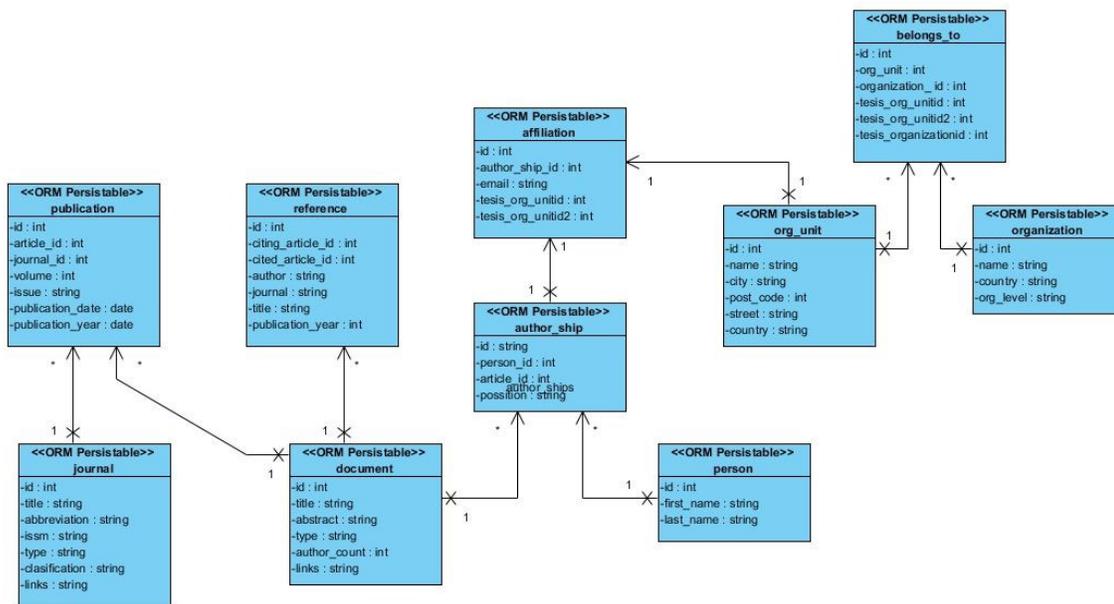


Ilustración 30: Diagrama de clases persistentes

3.2.5 Diseño de la base de datos

El modelado de datos permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades. Son un lenguaje gráfico para describir conceptos. Los elementos esenciales del diagrama son las entidades, los atributos y las relaciones entre las entidades.

- Las entidades son objetos de los que el sistema necesita guardar información.
- Los atributos son las características asociadas a una entidad. Estos pueden ser clasificados en obligatorios, opcionales, claves foráneas y claves primarias.
- Las relaciones, por su parte, muestran la forma en que dos entidades se asocian. Se representan mediante una línea que une a las dos entidades implicadas.

Para su realización se utilizó como guía el diagrama de clases persistentes visto anteriormente y además se agregaron las tablas: `auth_user_group`, `auth_group_permission` y `auth_user_user_permission` generadas a partir de las relaciones de mucho a mucho entre las clases: `group-user`, `group-permission` y `permission-django_content_type` respectivamente.

La (Ilustración 31) muestra el modelo físico de la base de datos.

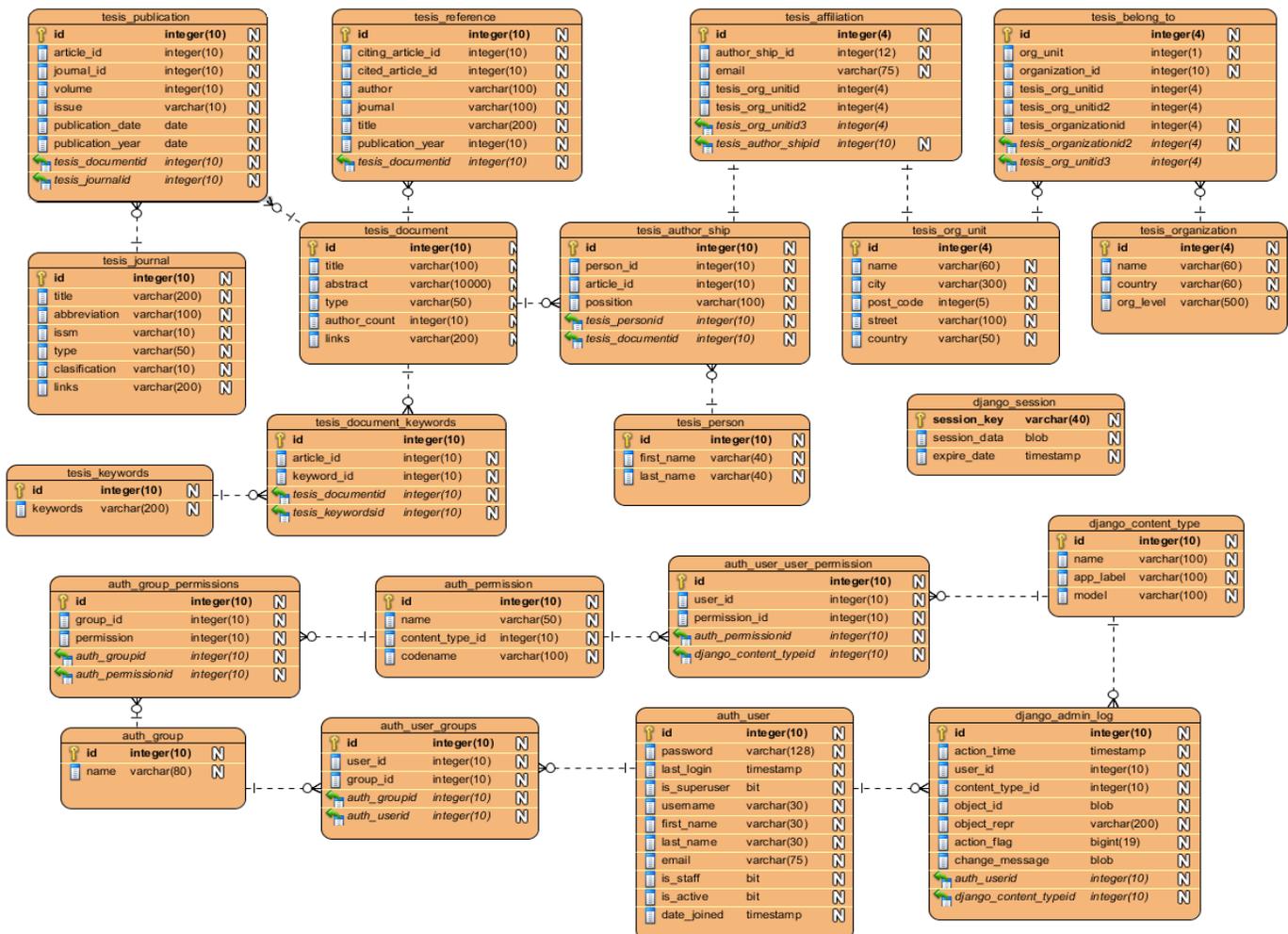


Ilustración 31: Modelo físico de la base de datos

3.3 Prototipos de interfaz de usuario

En la parte superior de la aplicación web RBC se observa que el usuario si lo desea puede volver a realizar una búsqueda, acceder a la administración o simplemente informarse sobre la herramienta. Las demás interfaces se muestran en los anexos.

La **(Ilustración 32)** representa la vista detallada de un documento seleccionado por el usuario. Se podrán observar los datos de ese documentos, además de las publicaciones del documento y un vínculo llamado (Ver Relación) que mostrará la relación del documento con otros en forma de grafo dirigido.

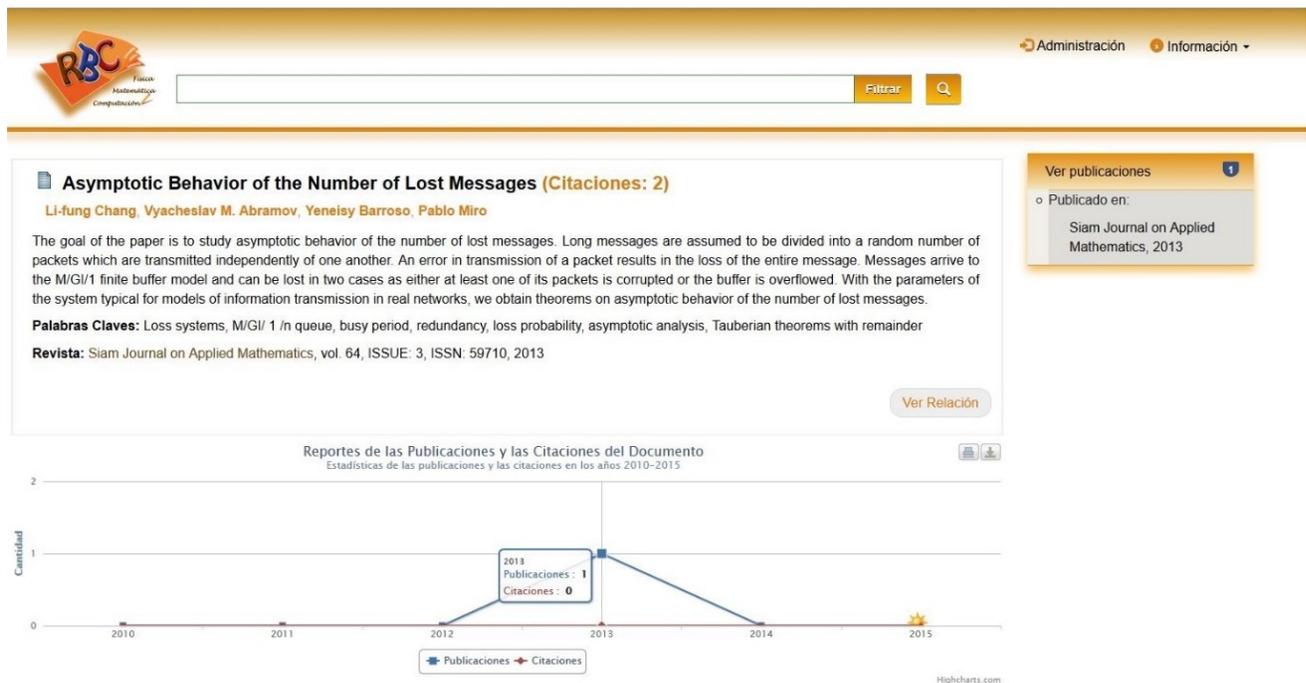


Ilustración 32: Vista detallada del documento seleccionado

La **(Ilustración 33)** muestra un grafo egocéntrico dirigido centrado en un documento seleccionado. Se puede observar el documento y los que se relacionan con él. Esta relación está dada a partir de las referencias (las citas realizadas por el documento específico hacia otros documentos y los documentos que lo han citado a él). En la parte superior derecha se muestra la leyenda de los aspectos del grafo y debajo las referencias enumeradas para guiar las relaciones del mismo. Si el documento seleccionado no tiene relación con otros documentos, el sistema mostrará el mensaje (No se puede mostrar la relación del documento, porque no tiene referencias).

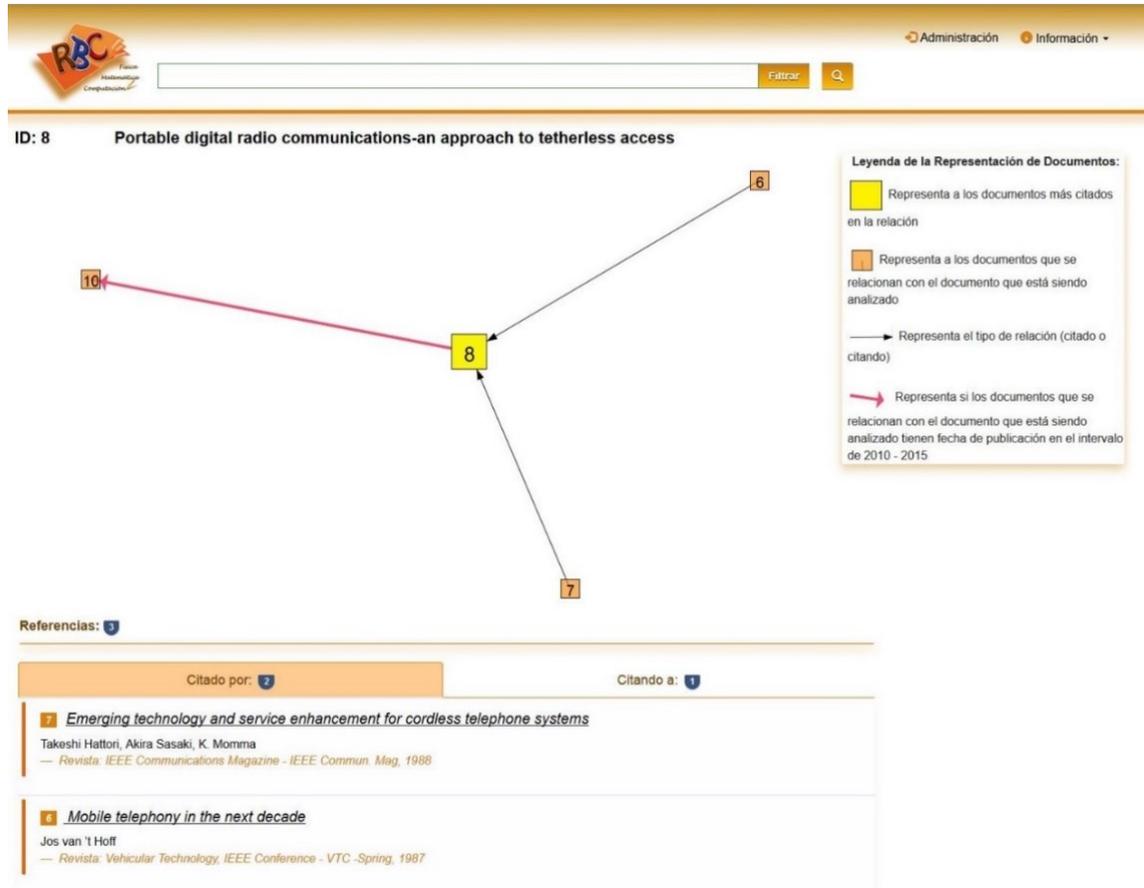


Ilustración 33: Grafo dirigido mostrando las relaciones entre documentos

3.4 Pruebas del Sistema

Las pruebas (testing) son una de las prácticas fundamentales de la metodología XP. El objetivo de esta fase es realizar la verificación y validación del sistema desarrollado, es decir, comprobar que el software realiza correctamente las tareas especificadas en los requisitos.

3.4.1 Métodos de pruebas

Los métodos de prueba definen la estrategia a seguir en función de la verificación y validación del sistema diseñados para descubrir fallos. Los métodos más significativos son las pruebas de caja blanca y las pruebas de caja negra.

3.4.1.1 Pruebas de Caja Blanca

“Es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba.” (Pressman, 2006)

Se encarga de comprobar los caminos lógicos del software a través de los casos de prueba que se ejerciten con conjuntos específicos de condiciones y/o bucles y sus límites, así como las estructuras



de datos. Se necesita conocimiento específico del código para analizar los resultados y son pruebas con acceso al código fuente (datos y lógica).

3.4.1.2 Pruebas de Caja Negra

“Denominada también prueba de comportamiento, se centran en los requisitos funcionales del software, tiene como objetivo verificar que la entrada se acepta correctamente y que ejerciten en su totalidad todos los requisitos funcionales de un sistema.” (Pressman, 2006)

Son pruebas funcionales sin acceso al código fuente de las aplicaciones, y se llevan a cabo sin tener conocimiento de la estructura y funcionamiento interno del sistema.

La prueba de caja negra intenta encontrar las siguientes categorías de errores (Pressman, 2006):

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externos.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

3.4.2 Estrategia de prueba seguida

La estrategia seguida para la realización de las pruebas a la aplicación web para la representación gráfica de datos bibliográficos contiene dos niveles: el primero lo constituyen las pruebas unitarias y el segundo las pruebas de aceptación. Las pruebas unitarias, son las encargadas de verificar el código y diseñada por los programadores, y las pruebas de aceptación (pruebas funcionales) son diseñadas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema, con el objetivo de verificar su correcto funcionamiento y que cumpla con los requisitos establecidos por el cliente.

3.4.2.1 Pruebas unitarias

Las pruebas unitarias son pruebas de caja blanca y fueron desarrolladas constantemente al concluir cada implementación, para comprobar el correcto funcionamiento de esta. Se utilizó la librería de Python (unit testing) y las pruebas fueron hechas dentro del archivo test.py, ya creado por el marco de trabajo Django. Las pruebas fueron realizadas de forma automática haciendo uso del sistema de prueba que posee Django y para ejecutar los test o iniciar el servidor de prueba se hace uso del comando “python manage.py test tesis”, pues en tesis es donde se encuentran todos los test y así probarlos llevaría menos tiempo.

A continuación la **(Ilustración 34)** muestra un ejemplo de prueba unitaria para el análisis de las clases Org_unit y Organization. Las demás pruebas se detallan en los anexos.

```

class Test_models(unittest.TestCase):
    def test_Org_unit(self):
        true_org=False
        try:
            org=Org_unit(name='unidad',city='la habana',post_code='10400',street='la franela',country='cuba')
            org.save()
            true_org=True
        except:
            pass
        self.assertTrue(true_org)

    def test_Organization(self):
        true_organization=False
        try:
            org=Organization(name='organo',country='cuba',org_level='4')
            org.save()
            org.country='editando pais'
            org.save()
            true_organization=True
        except:
            pass
        self.assertTrue(true_organization)
    
```

Terminal

```

+-----+
+ Ran 2 tests in 0.041s
+-----+
OK
Destroying test database for alias 'default'...
    
```

Ilustración 34: Resultado de las pruebas realizadas

3.4.2.2 Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra definidas por el cliente creadas a partir de las Historias de Usuario, en cada iteración del ciclo de desarrollo. Una Historia de Usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. El objetivo del cliente es verificar que el software esté listo y que puede ser usado por los usuarios finales para ejecutar aquellas funciones y tareas para las cuales fue construido.

A continuación se muestra el caso de prueba de aceptación correspondiente a la Historia de Usuario “Mostrar de forma gráfica la relación entre autores”. Los casos de prueba de aceptación definidos por el cliente para el resto de las Historias de Usuario se detallan en los anexos.

Tabla 9: Prueba de aceptación #19

Caso de Prueba de Aceptación	
Código: CP#19	HU#: 42 Mostrar de forma gráfica la relación entre autores
Responsable: Ivelisse Montero y Yeneisy Barroso	
Descripción: Prueba de funcionalidad para verificar que la aplicación muestra correctamente de forma gráfica la relación entre los autores.	
Condiciones de ejecución: El usuario debe hacer clic en el enlace Ver Relación.	

Entrada / Pasos de ejecución:

- Verificar que el enlace (Ver Relación) funcione correctamente.
- Verificar que el grafo muestre correctamente las relaciones entre los autores, centrándose en el autor seleccionado y sus colaboradores.

Resultado esperado: Se muestre en la interfaz gráfica el grafo no dirigido centrado en el autor seleccionado y los autores que se relacionen con él (colaboradores).

Evaluación de la Prueba: Prueba satisfactoria

3.4.3 Resultados de las pruebas

La (Ilustración 35) muestra los resultados obtenidos de las pruebas de aceptación a partir de los (20) casos de pruebas definidos por el cliente durante las (4) iteraciones de codificaciones planificadas. La primera iteración arrojó un total de 2 no conformidades (NC), la segunda 6 (NC), la tercera 5 (NC) y la cuarta 4 (NC) lo que equivale a un total de 17 no conformidades.

A las Historias de Usuario correspondientes a cada iteración se le realizaron 3 iteraciones de pruebas. Estas dificultades fueron solucionadas en un corto plazo, antes de pasar a la iteración posterior, cumpliéndose correctamente todos los requisitos solicitados por el cliente.

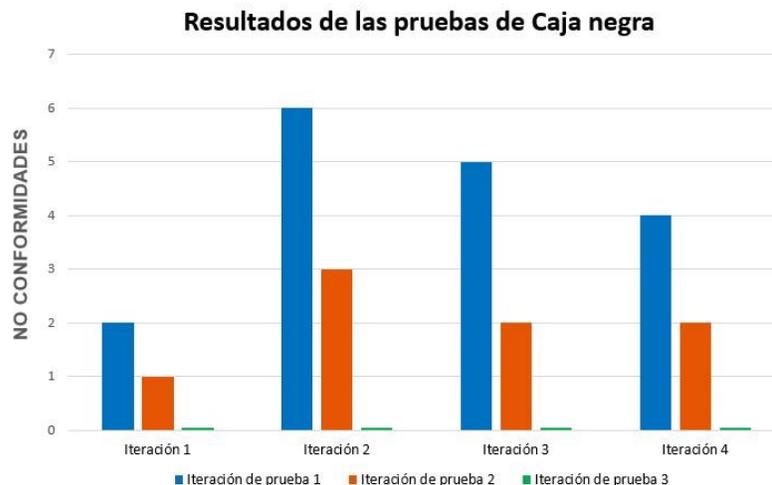


Ilustración 35: No conformidades detectadas durante las 4 iteraciones

3.4.4 Entornos de ejecución de las pruebas

Las pruebas de software realizadas se llevaron a cabo en diferentes entornos.

Sistema Operativo Windows 7 Ultimate con las siguientes características:

- Procesador: Intel Pentium 4 CPU 3.40 GHz.
- Memoria RAM: 1.00 GB.



Sistema Operativo Windows 8 Pro con las siguientes características:

- Procesador: AMD E2-1800 CPU 1.70 GHz.
- Memoria RAM: 4.00 GB.

Sistema Operativo Ubuntu 15.04 con las siguientes características:

- Procesador: Intel Core i3-4160 CPU 3.60 GHz.
- Memoria RAM: 4.00 GB.

Conclusiones parciales del capítulo

En el presente capítulo se mostraron los resultados de la implementación y pruebas realizadas a la aplicación web.

- Se definieron las tareas de ingeniería correspondientes a cada Historia de Usuario para obtener un nivel más detallado de la implementación de la aplicación.
- La utilización de los estándares de codificación permitieron la estructuración homogénea del código.
- Se realizaron pruebas unitarias a la implementación del código durante todo el desarrollo de la aplicación, lo que permitió comprobar en todo momento la validez del mismo.
- Se realizaron pruebas de aceptación que permitieron detectar (17) no conformidades durante las (4) iteraciones. Luego de ser corregidas en su totalidad, validaron su correcto funcionamiento en relación a los requisitos definidos por el cliente.



Conclusiones Generales

Una vez finalizada la fundamentación teórica que sustentó la presente investigación, definidas las características de la propuesta de solución y efectuado su desarrollo y validación, se obtuvieron resultados que permiten arribar a las siguientes conclusiones:

- ✓ El estudio de las principales tendencias relacionadas con la aplicación web demostró que las herramientas de búsqueda de información científica, tanto para el ámbito nacional como internacional, necesitan representar la información mediante técnicas de visualización de información, permitiendo explorar los datos y observar las relaciones entre ellos para mejorar la visualización.
- ✓ Se determinaron las herramientas, tecnologías y la metodología de desarrollo de software para el desarrollo de la aplicación web.
- ✓ En el desarrollo de la aplicación web Representación de Bibliografías Científicas se logró representar de forma gráfica los datos bibliográficos de publicaciones científicas en el área de las Ciencias Básicas y la Cibernética.
- ✓ Las pruebas realizadas arrojaron resultados satisfactorios, pues se identificaron un grupo de no conformidades que fueron corregidas, lo que posibilitó la verificación y validación de las funcionalidades del sistema.



Recomendaciones

Una vez concluido el desarrollo de la aplicación web Representación de Bibliografías Científicas y luego de haber cumplido los objetivos de la presente investigación, se recomienda:

- Para una mejor indexación de documentos, se debe utilizar un indexador web o motor de búsqueda, preferiblemente Lucene o Apache Solr debido al lenguaje utilizado en el desarrollo de la aplicación web.
- Emplear la técnica de nube de términos o nube de etiquetas para representar los criterios más buscados por los usuarios en la aplicación web.
- Extender el tamaño del grafo egocéntrico no dirigido centrado en el autor a más de un nivel de jerarquía, permitiendo conocer los autores que se relacionan con los colaboradores de un autor específico.



Referencias Bibliográficas

- Abellanas, Manuel, Aiello, Andrés y Silveira, Gregorio Hernández Peñalver and Rodrigo I. 2004.** Force-directed algorithms for drawing graphs with vertices that represent geographical regions. s.l. : Departamento de Matemática Aplicada, Facultad de Informática, Universidad Politécnica de Madrid, Spain. Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina., 02 de Diciembre de 2004.
- ACTIV. ACTIV.** Fundamentos de CSS3. [En línea] [Citado el: 08 de Noviembre de 2014.] <http://www.maestrosdelweb.com/editorial/css-3-mas-social-que-nunca/>.
- Alvaro, J. Henry. 2013.** HighCharts: Librería para creación de gráficos. [En línea] 1 de Marzo de 2013. [Citado el: 02 de Junio de 2015.] <https://enboliviacom.wordpress.com/2013/03/01/highcharts-libreria-para-creacion-de-graficos/>.
- Antón, Arturo Elias. 2009.** Python R2 . [En línea] 5 de Mayo de 2009. <https://pythonr2.wordpress.com/tag/ejemplo-cairo/>.
- Arrollo, Roberto Gamonal. 2013.** Historia y Comunicación Social. Madrid, España : s.n., Diciembre de 2013. Vol. 18, Esp, págs. 335-347.
- Astudillo, Marcello Visconti y Hernán. 2004.** *Fundamentos de Ingeniería de Software*. Departamento de Informática , Universidad Técnica Federico Santa María . 2004. pág. 48.
- Baeza-Yates, R., Castillo C., Marín y M., Rodríguez, A. 2005.** "Crawling a Country: Better Strategies than Breadth-First for Web Page Ordering.". 2005.
- Barry, Wellman y SD, Berkowitz. 1988.** *Structural analysis: from method and metaphor to theory and substance*. s.l. : Social structures: a network approach. Cambridge: Cambridge University Press, 1988. págs. 19-61.
- Bartomeo, Yanedi Abreu. 2012.** Biblioteca Digital: Un complemento al Sistema de Bibliotecas de las Universidades. La Habana, Cuba : s.n., 12 de Marzo de 2012.
- Beck, K. 2000.** Extreme Programming Explained. Embrace Change. [ed.] Addison Wesley. 2000. Traducido al español como: "Una explicación de la programación extrema. Aceptar el cambio".
- Begelman, G. 2006.** Automated Tag Clustering: Improving search and exploration in the tag space. Edinburgh, UK : s.n., 22-26 de Mayo de 2006. pág. 11.
- Bernd Klein. 2011.** Python Course. [En línea] 2011. <http://www.python-course.eu/numpy.php>.
- Bizagi. 2014.** BPMN. [En línea] 2014. <http://www.bizagi.com/docs/BPMNbyExampleSPA.pdf>.
- Borgatti, Steve y College, Boston. 2003.** Conceptos Básicos de Redes Sociales. Cancún : s.n., 14 de Febrero de 2003.



- Buono, Enrique. 2012.** Red Colaborativa Postgrado UCV. *Sitio de integración social y de aprendizaje.* [En línea] Julio de 2012. [Citado el: 14 de Abril de 2015.] http://kuainasi.ciens.ucv.ve/red_educativa/blogs/42?language_id=1.
- Cairo. 2014.** Cairo. [En línea] 20 de Noviembre de 2014. <http://cairographics.org/> .
- Candela, L. 2007.** D-Lib Magazine. *Setting the foundations of Digital Libraries: The DELOS Manifesto.* [En línea] marzo/abril de 2007. [Citado el: 4 de marzo de 2015.] <http://www.dlib.org/dlib/march07/castelli/03castelli.html>. ISSN 1082-9873.
- Carrasco, Juan Antonio y Miller, Eric J. 2006.** *Exploring the propensity to perform social activities: a social network approach.* s.l. : Transportation, 2006. págs. 463-480. Vol. 33. 5.
- Cervantes, Dr. Humberto. 2010.** Arquitectura de Software. Arquitectura. [En línea] Abril de 2010. [Citado el: 23 de Febrero de 2015.] <http://sg.com.mx/content/view/922>.
- Chacón, José Luis. 2005.** Introducción a la Teoría de Grafos. *Matemáticas Discreta.* 2005.
- Codina, L. 1996.** La investigación en sistemas de información. Santiago de Chile: CEPAL. : Comisión Económica para América Latina y el Caribe (CEPAL). Anuario estadístico de América Latina y el Caribe., 1996.
- Cortés, Oscar Hernando Guzmán. 2007.** Taxonomías de la visualización de información. [ed.] SISTEMAS & TELEMÁTICA. Cali, Colombia : Universidad Icesi, 2 de Octubre de 2007. págs. 87-125.
- Drake, José M. 2008.** *Proceso de desarrollo de aplicaciones software.* Santander : Santander, 2008.
- Escribano, Gerardo Fernández. 2002.** "Introducción a Extreme Programming". 9 de 12 de 2002. Free Download Manager. Paradigma Visual para UML. [En línea] [Citado el: 05 de Abril de 2015.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
- Gamma, Erich, et al. 1995.** *Design Patterns. Elements of Reusable Software.* [ed.] Addison-Wesley. USA : s.n., 1995.
- Giménez-Toledo, Elea y Rodríguez-Yunta, Luis. 2012.** ¿Por qué nadie habla ya de las bases de datos bibliográficas "clásicas"? *Anuario ThinkEPI.* 2012, Vol. v.6, págs. 130-134.
- . 2005. Lo que los usuarios piensan de las bases de datos bibliográficas y no se atreven a decir. IX Jornadas Españolas de Documentación Madrid, España : Primera versión de la Comunicación enviada a Infogestión, 2005.
- González, Atilio Bustos y Porcel, Antonio Fernández. 2007.** *Directrices para la creación de repositorios institucionales en universidades y organizaciones de educación superior.* s.l. : Asociación Columbus y Comisión Europea, 2007. págs. 5-27.



- Grosso, Andrés. 2011.** Prácticas de Software. *Experiencias sobre la Ingeniería y Management del Software*. [En línea] 21 de Marzo de 2011. <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
- Gutiérrez, C. 2008.** *Cómo funciona La Web*. Universidad de Chile. Santiago, Chile : Centro de Investigación de la Web, 2008.
- Holovaty, Adrian y Kaplan-Moss, Jacob. 2008.** El libro de Django. [En línea] 2008. <http://www.djangobook.com.757>.
- Hønsi, Torstein. 2011.** Highcharts product. [En línea] 08 de July de 2011. [Citado el: 02 de Junio de 2015.] <http://www.highcharts.com/products/highcharts>.
- Hu, Yifan. 2006.** Efficient, High-Quality Force-Directed Graph Drawing. s.l. : The Mathematica Journal. Wolfram Media, Inc., 2006. págs. 38-71.
- Igraph. 2012.** igraph. [En línea] 22 de Noviembre de 2012. <http://igraph.wikidot.com/>.
- . **2003.** python-igraph Manual. [En línea] The igraph core team, 2003. <http://igraph.org/python/doc/tutorial/tutorial.html>.
- . **2012.** The igraph library. [En línea] 2012. <http://cneurocv.s.rmki.kfki.hu/igraph/screenshots2.html>.
- Jesus, Joane De y Oliveras, Ámbar. 2010.** Visualización. 11 de Febrero de 2010. pág. 60.
- Kamada, T. y Kawai, S. 1989.** *An algorithm for drawing general undirected graphs*. s.l. : Information Processing, 1989. págs. 7-15. Vol. 31.
- . **1989.** *An algorithm for drawing general undirected graphs*. s.l. : Information Processing, 1989. págs. 7-15. Vol. 31.
- Keim, D. 2002.** Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*. January-March de 2002. Vol. 7, 1.
- Kowalski, Gerald. 2011.** *Information Retrieval Architecture and Algorithms*. USA : Springer Science+Business Media, 2011. DOI 10.1007/978-1-4419-7716-8.
- Kumar, Bharath. 2008.** *Online visualization of bibliography Using Visualization Techniques*. San Jose State University. 2008.
- 2004.** L. *UML y Patrones Introducción al análisis y diseño orientado a objetos*. CRAIG. *Lenguaje UML y Patrones Introducción al análisis y diseño orientado a objetos*. 2004. 8420534382.
- Lamarca, María Jesús. 2013.** Hipertexto: El nuevo concepto de documento en la cultura de la imagen. [En línea] 8 de diciembre de 2013. [Citado el: 20 de diciembre de 2014.] http://www.hipertexto.info/documentos/tipos_buscador.htm.
- Lic. Mercedes Vitturini, Lic. Karina Cenci, Mg. Silvia Castro. 2000.** Visualización de grandes volúmenes de datos. Bahía Blanca : Departamento de Ciencias de la Computación. Universidad Nacional de Sur, 2000.



- Maldonado, Ángeles. 2012.** Características de las bases de datos bibliográficas. [En línea] 20 de Enero de 2012.
- Marcum, Deanne. 2005.** La Biblioteca Digital: Requisitos. s.l. : Boletín de la Asociación Andaluza de Bibliotecarios, 2005. págs. 57-58. 20.
- Martig, Sergio R. y Castro, Silvia M. 2000.** *Visualización de Grafos*. Dpto. de Ciencias de la Computación , Universidad Nacional del Sur - Bahía Blanca . 2000. págs. 154-157.
- Méndez, Francisco J. Martínez. 2002.** *Propuesta y desarrollo de un modelo para la evaluación de la recuperación de información en internet*. 2002.
- Molina, Jose Luis Gutierrez. 2005.** *El estudio de las redes personales: contribuciones, métodos y perspectivas*. s.l. : EMPIRIA.Revita de Metodología en Ciencias Sociales, 2005. págs. 71-105. 10.
- Molina, Maria Pinto. 2011.** Organización, Filtración y Representación de la Información. [En línea] 13 de Abril de 2011. [Citado el: 23 de Enero de 2015.] http://www.mariapinto.es-comsor_con_elect.htm.
- Mozilla Developer Network and individual contributors. 2015.** Mozilla Foundation. Mozilla. [En línea] 2015. [Citado el: 08 de Noviembre de 2014.] <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>.
- Numpy. 2013.** Numpy developers. [En línea] 2013. <http://www.numpy.org/>.
- oa.usun, [ed.]. 2012.** *Visualización de la información: la base del conocimiento enlazando ideas de forma gráfica*. s.l. : Ibermatica RTD, 21 de Febrero de 2012.
- OCLC. 2011.** Perceptions of libraries,2010. *Context and community*. [En línea] 2011. [Citado el: 4 de marzo de 2015.] <http://www.oclc.org/reports/2010perceptions.html>.
- Pérez, Maidelyn Díaz. 2009.** Características de los sistemas de información que permiten la gestión oportuna de la información y el conocimiento institucional. [En línea] ISSN 1024-9435, 25 de Octubre de 2009. http://scielo.sld.cu/scielo.php?pid=S1024-94352009001100006&script=sci_arttext. 5.
- Pierri, Nicolás. 2005.** *Visualización de información*. Departamento de Investigación , Facultad de Ingeniería y Tecnología Informática Carrera de Licenciatura en Sistemas de Información. s.l. : UNIVERSIDAD DE BELGRANO, 2005. pág. 32, Tesis. 181.
- Polanco, X. y Zartl, A. 2002.** Information visualization. EICSTES Project. Deliverable 1.4 State of the art part c: WP9. [En línea] 2002. [Citado el: 10 de Marzo de 2015.] http://eictes.inist.fr/public/D1.4_Visualization_WP9.pdf.
- PostgreSQL. 2013.** The world's most advanced open source database. [En línea] 2013. www.postgresql.org/.
- Pressman, Roger S. 2006.** *Ingeniería del Software. Un enfoque Práctico*. Madrid : McGraw-Hill : Sexta Edición., 2006. ISBN: 970-10-5473-3.



Pullannagari, Deepthi. 2008. *Bibliography Data Mining and Data Visualization*. The Faculty of the Department of Computer Science, San Jose State University. s.l. : Master's Theses and Graduate Research at SJSU ScholarWorks, 2008. pág. 67.

—. 2008. *Bibliography Data Mining and Data Visualization*. San Jose State University. 2008.

PyCharm. 2015. JetBrains PyCharm Blog. [En línea] WordPress, 2015. <http://www.jetbrains.com/pycharm>.

Quesada, Benjamín Vargas. 2005. *Visualización y Análisis de grandes dominios científicos mediante redes Pathfinder(PFNET)*. Departamento de Biblioteconomía y Documentación, Departamento de Biblioteconomía y Documentación. Granada : Departamento de Biblioteconomía y Documentación, 2005. pág. 382, Tesis Doctoral.

Rodríguez-Yunta, Luis. 2001. *Bases de datos documentales: estructura y uso*. Madrid : CINDOC-CSIC, 2001.

Roldan, Diego. 2009. Diagrama de Componentes. [En línea] 26 de 2 de 2009. <http://exposicionuml40099.blogspot.com/>.

Roldán, P K. Suárez. 2003. *Animación y Visualización de Fenómenos Naturales*. Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas Puebla. 2003. Tesis Licenciatura. Ingeniería en Sistemas Computacionales.

Ruiz, Francisco y López, Patricia. 2011. *INGENIERÍA DEL SOFTWARE I*. s.l. : Universidad Cantabria. Facultad de Ciencias, 2011.

Sanchez, M. Mendoza. 2004. Informatizate. Metodologías De Desarrollo De Software. . [En línea] 7 de Junio de 2004. [Citado el: 06 de Noviembre de 2014.]

Silveira, Andrés Aiello y Ignacio, Rodrigo. 2004. *Trazado de grafos mediante métodos dirigidos por fuerzas: revisión del estado del arte y presentación de algoritmos para grafos donde los vértices son regiones geográficas*. Departamento de Computación , Facultad de Ciencias Exactas y Naturales. s.l. : Universidad de Buenos Aires, 2004. Tesis de Licenciatura en Ciencias de la Computación.

—. 2004. *Trazado de grafos mediante métodos dirigidos por fuerzas: revisión del estado del arte y presentación de algoritmos para grafos donde los vértices son regiones geográficas*. Departamento de Computación , Facultad de Ciencias Exactas y Naturales. s.l. : Universidad de Buenos Aires, 2004. Tesis de Licenciatura en Ciencias de la Computación.

Solana, V.H. 2006. [ed.] iRinformation research. *Metodologías para el desarrollo de interfaces visuales de recuperación de información: análisis y comparación*. Granada, España : Grupo SCImago, Universidad de Granada. Facultad de Biblioteconomía y Documentación, Abril de 2006. Vol. 11, 3.



Stoughton N. 2005. *"Update on Standards."*. 2005.

Tedeschi, Nicolás. 2010. ¿Qué es un Patrón de Diseño? [En línea] Analista de Sistemas de la Facultad de Ingeniería del Uruguay., 2010. [Citado el: 03 de Marzo de 2015.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

Tendero, D. Juan Pedro García. 2011. *Ingeniero Técnico en Topografía*. Máster en Geotecnologías Cartográficas en Ingeniería y Arquitectura , ESCUELA POLITÉCNICA SUPERIOR DE ÁVILA. Universidad de Salamanca. 2011. pág. 102, Maestría.

Universidad de Salamanca. 2015. Bibliotecas. [En línea] Universidad de Salamanca, 2015. http://bibliotecabiologia.usal.es/tutoriales/catalogos-repositorios-bibliosvirtuales/repositorios_digitales.html.

University of Birmingham. 2012. Computer Science SupportWeb. *Class KamadaGraphLayout RePast v. 2.0*. [En línea] 2012. <https://supportweb.cs.bham.ac.uk/documentation/java/repast/api/uchicago/src/sim/gui/KamadaGraphLayout.html>.

Vitaly Friedman. 2007. Data Visualization: Modern Approaches. *Smashing Magazine*. [En línea] Vitaly Friedman and Sven Lennartz, 2 de Agosto de 2007. <http://www.smashingmagazine.com/2007/08/02/data-visualization-modern-approaches/>.

Zafra, Salud María Jiménez. 2013. *Herramienta basada en la computación con palabras para la caracterización de usuarios en redes sociales*. Departamento de Informática , Universidad de Jaén. Escuela Politécnica Superior (Jaén). 2013. pág. 164, Tesis.