UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

CARRERA DE INGENIERÍA EN SISTEMAS

TEMA:

"DESARROLLO E IMPLEMENTACIÓN DE UN MOTOR DE BÚSQUEDA PARA LA RECUPERACIÓN DE INFORMACIÓN DE INTERNET, DESDE EL PORTAL DE LA UNIVERSIDAD NACIONAL DE LOJA"

Tesis previa a la obtención del Título de Ingeniero en Sistemas.

AUTORAS:

JOHANNA ALEXANDRA ABAD AYAVACA
CESILIA ABAD CONDE

DIRECTOR:
ING. GERMÁN PATRICIO VILLAMARÍN CORONEL

LOJA-ECUADOR
2010

CERTIFICACIÓN

Ing. Germán Patricio Villamarín Coronel

CATEDRÁTICO DE LA CARRERA DE INGENIERIA EN SITEMAS Y DIRECTOR DE TESIS

CERTIFICO:

Haber revisado el trabajo de tesis previo al grado de Ingeniería en Sistemas, presentado por las egresadas Cesilia Abad Conde y Johanna Alexandra Abad Ayavaca; bajo el título "Desarrollo e Implementación de un Motor de Búsqueda para la Recuperación de Información de Internet, desde el Portal de la Universidad Nacional de Loja", la misma que reúne los requisitos que exige el reglamento de la universidad, por tal razón autorizó su presentación.

Loja, Julio del 2010

_____

Ing. Germán Patricio Villamarín Coronel

DIRECTOR DE TESIS

AUTORIA

Las ideas y contenidos expuestos en el presente trabajo de Investigación, son de exclusiva responsabilidad de su autor.

…….. ……

JOHANNA ALEXANDRA ABAD AYAVACA ***CESILIA ABAD CONDE***

AGRADECIMIENTO

Nuestro agradecimiento a la Universidad Nacional de Loja, a la Carrera de Ingeniería en Sistemas, a través de sus Profesores, Directores, Compañeros; que nos brindaron sus conocimientos y apoyo incondicional en la culminación de nuestros estudios y en la realización de este proyecto.

Nuestro reconocimiento al Ingeniero Ing. Germán Patricio Villamarín Coronel, por su gestión al haber dirigido nuestro trabajo y apoyarnos incondicionalmente en la investigación y desarrollo del mismo.

Las Autoras.

DEDICATORIA

Este trabajo esta dedicado a los muchos años de sacrificio y dedicación de mis padres, y de manera especial a mi esposo e hijos que se convirtieron en el apoyo incondicional para cumplir este propósito, también se lo dedico a toda mi familia que me incentivaron a seguir adelante.

JOHANNA ALEXANDRA ABAD AYAVACA

Dedico este trabajo de manera especial y con mucha gratitud a mis padres Lauro e Imelda, cuyo respaldo y sacrificio me sirvieron para lograr y cumplir una de mis metas de ser una profesional, a mis hermanos con quienes comparto día a día y a todos quienes de alguna u otra forma colaboraron para la realización del mismo.

CESILIA ABAD CONDE

ÍNDICE DE TEMAS

CONTENIDOS
PÁGINA

1. RESUMEN

La presente investigación con el tema: “DISEÑO E IMPLEMENTACIÓN DE MOTOR DE BÚSQUEDA PARA LA RECUPERACIÓN DE INFORMACIÓN DE INTERNET, DESDE EL PORTAL DE LA UNIVERSIDAD NACIONAL DE LOJA”, está dirigida a solucionar la difícil tarea de buscar información en Internet y con el propósito siempre con contribuir al avance tecnológico de nuestra querida universidad.

El diseño de toda la aplicación está diseñada mediante el paradigma orientado a Objetos, por lo que fué necesario realizar los diagramas de Use Case, Diagrama de Clases, Prototipo de Pantallas, Diagramas de Robustez y Diagramas de Secuencia por cada Use Case, para ello se utilizó la herramienta de modelado UML, Enterprice Architect.

Cabe indicar que el diseño de nuestra aplicación está conformado de dos entornos de trabajo: el primero Web denominado Buscador y un segundo de Escritorio, programa RastradorUNL.

Como herramienta de programación se utilizó NetBeans 6.8, que es un IDE disponible de forma gratuita que ayuda a la facilidad de la programación en el lenguaje Java.

En cuanto al servidor Web, escogimos el servidor con el que trabaja la Universidad Nacional de Loja que es Jboss 5.1.0, pues es donde se alojara la aplicación.

Cabe destacar que esta aplicación está debidamente documentada con los manuales de programador y de usuario para una mejor comprensión, tanto en forma como en diseño.

SUMARY

The present investigation with the subject “DESIGN AND IMPLEMENTATION OF MOTOR SEARCH FOR the INFORMATION RETRIEVAL OF Internet, FROM the VESTIBULE OF the NATIONAL UNIVERSITY OF LOJA”, is directed to solve the difficult task always of looking for information in Internet and with the intention with contributing to the technological advance of our dear university.

The design of all the application is designed by means of the paradigm OO, reason why it was necessary to realise the diagrams of Uses CASE, Diagram of Classes, Prototype of Screens, Diagrams of Robustness and Diagrams of Sequence by each Use CASE, for it was used the tool of modeled UML, Enterprice Architect.

It is possible to indicate that the design of our application is conformed of two surroundings of work: the first denominated Web Seeking and a second, programs RastradorUNL.

As programming tool were used NetBeans 6,8, that is an IDE available free that helps to the facility of the programming in the Java language. As far as the Web server, we chose the servant with whom works the National University of Loja that is Jboss 5.1.0, because it is where the application lodged.

It is possible to emphasize that this application properly is documented with the manuals of programmer and user for one better understanding, as much in form as in design.

2.- INTRODUCCIÓN

La World Wide Web surgió en 1989 como una forma de distribuir información. El rápido crecimiento de la misma, sobre todo a partir de 1993, y su propia naturaleza no jerárquica creó la necesidad de herramientas que ordenaran la información disponible para posibilitar su fácil acceso. Se produce la generación de Motores de Búsqueda o buscadores, la cual se caracteriza por catalogar los documentos en base a la información interna de los mismos, utilizando técnicas provenientes de Recuperación de Información (Information Retrieval) tradicional.

En vista del avance tecnológico, surge la necesidad de implementar un Buscador para la recuperación de información de la Web, desde el Portal de la Universidad.

Además con el presente tema, pretendemos que el estudiante, docente o cualquier usuario que ingrese al Portal acceda al buscador, permitiéndole obtener de manera rápida y oportuna los resultados esperados.

La tendencia educativa actual hace de las tecnologías de la información y la comunicación, una herramienta fundamental para recuperar la información de Internet.

Nuestra propuesta informática es un aporte tecnológico para aquellas personas que hagan uso del Buscador, es por esta razón que el desarrollo de el Motor de Búsqueda requiere de conocimientos muy amplios que serán adquiridos en el transcurso del proceso de su desarrollo, así como aquellos conocimientos logrados en las aulas universitarias, que se basan en el ciclo de vida de un sistema informático, con la metodología Orientada a Objetos y además obtendremos experiencia en el campo profesional y la obtención del título universitario.

3.- METODOLOGÍA

Para hacer efectiva la realización del presente proyecto se recurrió a la aplicación de diferentes técnicas como la observación directa y análisis de los diferentes tipos de buscadores web, método investigativo y analizar una variada bibliografía que nos permitió obtener y ordenar la información de una manera adecuada, con el fin de cumplir con los objetivos planteados en la presente investigación.

El Método Descriptivo, mediante el cual se realizó el análisis de los resultados de las encuestas para su respectiva tabulación El análisis, la síntesis, la inducción, la entrevista, la analogía de acuerdo con las necesidades investigativas.

En el diseño del sistema se utilizó procesos de desarrollo de software orientados a objetos. Para el desarrollo del Buscador, se utilizó el proceso de desarrollo de software ICONIX el cual consta de las siguientes fases:

- Requerimientos.
- Modelo Conceptual.
- Prototipos.
- Diagramas de Casos de Uso.
- Diagrama de Robustez.
- Diagrama de Secuencia.
- Diagrama de Clases.
- Código.

La metodología de desarrollo que se implementó durante la ejecución del proyecto es ICONIX, que es utilizado en todos los proyectos intermedios y pequeños, la cual está basada en el modelado de casos de uso con UML, con diagramas que permitieron representar gráficamente la funcionalidad del buscador y las iteraciones con el usuario, que facilitó la programación requerida

ICONIX usa UML el cual genera un sistema mínimo de diagramas y algunas técnicas valiosas que llevaron los casos de uso al código en forma rápida y eficiente.

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. Se ha convertido en la notación estándar para organizar, definir y visualizar los elementos que intervienen en la arquitectura de un sistema.

Los requerimientos los obtuvimos en base al análisis que se realizó sobre los servicios que presta el portal de la Universidad hacia el usuario, y en base de los problemas que se presenta al momento de obtener información desde el portal de la Universidad, a nivel interno y de la Web, de esta información construimos el modelo conceptual que es el dominio del sistema, luego diseñamos los prototipos que son la interfaz entre el usuario y el sistema, de esta manera tuvimos una visión más amplia de los procesos, donde los usuarios interactúan con el buscador.

Posteriormente se realizó el diseño de la aplicación, la cual incluyen los diagramas de modelado, base de datos, casos de uso, robustez, secuencia, paquetes, y la arquitectura de un motor de búsqueda, es decir, el RastreadorUNL que recorre la Web buscando recursos de información y sus respectivas URLs para incorporarlas a una base de datos, en donde por medio de la Interfaz del Buscador UNL se recuperara la información de la base datos. Estos diagramas nos permitieron codificar el modelado lógico en la construcción de la interfaz de usuario, clases, métodos, eventos, conexiones y controles.

Se efectuaron dos tipos de pruebas: Personal Administrativo y Estudiantes, la primera para que ejecute el programa del motor de búsqueda denominado RastreadorUNL y analice el funcionamiento general de la aplicación y la segunda para medir tiempos de respuesta y valorando el sistema de acuerdo

al requerimiento del usuario sea la búsqueda mediante palabra clave, búsqueda avanzada y búsqueda por categorías.

REVISIÓN DE LA LITERATURA

4.- REVISION DE LA LITERATURA

4.1. ESTRUCTURA ACADÉMICA DE LA UNIVERSIDAD NACIONAL DE LOJA

El Sistema Modular de enseñanza-aprendizaje por objetos de transformación es un modelo pedagógico que permite problematizar el conocimiento en un proceso de investigación activa en el que intervienen profesores y alumnos para generar, recrear y aplicar ese mismo conocimiento.

Las características del sistema modular son diferentes al modelo por asignaturas que se ha venido practicando en forma tradicional: El estudiante es el sujeto del aprendizaje, el conocimiento no es algo acabado, la voz del profesor es sólo voz orientadora en la recreación del conocimiento. A las asignaturas les ha reemplazado el módulo que se orienta al tratamiento de un problema real en forma interdisciplinaria, el alumno descubre y constituye su identidad personal, la inteligencia y la reflexión son desarrolladas prioritariamente sin soslayar las otras facultades mentales como la memoria, sobrevalorada en el modelo por asignaturas.

Este modelo pedagógico innovador gira alrededor de un módulo que es una especie de unidad dialéctica autónoma estructurada interdisciplinariamente para resolver un problema de la realidad-objeto de transformación aprovechando bibliografía pertinente, la investigación participativa que permite unir la teoría con la práctica, la reflexión con la acción, la ciencia y la técnica con la ideología, y la universidad con la sociedad o ejerciendo así la docencia, la investigación y la extensión, obligación prioritaria de un centro de estudios contemporáneos.

4.1.1. Elementos Académicos Del Sistema Modular

El Sistema Modular por Objetos de Transformación vigente en la Universidad Nacional de Loja cuenta con elementos académicos estructurales.

Entre los elementos académicos tenemos el objeto de transformación, la matriz problemática, la matriz temática y el módulo.

Objetos de Transformación

Es un problema extraído de la realidad, social o natural, susceptible de ser investigado y solucionado. Al momento los objetos de transformación son tomados del perfil y de las prácticas profesionales que orientan la labor de las carreras universitarias facilitando así: la integración interdisciplinaria de varias áreas del conocimiento; la relación práctica-teoría-práctica; la vinculación docencia-investigación-extensión; y la interacción universidad-sociedad.

La Matriz Problemática

Es un proceso de ordenación y ascensión de los problemas inmersos en el objeto de transformación, en base del perfil y prácticas profesionales, que sirve de orientador en el escogitamiento de los problemas a tratarse.

La Matriz Temática

Es un proceso de abstracción y ascensión de conceptos y referentes teóricos e instrumentales que, partiendo de la explicación y manejo de las problemáticas, en una forma muy general, tiene por objeto elevarse hasta una explicación más compleja, más estructurada y científica en ánimo a cumplir con las destrezas, aptitudes, valores y conocimientos que demandan las prácticas de cada profesión.

El Módulo

Es un conjunto de elementos técnicos y prácticos que definen y orientan el proceso que se efectúa interdisciplinariamente en torno de un problema de la amplitud del objeto de transformación.

4.1.2. Sitio web de la UNL

Figura 4.1.-Ventana del Sitio Web de la Universidad Nacional de Loja.

En la actualidad la Universidad Nacional de Loja, cuenta con un Sitio Web cuya dirección Web es http:\www.unl.edu.ec, el mismo que tienen informado y permiten el acceso a la comunidad universitaria y todos quien visitan el portal, la oferta universitaria, noticias eventos y servicios online.

Conocedores del avance científico técnico de la Universidad Nacional de Loja con el fin de mejorar el proceso de enseñaza-aprendizaje se pretende con nuestro proyecto que en el Portal de la Universidad Nacional se pueda recuperar información de Internet mediante el desarrollo de un Motor de Búsqueda.

Mapa del Sitio - Universidad Nacional de Loja

- Inicio
- Áreas
  - EDUCATIVA

    - JURÍDICA
    - AGROPECUARIA
    - ENERGÍA
    - SALUD
- Asociaciones
    - APUL
    - AGEUL
    - FEUE
- Plan Gedes
- Autoridades
    - JUNTA UNIVERSITARIA
    - REPRESENTANTES
    - CAAS
    - DIRECTORES
- Estructura
- Mapa del Sitio

UNIVERSIDAD

- Quiénes Somos
    - Mensaje del Rector
- Misión y Visión
- Historia
- Calendario Académico
- Directorio UNL
- Egresados UNL
- Transparencia Institucional
    - Reglamentos y Estatutos
    - Procesos de Evaluación
    - Información Financiera
    - Estadísticas
- Bienestar Universitario

SERVICIOS EN LINEA

- Radio Universitaria
- Sistema Bibliotecario
    - Acerca de
    - Biblioteca Virtual
- Libro de Visitas
- Correo Electrónico
- Cursos Regionalización
- Modalidad a Distancia
- Centro de Recursos Idrisi

- SGA Estudiantes
- SGA Docentes

4.2. ASPECTOS GENERALES DE UN BUSCADOR.

4.2.1. Introducción.

El origen de los buscadores se remonta a abril de 1994, año en el que una pareja de universitarios norteamericanos (David Filo y Jerry Yang) decidieron crear una página web en la que se ofreciera un directorio de páginas interesantes clasificadas por temas, pensando siempre en las necesidades de información que podrían tener sus compañeros de estudios. Había nacido ***Yahoo!***. El éxito de esta página fué tan grande que una empresa decidió comprarla y convertirla en el portal que hoy conocemos. Además del buscador, hoy *Yahoo!* ofrece muchos más servicios

La manera más rápidas y moderna de buscar información, es por medio de la Internet. Hoy en día, existen millones de páginas en todo el mundo, las cuales contienen la más variada información posible. Es por lo mismo, que con la modernidad, la globalización y la tecnología, la búsqueda de información, dejó se de estar limitada sólo a las bibliotecas, para ahora llevarse a cabo, en los centenares de sitios que existen en Internet. Cuya cantidad se van incrementando a una gran velocidad, todos los años.

Según el portal *SearchEngineWath.com* se realiza unos 213 millones de búsquedas al día, con un total de 6.400 millones en Marzo de 2006 en EEUU. Por este motivo, el mundo de los buscadores mueve grandes cantidades de dinero (el 99% del cual es publicidad, según *El País*).

"Aunque es sumamente difícil medir el tamaño de la Web, se estima que hoy en día unos 1.000.000.000 usuarios utilizan la Web, y que esta contiene del orden de 4.000.000.000 documentos, un volumen equivalente a entre catorce y veintiocho millones de libros" [1]

Así, podemos admitir que **una estructuración de esta información es necesaria**. ¿Se imaginan el tiempo que tardaría una persona para encontrar

[1] Pablo Castells, *La Web Semántica*

todos los libros que contengan una palabra determinada en una pequeña biblioteca de cien libros? “Ahora se puede buscar a lo equivalente a 70 millas de altura en papel en menos de un segundo. Creemos que es fantástico” [2]

Los buscadores en Internet, son programas dentro de un sitio o página web, los cuales, al ingresar palabras claves, operan dentro de la base de datos del mismo buscador y recopilan todas las páginas posibles, que contengan información relacionada con la que se busca.

Por ende, en los buscadores, sólo se necesita ingresar la palabra clave o el concepto que se desea preguntar y el programa del buscador, entregará una lista de páginas que contienen aquella información.

Existen básicamente dos tipos de buscadores en Internet. Están los buscadores de Internet tipo directorio, y los que operan mediante robots o arañas. Los primeros, los buscadores en Internet tipo directorios, funcionan igual que cualquier directorio existente. Como por ejemplo, las páginas blancas o amarillas que pueden llegar a existir en los distintos países. Estos directorios clasifican y orden la información, según categorías preestablecidas. Dentro de los buscadores en Internet tipo directorios, los más famosos son Yahoo (el primero en gran escala de su tipo), y Dmoz o el open directory proyect, directorio cuya particularidad es que opera mediante editores voluntarios de todo el mundo.

Ahora, si hablamos de los buscadores en Internet que operan mediante robots, estamos hablando de sitios como Google, una de las páginas más visitadas y exitosas en la búsqueda de información. Es en estos tipos de sitios, en que las palabras claves, juegan un papel primordial, mayor que en los otros buscadores de Internet. Ya que por medio de estas palabras, es que el buscador va, valga la redundancia, buscando las páginas o documentos que contienen estas palabras o títulos, y las ordenan según su preponderancia o

[2] Lawrence Page, cofundador de Google

relevancia, en comparación a la palabra clave. Otros buscadores tipo robot conocidos, son Yahoo (además de poseer un directorio), y Ask Jeeves.

Otra denominación común para los buscadores que operan a través de robots es la de motores de búsqueda, que es lo mismo. Los robots o arañas son programas que escudriñan la web siguiendo los links o enlaces que van encontrando en las diferentes páginas. Estas arañas no descansan nunca, descubriendo cada vez nuevas páginas en la red. Unos de los robots más conocidos son el Googlebot, de Google, el Slurp, de Yahoo, y el MSN bot, de MSN search.

Ahora, también existen aquellos buscadores en Internet, que satisfacen zonas geográficas específicas. Los hay de tipo provincial, de una ciudad, de un país y aquellos internacionales, ya que buscan en páginas de distintos países. Incluso los grandes buscadores ya ofrecen la opción de búsqueda local, como en el caso de Google, que permite buscar dentro de las páginas de un determinado país.

Como se puede ver, es cosa de definir que es lo que deseamos buscar, e ingresarlo como palabra o frase clave en nuestro motor de búsqueda favorito para obtener la información que tanto deseamos.

4.2.2. Definiciones de Buscador:

- Un motor de búsqueda es un sistema informático que indexa archivos almacenados en servidores web gracias a su «spider» (o Web crawler).
- Servicio web que dispone de una base de datos que permite realizar búsquedas específicas en los contenidos de las páginas web publicadas en Internet, en función de los términos introducidos en el buscador por el usuario.
- Página en Internet que permite buscar información a través de ella, bien sea tecleando nosotros mismos una serie de palabras clave, o bien empleando el sistema de menús que la página incorpora. Cada día ofrecen más servicios, entre los que se incluyen noticias, chats, etc.

- Herramienta que permite ubicar contenidos en la Red, buscando en forma booleana a través de palabras clave. Se organizan en buscadores por palabra o índices (como Lycos o Infoseek) y buscadores temáticos o Directories (como Yahoo!).
- Es un programa, ubicado en un sitio de Internet, que recibe un pedido de búsqueda, lo compara con las entradas de su base de datos y devuelve el resultado. Algunos de los más conocidos: Yahoo, Altavista, Lycos, Infoseek.
- Servicio WWW que permite al usuario acceder a información sobre un tema determinado contenida en un servidor de información.
- Sitio de Internet que contiene una amplia base de datos sobre las páginas que se encuentran en la red.
- Aplicación que poseen algunos servidores y que permite la búsqueda de información en Usenet o WWW. Pueden ser "motores de búsqueda", como los llamados robots o arañas, que indexan la información recibida de forma automatizada y sin control humano directo.
- Herramienta de software utilizada para la localización de páginas disponibles en Internet. Constituye un índice generado de manera automática que se consulta desde la propia Red. .
- Instrumento que se acopla a un telescopio y que sirve de mira pata la ubicación rápida de un objeto en el cielo nocturno.
- Los buscadores (o motor de búsqueda) son aquellos que están diseñados para facilitar encontrar otros sitios o páginas Web. Existen dos tipos de buscadores, los spiders (o arañas) como Google y los directorios, como Yahoo.
- Herramienta diseñada específicamente para que los usuarios realicen búsquedas en Internet, introduciendo una o más palabras claves. Como resultado de ésta, el buscador devuelve una lista de resultados presentados en hipertexto, es decir en forma de enlace.
- Página web que, de diferentes maneras, ayuda al usuario a encontrar otras páginas que contengan la información que busca.

4.2.3. Tipos de buscadores

Existen varios tipos de buscadores, en función del modo de construcción y acceso a la base de datos, pero todos ellos tienen en común que permiten una consulta en la que el buscador nos devuelve una lista de direcciones de páginas web relacionadas con el tema consultado.

Los buscadores se pueden clasificar en tres tipos, según la forma de obtener las direcciones que almacenan en su base de datos. Cada tipo de buscador tiene sus propias características. Conocerlas puede ayudarnos a decidir cuál utilizar en función de las necesidades de nuestra búsqueda. No obstante, hoy en día todos los buscadores tienden a ofrecer el mayor número de servicios posible, con lo que sus ofertas de búsqueda se asemejan cada vez más, siendo difícil adivinar de qué tipo de buscador estamos hablando.

4.2.3.1. Índices De Búsqueda

Es el primer tipo de buscador que surgió. En los índices de búsqueda, la base de datos con direcciones la construye un **equipo humano**. Es decir, un grupo de personas va rastreando la red en busca de páginas. Vistas éstas son **clasificadas por categorías** ó temas y subcategorías en función de su contenido. De este modo, la base de datos de un índice de búsqueda contiene una lista de categorías y subcategorías relacionadas con un conjunto de direcciones de páginas web que tratan esos temas.

Yahoo www.yahoo.com	Directorio temático con más de 150 editores, 1.200.000 enlaces a sitios web ordenados en 14 categorías temáticas.
LookSmart www.looksmart.com	Guía interactiva con más de 1 millón de los sitios web. Actualizado a diario por 160 editores que seleccionan la información sobre el Web que cada sitio se repasa para la calidad y que se coloca en una de más de 70.000 categorías.
Open Directory dmoz.org	Proyecto abierto que produce un directorio comprensivo, con un extenso grupo de editores voluntarios. Posee 70.000 categorías temáticas, 500.000 enlaces y realiza 2.000 enlaces diarios
Snap www.snap.com	Recopila información de diversas fuentes, y enlaces en Internet. Las conexiones son agrupadas por categoría. Incluyen noticias, deportes, negocios, vida, salud, computación e Internet, las artes y humanismo, ciencia y tecnología, gente y sociedad, familia, educación, y más. Posee 60 editores, 50.000 categorías, 400.000 enlaces.
eBLAST www.eblast.com	Los editores de Britannica.com ofrecen más de 125.000 enlaces a sitios web evaluados y especializados en 16 categorías principales, incluida la Enciclopdia Británica.
BubleLink bubl.ac.uk/link	Reúne información sobre todas las disciplinas y ordena los temas según el sistema de clasificación decimal de Dewey. La selección, evaluación, catalogación y descripción del todos los items es realizada por especialistas.
Argus Clearinghouse www.clearinghouse.net	Creado por bibliotecarios y evaluado por especialistas ofrece guías temáticas de las siguientes áreas: Artes y Humanidades, Administración y Negocios, Comunicaciones, Computación y Tecnologías de Información,

Figura 4.2.- Cuadro de buscadores por categorías.

La consulta de un índice se realiza, pues, a través de categorías. La ventana de su versión en castellano tiene el aspecto de la imagen.

Figura 4.3.- Ventana de el Buscador Yahoo

Se puede observar que, a pesar de tratarse de un índice de búsqueda, ofrece también un espacio para introducir palabras clave (bajo el título de la web). Esto se debe a que todos los buscadores que ofrecen servicios en la red tienden a satisfacer al máximo las necesidades de los navegantes, de forma que intentan abarcar toda la gama de posibilidades.

Figura 4.4.- Ventana del Buscador Yahoo en donde se describe el directorio de sitios Web.

4.2.3.2. Motores de búsqueda

Temporalmente, los motores de búsqueda son posteriores a los índices. El concepto es diferente: en este caso, el rastreo de la web lo hace un programa, llamado araña ó motor (de ahí viene el nombre del tipo de buscador). Este programa va visitando las páginas y, a la vez, creando una base de datos en la que relaciona la dirección de la página con las 100 primeras palabras que aparecen en ella. Como era de esperar, el acceso a esta base de datos se hace por palabras clave: la página del buscador me ofrece un espacio para que yo escriba la ó las palabras relacionadas con el tema que me interesa, y como resultado me devuelve directamente un listado de páginas que contienen esas palabras clave.

Altavista www.altavista.com	Es una de las herramientas de mayor alcance en la Internet, con 250 millones de páginas indexadas. Permite consultar en más de 25 lenguas, indiza texto completo y de actualización diaria. Abierto desde 1995.
Excite www.excite.com	Uno de los servicios de búsqueda más populares en el Web. Abierto desde 1995, ofrece un índice que integra además, material no-Web. Posee 214 millones de páginas.
Northernlight www.northernlight.com	Uno de los buscadores preferidos entre investigadores. Con 240 millones de páginas, ofrece uno de los índices más grandes del Web, que además posee un conjunto de documentos de "colecciones especiales " difíciles de acceder en web. Abierto desde 1997.
Go/Infoseek www.go.com	Con un directorio de sitios web compilado por especialistas, proporciona resultados de calidad, utilizando un algoritmo de busca en las 50 millones de páginas que posee. INFOSEEK se inicio en 1995 y continuó oficialmente como Go en 1999.
Google www.google.com	Motor especialmente útil para recuperar buenos sitios web a través de búsquedas generales. Posee 200 millones de páginas.

Figura 4.5.- Cuadro de Motores de Búsqueda

Un buen ejemplo de motor de búsqueda es Google, aquí tenemos el aspecto de su página principal.

Figura 4.6.- Ventana de el buscador Google

Observando esta ventana vemos que, en la parte central-derecha hay una pestaña con el nombre Directorio. Si hacemos clic sobre ella nos llevará a otra página en la que se nos ofrece realizar la búsqueda por categorías. Como en el caso de los índices, los motores también tienden a ofrecer todos los servicios posibles al usuario, y le dan la posibilidad de realizar una búsqueda por categorías.

4.2.3.3 Metabuscadores

Los metabuscadores son páginas web en las que se nos ofrece una búsqueda sin que haya una base de datos propia detrás: utilizan las bases de varios buscadores ajenos para ofrecernos los resultados. Un ejemplo de metabuscador es Metacrawler.

http://www.c4.com	GoTo, Snap.com, WebCrawler, Yahoo, FindWhat.
InFind http://www.infind.com	Busca en WebCrawler, Yahoo, Lycos, Alta Vista, InfoSeek.
Metacrawler http://www.metacrawler.com	Busca en AltaVista, GoTo.com, InfoSeek, LookSmart, Lycos, Thunderstone, y WebCrawler.
ProFusion http://profusion.com	Busca en Alta Vista, Excite, InfoSeek, LookSmart, GoTo, Snap, WebCrawler, Yahoo, AllTheWeb.

Figura 4.7.- Cuadro de ejemplos de metabuscadores

4.2.4. Google español el Motor de Búsqueda más Potente.

Google se trata de un potente buscador con una amplia base de datos. Tal vez sea, junto con Altavista, uno de los más capaces que se nos ofrecen hoy en la red. Además, tiene versión en castellano. Su dirección URL es http://www.google.com/ y la vemos directamente en este idioma, ya que el buscador lo detecta en la versión de Windows en funcionamiento. El mercado actual está dominado por ***Google***, que **indexaba** hace un tiempo **el 71.16% de toda la red** y que cuenta con más de 9.500 empleados bajo un lema: “organizar la información mundial y hacerla universalmente accesible y útil”.

Una de las cosas más interesantes es su página de preferencias que permite elegir el interface de búsqueda entre 15 idiomas y los resultados entre 25, además se puede indicar cuantos resultados por página se quieren obtener, 10, 20, 30, 50 ó 100. Estas preferencias se guardan en nuestro navegador usando cookies y cada vez que conectemos con Google se activarán los valores seleccionados.

Este buscador utiliza la popularidad de los links como sistema primario para dar prioridad a sus resultados, es decir, muestra principalmente los sitios que más han elegido otros usuarios y por tanto los más populares. Básicamente, los usuarios están votando por los sitios más interesantes y por tanto es muy útil para búsquedas donde la popularidad es importante, como los viajes o los coches.

Poseen una de las bases de datos más grandes (suelen pelear por el primer puesto con FAST Search), con más de 1.000 millones de páginas indexadas por su "crawler", que provee de resultados a Yahoo y Netscape Search.

Utilizan Open Directory para mostrar su propio directorio en http://directory.google.com.

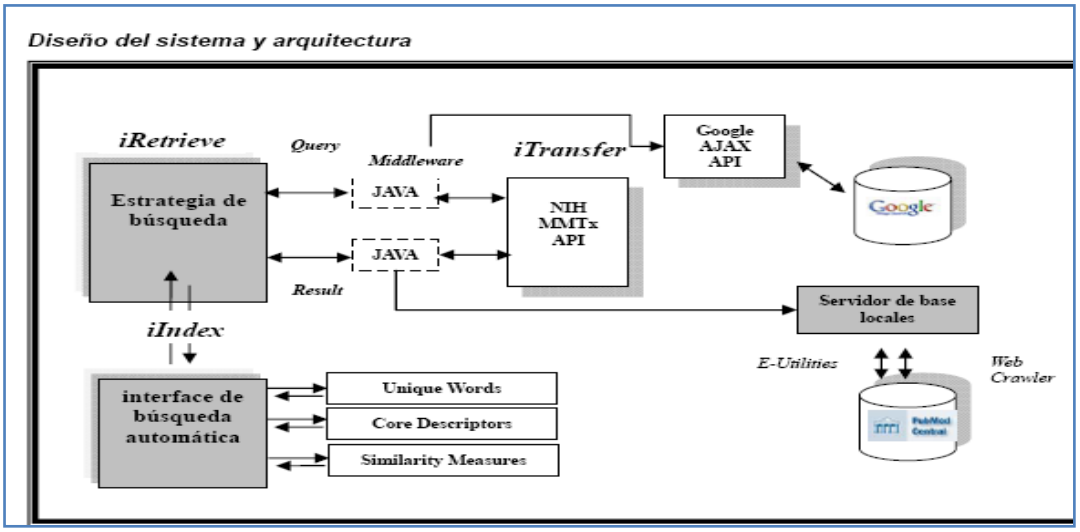


Figura 4.8.- Arquitectura del sistema de indización..

La página principal de este buscador es extremadamente sencilla y clara. Presenta de un vistazo todas sus herramientas sin desviar nuestra atención hacia detalles irrelevantes. Su aspecto es el siguiente:

Figura 4.9.- Ventana del Buscador Google

4.2.4.1. Búsqueda por palabras clave.

Es la que nos va a proporcionar resultados más numerosos, siempre que el objeto de la búsqueda sea concreto.

El resultado de una búsqueda de este tipo será una página en la que se nos ofrece un listado de páginas web que tratan el tema reflejado en las palabras clave. Pinchando en las distintas zonas señaladas de la imagen obtendrás información de su utilidad:

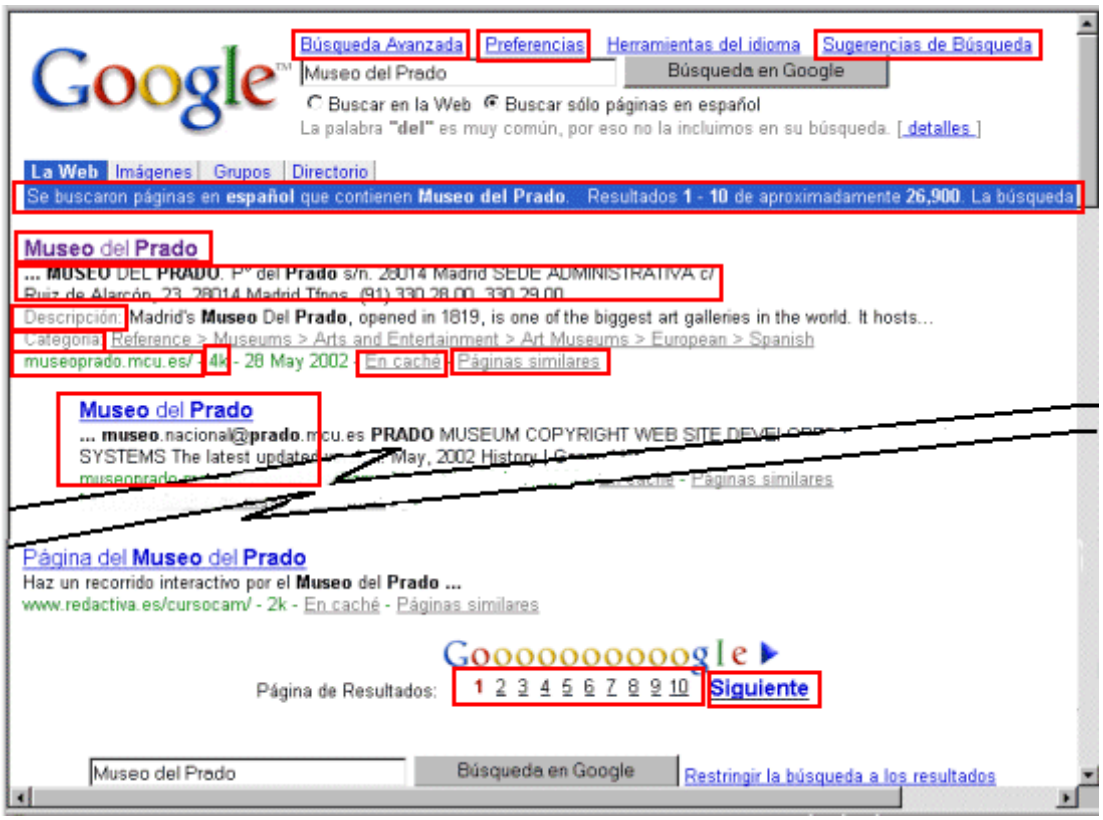


Figura. 4.10: Imagen de la ventana de google de búsqueda por palabra clave

4.2.4.2. Búsqueda por directorio

La búsqueda por directorio se utiliza cuando el tema de la búsqueda no es demasiado concreto. El directorio nos ofrece una lista de categorías divididas en subcategorías que nos permiten ir acotando el motivo de la búsqueda. Al final obtenemos un listado de páginas incorporadas al directorio de Google que nos ofrecen la información buscada. Pero además Google en particular nos permite hacer una búsqueda por palabras clave dentro de una categoría del directorio. Es decir, acotado parcialmente el tema de búsqueda (localizada la categoría) usamos las palabras clave para obtener las páginas que, en esa categoría, hacen referencia a lo que buscamos.

4.2.4.3. Búsqueda de imágenes

Esta es una opción que nos brinda Google y que resulta extremadamente útil. A partir de unas palabras clave que nosotros introducimos nos proporciona un

conjunto de imágenes relacionadas con esas palabras y que están contenidas en páginas web de su base de datos.

4.2.4.4. Búsquedas avanzadas

Cuando realizamos una búsqueda simple con Google, este localiza automáticamente páginas que contengan todas las palabras clave introducidas en la caja. Por ello, la forma de acotar la búsqueda, es decir, de reducir el número de páginas obtenido, es añadir más palabras clave. Si deseamos que encuentre páginas que lleven alguna de las palabras clave, o una expresión exacta (es decir, con las palabras en el orden que nosotros ponemos e incluyendo artículos, determinantes, etc...) debemos recurrir a la opción.

4.3. ELEMENTOS DE UN MOTOR DE BÚSQUEDA

Figura 4.11 Imagen de un modelo de buscador

Un motor de búsqueda está formado por cuatro elementos básicos:

- Un programa (también denominado robot, rastreador o webcrawler) que recorre el WWW buscando recursos de información y sus respectivas URLs.
- Un sistema automático de análisis de contenidos e indexación de los documentos localizados por el robot.
- Un programa que actúa de pasarela entre el servidor de documentos html y la base de datos.

4.3.1. **RASTREADOR WEB**

Definición: Un **Rastreador Web** (o araña de la web) es un programa que inspecciona las páginas del World Wide Web de forma metódica y automatizada.

Los **Web crawlers** se utilizan para crear una copia de todas las páginas web visitadas para su procesado posterior por un motor de búsqueda que indexa las páginas proporcionando un sistema de búsquedas rápido.

El índice de páginas generado por los **crawlers** es utilizado como parte central de cualquier sistema de acceso a la información en el WWW (como motores de búsqueda).

4.3.1.1 **Arquitectura de un crawler**

El funcionamiento de los web CRAWLER, lo podemos explicar a través del siguiente esquema de un buscador basado en crawler:

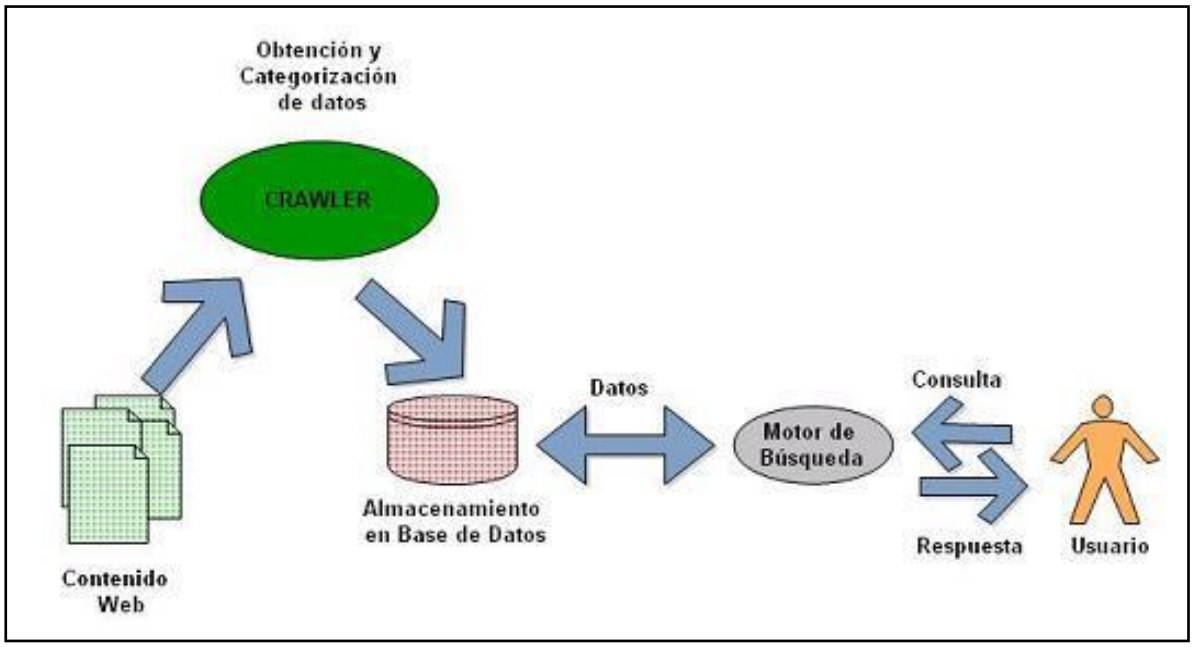


Figura 4.12: Diagrama de la Arquitectura de un WebCrawler

El programa llamado **crawler** o araña de la web (**spider**) no sólo tiene como objetivo realizar peticiones a los servidores para indexar, almacenar y categorizar la información. Antes que nada, visitan una URL específica provista por un servidor de URLs y almacenan su contenido en los discos de los buscadores, luego, aplicando distintos algoritmos, la "araña de la web" analiza y relaciona todo el contenido de los datos; no se limita al título y a las etiquetas meta del código html, porque esto puede llevar a indexar erróneamente el sitio, en cambio toma ciertas palabras dentro del documento tratando de contextualizar las mismas; también calcula la importancia de cada página en función de cuántas otras la enlazan y muchos otras variables y factores particulares que dependen ya de cada buscador.

Una vez analizado todo el contenido, generalmente, se los envía a otro programa encargado de almacenarlo en la Base de Datos.

Una diferencia en algunos buscadores se da en el hecho de que una vez indexado el sitio se elimina del disco del buscador y en cambio en otros como Google se guarda en los discos quedando esto como una especie de caché del sitio indexado y disponible en caso de que el sitio ya no esté más online.

Algo que la mayoría de los **crawler** tienen en cuenta es un archivo de texto en el directorio raíz del sitio que le indica que directorios no debe indexar. Esto último es muy importante ya que de no encontrar este archivo, los **crawler** indexarán todo el sitio de nuevo con toda la información que contenga.

4.3.2. Indexar En La Web

Indexar, es un **anglicismo** derivada de la palabra inglesa **index**, que significa índice.

Indexar es la acción que el buscador realiza para dar con una dirección, agregarla a su listado y categorizar su contenido. Nosotros no podemos indexar una web, "sugerimos" al buscador el hacerlo.

Entonces, cuando uno pregunta por el cómo Indexar un sitio, se refiere al, cómo hacer, para que el buscador tome en cuenta su petición y agregue su contenido.

4.3.2.1. Proceso de indexación:

Es el proceso por el cual una página o contenido se hace indexable (con posibilidad a ser indexado) a los buscadores Web, donde una página o dirección se incluye al listado de resultados o SERP (Search Engine Result Page). Indexar url e Indexar contenidos no significa necesariamente lo mismo El buscador hace una clara diferencia, entre la indexación de la dirección del sitio y la categorización e indexación del contenido. La primera se consigue en minutos a días y la segunda hasta varios meses. Por desgracia, ésto último no significa que todo el contenido se categorizará, habrá mucho contenido que no será tomado en cuenta.

Indexar no es un tema de certeza, intervienen diversos factores y esos factores son tratados por la SEO (Search Engine Optimization), donde se engloban los caminos más adecuados para volver indexable un sitio y su contenido, además del tema de las posiciones en los SERP, donde intervienen valores como el PageRank (PR) y la relación de los sitios con otros sitios (Vinculos unidireccionales o bidireccionales), el contenido estático del dinámico, el tipo de lenguaje web, entre muchos, muy diversos factores en la operación que realiza el algoritmo clasificador.

Así que la mejor forma de describir la palabra indexar incluye tomar en consideración que es un término que presenta un significado con un alto grado de ambigüedad.

4.3.3. Algoritmo Del Pagerank

PageRank[3] es una marca registrada y patentada por Google el 9 de enero de 1999 que ampara una familia de algoritmos utilizados para asignar de forma numérica la relevancia de los documentos (o páginas web) indexados por un motor de búsqueda. Sus propiedades son muy discutidas por los expertos en optimización de motores de búsqueda.

El sistema PageRank es utilizado por el popular motor de búsqueda Google para ayudarle a determinar la importancia o relevancia de una página. Fue desarrollado por los fundadores de Google, Larry Page y Sergey Brin, en la Universidad de Stanford.

PageRank confía en la naturaleza democrática de la web utilizando su vasta estructura de enlaces como un indicador del valor de una página en concreto. Google interpreta un enlace de una página A a una página B como un voto, de la página A, para la página B. Pero Google mira más allá del volumen de votos, o enlaces que una página recibe; también analiza la página que emite el voto. Los votos emitidos por las páginas consideradas "importantes", es decir con un PageRank elevado, valen más, y ayudan a hacer a otras páginas "importantes". Por lo tanto, el PageRank de una página refleja la importancia de la misma en Internet.

El algoritmo inicial del PageRank lo podemos encontrar en el documento original donde sus creadores presentaron el prototipo de Google: “The Anatomy

[3] *PageRank ha tomado su modelo del Science Citation Index (SCI) elaborado por Eugene Garfield para el Instituto de información científica (ISI) en los Estados Unidos durante la década de los 50.*

of a Large-Scale Hypertextual Web Search Engine":[2]

$$PR(A) = (1-d) + d * \sum_{i=1}^{n} \frac{PR(i)}{C(i)}$$

Donde:

- **PR(A)** es el PageRank de la página A.
- **d** es un factor de amortiguación que tiene un valor entre 0 y 1.
- **PR(i)** son los valores de PageRank que tienen cada una de las páginas **i** que enlazan a A.
- **C(i)** es el número total de enlaces salientes de la página i (sean o no hacia A).

Algunos expertos aseguran que el valor de la variable *d* suele ser 0,85.[Representa la probabilidad de que un navegante continúe pulsando links al navegar por Internet en vez de escribir una url directamente en la barra de direcciones o pulsar uno de sus marcadores y es un valor establecido por Google. Por lo tanto, la probabilidad de que el usuario deje de pulsar links y navegue directamente a otra web aleatoria es *1-d*.La introducción del factor de amortiguación en la fórmula resta algo de peso a todas las páginas de Internet y consigue que las páginas que no tienen enlaces a ninguna otra página no salgan especialmente beneficiadas. Si un usuario aterriza en una página sin enlaces, lo que hará será navegar a cualquier otra página aleatoriamente, lo que equivale a suponer que *una página sin enlaces salientes tiene enlaces a todas las páginas de Internet*.

El peso o importancia de una página es el resultado de una "votación" entre todas las demás páginas de la World Wide Web acerca del nivel de importancia que tiene esa página. Un hiperenlace a una página cuenta como un voto de apoyo. El PageRank de una página se define **recursivamente** y depende del número y PageRank de todas las páginas que la enlazan. Una página que está

enlazada por muchas páginas con un PageRank alto consigue también un PageRank alto. Si no hay enlaces a una página web, no hay apoyo a esa página específica. El PageRank de la barra de Google va de 0 a 10. Diez es el máximo PageRank posible y son muy pocos los sitios que gozan de esta calificación, 1 es la calificación mínima que recibe un sitio normal, y cero significa que el sitio ha sido penalizado o aún no ha recibido una calificación de PageRank.] Parece ser una escala logarítmica. Los detalles exactos de esta escala son desconocidos.

4.3.4. Parser-html de software libre para java implementado en el en programa Rastreador UNL del motor de búsqueda.

El parser-html o librerias.jar que utilizamos dentro del motor de búsqueda es jericho-html, pero antes de explicar porque se eligió este parser y su instalación, explicaremos que es un parser, su funcionamiento y algunos parsers-html de software libre para java con sus características y ventajas.

Para realizar la limpieza de las notas de prensa en el código fuente de los documentos html, es necesario utilizar alguna herramienta que permita recorrer el código html del documento y seleccionar aquellas partes que sean de utilidad. Una de las herramientas que permite hacer lo que se ha explicado anteriormente son los programas denominados HTML parsers o parsers HTML,

4.3.4.1. ¿Qué es un Parser HTML?

Un parser HTML es un programa que recorre el código fuente de un documento HTML y permite extraer información del documento de la manera que se estime oportuna para poder tratarla posteriormente. También pueden servir para manipular o modificar un documento HTML.

4.3.4.2. Funcionamiento de un Parser

Los parsers HTML están basados en los parsers para el lenguaje de marcado XML (HTML es un lenguage XML), pero están adaptados para poder recorrer código HTML en el que puede haber malformaciones tales como que haya etiqueta de apertura de un bloque pero no de cierre, que se cierre un bloque antes que otro cuando debería ser al revés o cualquier otro tipo de error que pueda existir en un documento HTML.

Los parsers cargan el fichero a escanear en memoria mediante alguna estructura de datos, una vez que el documento está cargado, se puede acceder a el mediante el DOM del documento, es decir mediante las etiquetas de marcado.

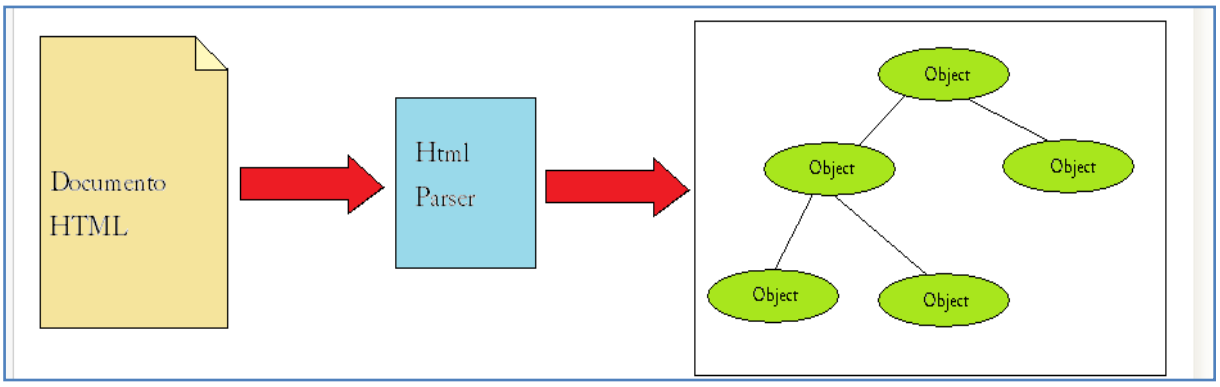


Figura 4.13: Imagen del Funcionamiento de un Parser HTML.

Lo primero que se hace es leer el documento html, a continuación se pasa a través de un parser html y este lo transforma en una estructura de datos como por ejemplo un árbol, desde el cual se puede acceder al contenido de cualquiera de las etiquetas del documento html.

4.3.4.3. Parser Html Software Libre Para Java

Existe gran cantidad de parsers bajo la licencia de software libre para poder usarlos en Java. En esta sección se mostrarán los más importantes de los mismos, así como sus principales características y así poder escoger entro uno u otro para poder usarlo para limpiar las notas de prensa para este proyecto.

❖ NekoHTML

NekoHTML es un escáner simple de código HTML y balanceador de etiquetas que permite parsear documentos HTML y acceder a la información usando interfaces de XML estándar. El programa puede explorar documentos HTML y obviar muchas de las equivocaciones que se cometen en la escritura de código HTML.

NekoHTML está escrito usando la Interfaz Nativa de Xerces (XNI) que está incluida en la implementación de Xerces2. Esto habilita a usar NekoHTML con cualquiera de las herramientas XNI existentes sin necesidad de variar su código.

Limitaciones

- Existe documentos que NekoHTML Parser no genera un flujo de eventos de XML bien formado, por ejemplo documentos con múltiples etiquetas <html>.
- El código añadido al núcleo DOM en Xerces-J 2.0.1 introduce un bug en la implementación de DOM HTML basado en él. El problema afecta a los usuarios de NekoHTML que usen el parser con Xerces-J 2.0.1.

❖ HTML Parser

HTML Parser es una biblioteca de Java usada para analizar el código HTML de una manera lineal o jerarquizada. Su uso está optimizado sobre todo para la trasformación o extracción de código HTML y ofrece además una serie de filtros para poder filtrar la información y adicción de nuevas etiquetas personalizadas. Se trata de un paquete rápido y robusto.

Características

En cuanto a la extracción de datos del código html que es lo que se necesita para este proyecto, es capaz de lo siguiente:

- Extracción de texto, por ejemplo para servir como entrada de motores de búsqueda de bases de datos.
- Extracción de enlaces, para poder acceder a las páginas web que apuntan o conseguir los correos electrónicos que contiene.
- Raspados de pantalla para una entrada de datos programada.
- Extracción de recursos como imagen o sonido
- Chequeo de enlaces, averiguando que enlaces son válidos

❖ **Jericho HTML Parser**

Jericho HTML Parser es una biblioteca Java simple pero a la vez muy potente para analizar y modificar documentos HTML, así como ciertas etiquetas del lado del servidor. Es capaz de reproducir partes desconocidas o inválidas del código HTML. Es una librería de código abierto bajo la licencia Eclipse Public License (EPL) y GNU Lesser General Public License (LGPL).

Principales características

Esta librería se distingue de otras herramientas de análisis de código HTML por las siguientes cualidades:

- La presencia de HTML mal formado no interfiere en el análisis del resto del documento, lo que hace a esta librería ideal para el análisis en el mundo real, cosa que no cumplen otras herramientas.
- Etiquetas de servidor como ASP, JSP, PSP, PHP y Mason son reconocidas explícitamente por el parser.
- No es un parser basado en árbol (como DOM) o eventos (como SAX), pero utiliza una combinación de búsqueda simple de texto, reconocimiento eficiente de etiquetas y una caché de posición de etiquetas. El texto del documento fuentes es cargado primero en memoria y entonces solamente se buscan los segmentos relevantes por los caracteres relevantes para operación de búsqueda.

- Comparado con un programa de análisis basado en árbol tal como DOM, el uso de memoria puede ser radicalmente menor si lo que se necesita analizar son secciones muy pequeñas de código.
- Comparado con un parser basado en eventos como SAX, su interfaz es más intuitiva y de más alto nivel.
- Las posiciones de comienzo y fin en el documento fuente de todas las partes analizadas son accesibles.
- El número de fila y columna de cada posición en el documento fuente es fácilmente accesible.
- Tipo personalizados de etiquetas pueden ser fácilmente definidas y registradas para poder ser reconocidas por el parser.
- Funcionalidad para extraer todo el texto del código HTML para poder alimentar motores de búsqueda por ejemplo.

❖ **JTidy**

JTidy es una interfaz de Java para HTML Tidy y un verificador de sintaxis de HTML. JTidy puede ser usado como una herramienta para limpiar html defectuoso y para formatear el código HTML. También provee de una interfaz DOM al documento que está siendo procesado.

Características

- Capacidad de eliminar errores en el código HTML tales como:
  - Ausencia de etiqueta de cierre de un elemento
  - Etiquetas de cierre escritas en el orden equivocado
  - Recuperación de un mezclado erróneo de etiquetas
  - Adición de "/" perdidas en las etiquetas de cierre
  - Perfeccionamiento de listas añadiendo etiquetas olvidadas
- Indenta el código Html correctamente
- Posibilidad de leer varios juegos de caracteres internacionales como US ASCII, ISO Latin-1, UTF-8 y la familia ISO 2022.
- Limpieza del marcado del código Html

- Posibilidad de añadir nuevas etiquetas
- Soporte limitado a ASP, PHP y JSTE
- Soporte limitado a XML

❖ **TagSoup**

TagSoup es un parser basado en SAX y escrito en Java y es capaz de analizar HTML mal formado. Proporciona una interfaz de SAX que permite a las herramientas para XML estándar sean capaces de leer HTML mal formado.

Características

- TagSoup está pensado como un parser HTML, no como un programa capaz de limpiar permanentemente HTML mal formado, si no que escanea este código mal formado al vuelo.
- No es capaz de convertir la presentación en HTML a CSS
- No depende de ningún otro framework aparte de SAX
- TagSoup no funciona correctamente con Java 5 y Java 6

❖ **HotSax**

Es un parser basado en SAX2 para el análisis de XML, HTML y XHTML. Puede ser usado por agentes web simples o spiders. Este parser es similar al parser para XML de Apache, Xerces, pero con eventos para código HTML mal formado.

Esta herramienta está diseñada para ayudar a construir otras herramientas útiles como spiders, raspadores de páginas y para conversores de HTML a otros formatos. Se puede insertar HotSAX en grandes aplicaciones como sistemas de gestión.

❖ **HtmlCleaner**

HtmlCleaner es un parser HTML de código abierto escrito en Java. Dado que los documentos Html usualmente están mal formados debido a errores de

programación, existen herramientas que no pueden leerlo a no ser que se arreglen estos fallos. Para ello existe esta herramienta que es capaz de leer el documento y arreglar los fallos existentes produciendo XML estándar.

HtmlCleaner implementa el conjunto de etiquetas estándar de HTML y reglas para realizar un balanceado correcto.

- **Java Mozilla Html Parser**

Mozilla Html Parser es un parser escrito en Java y basado en el parser de Mozilla. Actúa como puente desde clases de Java a clases de Mozilla y obtiene como salida un DOM desde una entrada de código HTML sucio.

Limitaciones

La mayor limitación conocida de Mozilla HTML Parser está relacionada con su funcionamiento ya que el programa de análisis serializa las peticiones. En el momento que el programa de análisis está funcionando y recibe una petición, la analiza y la pone en una cola la respuesta al solicitante.

4.3.4.4. Elección De Un Parser Html

Una vez visto distintas opciones para escoger un parser o analizador de HTML, el siguiente paso es escoger de entre todos ellos uno que permita hacer la tarea de limpiar los documentos de la manera más eficiente posible intentando no sobrecargar mucho la máquina.

- **Características del Parser HTML a elegir**

Antes de escoger un parser es necesario definir una serie de características que serían deseables que fuesen cumplidas. Las características son las siguientes:

1. Capacidad de leer Html mal formado
2. Bajo uso de memoria ya que se analiza una pequeña porción de código
3. Facilidad de uso

4. Extracción de los datos del documento fácilmente.

❖ **Elección del parser**

Observando las características antes definidas que deben cumplir los parser y verificando que las cumplen los parsers anteriores, se observa que el parser Jericho HTML Parser cumple todos estos requisitos.

Dado que es un parser específico para Html es de suponer que la primera carácterísticas si la cumpla. La segunda característica también la cumple ya que el uso de memoria puede ser radicalmente menor que un parser basado en árbol como DOM si lo que se necesita analizar son secciones pequeñas de código.

Se ha hecho alguna prueba con documentos html sencillos para comprobar el funcionamiento y se comprueba que es muy fácil su uso y bastante intuitivo. También es posible mediante esta librería extraer todos los datos incluidos en el código Html.

Se han probado otro parsers también pero son más dificiles de usar que este o no cumplen todas las características definidas por lo que el analizador de html que se usará en el proyecto será Jericho HTML Parser.

4.3.4.5. Instalación de Jericho HTML Parser

Como ya se ha comentado, el parser HTML a utilizar será Jericho HTML Parser, y antes de empezar a la construcción del programa que limpie los documentos es necesario descargar e instalar el parser en nuestro sistema. Para poder utilizar el parser hay que realizar los siguientes pasos:

1. **Descargar** el parser de la página http://sourceforge.net/projects/jerichohtml/
2. Como la descarga anterior es un fichero comprimido en formato "zip" es necesario **descomprimirlo** mediante alguna herramienta que lo permita.

3. Incluir en el programa la **librería jericho-html-3.2.0.jar** que se encuentra en el directorio lib.
4. Importar en el programa Java **au.id.jericho.lib.html.*;** para tener acceso a Jericho HTML parser.

4.3.4.6. Código De Jericho-Html-3.2.0.Jar Que Importa El Programa Rastreador Unl

En el paquete package unl.rastreador_web.modelo; de nuestro proyecto rastreadorUNL se encuentra la clase Lector pagina que utiliza métodos de la librería jericho-html-3.2.0.jar y que a continuación se los describe.

Clases y métodos jericho-html-3.2.0	Descripción
Source	
Element	
MicrosoftTagTypes.register();	
PHPTagTypes.register();	
PHPTagTypes.PHP_SHORT.deregister();	
MasonTagTypes.register();	
HTMLElementName	
TextExtractor	
StartTag	
CharacterReference	

Figura 4.14: Imagen de la descripción de jericho

4.4. Estrategias para que los resultados de una búsqueda sea exitosa

Para obtener buenos resultados dependen de elementos o signos y de la habilidad del usuario.

Buscar en Internet nunca fue fácil. Sin embargo, nuestro motor de búsqueda "Buscador BUNL" nos ayudarán a encontrar lo que deseamos mediante la "Búsqueda Avanzada", y si aprendemos su manejo y algunos trucos de búsqueda. Pero cabe recalcar que muchas veces en Internet los sitios Web no siempre están bien definidos ya encontraremos paginas mal construidas y con títulos que no tiene relación con el contenido de la página del sitio Web.

Un buen plan de busqueda.

- Identifique: los conceptos claves y el área a la que pertenece el objeto de su búsqueda.
- Después use un buscador BUNL para obtener información más específica.
- Por ejemplo: sí le interesan las nuevas tecnologías y la educación no universitaria , mire en la página del P.N.T.I.C. * o en Education World. Sí ese no fuera su tema, podrá encontarlos en los buscadores descritos en los dos apartados anteriores.

Obtener sólo los resultados deseados

Posibles soluciones a los tres problemas más frecuentes de los buscadores automáticos.

1. Demaciados Resultados
2. Sin resultados o demaciado pocos
3. Resultados demaciado lentos

4.4.1. Demasiados resultados

Sea más específico en la descripción del tema.

Usar más palabras claves y relacionarlas con el AND lógico.

Exigir la presencia de las palabras más relevantes.

Eliminar posibles palabras parecidas sin interés, mediante el NOT lógico

Usar frases en vez de palabras sueltas si es posible.

Restringuir la búsqueda a campos concretos. Por ejemplo:

Título (title)

Url

enlace (link)

Anfitrión (host)

Ponga en mayúsculas la primera letra de los nombres propios y use acentos.

Escríbala en castellano o cualquier otro idioma que no sea el inglés.

Si desea darle mayor consideración a cierta palabra, simplemente repítala.

4.4.2. Sin resultados o demasiado pocos

Quitar palabras claves dejando sólo las más relevantes.

Cambiar el AND por el OR lógico.

Compruebe su ortografía. Sobre todo si deberían haber mas resultados de los conseguidos.

Use sinónimos y variantes.

Cambie o incluya el otro número. Por ejemplo: libro a libros ; lápices a lápiz.

Ponga todas las palabras en minúsculas.

Use buscadores mas universales y use el inglés.

Es posible no haya mucha información sobre su tema.

4.4.3. Resultados demasiado lentos

Elimine las palabras comunes o frecuentes. No utilice palabras de pocas sílabas como los artículos pues no facilitarán la búsqueda y la prolongarán innecesariamente.

No use muchas palabras. Elimine las superfluas.

Cambie de buscador, quizás esté sobrecargado o realice la búsqueda en otro momento.

Ahorre esfuerzo.

Buscar en Internet es la manera rápida de obtener información básica en un tema; además puede investigar en varias fuentes a la vez.

Si la fuente no le resulta fiable sáltela.

Grabe las páginas de su interés sin leerlas completamente, busque solo en los títulos y palabras claves de los documentos. Utilice la opción *Buscar* de su visor. Guárdelos mediante:

Cortar y pegar alternando con un procesador de texto.

Grabar en formato HTML o Texto (ASCII).

No imprima por defecto, no es práctico y ni ecológico.

Guardar los sitios favoritos

Acuérdese de incluir los sitios más habituales en su propia colección de favoritos. Para asi ahorrar pasos intermedios.

Si encuentra una página que le parece interesante guardela como favorita y después cuando la lea totalmente (posiblemente sin estar conectado y si la grabó) decida si la mantendrá o eliminará de su lista de preferidos.

Vea las recopilaciones de su interés antes de decidir hacer una, es posible que alguien ya la haya realizado.

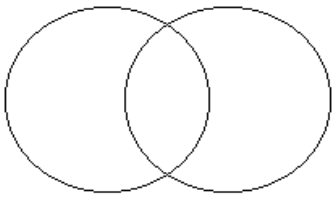
Si los contenidos de su búsqueda cambian rápidamente. Guarde la búsqueda para realizarla más adelante de forma cómoda. Recuerde que las cosas cambian en la Web.

Evite repetir los sitios visitados previamente. Un sitio se compone de varias páginas, si lo hemos desestimado al ver alguna de ellas evite ver las otras. Si son muchas puede eliminarlas de la página de resultados.

4.4.4. Operadores

Para obtener los resultados deseados el buscador deberá permitir el ajuste en la búsqueda para ampliarla , reducirla o dirigirla según la observación de los resultados o de nuestras previsiones iniciales. Existen diversas clases operadores que lo facilitan:

4.4.4.1 Operadores lógicos



Los operadores lógicos o booleanos nos faciltan este objetivo. Para los ejemplos siguientes usaremos dos conjuntos de elementos los *estudiantes* y los *europeos* que representaremos gráficamente con dos círculos.

1. Y lógico (AND)

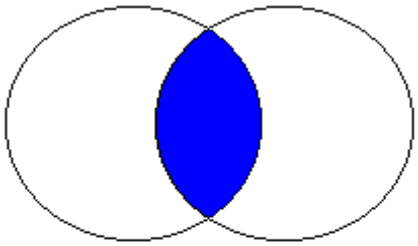
Es la intersección de los dos conjuntos de búsqueda. Apareciendo en el resultado únicamente los elementos que aparecen en los dos conjuntos. Es un operador de reducción. Ejemplo:

estudiantes AND europeos Es decir los *estudiantes europeos.*

En nuestro ejemplo:

- Sólo nos devolverá las páginas que contengan **ambos** elementos.

- No nos devolverá las páginas que sólo contengan uno de los dos o ninguno.
- También podemos usar **&** en vez de **AND**.



2. NO lógico (NOT)

Excluye los elementos los elementos de uno de los dos conjuntos de la búsqueda. Apareciendo en la búsqueda únicamente los elementos que no aparecen en el conjunto indicado. Es un operador de reducción. Ejemplo:

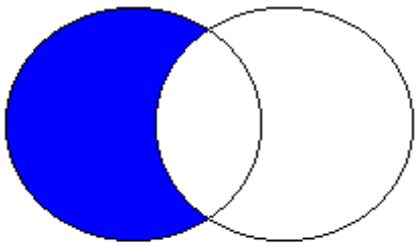
estudiantes AND NOT europeos Es decir los *estudiantes* que no sean *europeos*.

En nuestro ejemplo:

Sólo nos devolverá las páginas que contengan el primer elemento y no el segundo.

No nos devolverá las páginas en las que figure la palabra europeo.

También podemos usar **!** en vez de **NOT**.



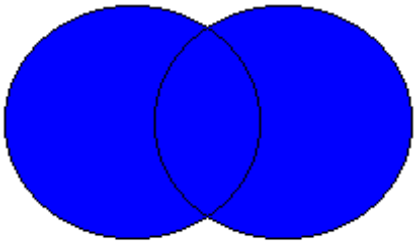
3. O lógico (OR)

Unión de los elementos de los dos conjuntos de la búsqueda. Apareciendo en la búsqueda todos los elementos. Es un operador de ampliación. Ejemplo:

estudiantes OR europeos Es decir los *estudiantes* o *europeos*.

En nuestro ejemplo:

- Nos devolverá las páginas que contengan algún elemento de los dos o los dos.
- No nos devolverá las páginas que no contengan ningún elemento de los dos.
- Tambien podemos usar **|** en vez de **OR**.
- Cuando no indicamos ningún operador entre palabras los buscadores lo interpretan como si hubiéramos colocado **OR**.



Es muy útil para buscar varias formas de escritura simultáneamente. Ejemplo:

España OR Spain OR Espagne

4.4.4.2. Operadores de proximidad

Para obtener unos resultados precisos el operador de proximidad nos permite especificar la relación entre elementos presentes de nuestra búsqueda.

1. Cerca (NEAR)

Es la intersección de los dos conjuntos de búsqueda. Se parece al Y lógico, pero además exige que entre ambas palabras no haya más de 10 palabras. Ejemplo:

estudiantes NEAR europeos

Es decir que las palabras *estudiantes* y *europeo* aparezcan relativamente juntas.

2. Junto (ADJ)

Es la intersección de los dos conjuntos de búsqueda que además están adyacentes. Se parece al Y lógico pero además exige que entre ambas palabras no haya otra palabra. Ejemplo:

estudiantes ADJ europeos

Es decir que las palabras *estudiantes* y *europeos* aparezcan juntas.

Trucos:

No lo tiene el buscador “BUNL” pero puede usar comillas con un resultado idéntico

"estudiantes europeos"

También puede usar un guión entre las palabras que desea juntar

estudiantes-europeos

3. Frases

Es la intersección de la palabras de búsqueda que además están adyacentes y en el orden en que se describen. Se parece al Y lógico pero además exige que la secuencia de palabras sea idéntica. Ejemplo:

"estudiantes europeos conectados a Internet"

Es decir que la *frase* aparezcan igual.

- Trucos:

Puede usar el guión - para concatenar y obtener un resultado idéntico

estudiantes-europeos-conectados-a-Internet

Cuando buscamos un texto concreto es muy útil. Operadores de existencia 1. Presencia .

Especifica palabras que deban aparecer en el resultado. Podemos exigir la presencia de alguna palabra clave en el documento, de tal forma que si no está presente en él no se incluirá en la lista de resultado.

Habitualmente se añade al inicio de la palabra clave el signo más (+) .No ponga ningún espacio entre el + y la palabra a buscar.

Por ejemplo:

+olímpico baloncesto fútbol voleibol balonmano

2. Ausencia

Podemos exigir la ausencia de alguna palabra clave en el documento, de tal forma que si está presente en él no lo incluirá en la lista de resultado.

Habitualmente se indica añadiendo el signo menos - al inicio de la palabra clave.

Por ejemplo:

juventud -racismo

Nota: Podemos obtenerlo también a través de estos dos operadores lógicos.

juventud AND NOT racismo

4.4.4.3 Operadores de exactitud

1. Familia**.**

Si quiere que la búsqueda localice también aquellos documentos donde las alabras tecleadas aparezcan como partes de otras palabras. ponga el signo (*****) después de la palabra a buscar para hacer que se expanda. De esta manera, una búsqueda de "educa" incluirá también en la respuesta "educador" y

"educativo". Esto es ideal si no se sabe como escribir la palabra a buscar exactamente.

2. Truncar la palabra.

Para encontrar sólo aquellas referencias que tengan la palabra exacta y no extensiones, use el punto **(.)** al final de una palabra en la búsqueda para limitarla. Por ejemplo "ciudad" encontrará referencias que contengan la palabra "ciudad" pero no así "ciudadano" ni "ciudadania".

4.5. HERRAMIENTAS DE DESARROLLO

4.5.1. JAVA

Java es un lenguaje orientado a objetos similar a C++ que nos permite desarrollar aplicaciones completas e independientes, no sólo para internet sino también para cualquier ámbito. Además, otra de las características más importantes del lenguaje java es la posibilidad de que las aplicaciones construidas sean operativas bajo diferentes plataformas.

Los programas java destinados a la Web se ejecutan dentro de los navegadores que estén preparados para ello, todos los navegadores modernos lo están, y son conocidos como applets (miniaplicaciones). Cuando al navegar se encuentra una página que alberga un applet java se pone en funcionamiento lo que es llamado "máquina virtual", que prepara al navegador para la ejecución automática de esta miniaplicación, que previamente es descargada desde el servidor al ordenador cliente.

El archivo java se guarda en el servidor, siendo descargado hacia el ordenador cliente cuando alguien se baja la página web que lo contiene. Una vez en el ordenador cliente, el applet java se verifica por seguridad y se guarda en una parte determinada de la memoria del ordenador. Finalmente se ejecuta el programa java.

4.5.1.1. Funcionamiento de una aplicación java:

a. Después de escribir y compilar el applet java, éste debe ser colocado en un servidor web. Contrariamente a lo que suele ocurrir con las secuencias CGI, las aplicaciones java pueden ser archivadas en cualquier directorio del servidor. Esto es así porque los scripts java se ejecutan en el ordenador cliente, mientras que los programas CGI se desarrollan en el propio servidor, siendo por lo tanto vulnerables a la intromisión ajena.
b. Cuando un usuario visita una página que contiene una aplicación java, en primer lugar ésta se descarga desde el servidor al ordenador cliente y a continuación se pone en funcionamiento el intérprete java del navegador.
c. Durante la interpretación del código del applet java se produce un proceso de verificación para detectar la existencia de virus y asegurar una ejecución segura.
d. Finalizada la verificación, los datos se colocan en una zona restringida del computador donde se ejecutan, favoreciéndose nuevamente la seguridad del proceso.
e. Por último, la aplicación java se ejecuta.

Para la implementación de nuestro sitio de búsquedas Web se ha estimado conveniente realizarlo sobre la plataforma de trabajo Linux, y las herramientas de desarrollo lenguaje de programación de páginas Web dinámicas PHP y base de datos MySQL. Ya que estas herramientas son utilizadas en la implementación del portal Web de nuestra universidad y permiten realizar nuestro motor de búsqueda

4.5.2. MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

MySQL es muy utilizado en aplicaciones Web como MediaWiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

4.5.3 JBoss

Es un servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte.

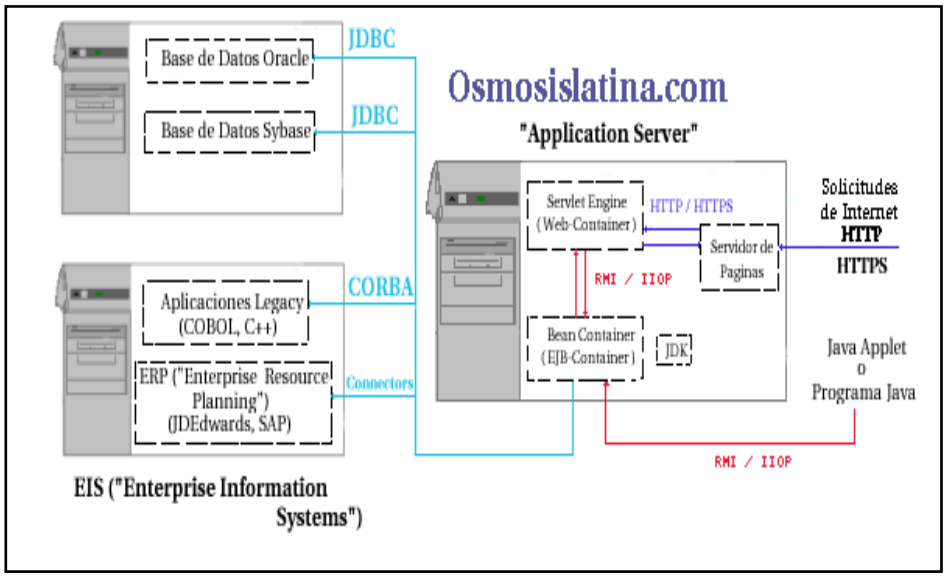


Figura 4.15: Imagen de la Arquitectura de JBoss

El producto JBoss es únicamente un "EJB Container" y es por esto que generalmente se utiliza en conjunción con un "Web-Container", el "Web-Container" puede ser cualquiera disponible en el mercado, sin embargo,

cuando obtenga se obtiene JBoss incluiye Tomcat proporcionado como "Web-Container", aunque lo anterior no restringe a JBoss para operar con otro "Web Container" como ServletExec , la única ventaja de utilizar el "Web Container" incluido con JBoss será en tiempo de coordinación/configuración entre JBoss|"x" Web-Container, y siendo que un ambiente utilizando EJB's es altamente complejo es preferible concentrarse en algo que ya ha sido utilizado y depurado.

4.5.3.1 Características

Las características destacadas de JBoss incluyen :

- Producto de licencia de código abierto sin coste adicional.
- Cumple los estandares.
- Confiable a nivel de empresa
- Incrustable, orientado a arquitectura de servicios.
- Flexibilidad consistente
- Servicios del middleware para cualquier objeto de Java
- Ayuda profesional 24x7 de la fuente
- Soporte completo para JMX

4.5.3.2. Estructura de JBoss

conf

Este directorio contiene las diferentes secciones de configuración utilizadas por JBoss, dependiendo de la modalidad utilizada este directorio puede contener distintos archivos, sin embargo, sus detalles serán descritos en configuración de JBoss .

data

Contiene distintos parámetros y archivos de configuración para las Bases de Datos proporcionadas con JBoss (Hypersonic y la implementación "Messaging" de JBoss)

deploy

Este directorio es ampliamente utilizado ya que aquí se colocan los EJB's para que sean ejecutados por JBoss, una vez colocado el archivo JAR (en forma de EJB) en este directorio, JBoss automáticamente expande y ejecuta el EJB.

lib

Contiene los archivos JAR's empleados por JBoss en base a la modalidad tratada.

log

Contiene los distintos registros ("Logs") generados por JBoss.

tmp

Contiene archivos creados por JBoss y utilizados de manera temporal.

work

Contiene las clases y archivos utilizados por JBoss para ejecución.

PROPUESTA ALTERNATIVA

FASE 1

ANALISIS

5. DESARROLLO DE LA PROPUESTA ALTERNATIVA

5.1. ANÁLISIS DE REQUERIMIENTOS

La Universidad Nacional de Loja, cuenta con un Portal, en el cual presenta los diferentes servicios y oferta académica, acorde a las nuevas tecnologías y con un alto nivel académico.

Para la recuperación de información de la Web, desde un navegador tendríamos que saber y grabarnos miles de direcciones web, por tal razón que en la actualidad existen los buscadores permitiendo la recuperación de información.

En el Portal de la Universidad Nacional de Loja se ve la necesidad de implementar un buscador, ya que es visitado por una gran cantidad de estudiantes, los mismos que podrán realizar diferentes tipos de búsquedas de acuerdo a sus requerimientos.

5.2. REQUERIMIENTOS FUNCIONALES

	Descripción	Categoría
RF001	La aplicación permitirá digitar la palabra clave en el campo de búsqueda.	EVIDENTE
RF002	El sistema validar que el campo se encuentre lleno.	OCULTA
RF003	El sistema permitirá realizar la búsqueda de la palabra clave ingresada por el usuario.	OCULTA
RF004	El sistema permitirá seleccionar categorías de búsqueda(salud, informática, ciencia, economía,	EVIDENTE

	educación, contabilidad)	
RF005	El sistema seleccionar las opciones de búsqueda avanzada	EVIDENTE
RF006	El sistema el enlace a google, yahoo y menssager	EVIDENTE
RF007	El sistema utilizará un programa denominado Rastreador UNL que recorrerá la Web buscando recursos de información y sus respectivas URLs, incorporándolas a una colección.	OCULTA
RF008	La aplicación la administración del Rastreador UNL.	EVIDENTE
RF009	El sistema ordenar las páginas de acuerdo a su relevancia	OCULTA
RF010	El sistema presentará los resultados de la búsqueda en la página del buscador.	EVIDENTE
RF011	El sistema permitirá modificar el horario de indexacion	

5.3. REQUERIMIENTOS NO FUNCIONALES

Código	Descripción
RNF001	El motor de búsqueda tendrá una interfaz gráfica de usuario amigable.
RNF002	El motor de búsqueda deberá ser una aplicación Web
RNF004	El motor de búsqueda deberá ser multiplataforma
RNF005	El motor de búsqueda se desarrollará en Java, JSP, base de datos MySQL y servidor JBoss.
RNF006	La arquitectura del motor de búsqueda será cliente – servidor.
RNF007	El sistema tendrá un tiempo de respuesta de búsqueda no más de 5 segundos
RNF008	El sistema deberá ser multiusuario

5.4 DEFINICIÓN DE ACTORES Y METAS

TERMINO	DEFINICION	CONCEPTO
USUARIO	Persona que haga uso del buscador	ACTOR
ADMINISTRADOR	Actor del sistema que tiene los atributos de manejar todos los procesos del sistema. Específicamente es el que se encargar de ejecutar el Rastreador UNL.	ACTOR
PORTAL DE UNIVERSIDAD	Es lugar donde va estar alojada la aplicación	**CONCEPTO**
BUSCADOR WEB	Programa que se encarga de facilitar la	**CONCEPTO**

	búsqueda de información en la WWW.	
PALABRA CLAVE	Palabra o frase con que inicia la búsqueda	**CONCEPTO**
BUSQUEDA AVANZADA	Opción para seleccionar una búsqueda avanzada con todas las palabras, frase completa o con alguna palabra.	**CONCEPTO**
BUSQUEDA CATEGORIAS	Opción para una búsqueda por categorías, con los siguientes criterios: Salud, Informática, Economía, Contabilidad, entre otras.	**CONCEPTO**
RESULTADOS DE BUSQUEDA	Grupo de ítems, presentados por la búsqueda	**CONCEPTO**
RASTREADOR	Programa que recorre la Web y recupera, almacena y categoriza URLs	**CONCEPTO**
INDEXACIÓN	Ordenar, clasificar la información	**CONCEPTO**
PAGERANKING	Mide la importancia y relevancia de una página en base al número y calidad de las páginas que la referencian.	**CONCEPTO**
CAMPO	Espacio vacío para digitar la palabra clave a buscar	**CONCEPTO**

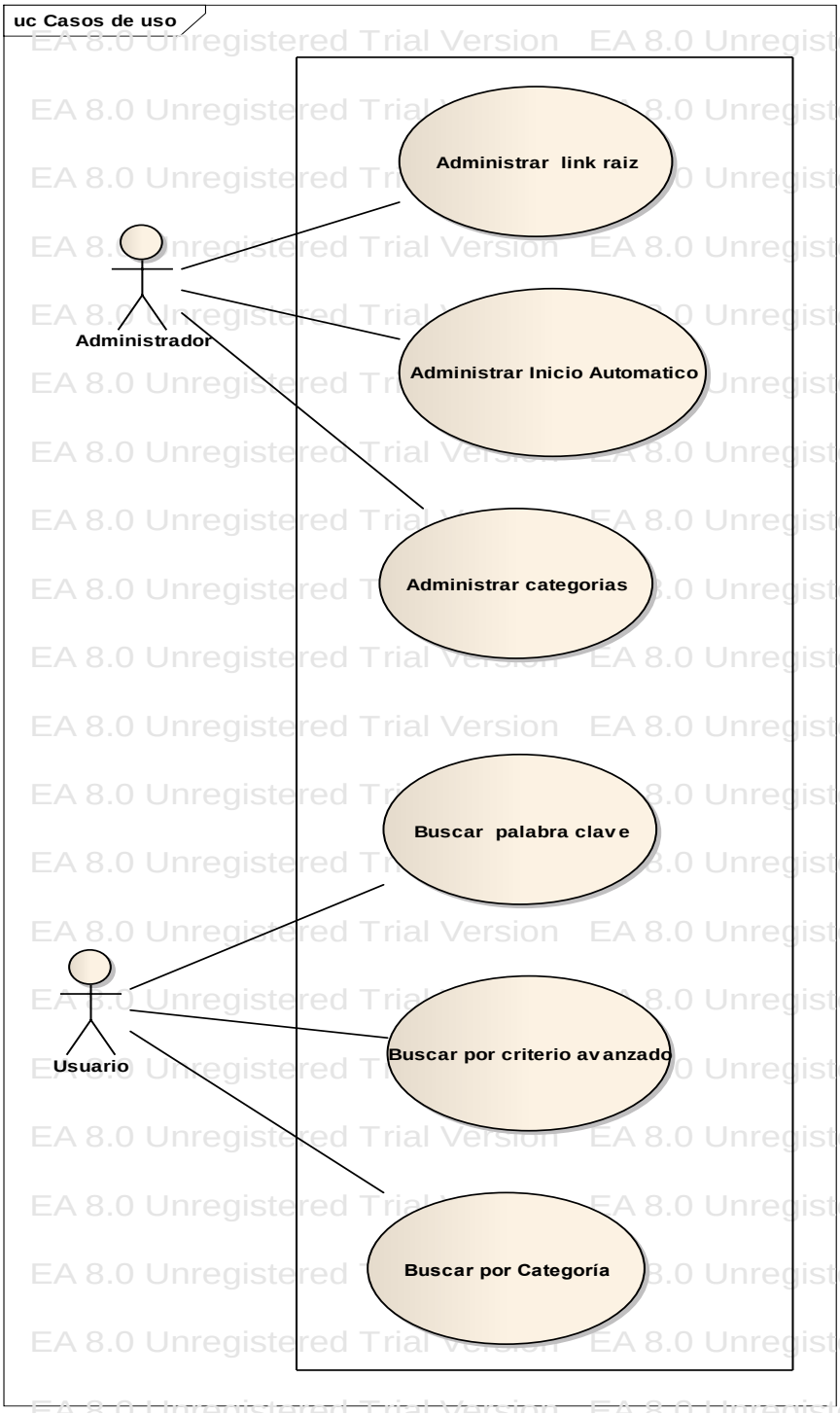
5.5 TABLA PARA LOS CASOS DE USO

TABLA PARA LOS CASOS DE USO		
ACTOR	**META**	**CASO DE USO**
Usuario	Ingresar al Buscador. Digitar la palabra clave. Ejecutar la búsqueda. Seleccionar categoría. Seleccionar búsqueda Avanzada.	Buscar palabra clave Buscar categoría Buscar avanzada
Administrador	Ingresar al programa Digitar el usuario y la calve de la cuenta de la Base de datos Digitar un Urls raíz Eliminar el Urls raíz Guardar Urls raíz Agregar horario Eliminar Horario Guardar Horario Digitar categoría Eliminar categoría Guardar categoría Ejecutar el Rastreador UNL Iniciar	Ingresar Link Raiz Administrar Horario Administrar Categorias

5.6. TABLA DE REFERENCIA DE CASOS DE USO

Nombre Caso de Uso	Referencia de Requerimientos
➢ **Ingresar Link raiz**	RF001, RF002,RF003,RF007, RF008, RF009
➢ **Administrar Horario**	RF011,
➢ **Administrar categoria**	RF007, RF008,
➢ **Buscar por palabra clave**	RF009, RF010
➢ **Buscar por criterio avanzado**	RF005
➢ **Buscar por categorías**	RF004

5.7. DIAGRAMA DE CASOS DE USO



FASE 2
DISEÑO

6. DISEÑO

Figura 6.1: Pantalla Ingreso al Rastreador-Unl

Figura 6.2:Pantalla de Entorno al Rastreador-Unl

6.1. DESCRIPCIÓN DE CASOS DE USO

6.1.1. CASO DE USO: INGRESAR LINKS RAICES

Nombre:	**Administrar link raíz**
Actor(es):	Administrador
Propósito:	Permitir agregar un nuevo link , guardar y eliminar.
Visión General	Proporcionar al administrador una forma fácil y sencilla de acceder a la aplicación, para su respectiva modificación, ingresar o eliminar un link.
Tipo:	Primario, esencial
Referencias:	RF001, RF002, RF003, RF007, RF008
Precondición(es)	El administrador(a) haya ingresado al programa rastreador.
Post-condiciones(es)	Presentar los links indexados

CURSO NORMAL DE EVENTOS

Programador	**Sistema**
1. El administrador presiona el botón **[Agregar Raices]** de la pantalla[**Rastreador UNL**]	**2.** El sistema presenta los campos de ingreso
3. El administrador digita el sitio en el campo [**raiz**] de la pantalla **Link Raiz**	
4. El administrador presiona el botón **[Guardar]** de la pantalla **Link Raíz**	
7.El administrador presiona el botón **[Aceptar]** de la pantalla **Link Raíz**	5.El Sistema presenta un mensaje de confirmación “el nuevo link se creó con satisfacción”.

	8.El sistema presenta el link agregado en la pantalla **[Rastreador UNL]**.
CURSO ALTERNO DE EVENTOS	
A.- EDITAR LINK RAÍZ	
A1. El Administrador selecciona el link ingresado haciendo clic en el botón **[editar]** de la pantalla Link Raíz A2. El sistema presenta el link en el campo raíz de la pantalla **Link Raíz** A3. El Administrador pulsa el botón aceptar de la pantalla **Link Raíz** A4. El sistema presenta un mensaje de confirmación "la actualización se realizo con éxito" A4. El sistema presenta el link modificado en la pantalla **Rastreador UNL**	
B.- ELIMINAR LINK RAÍZ	
B1. El Administrador selecciona el link y hace un clic en el botón **[eliminar]** de la pantalla Link Raíz B2. El sistema presenta un mensaje de confirmación "¿Está seguro que desea eliminar el link raíz?" B3. El Administrador acepta el mensaje de confirmación B4. El sistema presenta los resultados	

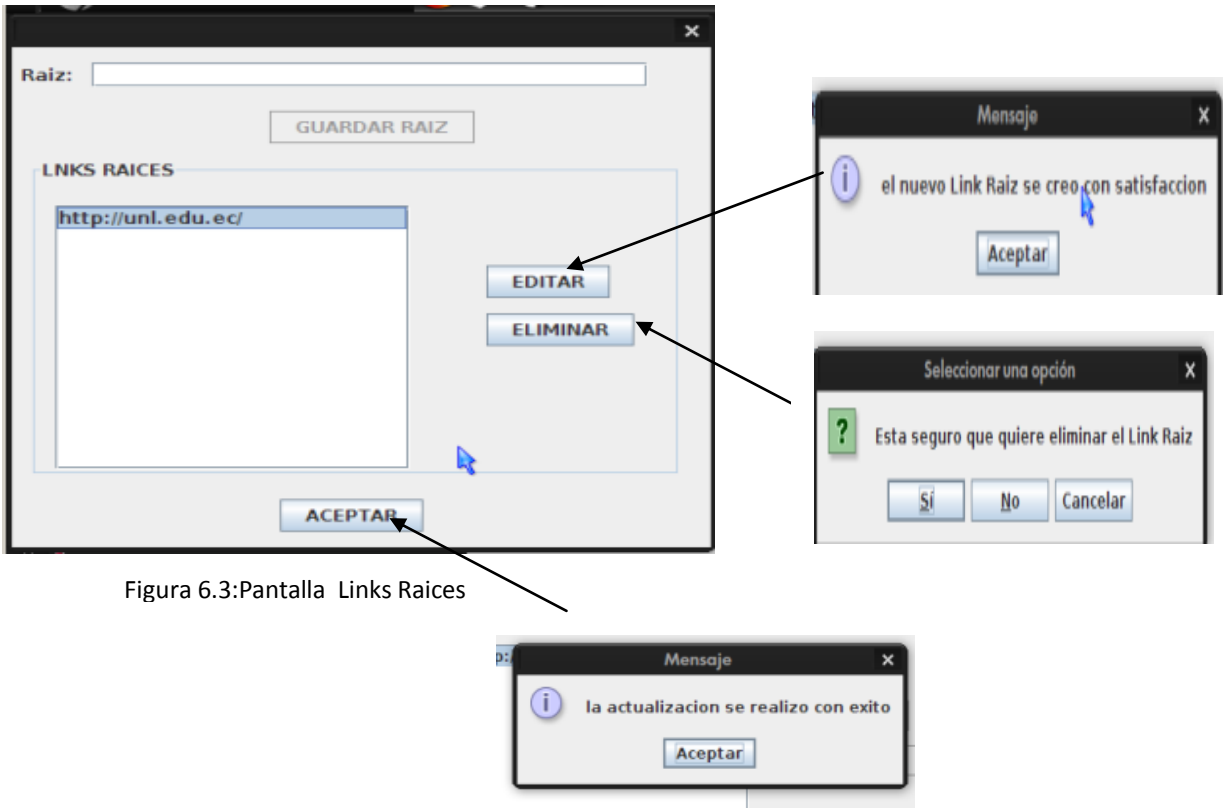
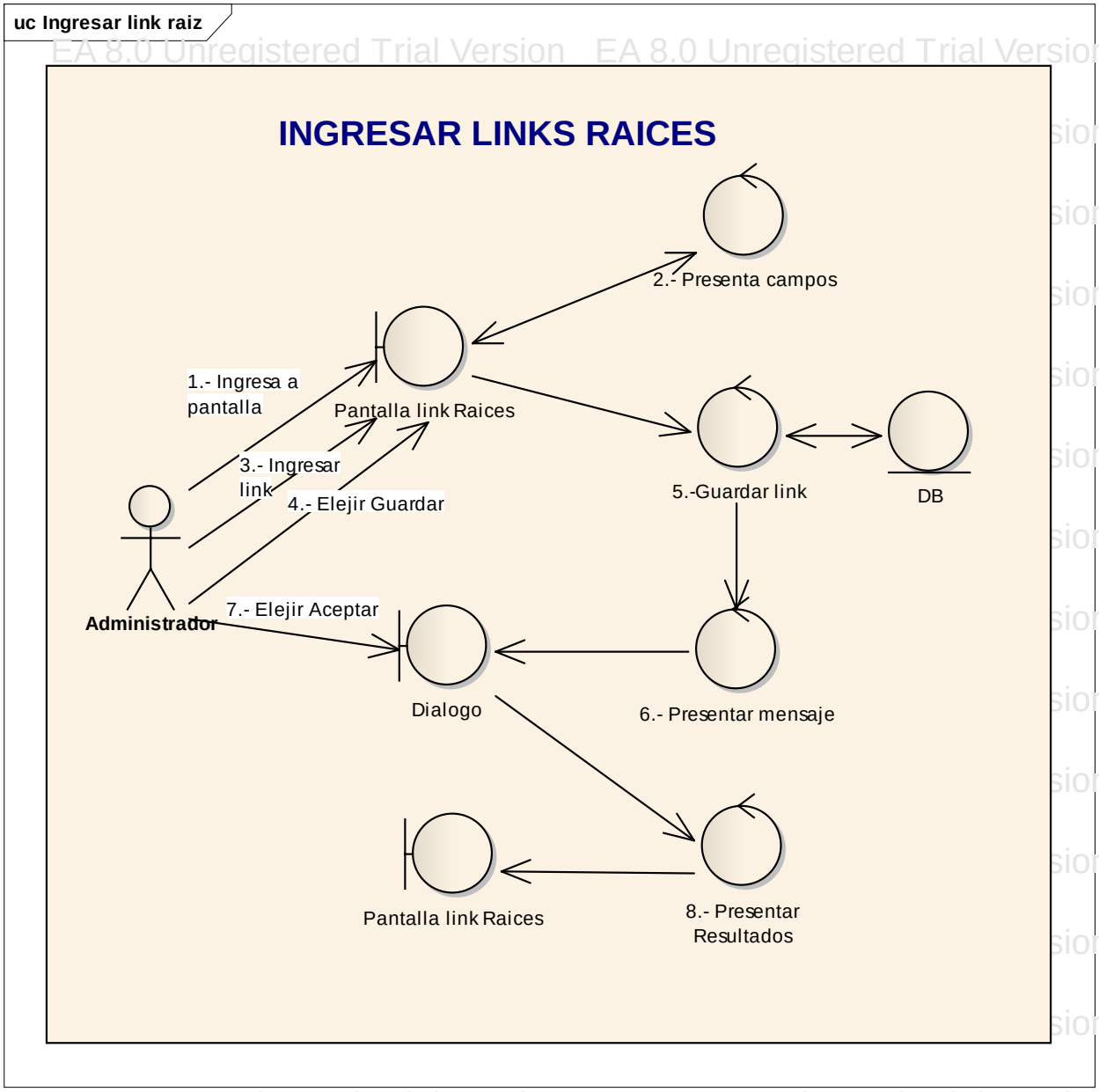


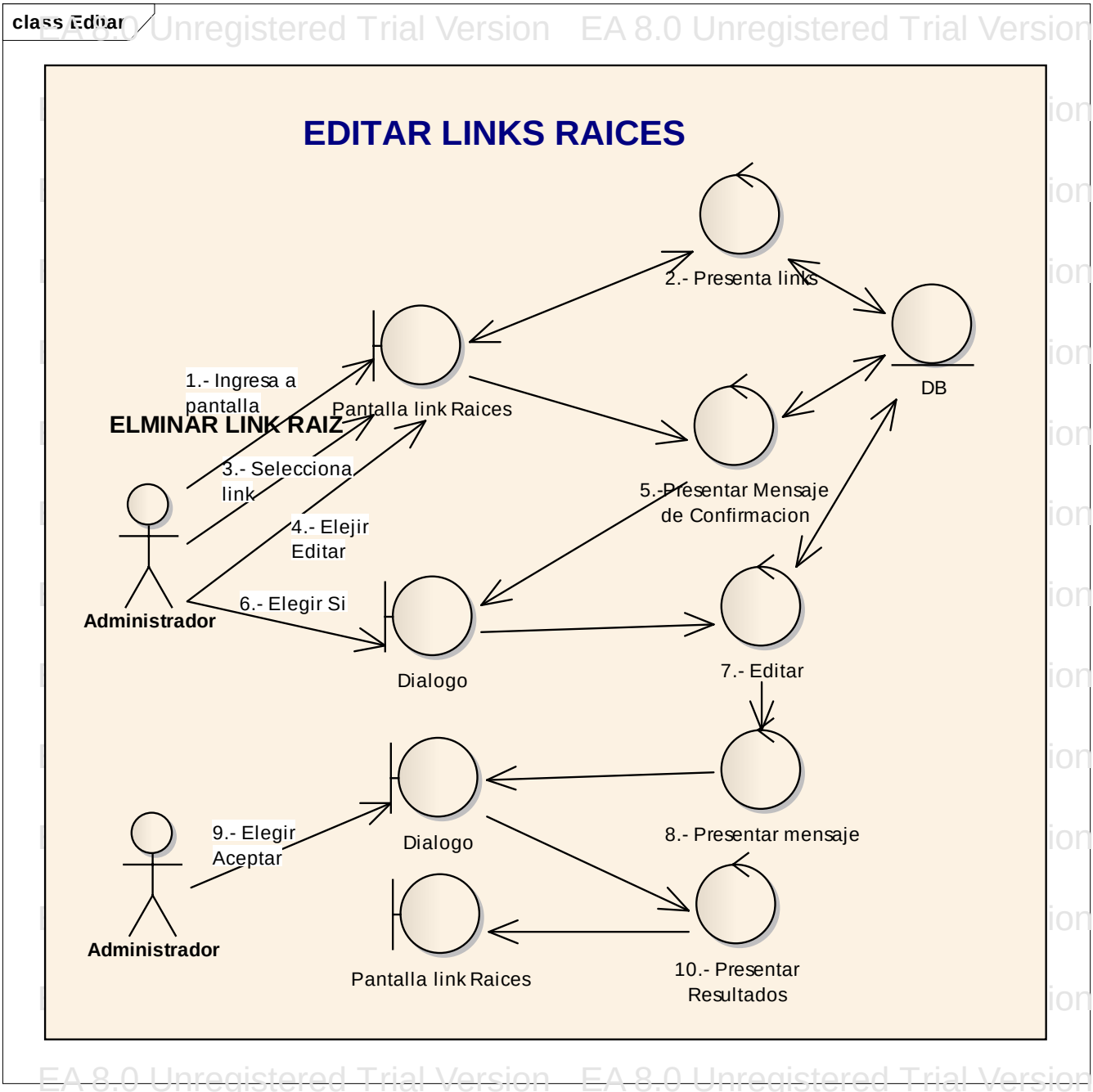
Figura 6.3:Pantalla Links Raices

CURSO NORMAL DE EVENTOS

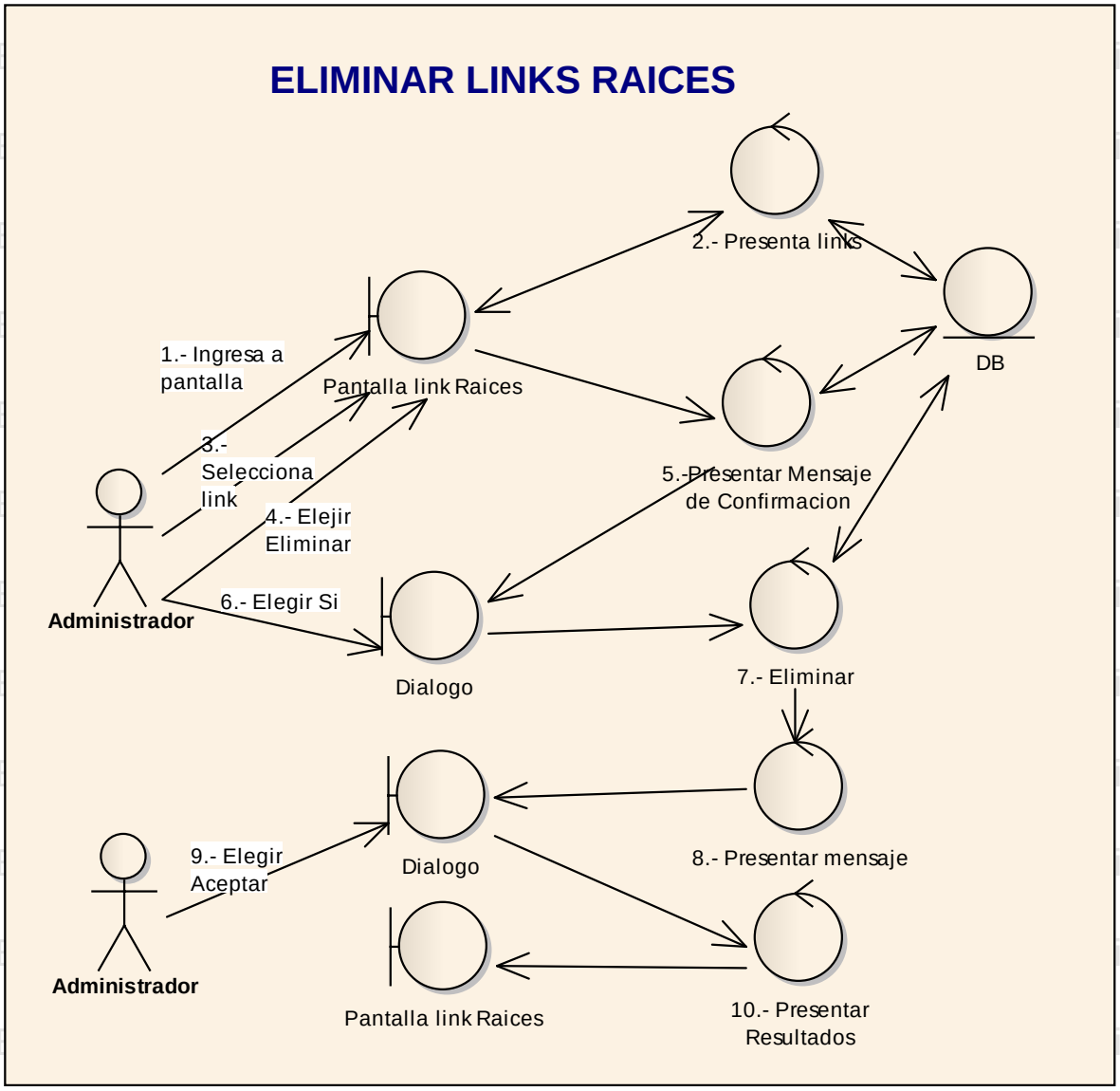


CURSO ALTERNO

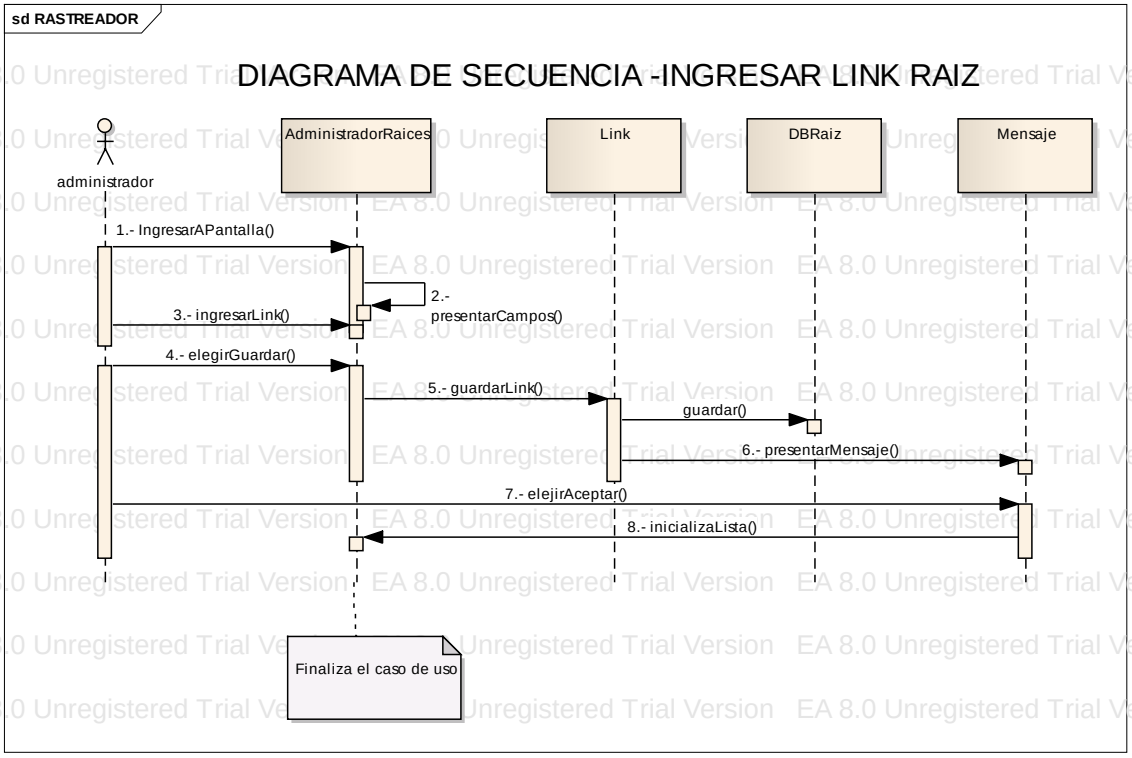
EDITAR LINK RAIZ

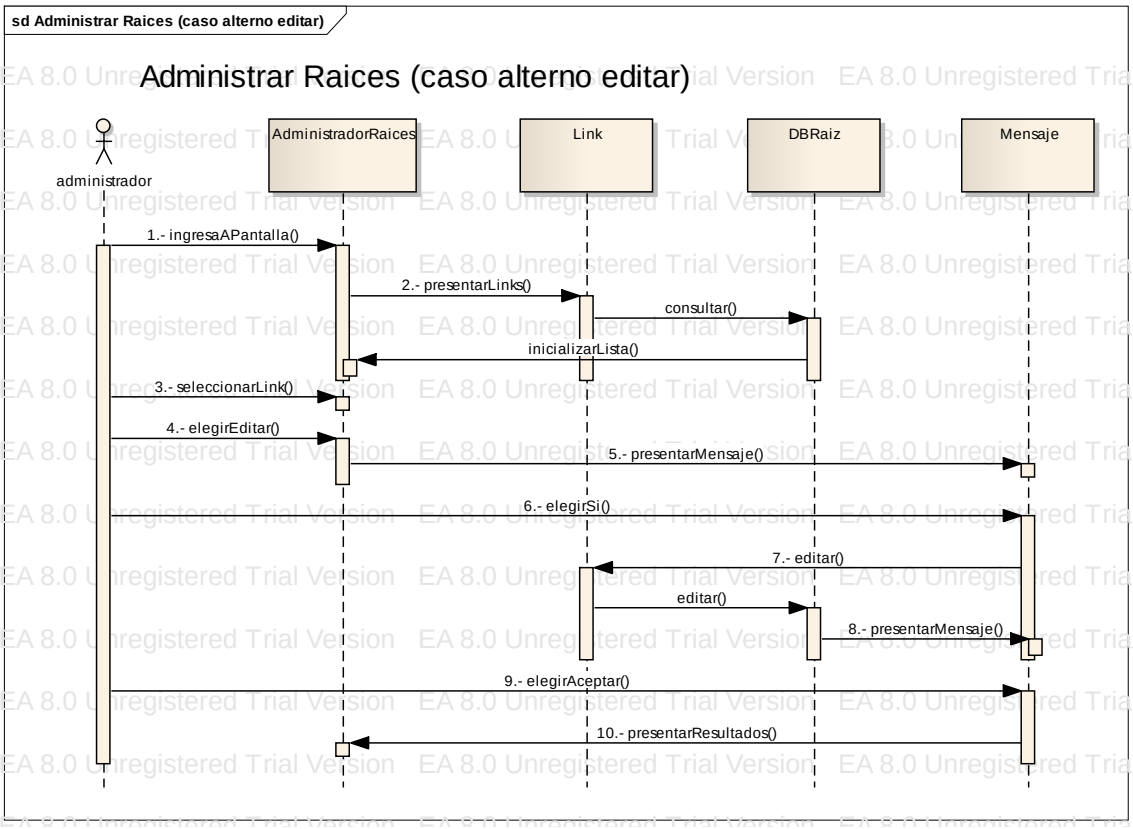


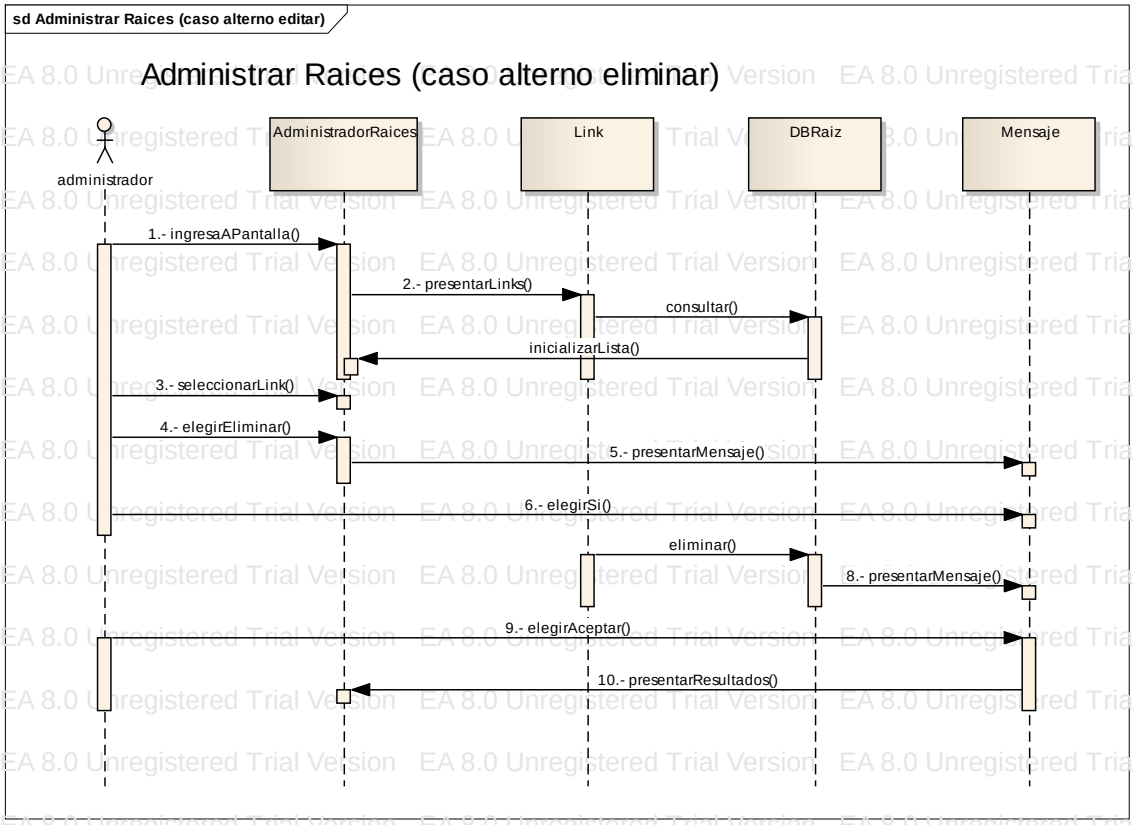
class Eliminar



ADMINISTRAR LINK RAIZ






sd Administrar Raices (caso alterno editar)
Administrar Raices (caso alterno eliminar)
administrador
AdministradorRaices
Link
DBRaiz
Mensaje
1.- ingresaAPantalla()
2.- presentarLinks()
consultar()
inicializarLista()
3.- seleccionarLink()
4.- elegirEliminar()
5.- presentarMensaje()
6.- elegirSi()
eliminar()
8.- presentarMensaje()
9.- elegirAceptar()
10.- presentarResultados()

6.1.2. CASO DE USO: ADMINISTRAR INICIO AUTOMATICO

Nombre:	**Administrar Inicio Automático**
Actor(es):	Administrador
Propósito:	Permitir seleccionar un horario de indexación.
Visión General	Proporcionar al administrador una forma fácil de trabajar para seleccionar un horario, fecha de inicio y finalización de indexación y guarda la programación.
Tipo:	Primario, esencial
Referencias:	RF11
Precondición(es)	El Administrador(a) haya ingresado al programa Rastreador U.N.L
Post-condiciones(es)	Activar el botón **[Iniciar]** de la pantalla Rastreador Unl

CURSO NORMAL DE EVENTOS	
Programador	**Sistema**
1. El administrador presiona el botón **[Configurar Automático]** de la pantalla[**Rastreador UNL**]	
	2. El sistema presenta los campos de ingreso
3.-El administrador selecciona el día y la hora que debe realizarse la indexación, en la pantalla **Horario**	
4. El Administrador pulsa el botón **[Agregar horario]** de la pantalla **Horario**	
	5.El Sistema presenta el horario
6. El Administrador selecciona el número	

de días a actualizar el rastreador, presionando el botón **[Periodo de Actualización]** de la pantalla **Horario**. 7. El Administrador presiona el botón **[Guardar Programación]** de la pantalla **Horario**	seleccionado 8.El sistema presenta el periodo de actualización en la pantalla **Rastreador Unl**
CURSO ALTERNO DE EVENTOS	
A.- ELIMINAR HORARIO.	
A1. El Administrador selecciona el horario incorrecto y lo elimina haciendo clic en el botón **[Eliminar Horario]** de la pantalla **Horario** A2. El caso de uso continua en el paso Nro.7 del curso normal de eventos	

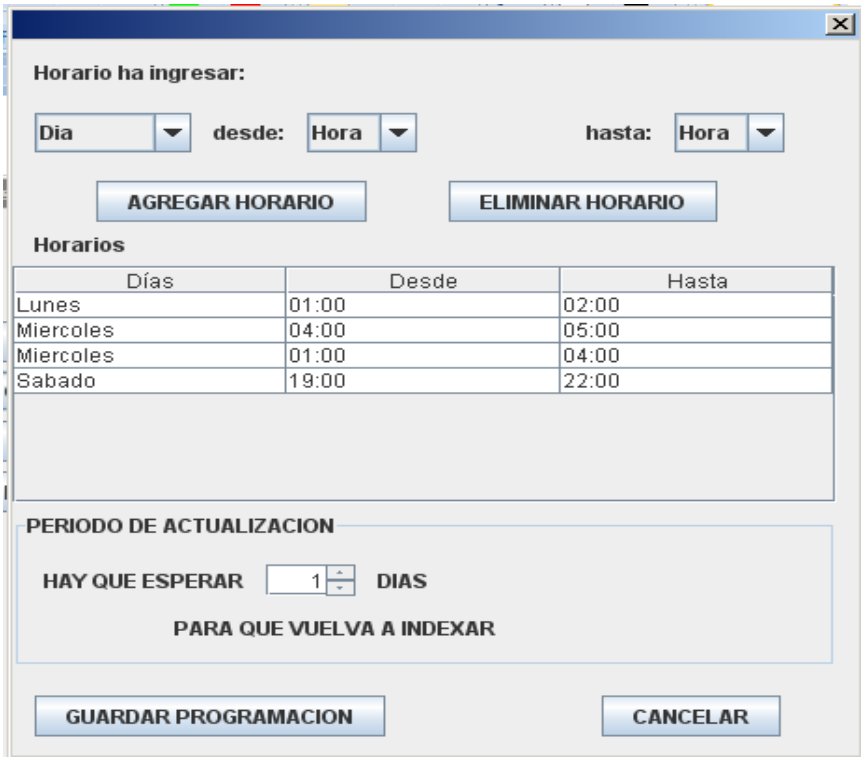
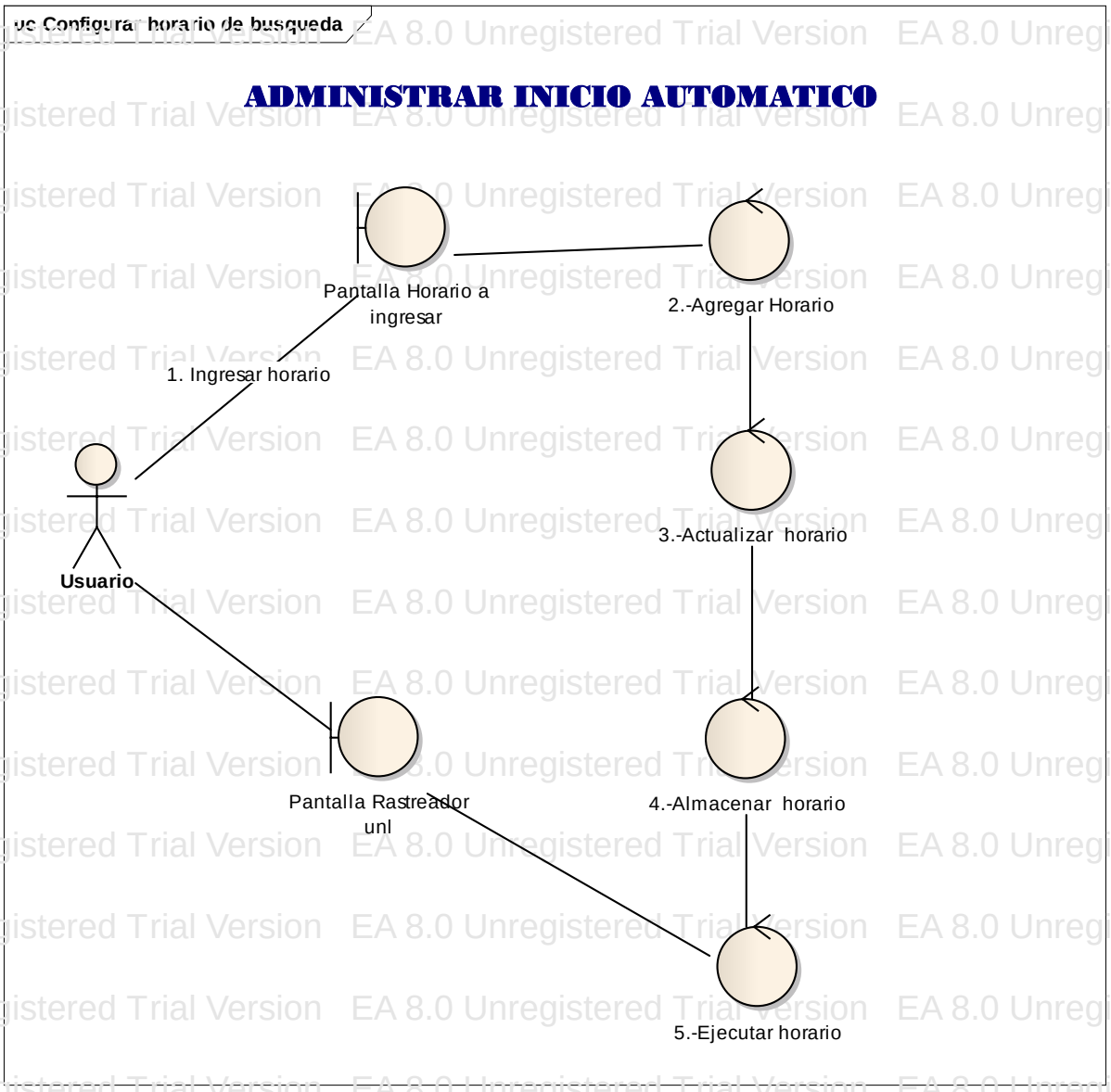


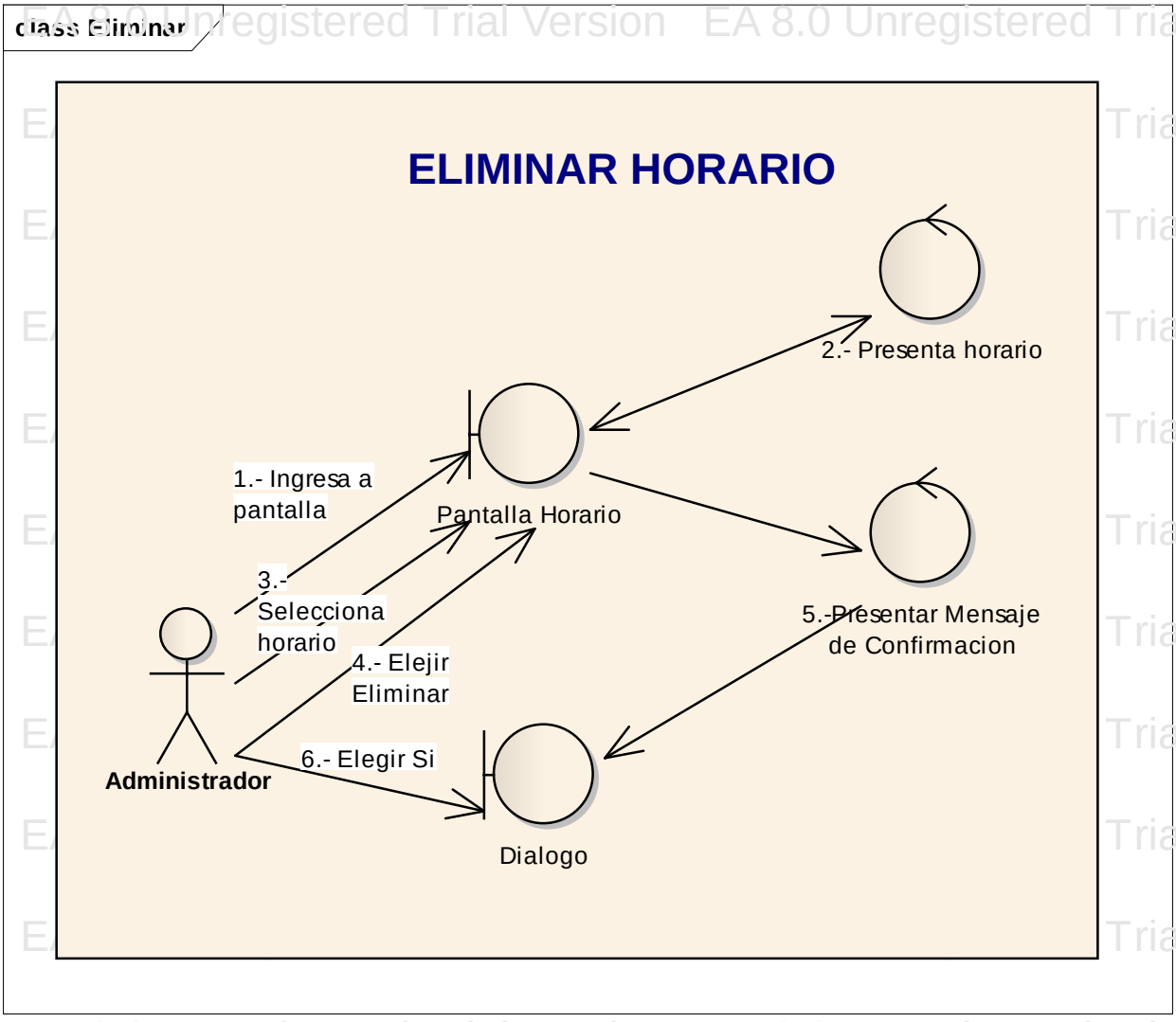
Figura 6.4:Administrar Inicio Automático

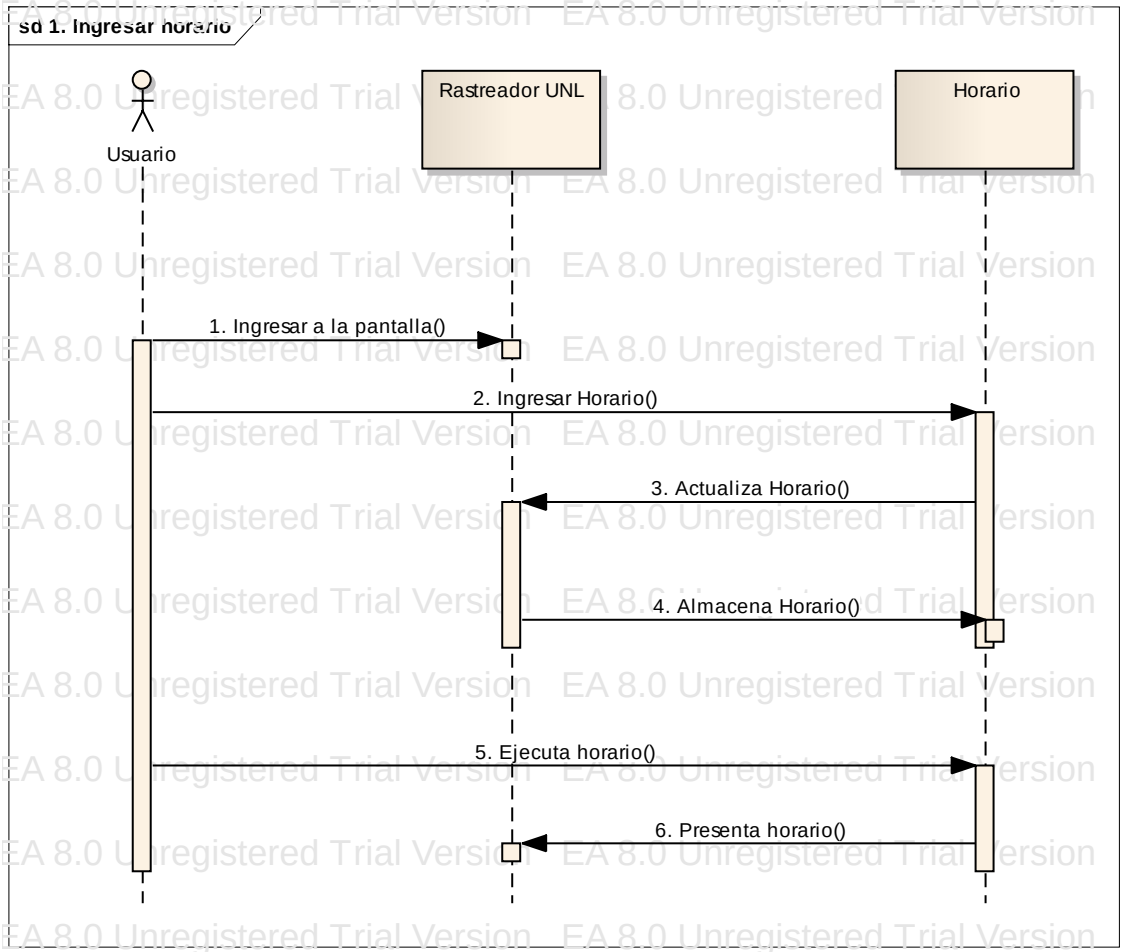
CURSO NORMAL DE EVENTOS



CURSO ALTERNO DE EVENTOS

ELIMINAR HORARIO





6.1.3. CASO DE USO: ADMINISTRAR CATEGORIAS

Nombre:	**Administrar Categorías**
Actor(es):	Administrador
Propósito:	El Administrador ingresa una categoría, guarda, edita y elimina una categoría.
Visión General	Proporcionar al administrador una forma fácil de trabajar para agregar, editar o eliminar una categoría, finalmente guarda la programación
Tipo:	Primario, esencial
Referencias:	RF007, RF008
Precondición(es)	El Administrador(a) haya ingresado al programa Rastreador U.N.L

Post-condiciones(es)	Presenta la categoría seleccionada

CURSO NORMAL DE EVENTOS

Programador	**Sistema**
1.El Administrador presiona el botón **[Administrar categorías]** de la pantalla **Rastreador UNL**	
	2. El sistema presenta el campo categoría y sinónimos en la pantalla **Categorías**
3.El Administrador digita una categoría en el campo de texto[**TextBox**] de la pantalla [**Categoria**]	
4.El administrador ingresa los sinónimos de la categoría en el campo de texto[**TextBox**]	
5. El administrador pulsa el botón **[Guardar Categoría]** de la pantalla [**Categoría**]	
	6. El sistema presenta un mensaje de confirmación “la nueva categoría se creó con satisfacción”
7. El Administrador acepta el mensaje de confirmación del cuadro de diálogo	
	8. El sistema presenta la nueva categoría en la pantalla **Categoría**
9. El Administrador pulsa el botón [**Aceptar**] de la pantalla **Categoría**	
	10.El Sistema retorna a la pantalla [**Rastreador UNL**]

CURSO ALTERNO DE EVENTOS	
A.- EDITAR CATEGORIA.	
A1. El Administrador selecciona la categoría y hace clic en el botón **[editar]** de la pantalla **Categorías** A2. El sistema presenta la categoría en los campos **[Categorías]** y **[Sinónimos]** de la pantalla **Categorías** A3. El Administrador pulsa el botón [**Guardar Categoría]** A4. El sistema presenta un mensaje de confirmación “la actualización se realizo con éxito” A5. El sistema presenta la categoría modificado en la pantalla **Categorías**	
B.- ELIMINAR CATEGORÍA	
B1. El Administrador selecciona la categoría y hace un clic en el botón **[eliminar]** de la pantalla **Categorias** B2. El sistema presenta un mensaje de confirmación “Está seguro que quiere eliminar esta categoría” B3. El Administrador selecciona una opción del cuadro de diálogo **[si] [no] [cancelar]** B4. El sistema presenta los resultados en la pantalla **[Categorias]**	

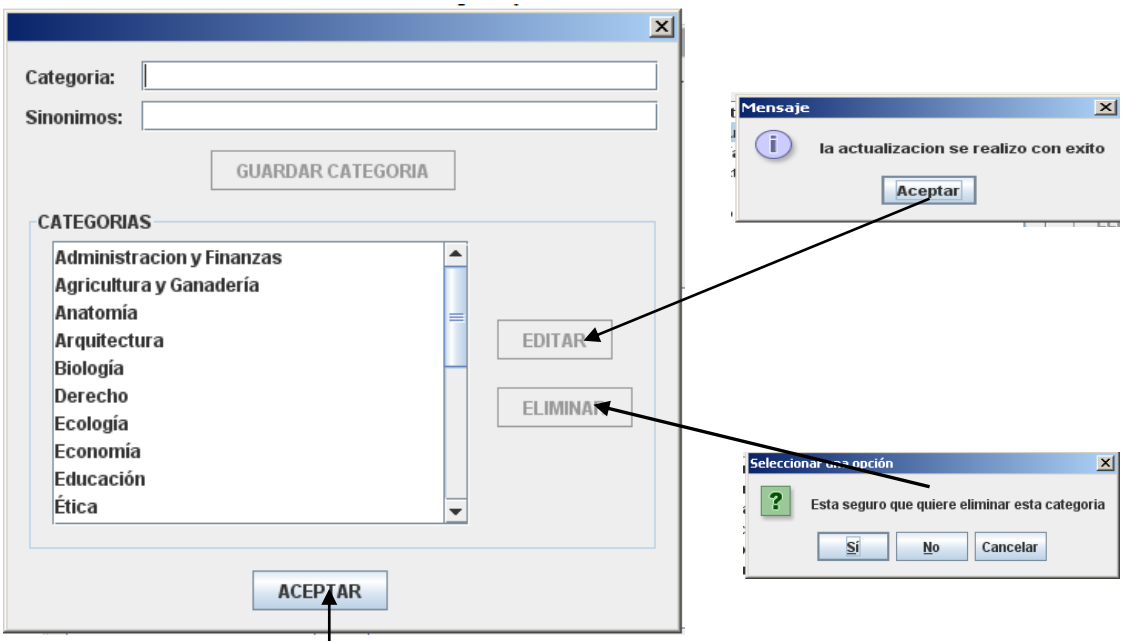
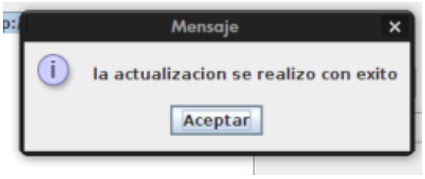
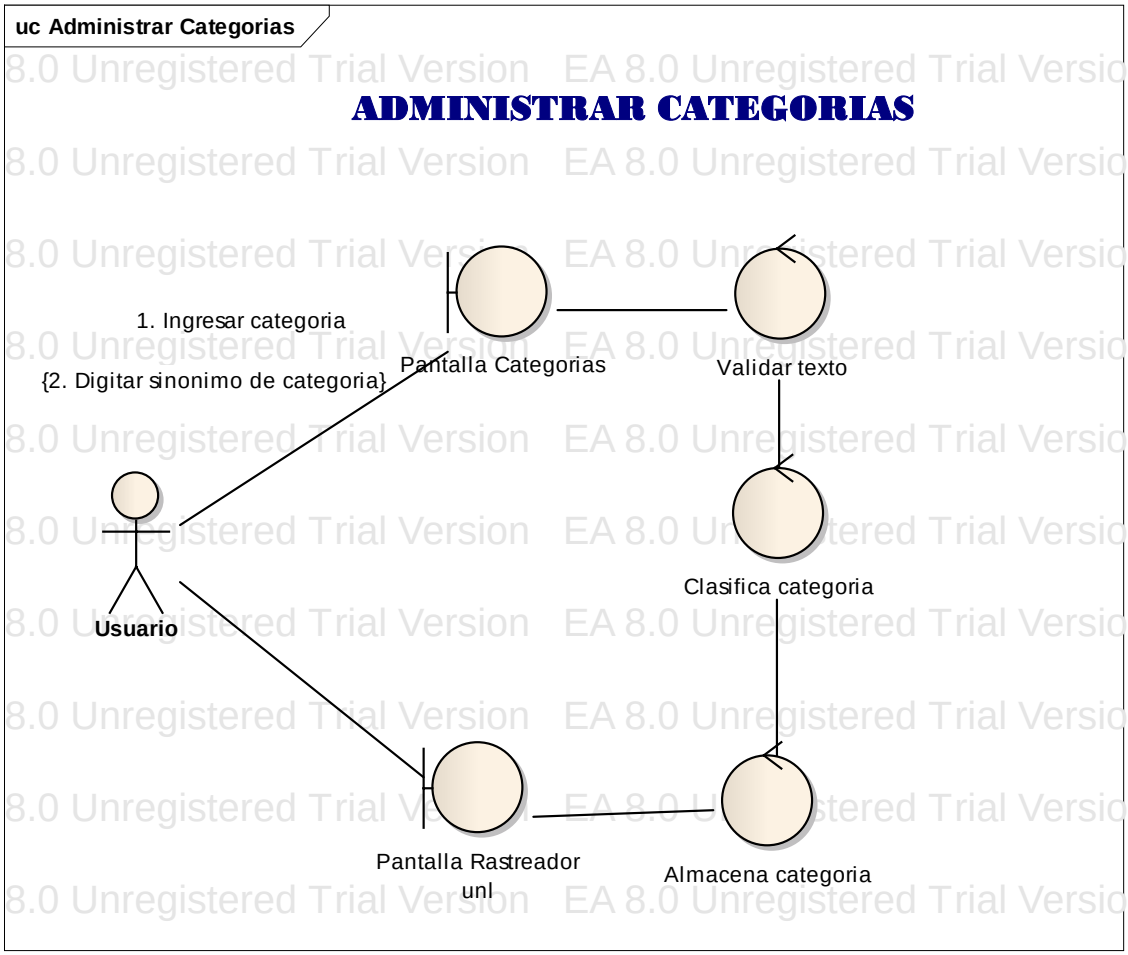
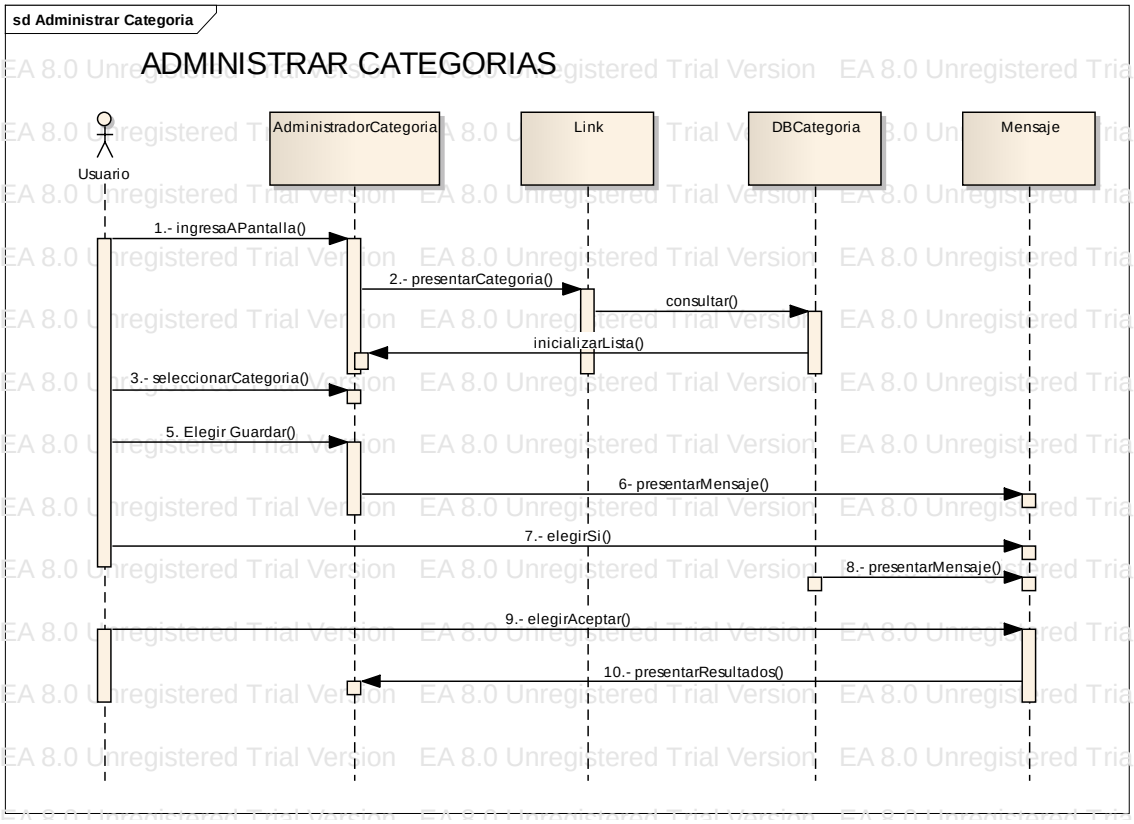


Figura 6.4:Administrar Categorias







6.1.4 **CASO DE USO: BUSCAR POR PALABRA CLAVE**

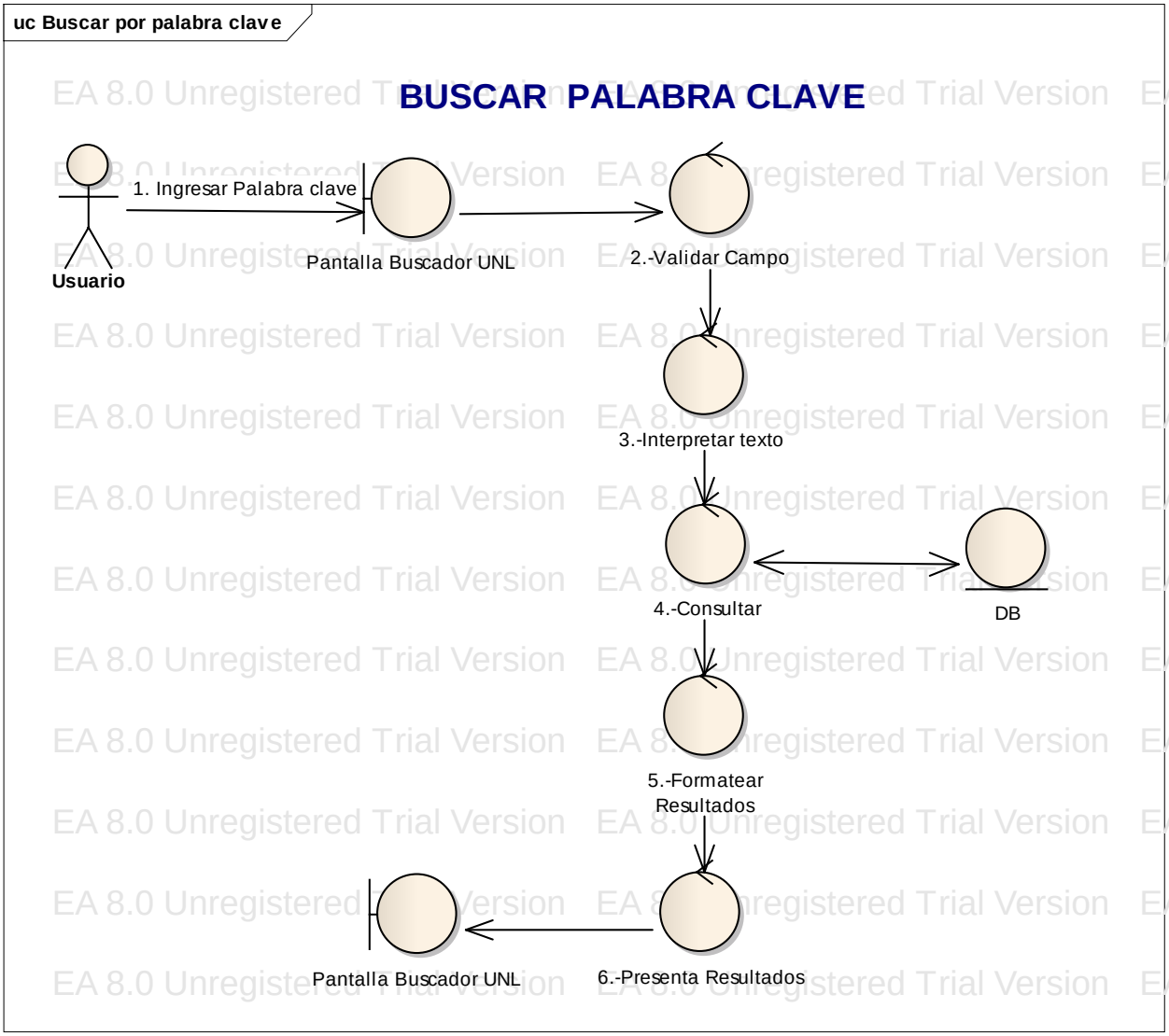
Nombre:	**Buscar por Palabra Clave**
Actor(es):	Usuario
Propósito:	El usuario ingresa la palabra clave y ejecuta la búsqueda
Visión General	Proporcionar al usuario una forma fácil de obtener información de acuerdo a sus requerimientos.
Tipo:	Primario, esencial
Referencias:	RF009, RF010
Precondición(es)	El usuario haya ingresado al programa Buscador
Post-condiciones(es)	Presenta los resultados de la búsqueda

CURSO NORMAL DE EVENTOS	
Programador	**Sistema**
1.El usuario ingresa a la pantalla	

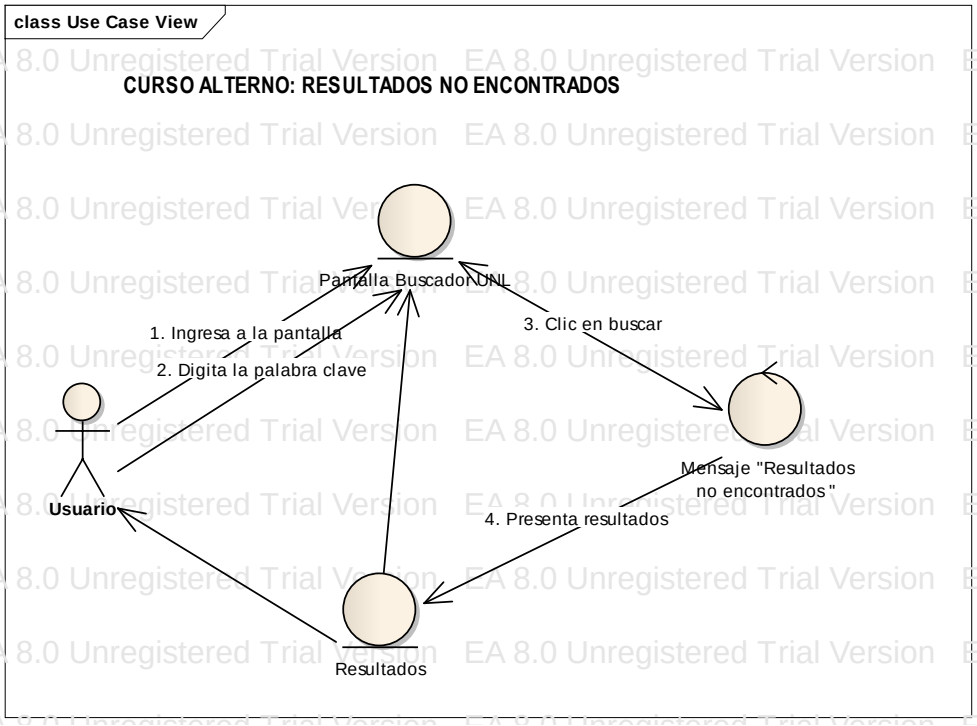
[Buscador] 3.El Usuario ingresa la palabra clave en el campo **[Texto]** de la pantalla **Buscador** 4. El usuario pulsa el botón **[Buscar]** de la pantalla **Buscador**	2. El sistema presenta el campo de búsqueda 5. El sistema recupera las URLs de la base de datos 6. El sistema presenta los resultados por grupos de items
CURSO ALTERNO DE EVENTOS	
A.- Resultados no Encontrados.	
A1. El sistema muestra un mensaje “Resultados no encontrados”	

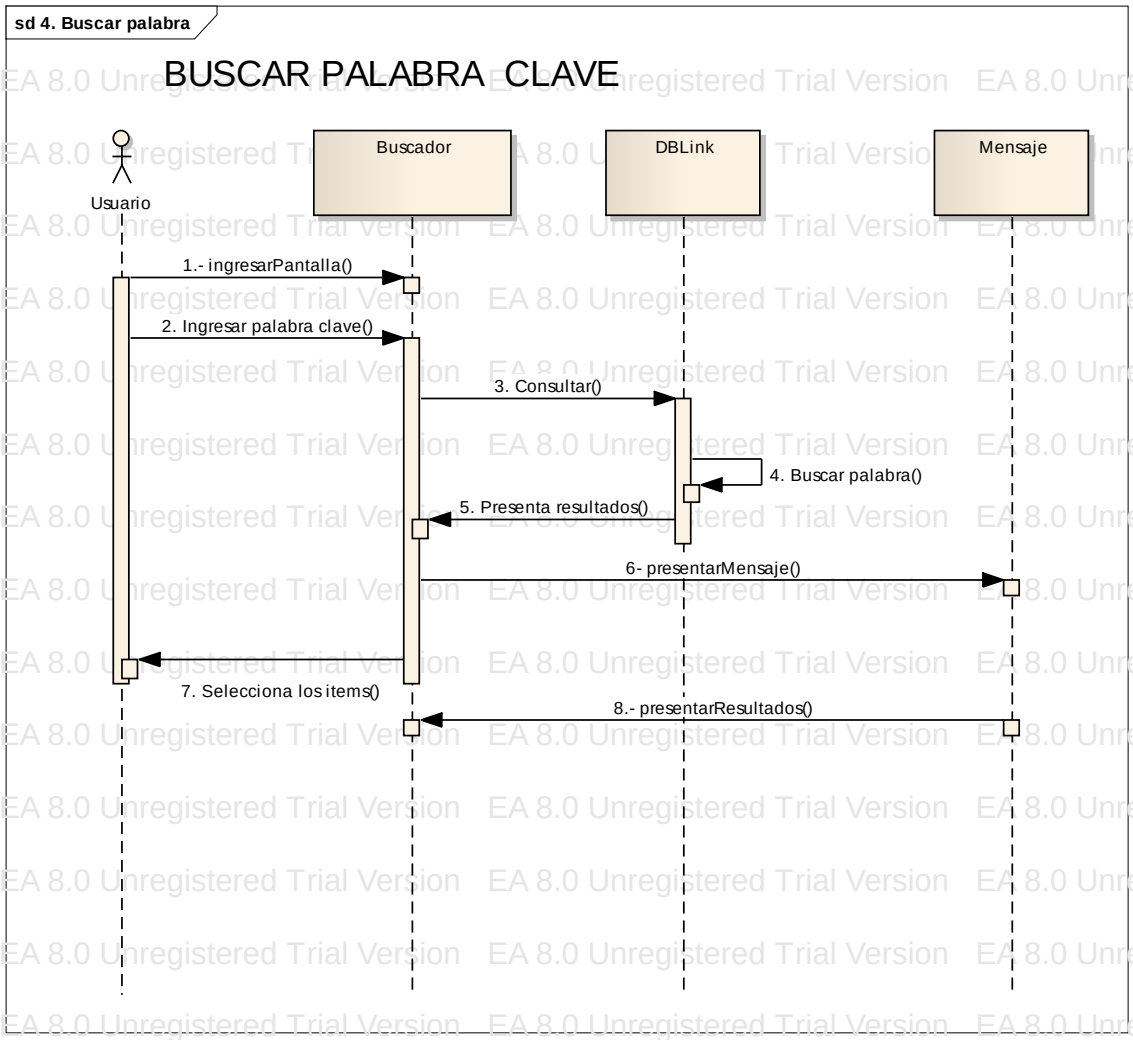
Figura 6.5:Buscar por palabra clave

CASO DE USO: BUSCAR POR PALABRA CLAVE



CURSO ALTERNO DE EVENTOS





6.1.5 CASO DE USO: BUSCAR POR CRITERIO DE BÚSQUEDA

Nombre:	**Buscar por Criterio de Búsqueda**
Actor(es):	Usuario
Propósito:	El usuario ingresa la palabra clave y selecciona un criterio de búsqueda
Visión General	Proporcionar al usuario una forma fácil de obtener información de manera más simplifica.
Tipo:	Primario, esencial
Referencias:	RF005
Precondición(es)	El usuario seleccione el criterio de búsqueda
Post-condiciones(es)	Presenta los resultados de acuerdo al criterio seleccionado

CURSO NORMAL DE EVENTOS

Programador	**Sistema**
1.El usuario ingresa a la pantalla **[Buscador]** **2.** El usuario pulsa el botón **[Búsqueda Avanzada]** de la pantalla **Buscador**	3. El sistema presenta el campo de búsqueda
4.El usuario digita la palabra clave a buscar en el campo [**Texto**] de la pantalla **Búsqueda Avanzada** 5. El usuario selecciona el criterio de búsqueda. ***5.1Frase completa:*** si selecciona esta opción el sistema presentara las	

páginas que tenga en su title o descripción las palabras de la frase a buscar. ***5.2Alguna de las palabras*** si selecciona esta opción el sistema presentara las páginas que tenga en su title o descripción alguna de las palabras a buscar. ***5.3Todas palabras*** si selecciona esta opción el sistema presentara las páginas que tenga en su title o descripción todas de las palabras a buscar. 6. El usuario pulsa el botón **[Buscar]**	7.El sistema recupera las URLs de la base de datos 8.El sistema presenta los resultados por grupos de items
CURSO ALTERNO DE EVENTOS	
A.- Resultados no Encontrados.	

A1. El sistema muestra un mensaje “Resultados no encontrados”

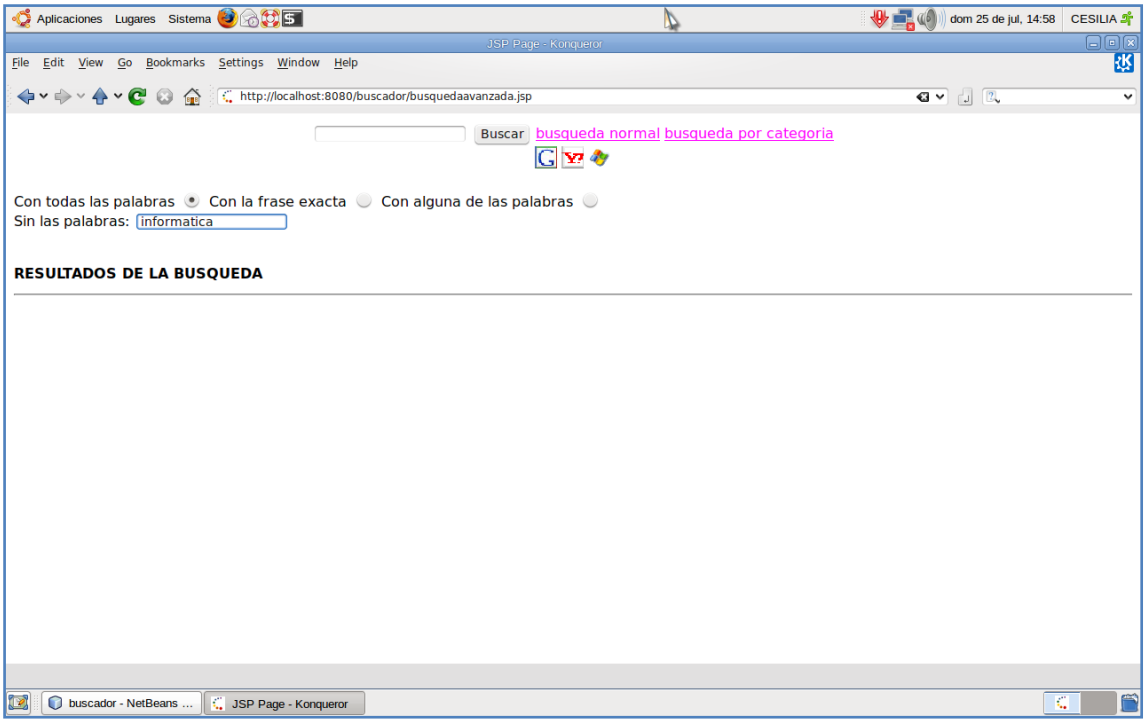
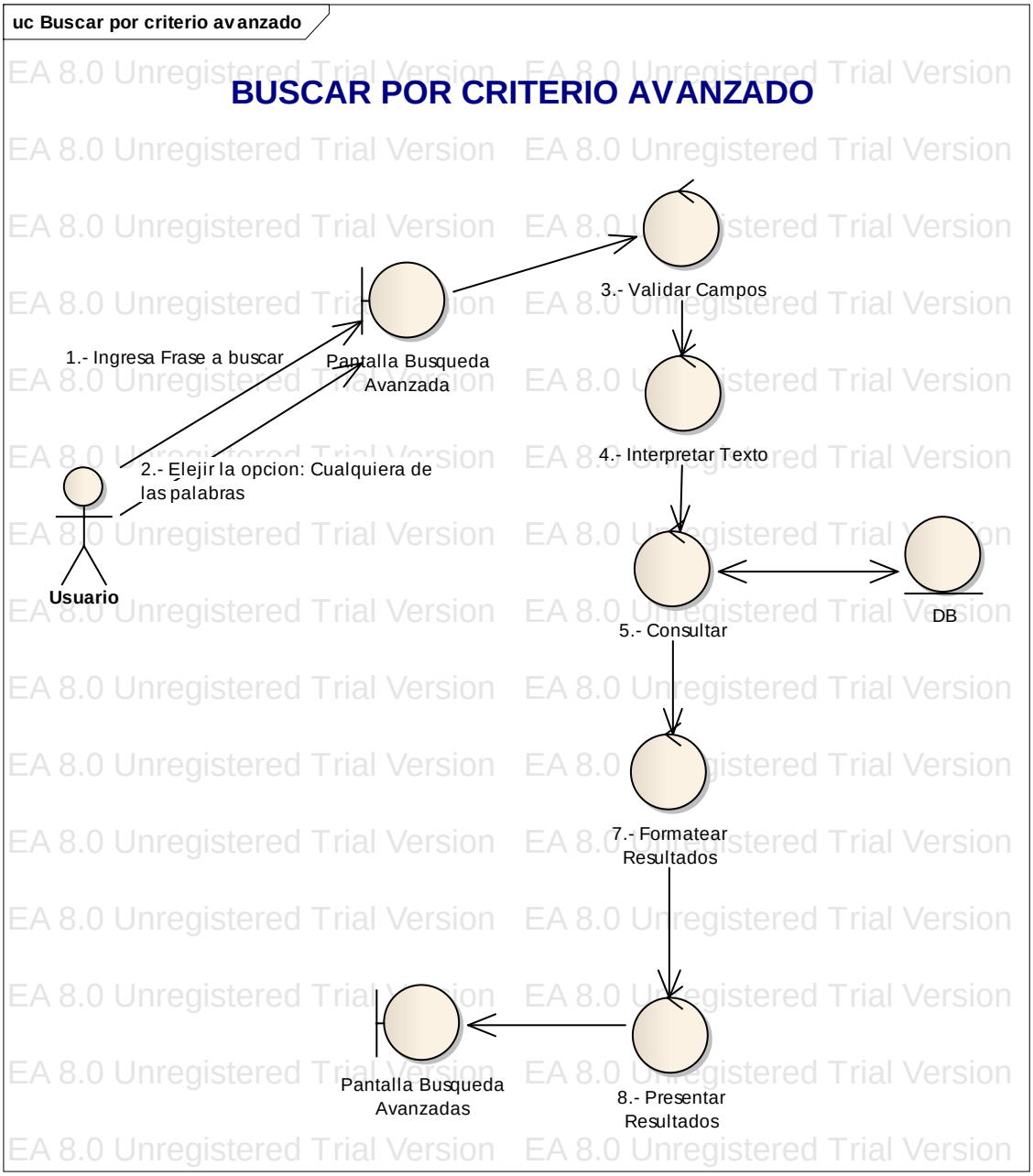
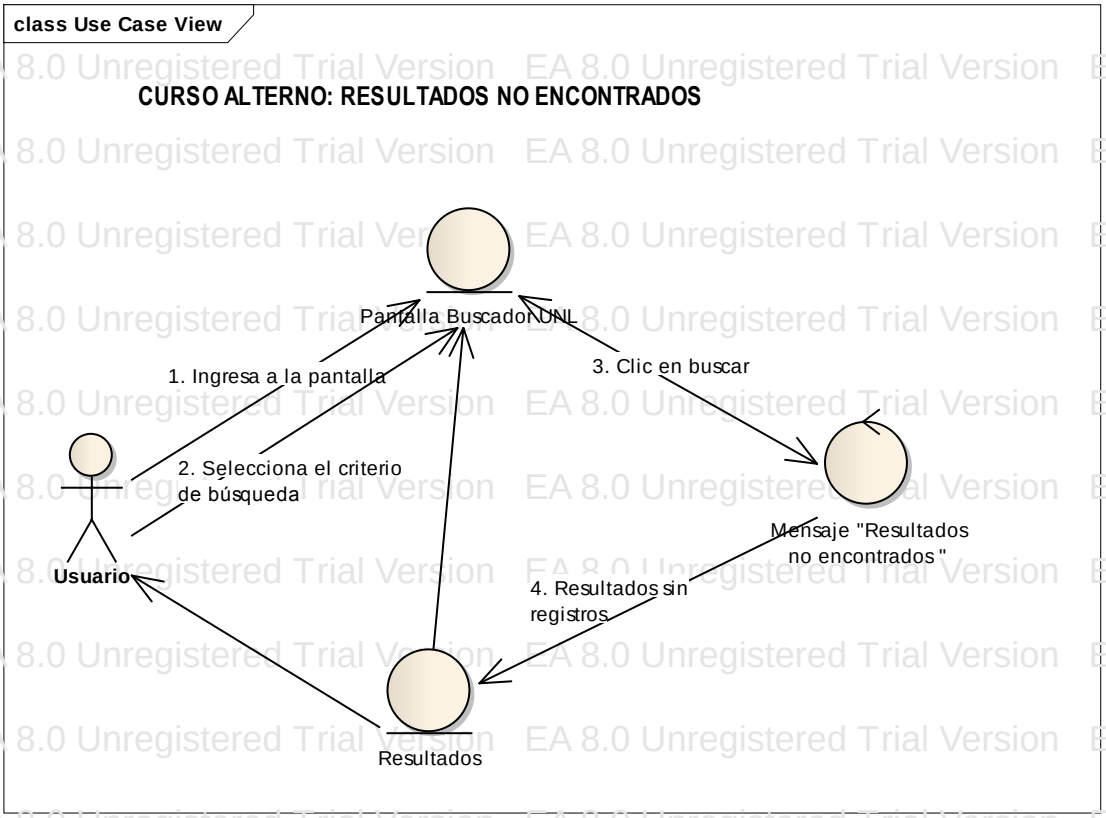


Figura 6.6:Buscar por criterio de Búsqueda





6.1.6 CASO DE USO: BÚSCAR POR CATEGORIAS

Nombre:	**Buscar por Categorías**
Actor(es):	Usuario
Propósito:	Obtener los resultados de acuerdo a la categoría seleccionada
Visión General	Proporcionar al usuario una forma fácil de obtener información por categoría.
Tipo:	Primario, esencial
Referencias:	RF004
Precondición(es)	El usuario seleccione la categoría
Post-condiciones(es)	Presenta los resultados de acuerdo a la categoría seleccionada
CURSO NORMAL DE EVENTOS	
Programador	**Sistema**

1.El usuario ingresa a la pantalla **[Buscador]** **2.** El usuario pulsa el botón **[Búsqueda por Categoría]** de la pantalla **Buscador**	3.El sistema presenta el campo de búsqueda
4.El usuario selecciona la categoría de la pantalla **Categorias** 5. El usuario pulsa el botón **[Buscar]**	6.El sistema recupera las URLs de la base de datos 7.El sistema muestra los resultado en grupos de ítems de acuerdo a la categoría seleccionada
CURSO ALTERNO DE EVENTOS	
A.- Resultados no Encontrados.	
A1. El sistema muestra un mensaje “No existe categoría”	

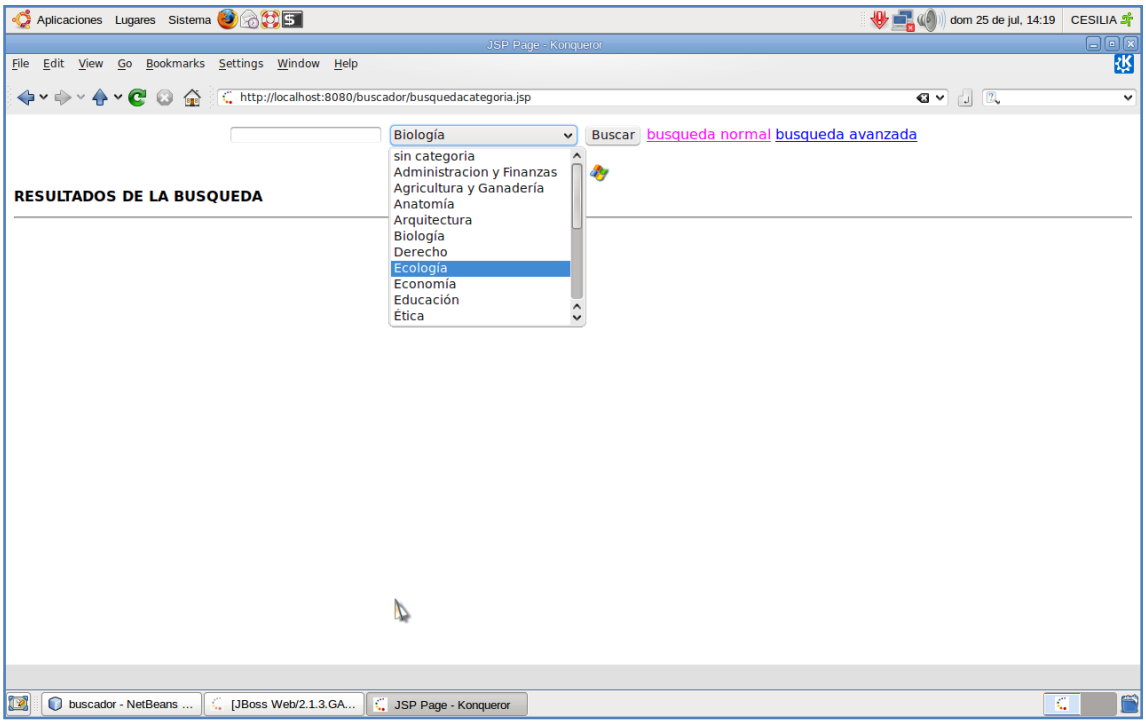
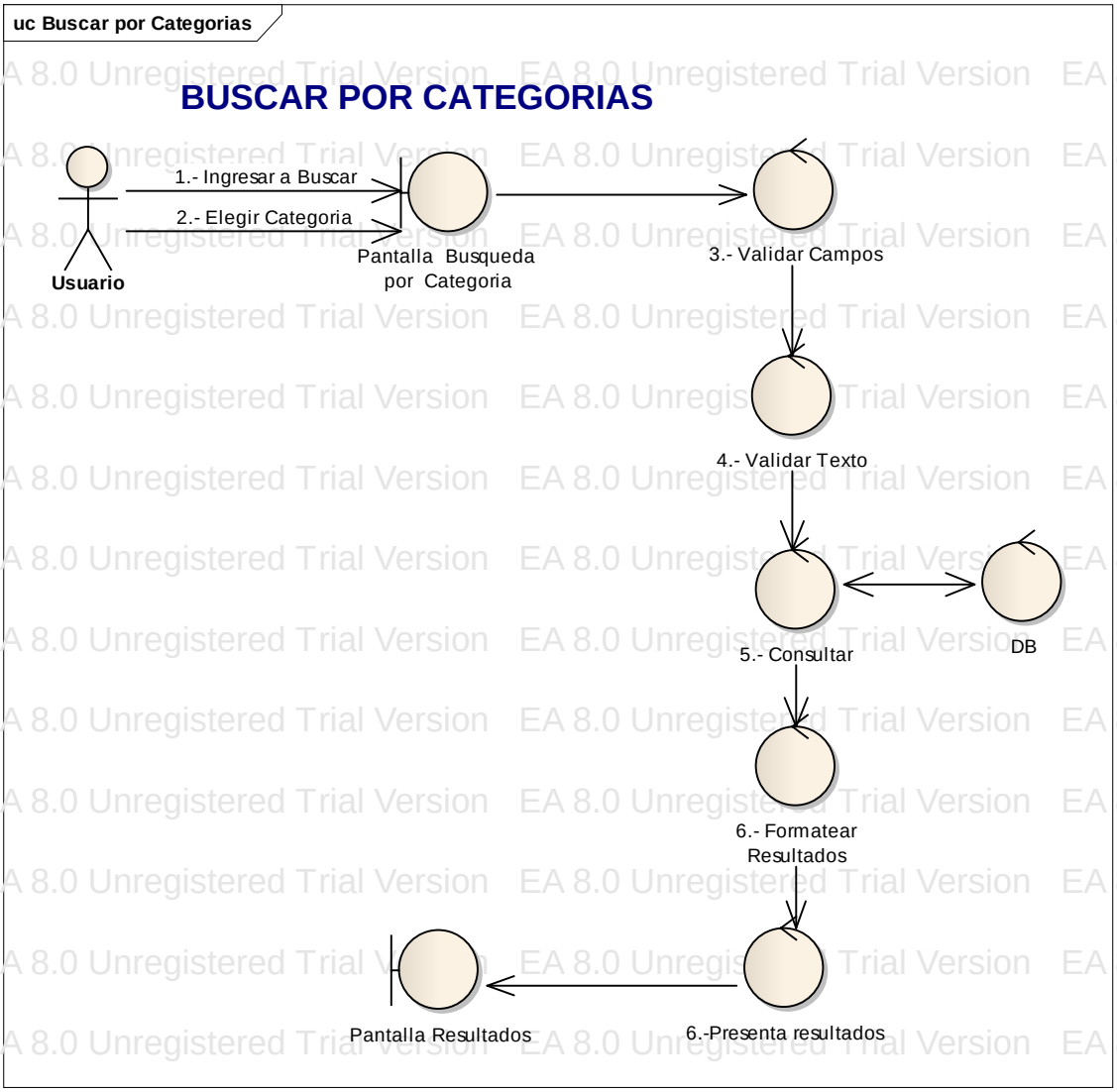
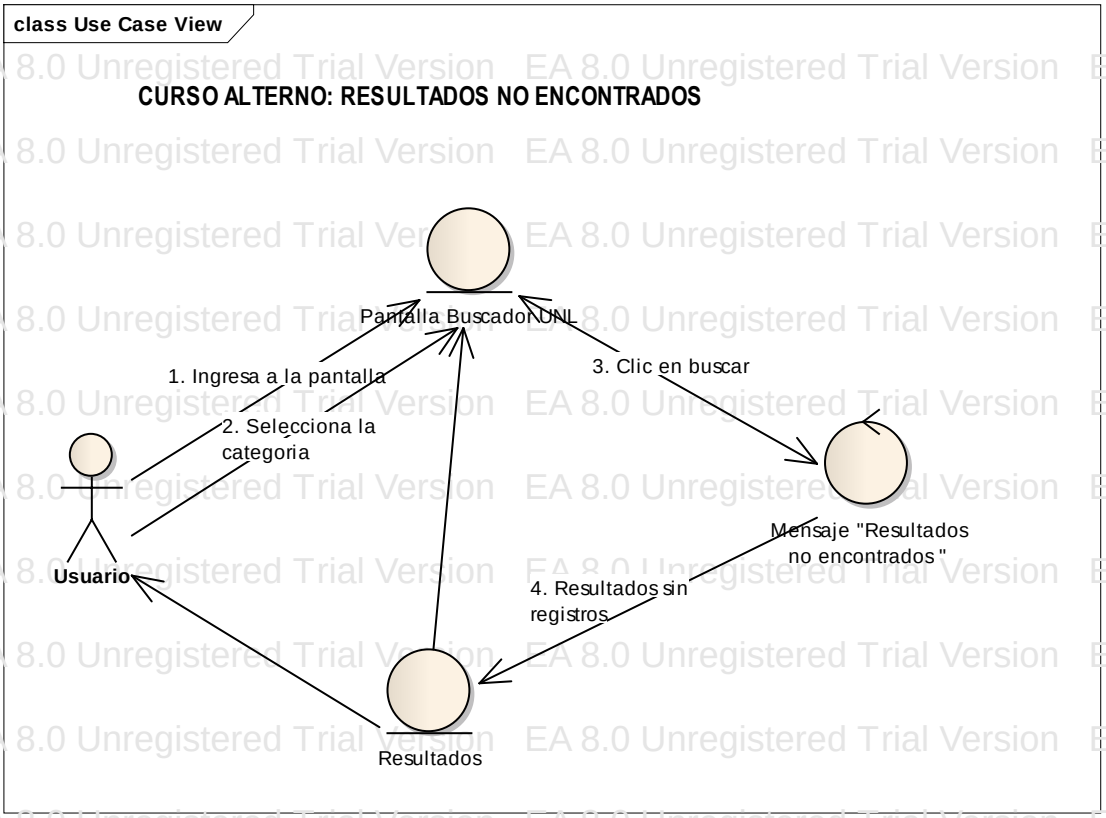
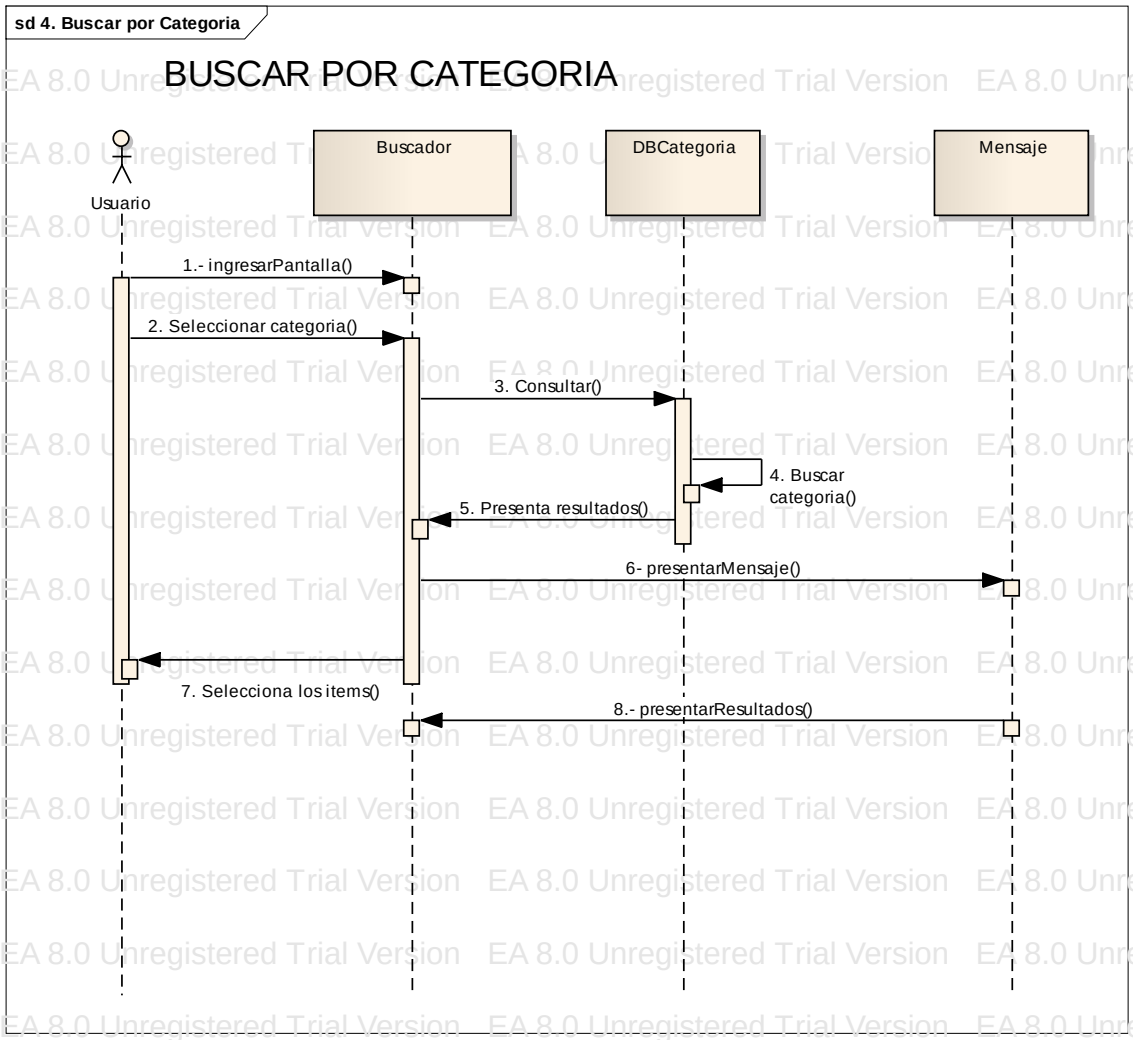


Figura 6.7:Buscar por Categoria

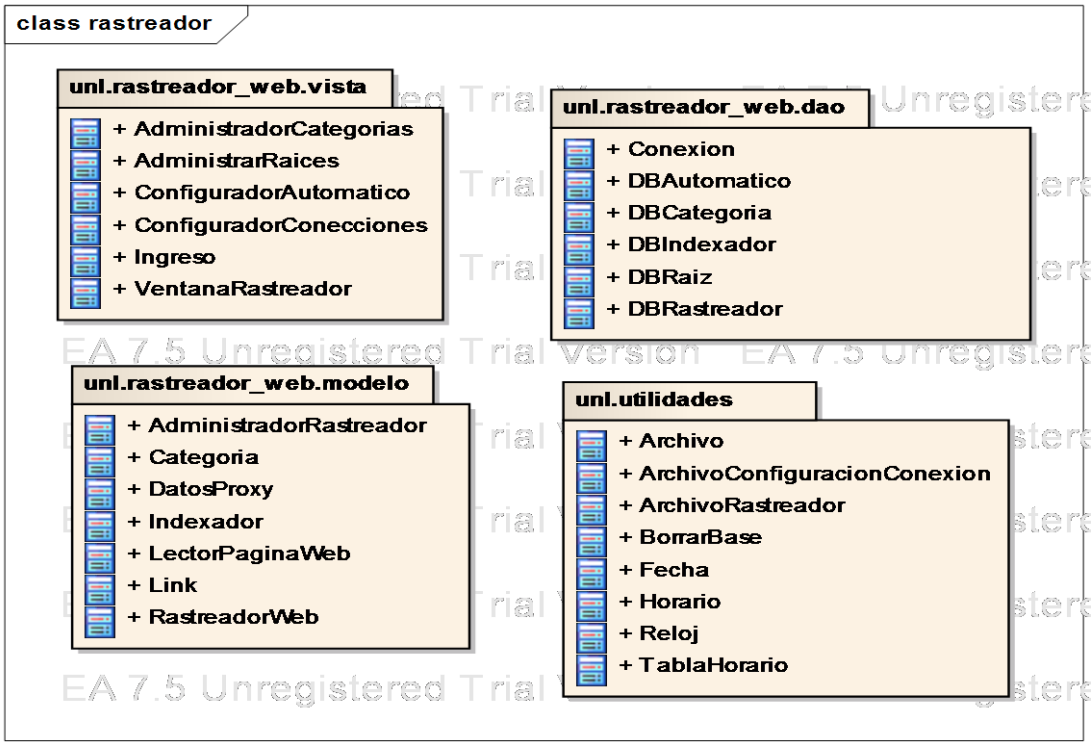




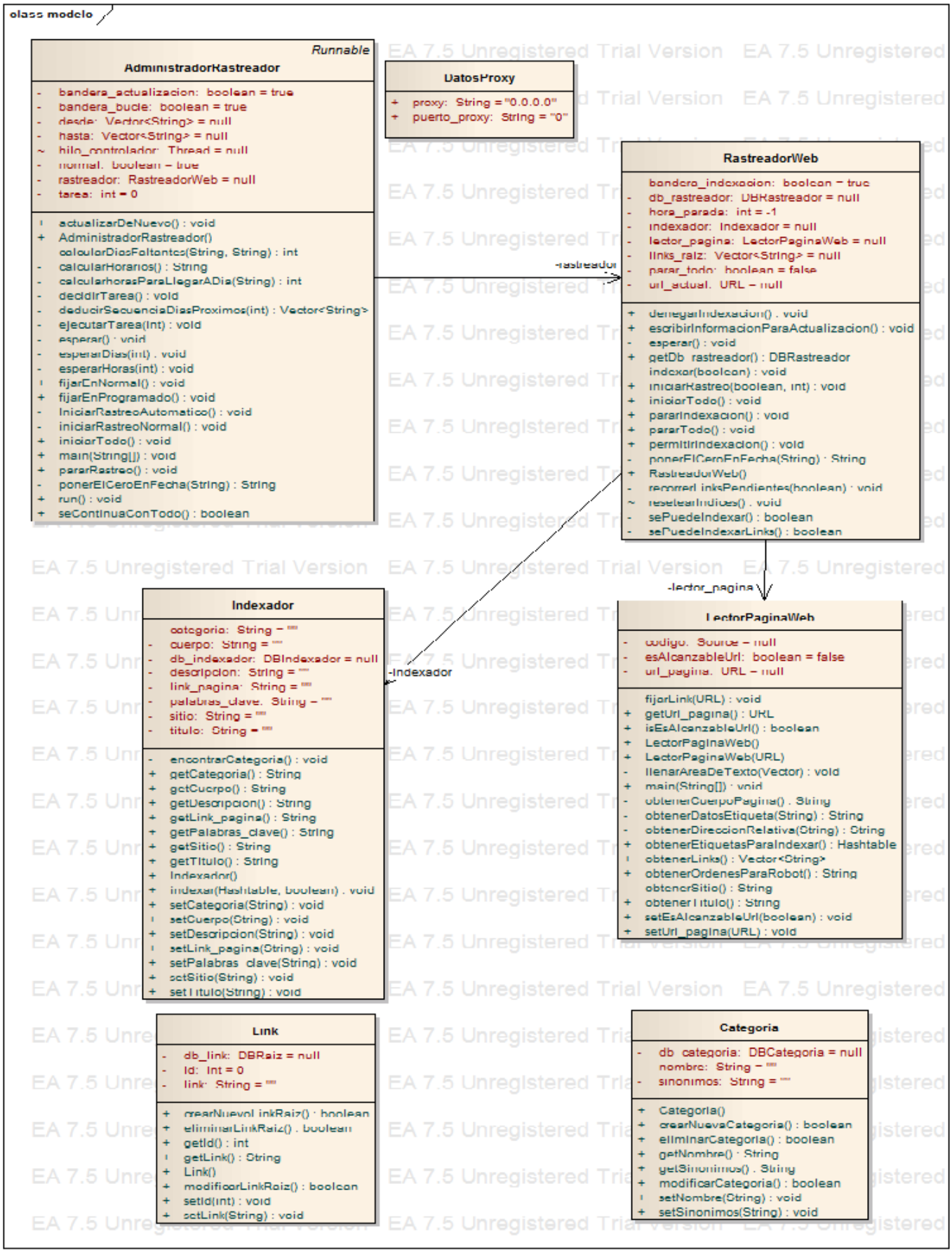


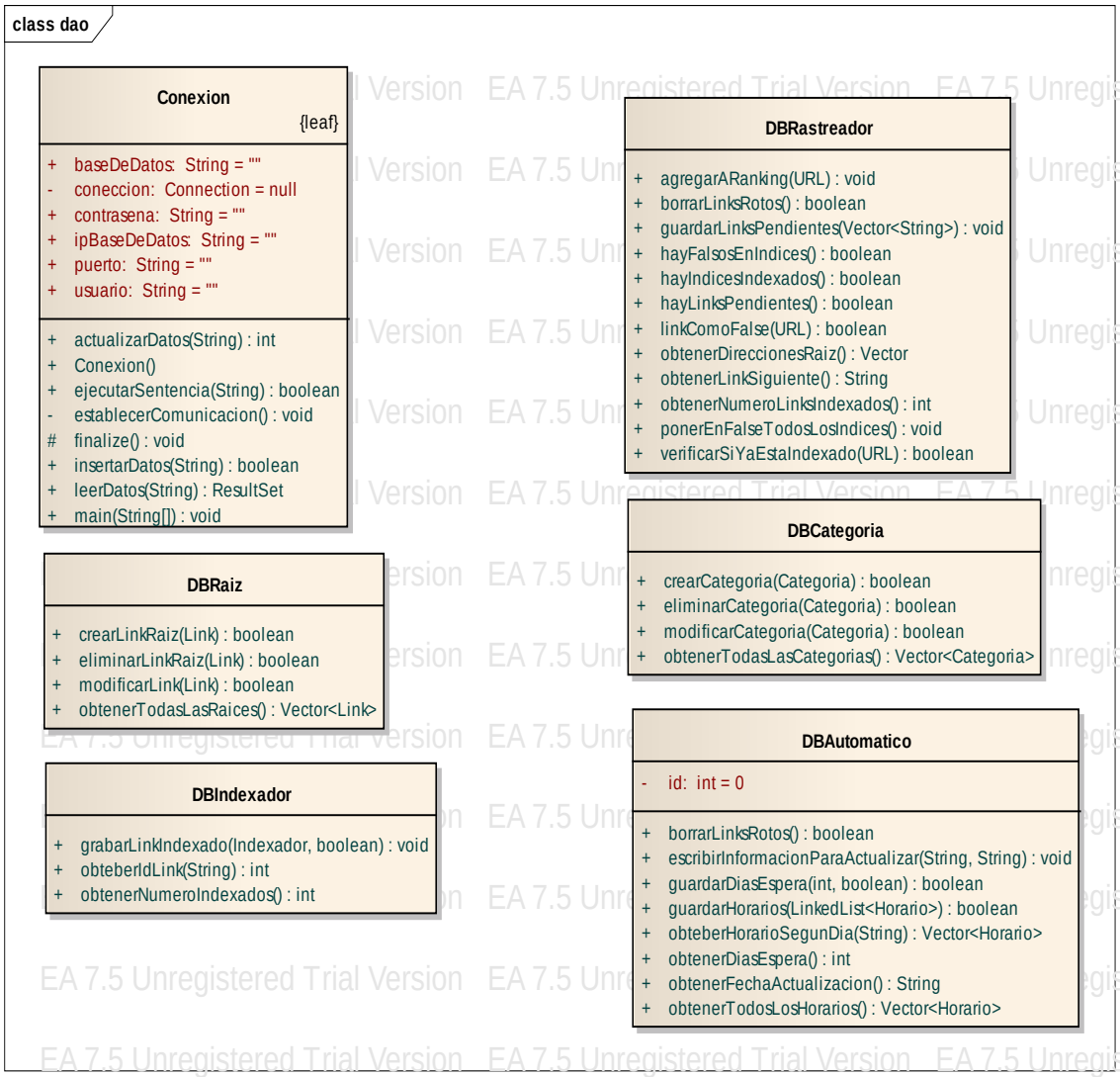
6.2 DIAGRAMA DE CLASES

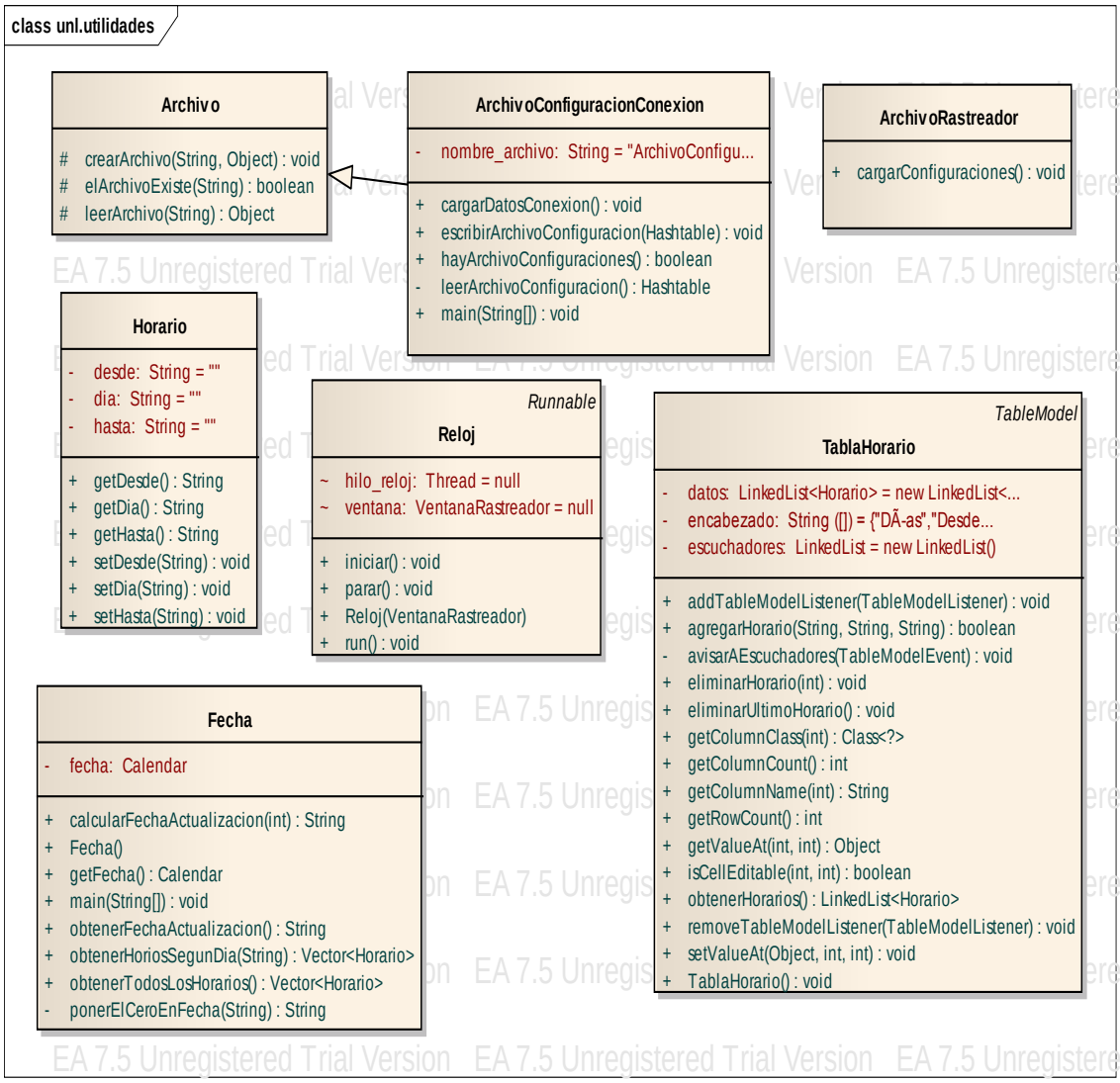
RASTREADOR



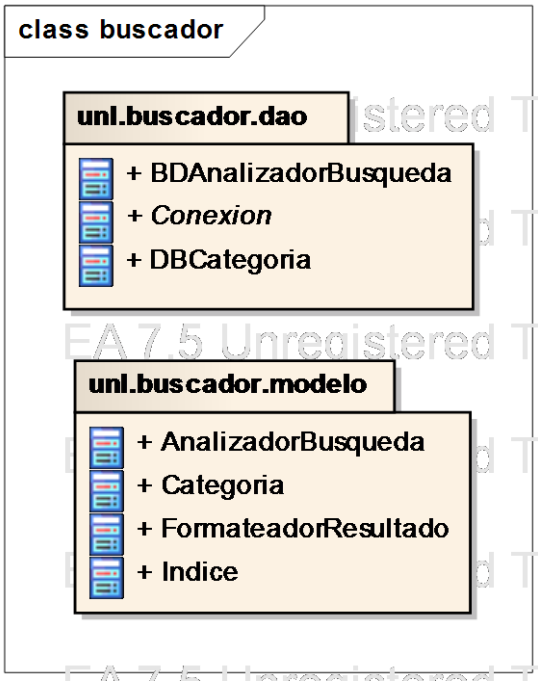

class vista
JDialog
AdministradorCategorias
- bandera_modificar: boolean = false
- btn_aceptar: javax.swing.JButton
- btn_editar: javax.swing.JButton
- btn_eliminar: javax.swing.JButton
- btn_guardar: javax.swing.JButton
- categoria_actual: Categoria = null
- categorias: Vector<Categoria> = null
- jLabel1: javax.swing.JLabel
- jLabel2: javax.swing.JLabel
- jPanel1: javax.swing.JPanel
- jScrollPane1: javax.swing.JScrollPane
- lst_categorias: javax.swing.JList
- txt_categoria: javax.swing.JTextField
- txt_sinonimos: javax.swing.JTextField
+ AdministradorCategorias(Frame, boolean)
- btn_aceptarActionPerformed(java.awt.event.ActionEvent) : void
- btn_editarActionPerformed(java.awt.event.ActionEvent) : void
- btn_eliminarActionPerformed(java.awt.event.ActionEvent) : void
- btn_guardarActionPerformed(java.awt.event.ActionEvent) : void
- inicializarLista() : void
- initComponents() : void
- lst_categoriasMouseClicked(java.awt.event.MouseEvent) : void
- txt_categoriaKeyReleased(java.awt.event.KeyEvent) : void
- txt_sinonimosKeyReleased(java.awt.event.KeyEvent) : void
JDialog
ConfiguradorAutomatico
- btn_agregar: javax.swing.JButton
- btn_cancelar: javax.swing.JButton
- btn_eliminar: javax.swing.JButton
- btn_guardar: javax.swing.JButton
- cmb_desde: javax.swing.JComboBox
- cmb_dia: javax.swing.JComboBox
- cmb_hasta: javax.swing.JComboBox
- db_automatico: DBAutomatico = new DBAutomatico()
- dias_espera: boolean = false
- jLabel1: javax.swing.JLabel
- jLabel2: javax.swing.JLabel
- jLabel3: javax.swing.JLabel
- jLabel4: javax.swing.JLabel
- jLabel5: javax.swing.JLabel
- jLabel6: javax.swing.JLabel
- jLabel7: javax.swing.JLabel
- jPanel1: javax.swing.JPanel
- jsc_tabla: javax.swing.JScrollPane
- spr_esperar: javax.swing.JSpinner
- tabla_horario: TablaHorario = null
- tbl_horario: javax.swing.JTable
- btn_agregarActionPerformed(java.awt.event.ActionEvent) : void
- btn_cancelarActionPerformed(java.awt.event.ActionEvent) : void
- btn_eliminarActionPerformed(java.awt.event.ActionEvent) : void
- btn_guardarActionPerformed(java.awt.event.ActionEvent) : void
+ ConfiguradorAutomatico(Frame, boolean)
- initComponents() : void
+ main(String[]) : void
javax.swing.JFrame
VentanaRastreador
- administrador_rastreador: AdministradorRastreador
- btn_automatico: javax.swing.JButton
- btn_categorias: javax.swing.JButton
- btn_conecciones: javax.swing.JButton
- btn_inicio: javax.swing.JButton
- btn_parar: javax.swing.JButton
- btn_programado: javax.swing.JButton
- btn_raiz: javax.swing.JButton
- jLabel1: javax.swing.JLabel
- jLabel11: javax.swing.JLabel
- jLabel13: javax.swing.JLabel
- jLabel14: javax.swing.JLabel
- jLabel15: javax.swing.JLabel
- jLabel2: javax.swing.JLabel
- jLabel4: javax.swing.JLabel
- jLabel9: javax.swing.JLabel
- jScrollPane1: javax.swing.JScrollPane
- jSeparator1: javax.swing.JSeparator
- jSeparator2: javax.swing.JSeparator
+ lb_reloj: javax.swing.JLabel
+ lbl_encontrados: javax.swing.JLabel
+ lbl_estado: javax.swing.JLabel
+ lbl_fecha: javax.swing.JLabel
+ lbl_indexados: javax.swing.JLabel
+ lbl_linkProceso: javax.swing.JLabel
- lbl_tiempo: javax.swing.JLabel
- reloj: Reloj
+ txt_linkEncontrados: javax.swing.JTextArea
- activarBotones() : void
- btn_automaticoActionPerformed(java.awt.event.ActionEvent) : void
- btn_categoriasActionPerformed(java.awt.event.ActionEvent) : void
- btn_coneccionesActionPerformed(java.awt.event.ActionEvent) : void
- btn_inicioActionPerformed(java.awt.event.ActionEvent) : void
- btn_pararActionPerformed(java.awt.event.ActionEvent) : void
- btn_programadoActionPerformed(java.awt.event.ActionEvent) : void
- btn_raizActionPerformed(java.awt.event.ActionEvent) : void
- desactivarBotones() : void
- initComponents() : void
+ main(String[]) : void
+ VentanaRastreador()
JDialog
Ingreso
- btn_ingresar: javax.swing.JButton
- jLabel1: javax.swing.JLabel
- jLabel2: javax.swing.JLabel
- txt_contrasena: javax.swing.JPasswordField
- txt_usuario: javax.swing.JTextField
- btn_ingresarActionPerformed(java.awt.event.ActionEvent) : void
+ dispose() : void
+ Ingreso(Frame, boolean)
- initComponents() : void
+ main(String[]) : void
- txt_contrasenaKeyReleased(java.awt.event.KeyEvent) : void
- txt_usuarioKeyReleased(java.awt.event.KeyEvent) : void
JDialog
AdministrarRaices
- bandera_modificar: boolean = false
- btn_aceptar: javax.swing.JButton
- btn_editor: javax.swing.JButton
- btn_eliminador: javax.swing.JButton
- btn_guardar: javax.swing.JButton
- jLabel1: javax.swing.JLabel
- jPanel1: javax.swing.JPanel
- jScrollPane1: javax.swing.JScrollPane
- link_actual: Link = null
- links: Vector<Link> = null
- lst_raices: javax.swing.JList
- txt_raiz: javax.swing.JTextField
+ AdministrarRaices(Frame, boolean)
- btn_aceptarActionPerformed(java.awt.event.ActionEvent) : void
- btn_editorActionPerformed(java.awt.event.ActionEvent) : void
- btn_eliminadorActionPerformed(java.awt.event.ActionEvent) : void
- btn_guardarActionPerformed(java.awt.event.ActionEvent) : void
- inicializarLista() : void
- initComponents() : void
- lst_raicesMouseClicked(java.awt.event.MouseEvent) : void
+ main(String[]) : void
- txt_raizKeyReleased(java.awt.event.KeyEvent) : void
JDialog
ConfiguradorConecciones
- btn_guardar: javax.swing.JButton
- btn_sin_proxy: javax.swing.JButton
- jButton2: javax.swing.JButton
- jLabel1: javax.swing.JLabel
- jLabel2: javax.swing.JLabel
- jLabel3: javax.swing.JLabel
- jLabel4: javax.swing.JLabel
- jLabel5: javax.swing.JLabel
- jLabel6: javax.swing.JLabel
- jLabel7: javax.swing.JLabel
- jSeparator1: javax.swing.JSeparator
- txt_base: javax.swing.JTextField
- txt_contrasena: javax.swing.JPasswordField
- txt_ipServidor: javax.swing.JTextField
- txt_proxy: javax.swing.JTextField
- txt_puerto: javax.swing.JTextField
- txt_puertoProxy: javax.swing.JTextField
- txt_usuario: javax.swing.JTextField
- btn_guardarActionPerformed(java.awt.event.ActionEvent) : void
- btn_sin_proxyActionPerformed(java.awt.event.ActionEvent) : void
- cargarConfiguraciones() : void
+ ConfiguradorConecciones(Frame, boolean)
- initComponents() : void
- jButton2ActionPerformed(java.awt.event.ActionEvent) : void
+ main(String[]) : void

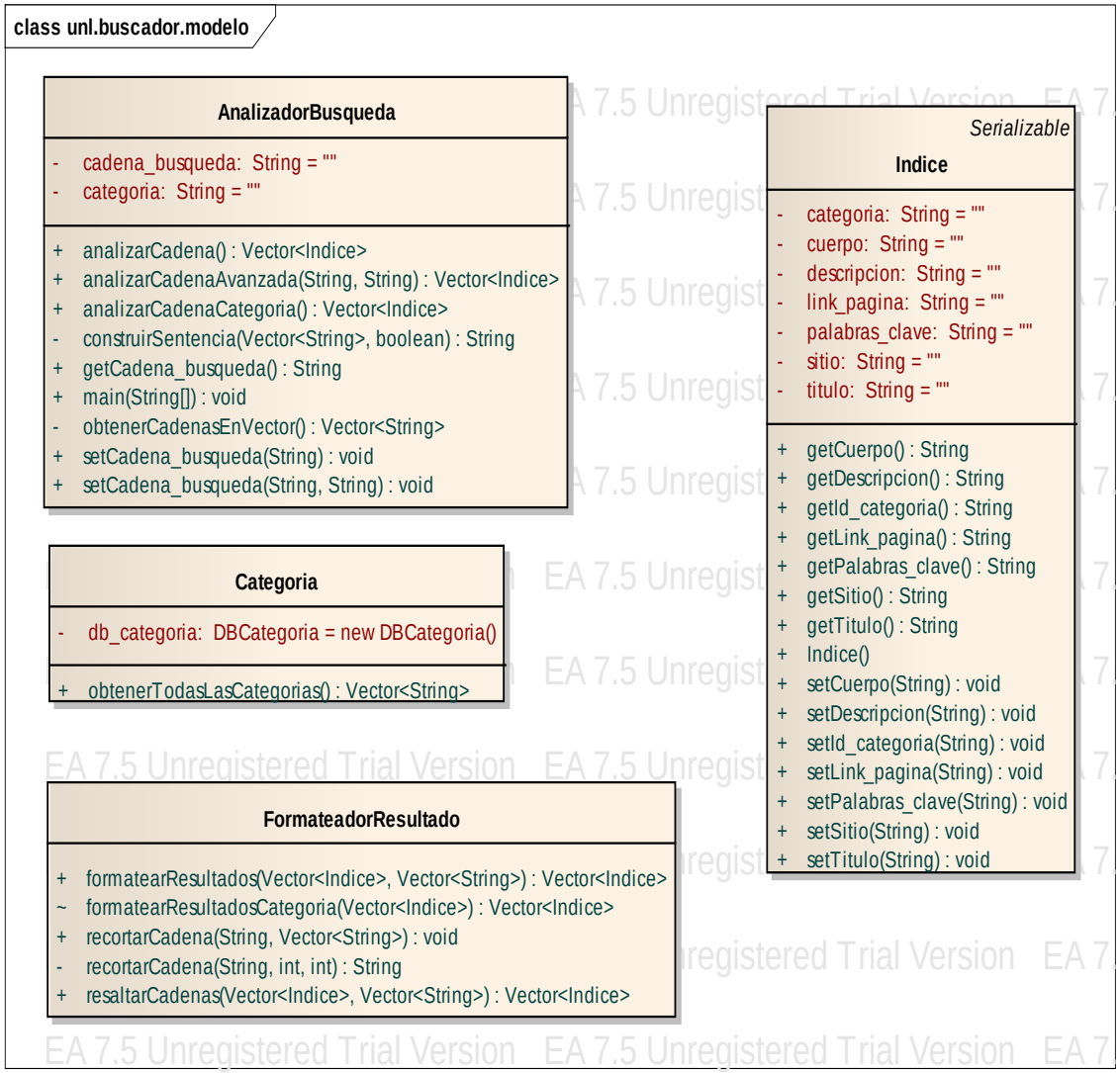

class modelo
AdministradorRastreador
Runnable
- bandera_actualizacion: boolean = true
- bandera_bucle: boolean = true
- desde: Vector<String> = null
- hasta: Vector<String> = null
~ hilo_controlador: Thread = null
- normal: boolean = true
- rastreador: RastreadorWeb = null
- tarea: int = 0
actualizarDeNuevo() : void
+ AdministradorRastreador()
calcularDiasFaltantes(String, String) : int
- calcularHorarios() : String
- calcularHorasParaLlegarADia(String) : int
- decidirTarea() : void
- deducirSecuenciaDiasProximos(int) : Vector<String>
- ejecutarTarea(int) : void
- esperar() : void
- esperarDias(int) : void
- esperarHoras(int) : void
fijarEnNormal() : void
+ fijarEnProgramado() : void
- IniciarRastreoAutomatico() : void
- iniciarRastreoNormal() : void
+ iniciarTodo() : void
+ main(String[]) : void
+ pararRastreo() : void
- ponerElCeroEnFecha(String) : String
+ run() : void
+ seContinuaConTodo() : boolean
DatosProxy
+ proxy: String = "0.0.0.0"
+ puerto_proxy: String = "0"
RastreadorWeb
bandera_indexacion: boolean = true
- db_rastreador: DBRastreador = null
- hora_parada: int = -1
- indexador: Indexador = null
- lector_pagina: LectorPaginaWeb = null
- links_raiz: Vector<String> = null
- parar_todo: boolean = false
- url_actual: URL = null
+ denegarIndexacion() : void
+ escribirInformacionParaActualizacion() : void
- esperar() : void
+ getDb_rastreador() : DBRastreador
indexar(boolean) : void
+ iniciarRastreo(boolean, int) : void
+ iniciarTodo() : void
+ pararIndexacion() : void
+ pararTodo() : void
+ permitirIndexacion() : void
- ponerElCeroEnFecha(String) : String
+ RastreadorWeb()
- recorrerLinksPendientes(boolean) : void
~ reiniciarIndices() : void
- sePuedeIndexar() : boolean
- sePuedeIndexarLinks() : boolean
-rastreador
-lector_pagina
-indexador
Indexador
categoria: String = ""
- cuerpo: String = ""
- db_indexador: DBIndexador = null
- descripcion: String = ""
- link_pagina: String = ""
- palabras_clave: String = ""
- sitio: String = ""
- titulo: String = ""
- encontrarCategoria() : void
+ getCategoria() : String
+ getCuerpo() : String
+ getDescripcion() : String
+ getLink_pagina() : String
+ getPalabras_clave() : String
+ getSitio() : String
+ getTitulo() : String
+ Indexador()
+ indexar(Hashtable, boolean) : void
+ setCategoria(String) : void
setCuerpo(String) : void
+ setDescripcion(String) : void
setLink_pagina(String) : void
+ setPalabras_clave(String) : void
+ setSitio(String) : void
+ setTitulo(String) : void
LectorPaginaWeb
- codigo: Source = null
- esAlcanzableUrl: boolean = false
- url_pagina: URL = null
fijarLink(URL) : void
+ getUrl_pagina() : URL
+ isEsAlcanzableUrl() : boolean
+ LectorPaginaWeb()
+ LectorPaginaWeb(URL)
- llenarAreaDeTexto(Vector) : void
+ main(String[]) : void
- obtenerCuerpoPagina() : String
- obtenerDatosEtiqueta(String) : String
- obtenerDireccionRelativa(String) : String
+ obtenerEtiquetasParaIndexar() : Hashtable
obtenerLinks() : Vector<String>
+ obtenerOrdenesParaRobot() : String
obtenerSitio() : String
+ obtenerTitulo() : String
+ setEsAlcanzableUrl(boolean) : void
+ setUrl_pagina(URL) : void
Link
- db_link: DBRaiz = null
- id: int = 0
- link: String = ""
+ crearNuevoLinkRaiz() : boolean
+ eliminarLinkRaiz() : boolean
+ getId() : int
getLink() : String
+ Link()
+ modificarLinkRaiz() : boolean
+ setId(int) : void
+ setLink(String) : void
Categoria
- db_categoria: DBCategoria = null
nombre: String = ""
- sinonimos: String = ""
+ Categoria()
+ crearNuevaCategoria() : boolean
+ eliminarCategoria() : boolean
+ getNombre() : String
+ getSinonimos() : String
+ modificarCategoria() : boolean
setNombre(String) : void
+ setSinonimos(String) : void

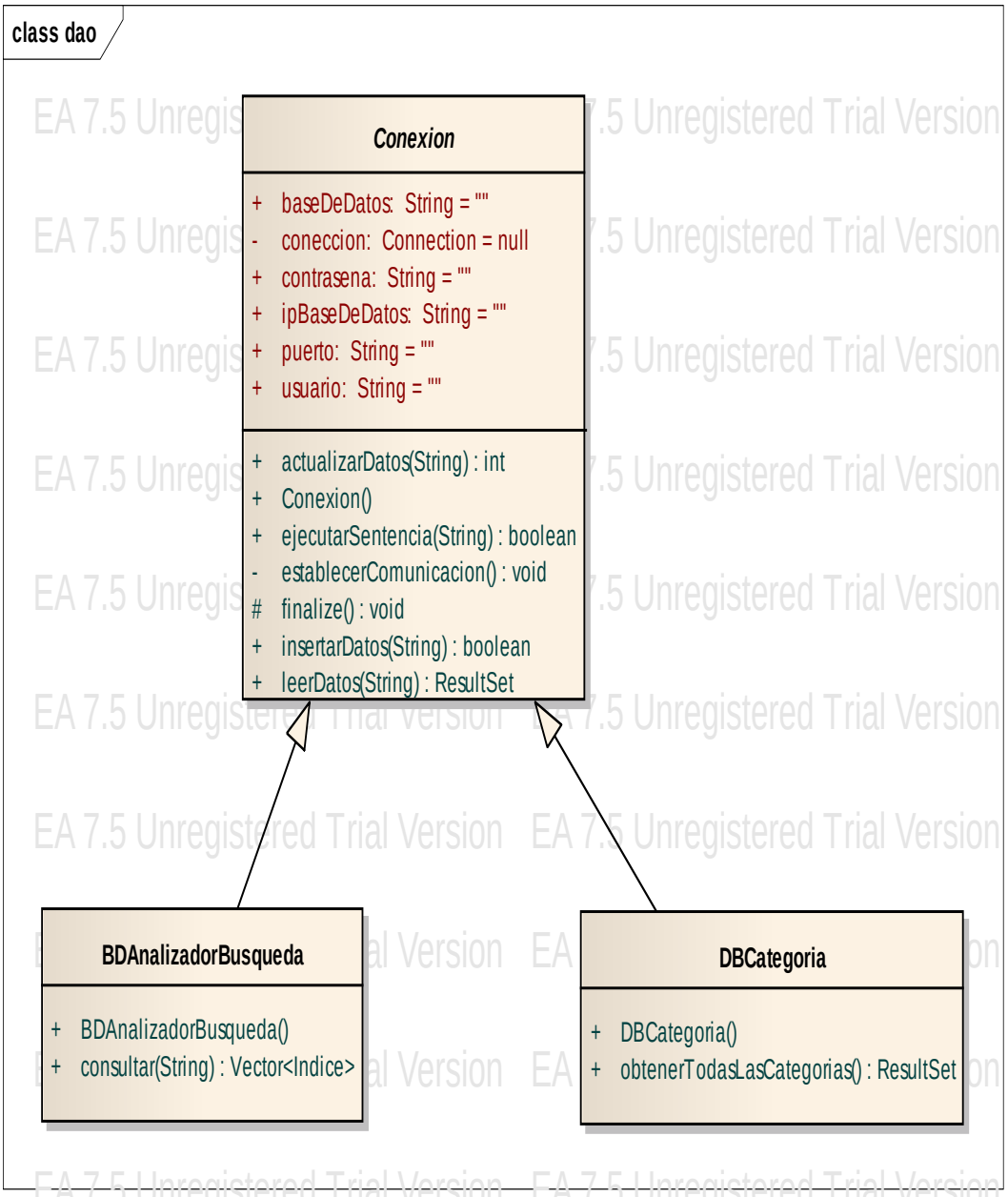

class dao
Conexion
{leaf}
+ baseDeDatos: String = ""
- coneccion: Connection = null
+ contrasena: String = ""
+ ipBaseDeDatos: String = ""
+ puerto: String = ""
+ usuario: String = ""
+ actualizarDatos(String) : int
+ Conexion()
+ ejecutarSentencia(String) : boolean
- establecerComunicacion() : void
finalize() : void
+ insertarDatos(String) : boolean
+ leerDatos(String) : ResultSet
+ main(String[]) : void
DBRastreador
+ agregarARanking(URL) : void
+ borrarLinksRotos() : boolean
+ guardarLinksPendientes(Vector<String>) : void
+ hayFalsosEnIndices() : boolean
+ hayIndicesIndexados() : boolean
+ hayLinksPendientes() : boolean
+ linkComoFalse(URL) : boolean
+ obtenerDireccionesRaiz() : Vector
+ obtenerLinkSiguiente() : String
+ obtenerNumeroLinksIndexados() : int
+ ponerEnFalseTodosLosIndices() : void
+ verificarSiYaEstaIndexado(URL) : boolean
DBRaiz
+ crearLinkRaiz(Link) : boolean
+ eliminarLinkRaiz(Link) : boolean
+ modificarLink(Link) : boolean
+ obtenerTodasLasRaices() : Vector<Link>
DBCategoria
+ crearCategoria(Categoria) : boolean
+ eliminarCategoria(Categoria) : boolean
+ modificarCategoria(Categoria) : boolean
+ obtenerTodasLasCategorias() : Vector<Categoria>
DBIndexador
+ grabarLinkIndexado(Indexador, boolean) : void
+ obteberIdLink(String) : int
+ obtenerNumeroIndexados() : int
DBAutomatico
- id: int = 0
+ borrarLinksRotos() : boolean
+ escribirInformacionParaActualizar(String, String) : void
+ guardarDiasEspera(int, boolean) : boolean
+ guardarHorarios(LinkedList<Horario>) : boolean
+ obteberHorarioSegunDia(String) : Vector<Horario>
+ obtenerDiasEspera() : int
+ obtenerFechaActualizacion() : String
+ obtenerTodosLosHorarios() : Vector<Horario>
EA 7.5 Unregistered Trial Version


class unl.utilidades
Archivo
crearArchivo(String, Object) : void
elArchivoExiste(String) : boolean
leerArchivo(String) : Object
ArchivoConfiguracionConexion
- nombre_archivo: String = "ArchivoConfigu...
+ cargarDatosConexion() : void
+ escribirArchivoConfiguracion(Hashtable) : void
+ hayArchivoConfiguraciones() : boolean
- leerArchivoConfiguracion() : Hashtable
+ main(String[]) : void
ArchivoRastreador
+ cargarConfiguraciones() : void
Horario
- desde: String = ""
- dia: String = ""
- hasta: String = ""
+ getDesde() : String
+ getDia() : String
+ getHasta() : String
+ setDesde(String) : void
+ setDia(String) : void
+ setHasta(String) : void
Runnable
Reloj
~ hilo_reloj: Thread = null
~ ventana: VentanaRastreador = null
+ iniciar() : void
+ parar() : void
+ Reloj(VentanaRastreador)
+ run() : void
TableModel
TablaHorario
- datos: LinkedList<Horario> = new LinkedList<...
- encabezado: String ([]) = {"DÃ-as","Desde...
- escuchadores: LinkedList = new LinkedList()
+ addTableModelListener(TableModelListener) : void
+ agregarHorario(String, String, String) : boolean
- avisarAEscuchadores(TableModelEvent) : void
+ eliminarHorario(int) : void
+ eliminarUltimoHorario() : void
+ getColumnClass(int) : Class<?>
+ getColumnCount() : int
+ getColumnName(int) : String
+ getRowCount() : int
+ getValueAt(int, int) : Object
+ isCellEditable(int, int) : boolean
+ obtenerHorarios() : LinkedList<Horario>
+ removeTableModelListener(TableModelListener) : void
+ setValueAt(Object, int, int) : void
+ TablaHorario() : void
Fecha
- fecha: Calendar
+ calcularFechaActualizacion(int) : String
+ Fecha()
+ getFecha() : Calendar
+ main(String[]) : void
+ obtenerFechaActualizacion() : String
+ obtenerHoriosSegunDia(String) : Vector<Horario>
+ obtenerTodosLosHorarios() : Vector<Horario>
- ponerElCeroEnFecha(String) : String

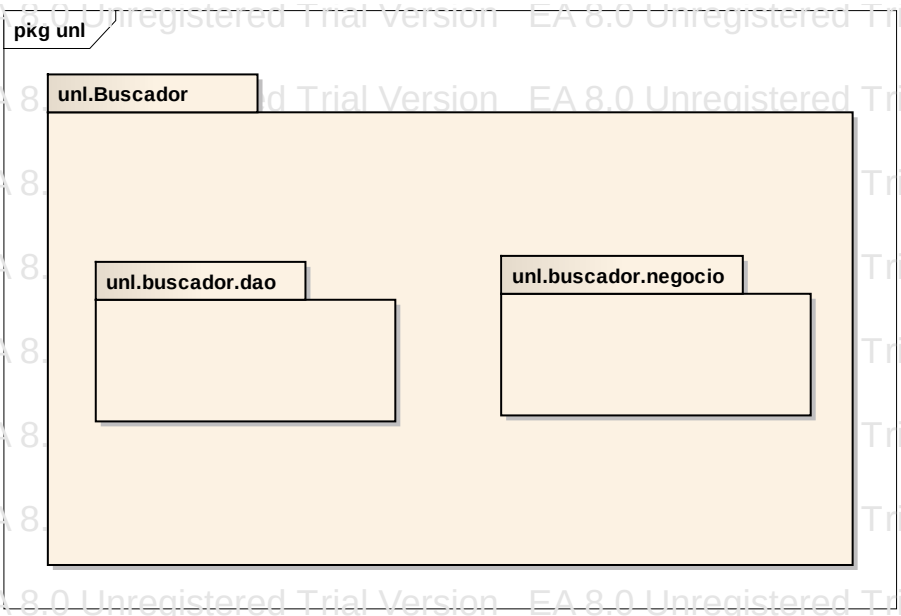
BUSCADOR

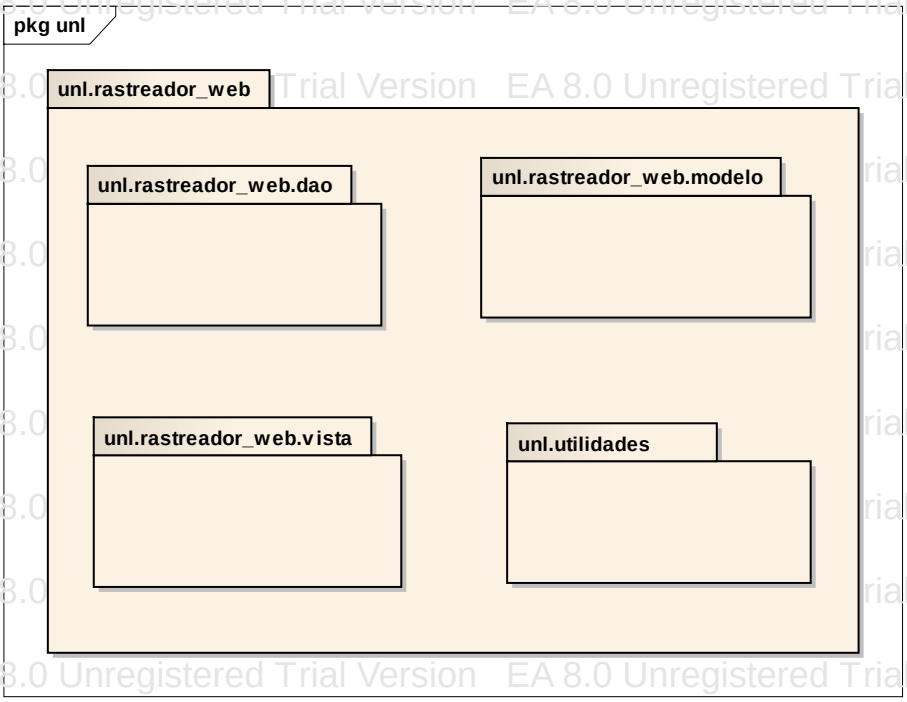

class buscador
unl.buscador.dao
+ BDAnalizadorBusqueda
+ Conexion
+ DBCategoria
unl.buscador.modelo
+ AnalizadorBusqueda
+ Categoria
+ FormateadorResultado
+ Indice


class unl.buscador.modelo
AnalizadorBusqueda
- cadena_busqueda: String = ""
- categoria: String = ""
+ analizarCadena() : Vector<Indice>
+ analizarCadenaAvanzada(String, String) : Vector<Indice>
+ analizarCadenaCategoria() : Vector<Indice>
- construirSentencia(Vector<String>, boolean) : String
+ getCadena_busqueda() : String
+ main(String[]) : void
- obtenerCadenasEnVector() : Vector<String>
+ setCadena_busqueda(String) : void
+ setCadena_busqueda(String, String) : void
Serializable
Indice
- categoria: String = ""
- cuerpo: String = ""
- descripcion: String = ""
- link_pagina: String = ""
- palabras_clave: String = ""
- sitio: String = ""
- titulo: String = ""
+ getCuerpo() : String
+ getDescripcion() : String
+ getId_categoria() : String
+ getLink_pagina() : String
+ getPalabras_clave() : String
+ getSitio() : String
+ getTitulo() : String
+ Indice()
+ setCuerpo(String) : void
+ setDescripcion(String) : void
+ setId_categoria(String) : void
+ setLink_pagina(String) : void
+ setPalabras_clave(String) : void
+ setSitio(String) : void
+ setTitulo(String) : void
Categoria
- db_categoria: DBCategoria = new DBCategoria()
+ obtenerTodasLasCategorias() : Vector<String>
FormateadorResultado
+ formatearResultados(Vector<Indice>, Vector<String>) : Vector<Indice>
~ formatearResultadosCategoria(Vector<Indice>) : Vector<Indice>
+ recortarCadena(String, Vector<String>) : void
- recortarCadena(String, int, int) : String
+ resaltarCadenas(Vector<Indice>, Vector<String>) : Vector<Indice>
EA 7.5 Unregistered Trial Version

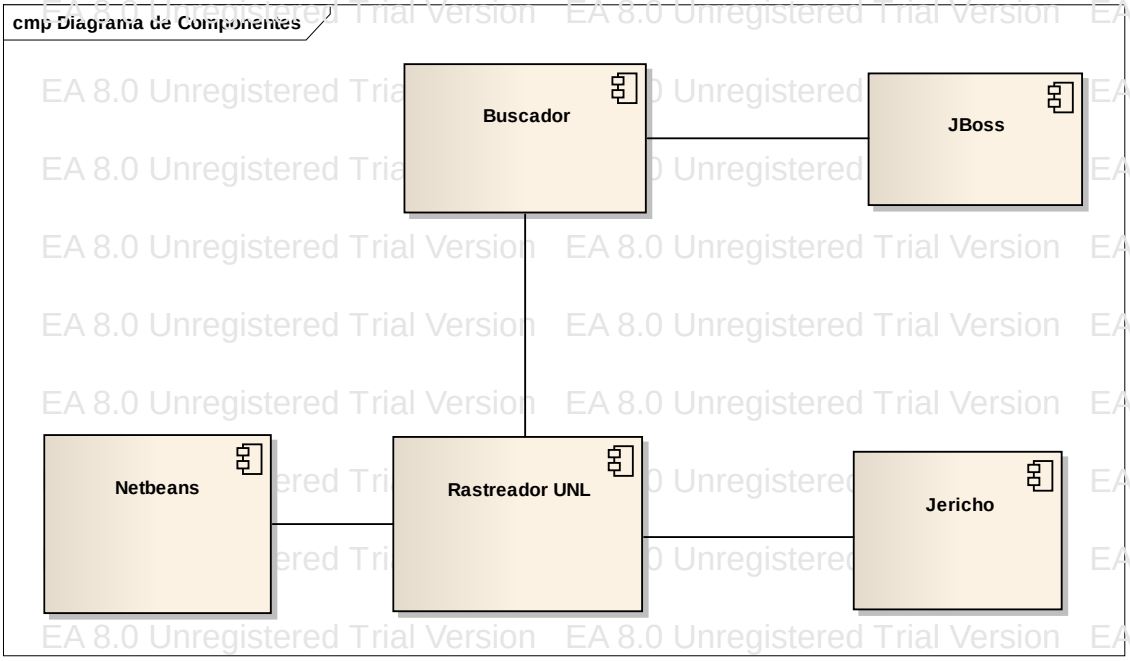

class dao
Conexion
+ baseDeDatos: String = ""
- coneccion: Connection = null
+ contrasena: String = ""
+ ipBaseDeDatos: String = ""
+ puerto: String = ""
+ usuario: String = ""
+ actualizarDatos(String) : int
+ Conexion()
+ ejecutarSentencia(String) : boolean
- establecerComunicacion() : void
finalize() : void
+ insertarDatos(String) : boolean
+ leerDatos(String) : ResultSet
BDAnalizadorBusqueda
+ BDAnalizadorBusqueda()
+ consultar(String) : Vector<Indice>
DBCategoria
+ DBCategoria()
+ obtenerTodasLasCategorias() : ResultSet

6.3 DIAGRAMA DE PAQUETES

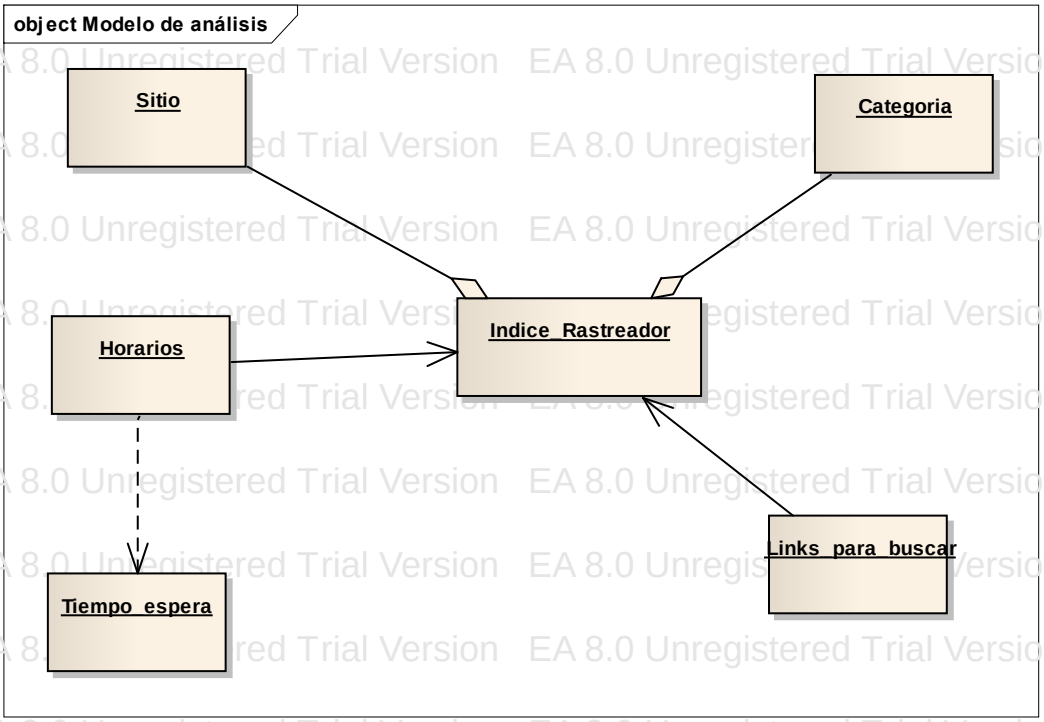




6.4. DIAGRAMA DE COMPONENTES



6.5 DIAGRAMA ENTIDAD-RELACION



FASE 3
CONSTRUCCION DEL MOTOR DE BUSQUEDA

7. CODIFICACIÓN.

Esta fase está contemplada la construcción de la aplicación BUNL, el script y la estructura de la base de datos, el programa de administración del rastreador UNL (Ambiente Escritorio), interfaz del cliente (Ambiente Web), y la configuración del servidor web

7.1. Arquitectura tres Capas.

Se trata de realizar un diseño del que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Figura 7.1: Imagen de la Arquitectura tres capas

A continuación describimos como se encuentra estructurada la aplicación:

7.1.1. Programa Rastreador UNL(Programa Escritorio)

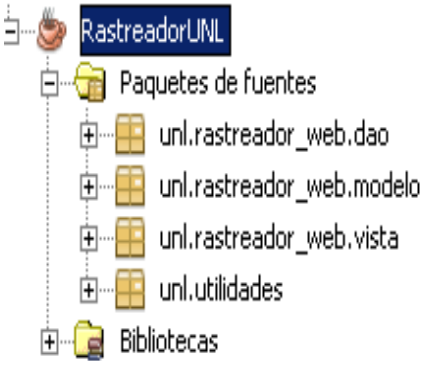


Figura 7.2: Imagen de paquetes en tres capas del programa rastreadorUNL

Este programa Rastreador UNL, utiliza un Api jericho 3.2 que sirve como pasarela y realiza peticiones a los servidores para que el Rastreador UNL puede indexar, almacenar y categorizar la información. Antes que nada, visitan una URL específica provista por un servidor de URLs y almacenan su contenido en los discos de los buscadores, luego, aplicando distintos algoritmos, el rastreador UNL analiza y relaciona todo el contenido de los datos; como título y a las etiquetas meta del código html, contenido ; también calcula la importancia de cada página en función de cuántas otras la enlazan.

La técnica de la indexación se basa principalmente en buscar en el titulo descripción, y contenido los sinónimos de las categorías y asignar la categoría correspondiente así también el link mas referenciado el Page Ranking.

7.1.2. Buscador (Programa WEB)

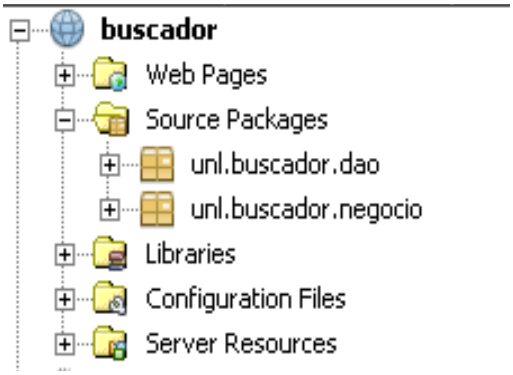


Figura 7.3: Imagen de paquetes en tres capas del Buscador

Es la Interfaz Web, el cual contiene un campo de búsqueda de palabras claves ingresadas por el usuario, petición de búsqueda que se realizara en la base de datos mediante el programa de búsqueda se presentan los resultados como titulo de la página o descripción y su URL , ordenados según su PageRanking.

FASE 4

IMPLEMENTACIÓN Y VALIDACIÓN

8. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA.

8.1. INSTALACIÓN

La etapa final del presente proyecto de tesis consiste en la implementación del Buscador BUNL 1.0.1. en el portal de la Universidad Nacional de Loja. La maquina donde se instalara nuestro proyecto contiene la siguientes características en cuanto al software necesario, para el correcto funcionamiento del sistema Informático.

El Buscador UNL 1.0.1 como ya lo hemos descrito anteriormente contiene dos subproyectos Buscador y RastreadorUNL que serán instalados en el servidor del Área de energía, las industrias y los recursos naturales no renovables; recuperando y almacenando información en la base de datos respectivamente.

8.1.1. Configuración de la base de datos

Para configuración de la base de datos se ingresa a la siguiente dirección:

http://aeirnnr.unl.edu.ec/phpmyadmin/index.php

Figura 8.1: Imagen de configuración de la base de datos

Se crea la base de datos y su respectiva contraseña luego se procede a importa el script de la base de datos con la opción Import.

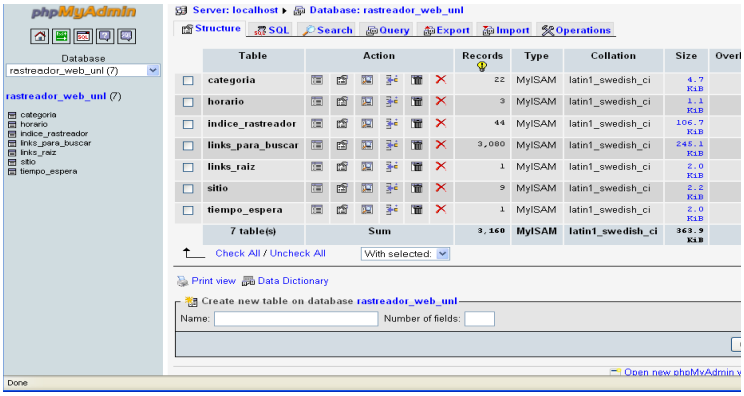


Figura 8.2: Imagen estructura de la base de datos

8.1.2. **Instalación y configuración del buscador.war**

Después de la configurar de la base de datos el siguiente paso es levantar la aplicación Buscador copiando el .war en el directorio deploy del servidor Web Jboss 5.2.0 y la configuración del mismo, con ello el Buscador UNL 1.0.1 podrá ser utilizado, accediendo a él mediante la dirección:

http://aeirnnr.unl.edu.ec:8081/buscador

8.1.3. **Instalación y configuración del rastreadorunl.jar**

Después de la configurar de la base de datos y el siguiente paso es instalar subproyecto Rastredor UNL.jar

Para su ejecución se hacer clic en el siguiente icono rastreadorUnl .sh o desde la consola con el comando:

Servidor $ java –jar /home/buscador/dist/rastreadorUnl.jar

El rastredorUnl.jar genera un archivo de configuración.dll ;cuando es ejecutado por primera vez y en el momento que se ingresan los datos de conexión a la base de datos.

La instalación en el servidor de rastreadorUnl.jar no será analizada con profundidad debido a que se trabajo mas con ambos subproyectos desde una maquina personal.

8.2. PRUEBAS DE VALIDACIÓN.

La sección final del presente proyecto de tesis se centrará en el proceso de pruebas de software.

Toda prueba de software puede detectar errores de diversa naturaleza dentro del sistema, sin embargo, la ausencia de los mismos no indica que le sistema no los posea.

Por ello el mejor resultado de las pruebas es demostrar que no existen errores fatales ocultos en el producto lo que muy diferente a decir que el sistema está libre de errores, como consecuencia de esto se considera que todo producto de software terminado siempre poseerá errores, sin embargo se estima que estos poseen una incidencia relativamente baja sobre el producto final y no deben ser considerados un razón para rechazar el producto.

Dentro del Buscador **BUNL** se realizo un proceso de pruebas estándar partiendo desde las unidades más pequeñas de software hasta el sistema completo en la figura 8.3 se observa la relación entre las actividades del desarrollo y las diversas pruebas se software aplicadas en el presente proyecto.

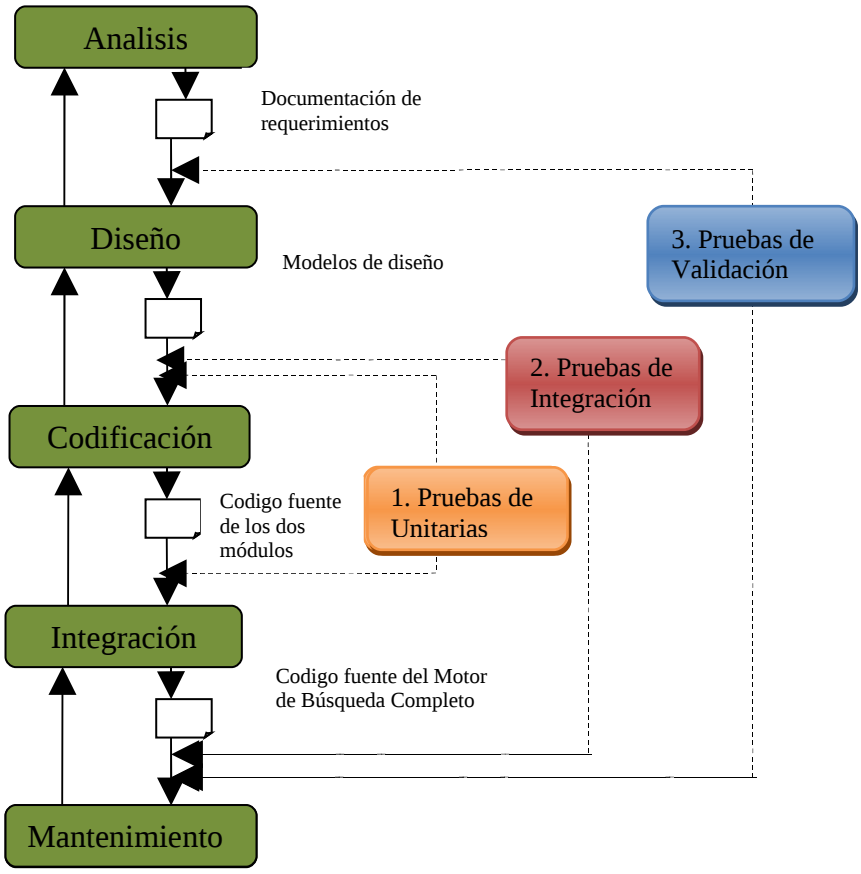


Figura 8.3: Pruebas del Motor de Búsqueda en las actividades de desarrollo

El proceso de pruebas de software descrito en figura 8.3 mantiene un orden jerárquico la primera etapa consistió en verificar la funcionalidad individual de las unidades de software que conforma el sistema a través de las denominadas pruebas unitarias, una vez concluidas se realizaron las pruebas de integración para comprobar que las unidades se software verificados trabajen bien en conjunto, al finalizar esta etapa el sistema se pude considerar completo siendo

apto para las pruebas de sistemas en general conocidas como pruebas de validación.

En el presente análisis se omite los resultados de las pruebas unitarias y de integración debido a que su relevancia se orienta más hacia los nosotras las desarrolladoras, y centra en los datos obtenidos en las pruebas de validación.

8.2.1. Plan de pruebas de validación

Terminada la fase de pruebas unitarias y de integración del sistema la siguiente fase del proceso de evaluación es determinar si el producto de software cumple con las especificaciones del diseño.

La fase de pruebas y validación del Motor de Búsqueda, se llevo a cabo mediante encuestas modelo que se detalla en el anexo 2. Las encuestas están dirigidas a los docentes y estudiantes de la Carrera de Ingeniería en Sistemas en el Área de Energía, las Industrias y los Recursos Naturales No Renovables, quienes observaran aspectos como: la Interfaz, la navegación en las ventanas o links, entrada de datos, indexación, obtención de resultados.

Dentro de este proceso de validación del Buscador las encuestas fueron aplicadas a 2 ingenieros de la carrera, y 25 estudiantes de la carrera de Ingeniería en sistemas los cuales harán uso de esta aplicación, siendo un total de 27 encuestas realizadas.

En el presente proyecto de tesis las pruebas de validación se centraran en dos aspectos; el rendimiento y la funcionalidad de ambos subproyectos, que conforman el Motor de Búsqueda BUNL.

A continuación se detalla los resultados de la tabulación de las pruebas de **FUNCIONALIDAD Y RENDIMIENTO** del sistema:

VARIABLES DE CALIFICACIÓN.-

- **EX = Excelente.**
- **MB = Muy Buena.**
- **B = Buena.**
- **R = Regular.**

DOCENTES Y ESTUDIANTES

FUNCIONALIDAD	Ex	MB	B	R	TOTAL
Cuál sería su calificación acerca del Ambiente del Desarrollo Buscador BUNL 1.1.0 en lo que corresponde a su Entorno Gráfico.	17	9	1	0	27
El buscador BUNL al usuario es de fácil acceso e interfaz amigable	27	0	0	0	27
De acuerdo a su apreciación cual sería su calificación de la distribución de botones de búsquedas.	9	18	0	0	27
El esquema de presentar los resultados en el Buscador los considera.	20	5	2	0	27
La opción *categorías* permite la búsqueda por categorías usted como la calificaría.	15	11	1	0	27
De acuerdo a su ejecución La *búsqueda avanzada* como la considera dentro de un buscador.	27	0	0	0	27
Cree que el Buscador UNL 1.1.0 es una aplicación de investigación necesaria para el usuario recupere información de Internet.	27	0	0	0	27
	RENDIMIENTO				
En cuanto al rendimiento cómo calificaría a la aplicación. Buscador BUNL 1.1.0	21	5	1	0	27
Los resultados obtenidos en la búsqueda tienen similitud con la palabra clave buscada	24	2	1	0	27
Cuál sería su calificación sobre los datos que requiere el programa Rastreador UNL para su ejecución.	22	3	2	0	27
Cuál sería su calificación en lo referente a los links visitados e indexados.	18	2	2	5	27
La ejecución del buscador BUNL 1.1.0 en la plataforma Windows como usted la calificaría.	23	4	0	0	27
La ejecución del buscador BUNL 1.1.0 en la plataforma Linux como usted la calificaría.	20	1	6	0	27
RESULTADOS	**269**	**60**	**16**	**5**	**351**

Cuadro 8.1: Resultados de las pruebas de validación

8.2.1.1. Análisis de resultados

Guía que se utilizó para realizar las pruebas a los usuarios del sistema

Para saber si la aplicación cumple con las perspectivas propuestas al inicio del proyecto se realizaron pruebas de validación y de rendimiento las cuales se las aplicó a los docentes y estudiantes, quienes siguieron ciertos pasos para llegar a realizar búsquedas avanzadas y por categorías, por otra parte la ejecución del programa Rastreador UNL es fundamental para indexación de páginas en la base de datos.

A continuación se presenta los procesamientos o pasos:

Para ejecución del buscador:

- Se ingresa a la pantalla del Buscador UNL 1.1.0.
- Se ingresa la palabra clave en el Cuadro texto, luego se pulsa el botón "*Buscar".* Se visualiza los resultados con información de la pagina como (SitioWeb al que pertenece, descripción y el page ranking).
- Para realizar la búsqueda avanzada se pulsa en el link *"Búsqueda Avanzada"* y se digita la palabra clave a buscar, luego se selecciona una de las dos opciones de búsqueda avanzada *"Todas las palabras"* o "Una de las palabras" y después se pulsa el botón "*Buscar".*

  Se visualiza los resultados con información de la pagina como

  (SitioWeb al que pertenece, descripción y el page ranking).
- Para la búsqueda por categorías se pulsa en el link *"Búsqueda* Categorías*"* y se selecciona una categoría.

- Se visualiza los resultados con información de la pagina como

  (SitioWeb al que pertenece, descripción y el page ranking).

Para ejecución del programa Rastreador UNL:

- Se Ingresa los datos de la conexión a la base de datos
- Una vez en el programa se ingresa el link raíz
- Se puede seleccionar Inicio automática especificando el rango de tiempo que el rastreador deberá indexar.
- Ingresar nuevas categorías de indexación
- Para ejecutar al programa para indexación y rastreo de paginas pulsar el Iniciar y se visualizara los links que se van analizando del link raíz.

Una vez termina las pruebas realizadas a los docentes y estudiantes se obtienen los siguientes resultados.

BUSCADOR UNL 1.1.0.

1. **¿ Cuál sería su calificación acerca del Ambiente del Desarrollo Buscador BUNL 1.1.0 en lo que corresponde a su Entorno Gráfico.?**

Nro	OPCIONES	F	%
1	Excelentes.	17	63
2	Muy Buena.	9	33
3	Buena.	1	4
4	Regular.	0	0
	TOTAL	27	100

Cuadro 8.2: Cuadro estadístico de la pregunta Nº1

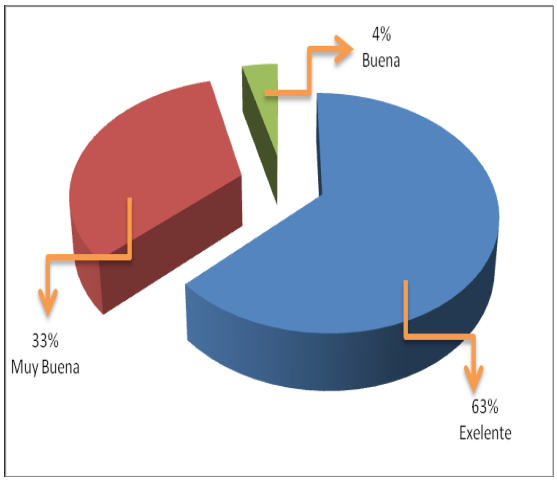


Figura 7.4: Grafico estadístico de la pregunta Nº1

- En lo que respecta a la distribución de las diferentes paginas y diseño que tiene BUNL se puede evidenciar en el cuadro y gráfico estadístico que predomina la excelencia con un 63%, Muy buena con un 33,0%, buena con un 4% y Regular con un 0%.

- Esto nos quiere decir que BUNL tiene una excelente aceptación en su entorno de presentación.

2. ¿El buscador BUNL 1.1.0 al usuario es de fácil acceso e interfaz amigable?

Nro	OPCIONES	F	%
1	Si.	27	100
2	No.	0	0
	TOTAL	27	100

Cuadro 8.3: Cuadro estadístico de la pregunta Nº2

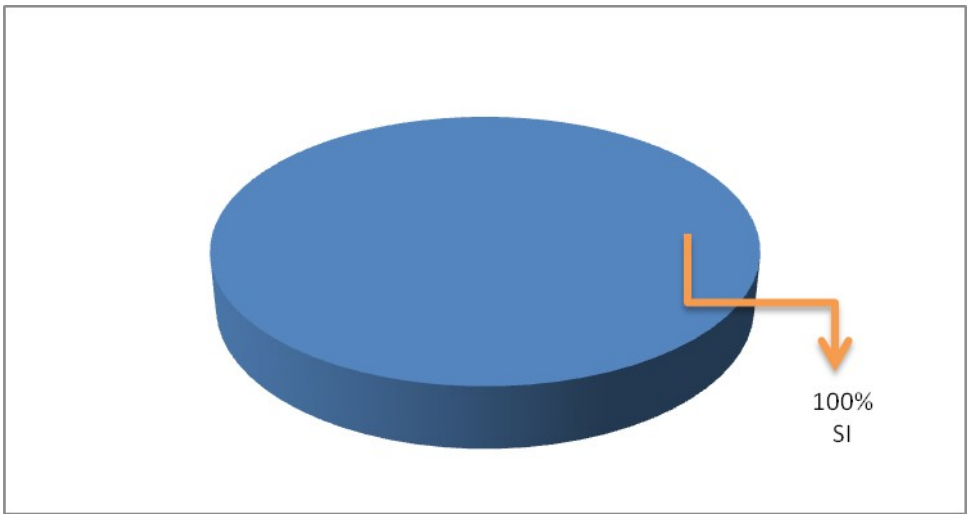


Figura 7.5: Grafico estadístico de la pregunta Nº2

- Como se puede evidenciar en el cuadro y gráfico estadístico predomina que el buscador BUNL es amigable al usuario con un 100%.

- Entonces podemos afirmar el buscador BUNL es amigable al usuario en el sentido que se pueden acceder fácilmente e ingresar las palabras a buscar.

3. ¿De acuerdo a su apreciación cual sería su calificación de la distribución de botones de búsquedas?

Nro	OPCIONES	F	%
1	Excelentes.	9	33,3
2	Muy Buena.	18	66.7
3	Buena.	0	0
4	Regular.	0	0
	TOTAL	27	100

Cuadro 8.4: Cuadro estadístico de la pregunta Nº3

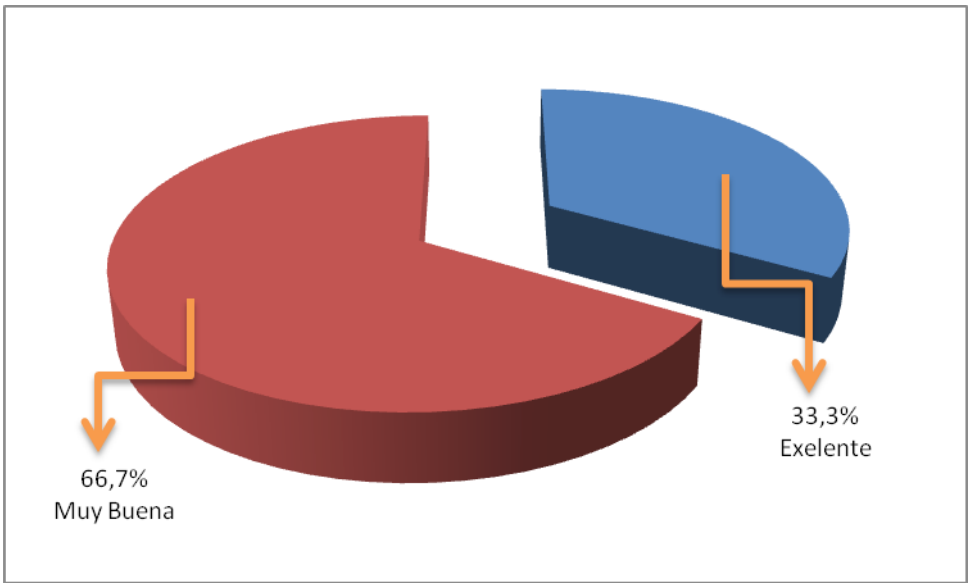


Figura 7.6: Grafico estadístico de la pregunta Nº3

- Como se puede evidenciar en el cuadro y gráfico estadístico tiene muy buena distribución de botones de búsquedas con un 66,70 % de los encuestados que lo afirman, y un 33,30% de los encuestados que dicen que la distribución es excelente.

- En este caso la distribución de botones de búsquedas el buscador BUNL es muy buena ya que los botones de búsquedas están agrupadas según la búsqueda a seleccionar.

4. ¿El esquema de presentar los resultados en el Buscador los considera?

Nro	OPCIONES	f	%
1	Rápida.	21	77,78
2	Normal.	5	18,52
3	Lenta.	1	3,70
	TOTAL	27	100

Cuadro 8.5: Cuadro estadístico de la pregunta Nº4

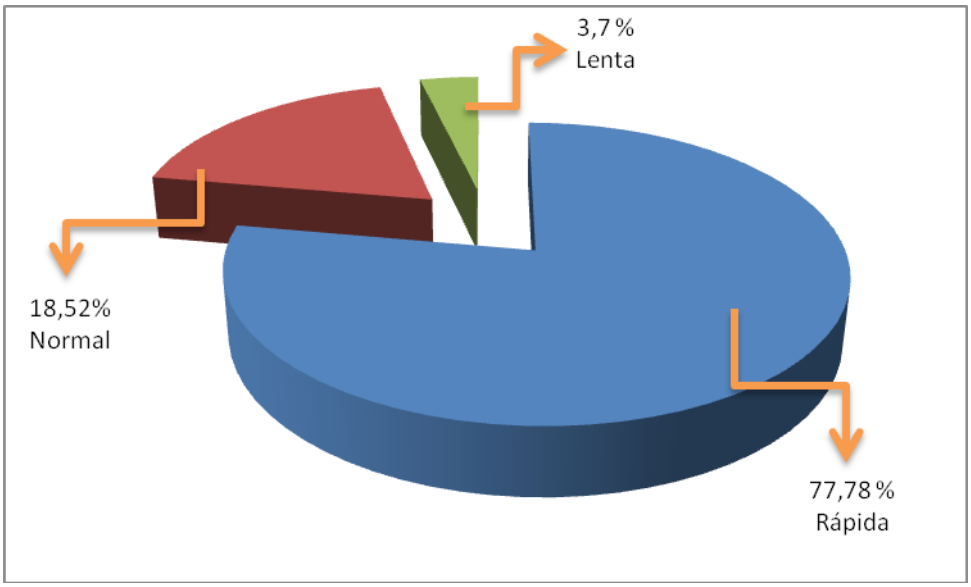


Figura 7.7: Grafico estadístico de la pregunta Nº4

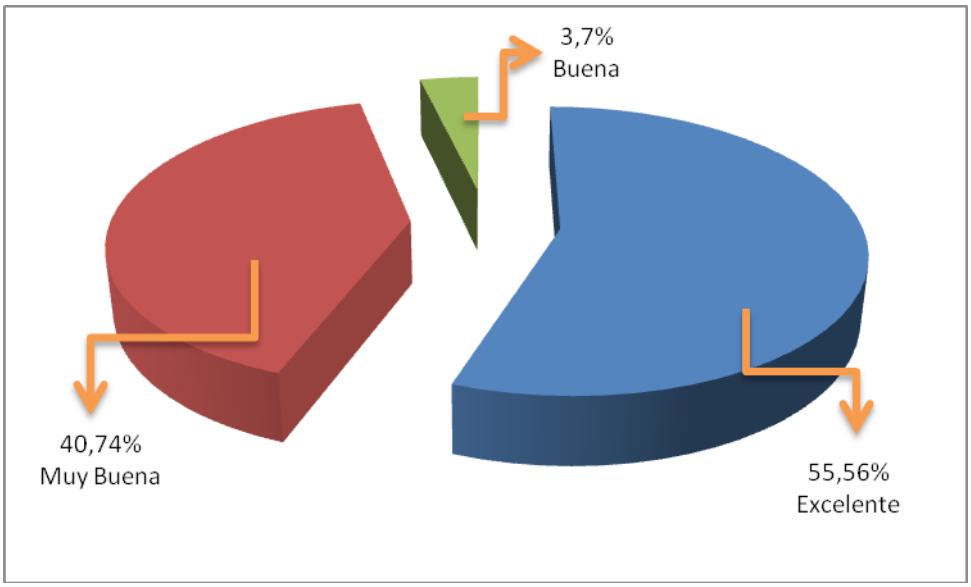
- Como se puede observar en el cuadro y gráfico estadístico existe una mayoría de usuarios que lo califican a BUNL como un buscador rápido con un 77,78% en la obtención de los resultados, un 18,52% de los encuestados opinan el rendimiento es normal y 3,7% Lenta.

- Con los porcentajes expuestos se puede afirmar que el buscador BUNL tiene un rendimiento aceptable en la obtención de los resultados.

5. ¿ La opción *categorías* permite la búsqueda por categorías usted como la calificaría.?

Nro	OPCIONES	f	%
1	Excelente.	15	55,56
2	Muy Buena.	11	40,74
3	Buena.	1	3,70
4	Regular.	0	0
	TOTAL	27	100

Cuadro 8.6: Cuadro estadístico de la pregunta Nº5



Cuadro 8.8: Cuadro estadístico de la pregunta Nº5

- Como se puede observar en el cuadro y gráfico estadístico existe un 55,56% de los encuestados que afirman que buscador BUNL tiene una excelente opción para la búsqueda por categorías, un 40,74% nos dice que opción para la búsqueda por categorías es muy buena y un 3,70% dice que opción para la búsqueda por categorías es bueno.

- Con los datos estadísticos presentados podemos decir el buscador BUNL debería realizar búsquedas por categorías.

6. ¿De acuerdo a su ejecución La *búsqueda avanzada* como la considera dentro de un buscador?

Nro	OPCIONES	F	%
1	Si.	27	100
2	No.	0	0
	TOTAL	27	100

Cuadro 8.7: Cuadro estadístico de la pregunta Nº6

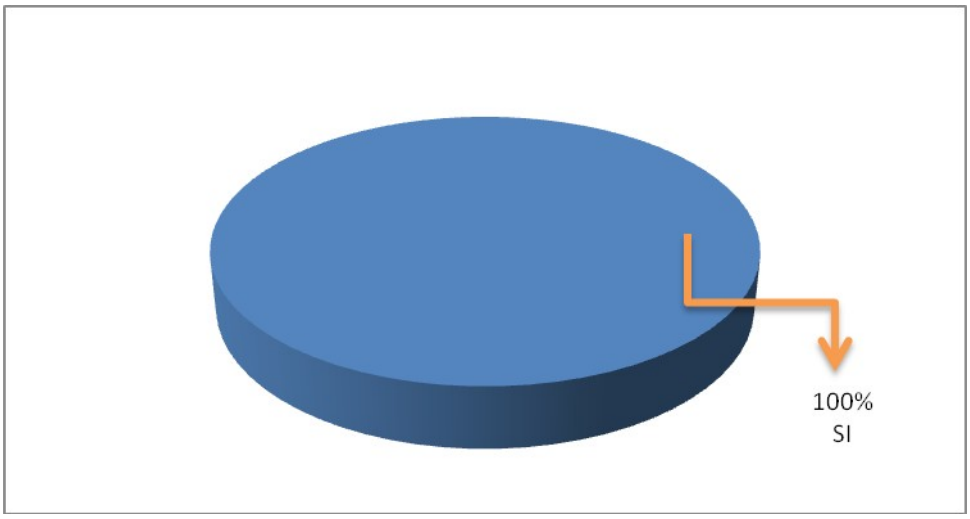


Figura 7.9: Grafico estadístico de la pregunta Nº6

- Como se puede evidenciar en el cuadro y gráfico estadístico predomina que el buscador BUNL debería realizar búsquedas avanzadas con un 100%.

- Entonces podemos afirmar el buscador BUNL debe realizar búsquedas avanzadas, permitiendo al usuario encontrar resultados más acorde con la palabra buscar.

7. ¿ Cree que el Buscador UNL 1.1.0 es una aplicación de investigación necesaria para el usuario recupere información de Internet.?

Nro	OPCIONES	F	%
1	Si.	27	100
2	No.	0	0
	TOTAL	27	100

Figura 7.8: Grafico estadístico de la pregunta Nº7

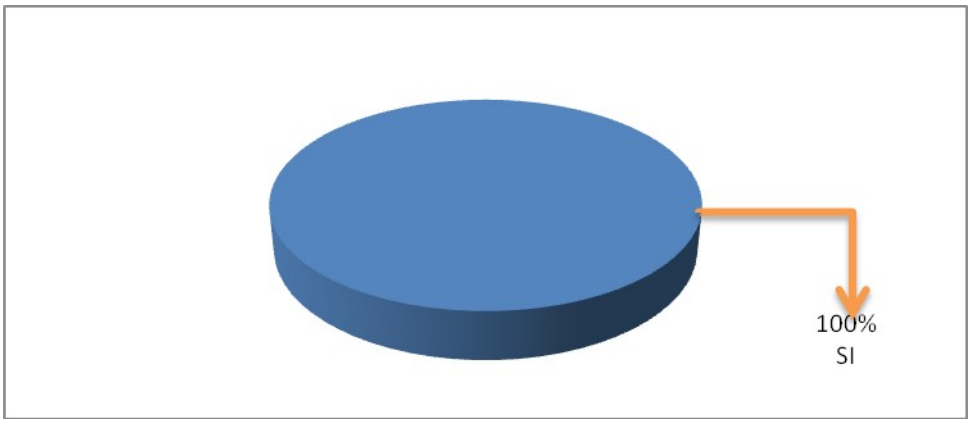
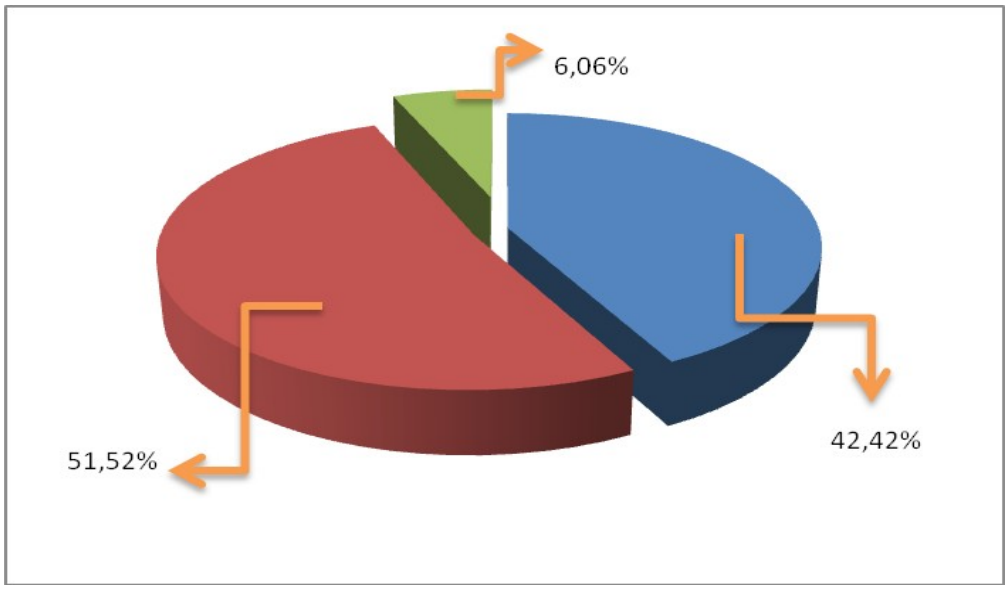


Figura 7.10: Grafico estadístico de la pregunta Nº7

Nro	OPCIONES	F	%
1	Presenta opciones búsqueda	14	42,42
2	Presenta rápido los resultados de búsqueda	17	51,52
3	Ahorra tiempo	2	6,06
	TOTAL	33	100

Porque:

Cuadro 8.9: Cuadro estadístico de la pregunta Nº7



Cuadro 8.11: Cuadro estadístico de la pregunta Nº7

- Observando el cuadro y gráfico estadístico se puede decir que la totalidad de los encuestados creen que buscador BUNL es una aplicación de investigación necesaria para que el usuario recupere información de Internet
- Presentados los datos estadísticos se puede afirmar BUNL es una aplicación de investigación necesaria para que el usuario recupere información de Internet por que permite:

  - **Opciones búsqueda**
  - **Presenta rápidos resultados de búsqueda**
  - **Ahorrar el tiempo en la búsqueda de información por palabras claves.**

8. **¿En cuanto al rendimiento cómo calificaría a la aplicación Buscador BUNL 1.1.0?**

Nro	OPCIONES	f	%
1	Excelente.	20	74,07
2	Muy Buena.	5	18,51
3	Buena.	2	7,40
4	Regular.	0	0
	TOTAL	27	100

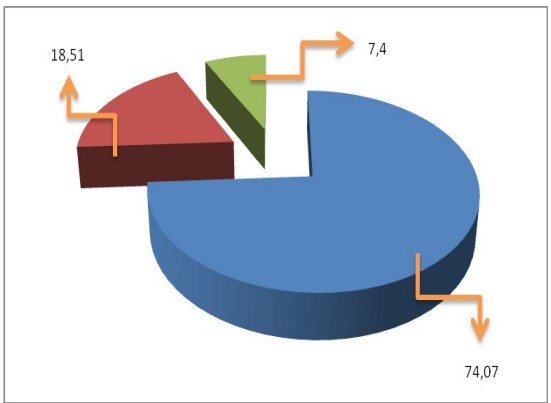


Figura 7.12: Grafico estadístico de la pregunta Nº8

- Como se puede observar en el cuadro y gráfico estadístico un 74,07% de los encuestados dicen que el buscador BUNL en cuanto su rendimiento es excelente, un 18,51% dicen que es muy bueno y un 7,4% dicen que es bueno.

- Se puede afirmar que el buscador BUNL en cuanto su rendimiento es aceptable.

9. **¿Los resultados obtenidos en la búsqueda tienen similitud con la palabra clave buscada?**

Nro	OPCIONES	f	%
1	Excelentes.	24	88,88
2	Muy Buena.	2	7,40
3	Buena.	1	3,70
4	Regular.	0	0
	TOTAL	27	100

Cuadro 8.11: Cuadro estadístico de la pregunta Nº9

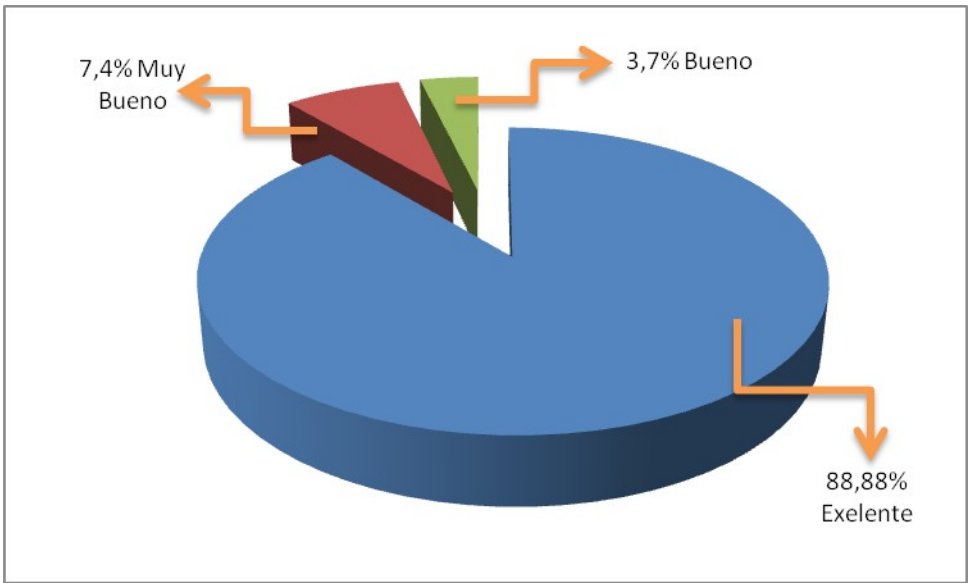


Figura 7.13 : Grafico estadístico de la pregunta Nº9

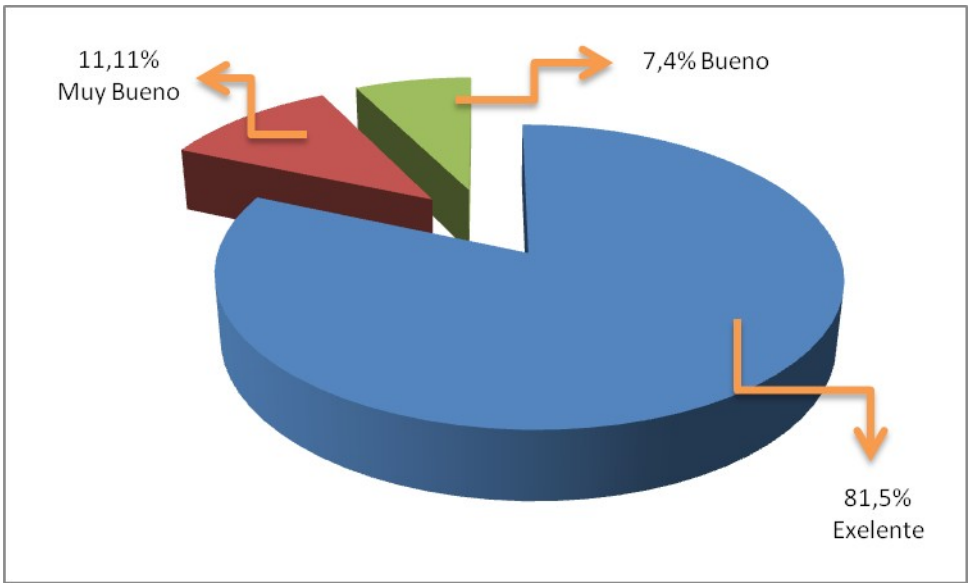
- En el cuadro y gráfico estadístico se puede observar que un 88,8% de los encuestados opinan que los resultados obtenidos en la búsqueda con relación a la palabra clave buscada es excelente, un 7,4% opina que son muy buenos y un 3,7% son buenos.

- Con la presentación de los resultados afirmamos que la manera Los resultados obtenidos en la búsqueda tienen similitud con la palabra clave buscada

10. **¿Cuál sería su calificación en lo referente a los links visitados e indexado. ?**

Nro	OPCIONES	f	%
1	Excelentes.	22	81,5
2	Muy Buena.	3	11,11
3	Buena.	2	7,40
4	Regular.	0	0
	TOTAL	27	100

Cuadro 8.12: Cuadro estadístico de la pregunta Nº10



Cuadro 8.14: Cuadro estadístico de la pregunta Nº10

- Como se puede observar en el cuadro y gráfico estadístico un 81,5% de los encuestados dicen que la indexación de los links visitados es exelente, un 11,11% de los encuestados dicen que es muy bueno y un 7,4% de los encuestados nos dicen que es bueno.

- Presentados los datos estadísticos podemos decir que la indexación de los links visitados es excelente ya que es comprensible al usuario.

11. **¿La ejecución del buscador BUNL 1.1.0 en la plataforma Windows como usted la calificaría?**

Nro	OPCIONES	f	%
1	Excelente.	18	66,7
2	Muy Bueno.	2	7,40
3	Bueno.	2	7,40
4	Regular.	5	18,5
	TOTAL	27	100

Cuadro 8.13: Cuadro estadístico de la pregunta Nº11

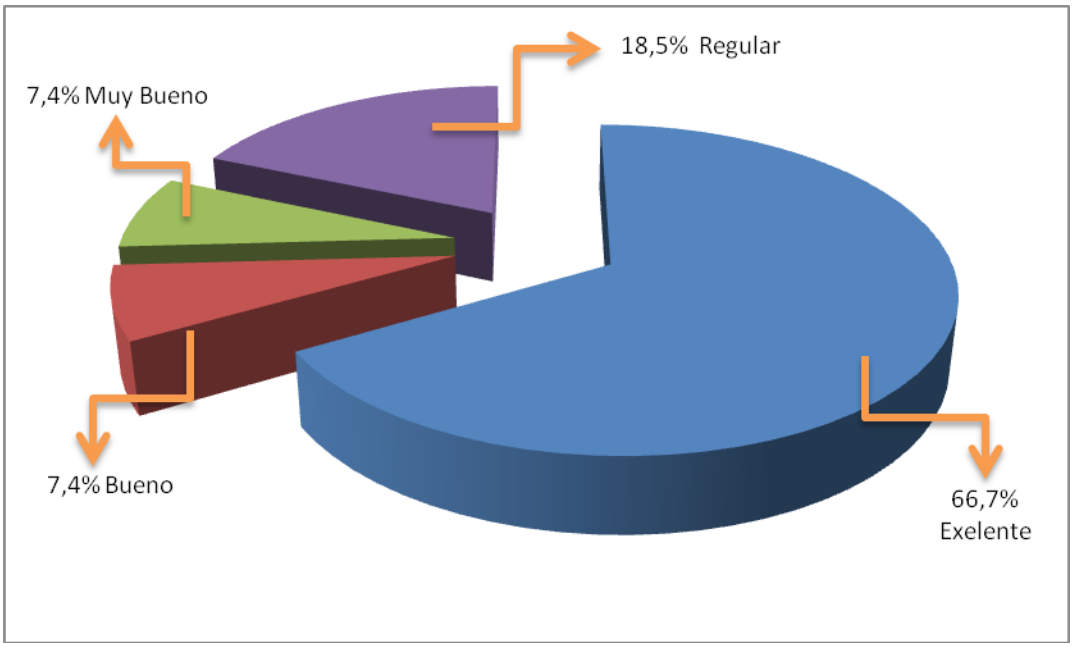


Figura 7.14: Grafico estadístico de la pregunta Nº11

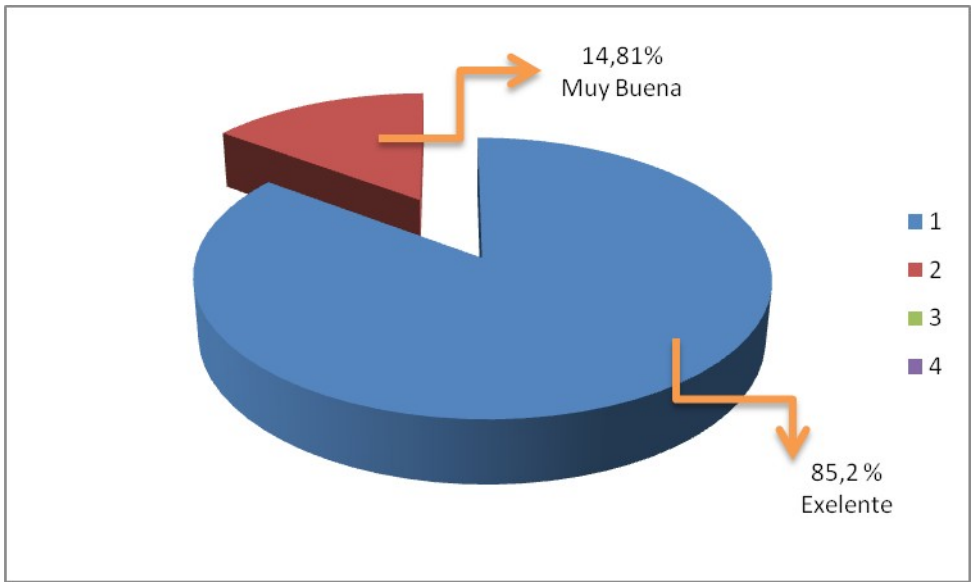
- Como se puede apreciar en el cuadro y gráfico estadístico un 66,7% de los encuestados opinan que la ejecución de buscador BUNL en la Plataforma Windows es excelente, un 18,5% de los encuestados opinan que es regular, un 7,4% de los encuestados opinan que es muy bueno, y un 7,4% de los encuestados opinan que es bueno.

- Se puede evidenciar que la ejecución de buscador BUNL en la plataforma Windows es muy aceptable ya que el tiempo de ejecución con respecto a otros entornos similares es inferior.

12. ¿La ejecución del buscador BUNL 1.1.0 en la plataforma Linux como usted la calificaría?

Nro	OPCIONES	f	%
1	Excelentes.	23	85,2
2	Muy Buena.	4	14,81
3	Buena.	0	0
4	Regular.	0	0
	TOTAL	27	100

Cuadro 8.14: Cuadro estadístico de la pregunta Nº12



Cuadro 8.15: Cuadro estadístico de la pregunta Nº11

- Como se puede apreciar en el cuadro y gráfico estadístico un 85,2% de los encuestados opina que la ejecución de buscador BUNL 1.1.0 en la plataforma Linux es excelente y un 14,81% de los encuestados opinan que es muy bueno.

- Presentados los datos estadísticos se puede afirmar que la ejecución de buscador BUNL 1.1.0 en la plataforma Linux es excelente y que trabaja de manera normal.

8.2.4. Pruebas de aceptación:

Si las pruebas de validación son aprobadas el producto de software está listo para ser entregado, pero el proceso de pruebas todavía no termina ya que el usuario final ésta en posición de realizar un conjunto de pruebas denominadas de aceptación; estas pruebas de aceptación pueden ser similares a las de validación y de hecho son realizadas sobre el producto fina, pero a diferencia de las anteriores etapas de pruebas esta vez el grupo de desarrolladores o responsables del proyecto no participan directamente.

En el presente proyecto de tesis el BUNL no ha planificado una fase de pruebas de aceptación de forma explícita, se ha considerado que la participación del usuario final en las pruebas de validación es una manera propicia de confirmar la calidad del proyecto del motor de Búsqueda, los cambios en migración de información y privacidad de acceso de la misma, que se realice en el servidor de la universidad, no contempla dentro del plan de pruebas y tampoco del proceso de pruebas en sí, ya que no se puede documentar ni dirigido de madera adecuada.

Con lo expuesto en el párrafo anterior, la culminación formal del proyecto de tesis: **"DISEÑO E IMPLEMENTACIÓN DE UN MOTOR DE BÚSQUEDA PARA LA RECUPERACIÓN DE INFORMACIÓN, DESDE EL PORTAL DE LA UNIVERSIDAD NACIONAL DE LOJA",** se daría con a la aprobación de las pruebas de validación.

9. VALORACIÓN TÉCNICA ECONÓMICA.-

En la culminación del presente proyecto se ha evidenciado que la aplicación cumple con los objetivos planteados al inicio del mismo, entre las características que posee la aplicación están:

- Recorre la Web con el rastreadorunL realizando peticiones a los servidores, almacenando e indexando las páginas.
- Permite al usuario realizar búsquedas por categorías.
- Permite al usuario realizar búsquedas avanzadas.
- En primera instancia realiza las búsquedas en el portal de la universidad nacional de Loja.
- El rastreador utiliza la técnica del PageRanking para medir la importancia de cada página de acuerdo links mas referenciado.
- Presentan resultados en grupo de ítems con la descripción de la Urls.
- La aplicación es gran aceptación por los estudiantes de la carrera.

En lo que concierne a la inversión económica del presente proyecto se ha evidenciado que los costos son regulares, debido a que las herramientas de trabajo es software Libre GNU.

A continuación se detalla los recursos utilizados para la elaboración del proyecto:

9.1.- RECURSOS.-

- **HUMANOS.-** A continuación detallamos los recursos humanos que intervienen directa e indirectamente en el proyecto:

RECURSOS HUMANOS

Descripción	Cantidad	# Horas	Valor Unitario	Costo Total
Director de Tesis	**1**	**0**	**0.00**	**0.00**
Programadores Analistas	**2**	**500**	**1.50**	**1500.00**
			Total =	**$1,500.00**

- **TÉCNICOS.-** Los recursos técnicos que se utilizarán en el desarrollo de nuestro proyecto se describe a continuación:

Descripción	Cantidad	Valor Unitario	Costo Total
HARDWARE			
Computador Portátil HP	**2**	**1,500.00**	**3,000.00**
Impresora	**1**	**60.00**	**60.00**
Dispositivos USB	**2**	**30.00**	**60.00**
Disco externo	**1**	**80.00**	**80.00**
SOFTWARE			
Netbeans 6.8	**1**	**0.00**	**0.00**
Plataforma Java (**J2SDK**)	**1**	**0.00**	**0.00**
Open Office Write	**1**	**0.00**	**0.00**
Enterprise UML	**1**	**0.00**	**0.00**
	Total =		**$ 3,200.00**

❖ **MATERIALES.-** A continuación se describe los recursos materiales que se utilizarán para el desarrollo del proyecto.

9.2 RECURSOS MATERIALES

Descripción	Cantidad	# Horas	Valor Unitario	Costo Total
Cartuchos de Tinta	10	0	2.50	25
Transporte			100	100
Resma de Papel	4	0	5.00	20.00
Copias	500	0	0.03	15.00
Internet	1	500	0.80	400.00
Anillados	5	0	1.00	5.00
Libros	2	0	20.00	40.00
			Total =	605.00

❖ **ECONOMICOS.-**

Los Recursos Económicos para la elaboración del Proyecto son los siguientes:

- ✓ **Recursos Humanos** $ 15000 .00
- ✓ **Recursos Técnicos** $ .00
- ✓ **Recursos Materiales** $ .00

Total $ .00

❖ **TECNOLÓGICOS.-**

Los recursos tecnológicos que se utilizarán para el desarrollo del proyecto serán despreciables con respecto al beneficio que se obtendrá, debido a que estas tecnologías son en algunos de los casos gratuitas.

- ✓ Tecnología JAVA (**J2SDK 1.6.0**).-
- ✓ Editores Desarrollo Netbeans 6.8 Lenguajes Java.
- ✓ Jericho_Html
- ✓ Jboss 5.2

CONCLUSIONES Y RECOMENDACIONES

10.- CONCLUSIONES

Al finalizar el presente proyecto de tesis se ha llegado a las siguientes conclusiones:

- El objetivo principal se ha sido cumplido , el BUNL 1.0.1., ahora cuenta con un programa rastreador, planteado al inicio del proyecto: la instalación del Rastreador pemitira recorrer la Web buscando recursos de información y sus respectivas Urls, para incorporarlos en una base datos, reduciendo el tiempo en la recuperacion de información desde el buscador.
-
- En cuanto al proceso indexación, el rastreador utilizo un framework jericho-html 3.2.0 el mismo que acta como parse entre el servidor Web y el programa restreadorUNL, anizando el codigo html de cada link visitado, almacenadolos en memoria para que el rastreador complete la indexación asignando una categoria a cada links, de acuerdo a las considencias de los sinonimos de las categorias.

- En cuando a la recuperación de informacion el Motor de Busqueda BUNL cuenta con una interfaz amigable al usuraio, el cual podra realizar busquedas normales, avanzadas y por categorias, BUNL interpreta y recupera la información, en la base de datos, de acuerdo al requerimiento del usuario.
- La utilización del PageRanking nos permite medir la importancia o relevancia de una pagina en base al número y calidad de las páginas que la referencian, es decir una página que sea citada por 10 páginas tendrá menor pagerankin que otra página que sea citada por 1000 páginas al mometo de la indexacion, y visulizada en los resultados de la busqueda.
- En cuento a la inversion, BUNL no debe asumir los costos de uso del servidor, esto sumando a los servicios por el programa rastreadorUNL, ademas de haber utilizado software libre y open source para el desarrollo, han proveído al BUNL de una infraestructura y administracion muy sencilla a un costo relativammente nulo.
- En su totalilidad el proyecto BUNL empleo la plataforma Java para para su desarrollo, fruto de ello se logro que los

programas que lo componen, posean ventajas como ser multiplataforma, ser escalables y disponer de varias herramientas para las diferentes etapas del ciclo de vida del software, estas y otras ventajas adicionales han sevido para que Java se posicione como una platarforma lider durante la ultima decada, pero actualmente estas ventajas tambien son proporcionadas por otras plataformas de desarrollo, esto ha obligado a Java a evolucionar, la implementacion del presente proyecto trato de utilizar los conceptos mas recientes en el medio, de alli la utilizacion de Jboss 5.0.1. como sevidor Web.

- En cuanto a la ejecucion del programa del Rastreador Unl y la optimización de sus procesos e indexación de información hemos concluido que depende del ancho de banda del internet, ademas del diseño de cada pagina visitada a continuacion se presenta un cuadro comparativo de lo expuesto.

Servicio de Internet	Maquina	Ancho de banda	Links Indexados	Sitios de Web
UNL	Marca HP Memory 2Ghz Procesador Disco	4Ghz	4 por minnuto 10 por minuto 9 por minuto	http://www.unl.edu.ec. http://www.rincondelbago.com http://www.monografias .com

PORTA GSM	Marca HP Memory 2Ghz Procesador Disco	..	8 por minnuto 15 por minuto 13 por minuto	http://www.unl.edu.ec. http://www.rincondelbago.com http://www.monografias .com
NETPLUS	Marca HP Memory 2Ghz Procesador Disco	350 k	7 por minnuto 13 por minuto 10 por minuto	http://www.unl.edu.ec. http://www.rincondelbago.com http://www.monografias .com

11. RECOMENDACIONES

Con la finalizacion e inplementacion de Buscador BUNL, se hicieron visibles algunos aspectos que requieren de mejoras para obtener el rendimiento y las pretaciones adecuadas del sitema implemnetado a continuacion se realizara algunas recomendaciones par hacer frente a dichos aspectos.

- ✓ En la actualidad BUNL, especificamente el programa Rastrador requiere de du ejecución con acceso a internet es obligatoria, la Univerdidad debe permitir y dar los permisos par la indexaxion de Urls funcion necesaria para alimntar la base de datos, donde recupera la informacion BUNL.

- ✓ Con la infraestructura que ahora maneja la institucion es recomndable crear un departamnto o designar a una persona que cumpla las funciones de administrador del sistema y que se encargue de tares como manteniemiento, soporte, pespaldo de informacion, entre otras, ademas de estas tareas basicas debe encargarse de establecer politicas de seguridad par el servidor y esblecer siempre la actulizacion de BUNL.

- ✓ Se recomienda iniciar un plan progresivo de capacitación para los usuarios del BUNL y comtemplar como una politica de la Univerdidad instruir alos nuevos miembros en el uso de los servicios y herraminetas proveidos por el sitema implementado.

- ✓ Ademas BUNL con su programa rastreador deben ser monitoreados regularmente, primero es la administarcion de

- ✓ Dentro del programa académico que se dicta en la carrera de Ingeniería en Sistemas, se debería enfocar en el desarrollo de aplicaciones Web y redescubrir el mundo del Internet, que ayude al desarrollo del proceso de enseñanza-aprendizaje de nuestra carrera de Ingeniería En sistemas de la Universidad nacional de Loja.

BIBLIOGRAFIA

11.- Bibliografía

LIBROS:

- Larman, Craig. **UML Y PATRONES** Introducción al análisis y diseño orientado a objetos, PRENTICE HALL, México, 1999.

- Jacobson, Ivar y otros; **EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE**, PEARSON EDUCACIÓN, S.A., Madrid, 2000.

INTERNET:

- http://www.wikipedia.org/buscadores
- http://www.wikipedia.org/motoresdebuscadores
- http://www.wikipedia.org/servidores
- http://www.wikipedia.org/basedatos

- http://www.netbeans.org/community/index.html
- http://www.planetnetbeans.org
- www.webdelprogramdor.com/serial-port.htm
- http://www.rincondelvago.com/vb/gen/vb_misc/algorithms/article.php
- http://www.yoprogramo.com/Programming
- http://www.sans.com
- http://www.eforses.com/servicios/desrrollo
- http://www.media.mit.edu/piople/pattieas
- http://www.osmolatina.com
- http://www.mobtemplate.com
- http://www.navactiva.com/web/es/atic/doc/glosario/internet/
- http://www.cnti.gob.ve/index.php/
- http://www.tayabeixo.org/glosario/letra_b.htmteleenfermeria.iespana.es/
- http://www.cem.itesm.mx/dacs/publicaciones/logos/comunicarte/2007/febrero.html
- http://www.wikipedia.org/wiki/Buscador

ANEXOS

ANEXO 2: Formato de Encuesta.

UNIVERSIDAD NACIONAL DE LOJA

AREA DE ENERGIA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES.

Con la presenta encuesta dirigida a los estudiantes de la Carrera de Ingeniería en Sistemas se pretende ver el grado de aceptación del Proyecto de Tesis denominado:

"Diseño e implementación de un motor de búsqueda para la recuperación de información de internet, desde el portal de la Universidad Nacional de Loja"

Nombre de la Aplicación: **BUNL 1.1.0.**

DATOS PERSONALES

Nombres y Apellidos:……..
Número de Cédula:.....
Cargo:……..

1. **¿** Cuál sería su calificación acerca del Ambiente del Desarrollo Buscador BUNL 1.1.0 en lo que corresponde a su Entorno Gráfico.**?**

Excelente ( )
Muy Buena ( )
Buena ( )
Regular ( )

2. ¿ El buscador BUNL 1.1.0 al usuario es de fácil acceso e interfaz amigable**?**

Excelente ( )
Muy Buena ( )
Buena ( )
Regular ( )

3. **¿** De acuerdo a su apreciación cual sería su calificación de la distribución de botones de búsquedas.**?**

Excelente ( )
Muy Buena ( )
Buena ( )
Regular ( )

4. ¿ El esquema de presentar los resultados en el Buscador los considera.**?**

Rápida ( )
Lenta ( )
Muy Lenta ( )

5. **¿** La opción *categorías* permite la búsqueda por categorías usted como la calificaría.**?**

Excelente ( )
Muy Buena ( )
Buena ( )
Regular ( )

6. ¿ Cree que el Buscador UNL 1.1.0 es una aplicación de investigación necesaria para el usuario recupere información de Internet.**?**

Excelente ( )
Muy Buena ( )
Buena ( )
Regular ( )

7. **¿**En cuanto al rendimiento cómo calificaría a la aplicación. Buscador BUNL 1.1.0**?**

Excelente ( )
Muy Buena ( )
Buena ( )
Regular ( )

8. **¿**En cuanto al rendimiento cómo calificaría a la aplicación. Buscador BUNL 1.1.0**?**

Excelente ( )
Muy Buena ( )
Buena ( )
Regular ( )

9. **¿**Cuál sería su calificación en lo referente a los links visitados e indexado.**?**

Excelente ( )
Muy Buena ( )
Buena ( )
Regular ( )

10. **¿**La ejecución del buscador BUNL 1.1.0en la plataforma Windows como usted la calificaría.?

Excelente ( )
Muy Buena ( )
Buena ( )
Regular ( )

12. ¿La ejecución del buscador BUNL 1.1.0 en la plataforma Linux como usted la calificaría.?

Excelente ( )
Muy Buena ( )
Buena ( )
Regular ( )

Comentarios y sugerencias.

……
……
……
……
……
……

Gracias por su colaboración

ANEXO 3.