

**MODELO DE BÚSQUEDA Y RECUPERACIÓN SEMÁNTICA DE
INFORMACIÓN PARA GUIDESIT APLICANDO PROCESAMIENTO DEL
LENGUAJE NATURAL Y PREFERENCIAS DEL USUARIO**

**KATHERINE LOLAIN LORA GARIZABALO
JUAN CARLOS MARTINEZ LINDADO**



**UNIVERSIDAD
DE LA COSTA**
1970

**UNIVERSIDAD DE LA COSTA CUC
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BARRANQUILLA
2013**

**MODELO DE BÚSQUEDA Y RECUPERACIÓN SEMÁNTICA DE
INFORMACIÓN PARA GUIDESIT APLICANDO PROCESAMIENTO DEL
LENGUAJE NATURAL Y PREFERENCIAS DEL USUARIO**

**KATHERINE LOLAIN LORA GARIZABALO
JUAN CARLOS MARTINEZ LINDADO**

Trabajo de Grado presentado como requisito parcial para optar el título de:

INGENIERO DE SISTEMAS

Director: Ing. Jorge Eliecer Hernández Mercado

**UNIVERSIDAD DE LA COSTA CUC
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BARRANQUILLA
2013**



Barranquilla 17 de Septiembre de 2013

Señores: Corporación Universidad de la Costa (CUC)

Atención: Departamento Admisiones y Registros y Biblioteca

Asunto: Recepción de proyecto de grado

El día 17 de Septiembre de 2013, el estudiante **Katherine Lolaine Lora Garizabalo** identificado con **C.C. 1143436392**, **Juan Carlos Martinez Lindado** identificado con **C.C. 1140835428**, hicieron la entrega de los documentos y monografía de proyectos de grado junto con los anexos respectivos al programa de ingeniería de sistemas. A continuación se relaciona la información del proyecto de grado:

Título del proyecto: Modelo de Búsqueda y Recuperación Semántica de Información para GUIDESIT Aplicando Procesamiento del Lenguaje Natural y Preferencias del Usuario

Calificación: 4.4

Decisión del jurado evaluador: Aprobado

Tutor: Jorge Hernandez Mercado

Cotutor: Roberto Morales Ortega

El jurado calificó el proyecto presentado como un proyecto destacable dado los objetivos alcanzados con el desarrollo del trabajo de grado. Esta calificación incluye el desempeño durante todo el proyecto y la sustentación final asignada por los jurados que por su director de proyecto de grado lo que incluye:

Atentamente

Paola Ariza Colpas

Ing. Paola Patricia Ariza Colpas

Director de Programa

Ingeniería de Sistemas

NOTA DE ACEPTACIÓN

Nota

Ing. Jorge Hernández Mercado
Asesor

Ing. Alexis de la Hoz
Jurado 1

Ing. Jorge Sepúlveda
Jurado 2

Barranquilla, 17 de septiembre de 2013

DEDICATORIA

A Mis padres y mis hermanas por el amor,
Creencia y apoyo emocional, sin ustedes no sería posible
Mis avances en la trayectoria de mi existencia y educación intelectual y
empresarial.

Katherine Lora

A mis padres por su amor y apoyo incondicional en todo momento
En mi formación y a todas las personas que compartieron conmigo
Y aportaron de alguna u otra forma parte en mi formación
Integral como persona y como profesional.

Juan Carlos Martínez

AGRADECIMIENTOS

Agradecemos de todo corazón a Dios todopoderoso por ayudarnos a formarnos poco a poco a lo largo de nuestra trayectoria educativa.

A la universidad de la costa, CUC, en especial al programa de ingeniería de sistema, y la planta docente, quienes durante los años de nuestro estudio de educación superior suplieron ser guías y personas de confianza ayudando a impartir sus conocimientos.

Al Ingeniero Jorge Hernández en su condición de asesor, quien nos brindó todo el apoyo y nos motivó a investigar y a enfocarnos en la elaboración total de nuestro proyecto.

Al profesor Víctor Montaña por su confianza de apoyar y prestar su libro para la investigación y presentación en mejora de este trabajo.

A todas aquellas personas y amistades que de una u otra manera contribuyeron a la ejecución de este trabajo.

RESUMEN

El artículo presenta un modelo de búsqueda de información para GuidesIT usando las tecnologías de la web semántica. GuidesIT es un conjunto de guías diseñadas para la adecuada planificación de un buen gobierno de TI, basado en el alineamiento del modelo COBIT y norma ISO 27000. Como resultado se diseñó una ontología que permite la construcción de una base de conocimiento para representar GuidesIT usando OWL. GuidesIT se encuentra en papel, por lo que el objetivo es sistematizarla. La metodología aplicada para este proyecto se basa en la asociación de Methontology con el proceso unificado. Además, se desarrolló un prototipo para validar el modelo de búsqueda incluyendo aspectos técnicos como el procesamiento del lenguaje natural y las Preferencias del usuario con la intención de alcanzar un entorno personalizado. En conclusión se realizó una investigación en busca de un desarrollo óptimo de GuidesIT.

Palabras claves: ontología, base de conocimiento, web semántica, OWL, Methontology, proceso unificado, GuidesIT.

ABSTRACT

The article presents a model of search of information for GuidesIT using the technologies of the semantic web. GuidesIT is a set of guides designed for a suitable planning of a good IT government, based on the alignment of the model COBIT and norm ISO 27000. As result we designed an ontology that allows the construction of a knowledge base to represent GuidesIT using OWL. GuidesIT are in books, therefore the target is to systematize it. The methodology applied for this project is based on the association of Methontology with the unified process. Also, a prototype developed to validate the model of search including technical aspects as the prosecution of the natural language and the Preferences of the user with the intention of reaching a personalized environment. In conclusion an investigation was realized in search of an ideal GuidesIT development.

Keywords: ontology, knowledge base, semantic web, OWL, Methontology, unified process, GuidesIT.

CONTENIDO

	Pág.
1 INTRODUCCIÓN.....	19
1.1 DESCRIPCIÓN DE LA COMPOSICIÓN DEL PROYECTO.....	19
1.2 PARTICIPANTES DEL PROYECTO.....	20
CAPITULO I DEFINICIÓN DEL PROBLEMA.....	21
2.1 INTRODUCCIÓN.....	21
2.2 PLANTEAMIENTO DEL PROBLEMA.....	21
2.2.1 Área Temática.....	21
2.2.2 Formulación del problema.....	21
2.2.3 Identificación y Delimitación del problema.....	22
2.2.3.1 Delimitación Temporal.....	22
2.2.3.2 Delimitación Espacial.....	22
2.3 FINES DEL PROYECTO.....	23
2.3.1 Justificación.....	23
2.3.1.1 Beneficios tangibles.....	23
2.3.1.2 Beneficios Intangibles.....	23
2.4 OBJETIVOS.....	24
2.4.1 Objetivos Generales.....	24
2.4.2 Objetivos Específicos.....	24
2.5 METODOLOGIA USADA.....	24

2.5.1 Methontology.....	25
2.5.2 Proceso Unificado.....	25
2.5.3 Fases del Proyecto.....	26
2.5.3.1 Gestión.....	27
2.5.3.2 Especificación.....	29
2.6 ESTADO DEL ARTE.....	30
2.7 PROPUESTA DE SOLUCIÓN.....	32
CAPITULO II ESPECIFICACIÓN DE REQUISITOS.....	32
3.1 INTRODUCCIÓN.....	32
3.1.1 Alcance y Propósito.....	32
3.1.2 Ámbito.....	33
3.1.3 Definiciones, Acrónimos y Abreviaturas.....	33
3.1.4 Referencias.....	34
3.2 DESCRIPCIÓN GENERAL.....	35
3.2.1 Perspectiva del Producto.....	35
3.2.2 Funciones del Producto.....	35
3.2.3 Características del Usuario.....	35
3.2.3.1 Perfil del Usuario.....	35
3.2.3.2 Jerarquía de Usuarios.....	36
3.2.4 Restricciones Generales.....	36

3.2.4.1 Políticas Regulatoras.....	36
3.2.4.2 Limitaciones de Hardware.....	36
3.2.4.3 Funciones de Control.....	36
3.2.4.4 Requisitos del Lenguaje.....	36
3.2.4.5 Credibilidad de la Aplicación.....	36
3.2.4.6 Consideraciones de Seguridad.....	37
3.2.4.7 Dependencias.....	37
3.3 REQUERIMIENTOS ESPECÍFICOS.....	37
3.3.1 Requerimientos Funcionales.....	37
3.3.1.1 Ingreso a la Aplicación.....	37
3.3.1.2 Gestión del Usuario y la KB.....	38
3.3.1.3 Requisitos de Desarrollo.....	40
3.3.1.4 Atributos del Software del Sistema.....	41
3.3.2 Obligaciones del Diseño.....	42
3.3.2.1 Estándares Cumplidos.....	42
3.3.3 Requisitos no Funcionales.....	42
CAPITULO III ANÁLISIS.....	42
4.1 INTRODUCCIÓN.....	42
4.2 DIAGRAMA DE CLASES.....	43
4.2.1 Diccionario de Entidades.....	44

4.3 CASOS DE USO.....	52
4.3.1 Resumen Actores.....	52
4.3.1.1 Actores.....	52
4.3.2 Casos de Uso del Usuario en General.....	53
4.3.3 Casos de Uso del Usuario Registrado.....	53
CAPITULO IV DISEÑO DEL SISTEMA.....	54
5.1 INTRODUCCION.....	54
5.2 ARQUITECTURA DEL SISTEMA.....	54
5.3 CAPAS DE LA APLICACIÓN.....	55
5.3.1 Descripción de la Arquitectura.....	55
5.3.2 Diagrama Entidad-Relación.....	55
5.4 Diagrama de Interfaz.....	56
5.4.1 Interfaz Principal.....	56
5.4.2 Interfaz para Buscar.....	56
5.4.2 Interfaz para las Preferencias del Usuario.....	57
CAPITULO V IMPLEMENTACIÓN Y DESARROLLO.....	57
6.1 INTRODUCCION.....	57
6.2 CONCEPTOS UTILIZADOS.....	57
6.2.1 Ingeniería del Conocimiento.....	57
6.2.2 Base De Conocimientos.....	58

6.2.3 Inteligencia Artificial (IA).....	58
6.2.4 Web Semántica.....	59
6.2.5 Ontologías, OWL y Protégé.....	61
6.2.6 Recuperación de la Información (RI).....	63
6.2.7 Procesamiento del Lenguaje Natural (PLN).....	64
6.2.8 Algoritmos Genéricos.....	66
6.2.8.1 Levenshtein	66
6.2.8.2 LetterpairSimilarity.....	67
6.2.9 Preferencias del Usuario.....	67
6.3 TECNOLOGIAS UTILIZADAS.....	68
6.3.1 Interfaz De Usuario.....	68
6.3.1.1 XML Y HTML.....	68
6.3.1.1.1 CSS.....	70
6.3.1.2 Controles.....	71
6.3.1.2.1 JavaScript.....	71
6.3.1.2.2 JQuery.....	71
6.3.1.2.3 Ajax.....	71
6.3.1.2.4 Servlet.....	72
6.3.1.2.5 JSP.....	72
6.3.1.3 MODELO.....	73



6.3.1.3.1 Java.....	73
6.3.1.3.2 Librerías.....	74
6.3.1.3.3 OpenNLP 1.5.3.....	74
6.3.1.3.4 OWLAPI 3.4.4.....	75
6.3.1.3.5 JESXML.....	75
6.3.1.3.6 Json-LIB.....	76
6.3.2 SOFTWARE.....	77
6.3.2.1 Netbeans 7.3.1.....	77
6.3.2.2 Protégé 4.1.....	77
6.3.2.3 Glassfich Server 3.1.2.2	78
6.3.3 Datos Y Metadatos.....	78
6.4 DESCRIPCIÓN DEL PROYECTO.....	79
6.4.1 Implementación de la Ontología.....	79
6.4.2 Modelo de Búsqueda y Recuperación Semántica.....	81
6.4.3 Componente de Consulta.....	81
6.4.4 Componente de Abstracción de Conceptos.....	82
6.4.5 Componente Comparativo de Conceptos.....	83
6.4.6 Componente de Visualización de la Información.....	86
6.4.7 Componente de Gestión de Preferencias del Usuario.....	88
6.5 Escenario Ilustrativo.....	89

CAPITULO VI EVALUACIÓN Y PRUEBAS	91
7.1 INTRODUCCION.....	91
7.2 VALIDACIONES.....	91
7.2.1 Validación de CSS.....	91
7.2.2 Validación de Enlaces.....	92
7.2.3 Validación de Navegadores.....	93
7.3 CASOS DE PRUEBA.....	94
8 CONCLUSIONES.....	98
8.1 Conclusiones y Trabajo Futuro.....	98
9. BIBLIOGRAFIA.....	99

ANEXOS

ÍNDICE DE GRÁFICOS

	Pág.
Gráfico N°1 Fases del Proyecto	26
Gráfico N°2 Cronograma de Actividades.....	28
Gráfico N°3 Jerarquía de Usuarios.....	36
Gráfico N°4 Diagrama de Clases del Proyecto	43
Gráfico N°5 Actores.....	52
Gráfico N°6 Usuario en General (No registrado).....	53
Gráfico N°7 Usuario Registrado	53
Gráfico N°8 Programación por Capas	54
Gráfico N°9 Modelo Entidad- Relación de la Base de Conocimiento.....	55
Gráfico N°10 Vista Preliminar de la Ventana Principal.....	56
Gráfico N°11 Vista Preliminar de la Ventana de Búsqueda.....	56
Gráfico N°12 Preferencias del Usuario Acerca de los Dominios y Procesos de GuidesIT.....	57
Gráfico N°13 Jerarquía de las Clases Realizado en Protégé 4.1.....	80
Gráfico N°14 Esquema General de la Ontología Realizado en Protégé 4.1.....	80
Gráfico N°15 Modelo para la Búsqueda Semántica de GuidesIT.....	81
Gráfico N°16 Vista preliminar de la Ventana de Búsqueda.....	82
Gráfico N°17 Algoritmo de Recorrido.....	84

Gráfico N°18 Algoritmo Comparativo de Palabras del Usuario y Guías Encontradas.....	85
Gráfico N°19 Visualización de Resultados de Búsqueda.....	86
Gráfico N°20 Visualización Completa de una Guía Específica.....	87
Gráfico N°21 Componente de Gestión de Preferencias del Usuario.....	88
Gráfico N°22 Validación del CSS (login.css) Relacionado al Módulo de Login del Sistema.....	91
Gráfico N°23 Validación del CSS (styles.css) Relacionado al Módulo de Búsqueda y Preferencias del Usuario.....	91
Gráfico N°24 Validación del CSS (default.css) relacionado al componente de fecha de nacimiento del módulo de preferencias del usuario.....	92
Gráfico N°25 Validador de Enlaces del Sistema.....	92
Gráfico N°26 Ejecución de la Aplicación en Internet Explorer v10.....	93
Gráfico N°27 Ejecución de la Aplicación en Google Chrome v28.....	93
Gráfico N°28 Ejecución de la Aplicación en Mozilla Firefox v22.....	94

ÍNDICE DE TABLAS

	Pág.
Tabla N°1 Resumen de Actividades en las Fases del Proyecto.....	27
Tabla N°2 Actividades de Administración del proyecto.....	28
Tabla N°3 Especificación de Requerimientos.....	29
Tabla N°4 Requerimiento Funcional: Ingreso a la Aplicación.....	37
Tabla N°5 Requerimiento Funcional: Creación de Usuario.....	38
Tabla N°6 Requerimiento Funcional: Modificación de Usuario.....	38
Tabla N°7 Requerimiento Funcional: Búsqueda de Información de la Ontología.....	39
Tabla N°8 Requerimiento Funcional: Recopilación de la Información de la Ontología.....	39
Tabla N°9 Requerimiento Funcional: Cantidad de Usuarios Concurrentes.....	40
Tabla N°10 Requerimiento Funcional: Cantidad de Información Almacenada.....	40
Tabla N°11 Requerimiento Funcional: Disponibilidad del Sistema.....	41
Tabla N°12 Requerimiento Funcional: Seguridad de la Contraseña.....	41
Tabla N°13 Descripción Clase: GestionOntologia.....	44
Tabla N°14 Descripción Método: recorrerOntologia.....	45
Tabla N°15 Descripción Método: obtenerValorAtributo.....	46

Tabla N°16 Descripción Método: obtener_guia.....	46
Tabla N°17 Descripción Método: getResultadosBusqueda.....	47
Tabla N°18 Descripción Clase: ProcesamientoPalabras.....	47
Tabla N°19 Descripción Método: Tagging.....	48
Tabla N°20 Descripción Método: GradodeAceptabilidadConceptosClaves.....	48
Tabla N°21 Descripción Método: eliminarRepetidos.....	49
Tabla N°22 Descripción Método: remover_acentos.....	49
Tabla N°23 Descripción Clase: GestionUsuarios.....	50
Tabla N°24 Descripción Método: validarUsuario.....	50
Tabla N°25 Descripción Método: crearUsuario.....	50
Tabla N°26 Descripción Método: modificarInfo.....	51
Tabla N°27 Descripción Método: consultarPreferenciasUsuario.....	51
Tabla N°28 Características de la librería JSON-LIB.....	76
Tabla N°29 Casos de Prueba I: CO_001 y CO_002.....	94
Tabla N°30 Casos de Prueba II: CO_003.....	95
Tabla N°31 Casos de Prueba II: CO_004 y CO_005.....	96
Tabla N°32 Casos de Prueba II: CO_006 y CO_007.....	97

INTRODUCCIÓN

El desarrollo de las tecnologías de la información invade nuestro quehacer cotidiano. El sector de administración empresarial tampoco se escapa a la integración de las TI como solución a las necesidades de implantar herramientas sistematizadas.

Este documento presenta el trabajo realizado en el proyecto de grado del programa de ingeniería de sistemas. El proyecto consiste en el desarrollo de un “Modelo De Búsqueda Y Recuperación Semántica De Información Para GuidesIT” mediante las herramientas y métodos para la construcción de un sistema informático. El conocimiento se representa mediante las técnicas de la Ingeniería del Conocimiento, las cuales comprenden: la adquisición del conocimiento, la familiarización con el dominio, el análisis, el diseño de la solución y la validación del conocimiento, hasta que el conocimiento acumulado de un dominio sea traducido en un código probado y refinado. La base de conocimiento del estudio realizado por los expertos se refiere a la articulación del Modelo COBIT con las Norma ISO 2000 para la exitosa implantación de un Gobierno de TI.

1.1 DESCRIPCIÓN DE LA COMPOSICIÓN DEL PROYECTO

En el capítulo I se establece la *Definición del Problema*, donde muestra la definición, planteamiento y delimitación del problema que motivo al equipo a desarrollar una solución para ella. Así mismo, plantea la justificación, los beneficios y los objetivos del proyecto.

El en capítulo II constituye la *Especificación de Requisitos*, estableciendo todo lo relacionado con la descripción general y restricciones del sistema de una forma detallada. Se redactó de una manera global una primera visión del proyecto donde señalamos los requisitos que debía cumplir.

En el capítulo III *Análisis*, conforma la parte conceptual, que se representa en los diagramas de clases y diagramas de casos de uso En esta etapa se especifica qué debe hacer la aplicación pero no cómo debe hacerlo.

En el capítulo IV establece el *Diseño* donde se utilizaron los elementos y modelos obtenidos durante la etapa de análisis con el fin de ir transformándolo en mecanismos que puedan ser utilizados en un entorno web.

En el capítulo V se presenta la *Implementación y Desarrollo* del Proyecto, Se utilizaron los elementos obtenidos en el diseño para permitir la elaboración del producto o prototipo funcional, en el que luego de esto puede ser sometido a pruebas. En esta etapa se establece el desarrollo del plasmado en las etapas iniciales como análisis y diseño.

En el capítulo VI *Evaluación y Pruebas* busca obtener fundamentos teóricos de todo lo utilizado en el desarrollo del software, Así mismo esta etapa se centró en la correcta comprobación del funcionamiento de software mediante las pruebas pertinentes.

Luego de la descripción de los capítulos se da por terminado el proyecto de grado, En el capítulo VII muestra la *Conclusión* del proyecto y la *Bibliografía* utilizada para la realización del mismo.

1.2 PARTICIPANTES DEL PROYECTO

Para llevar adelante el presente trabajo se cuenta con los siguientes participantes:

- Ingenieros de Sistemas:

Katherine Lora y Juan Carlos Martínez, autores del presente Trabajo.

- Usuarios especialistas que se integrarán al proyecto, en calidad de usuarios especialistas, los cuales son las siguientes personas:

Jorge Hernández Mercado – Ingeniero de sistemas, Máster en MIAC - Universidad de Girona - Tutor de la tesis.

Víctor Montaña Ingeniero de sistemas y auditor de sistemas de información - Universidad de la Costa - Asesor conceptual

CAPITULO I

DEFINICIÓN DEL PROBLEMA

2.1 INTRODUCCIÓN

El presente capítulo plasma el dominio general y específico del problema. En este capítulo también presenta los inconvenientes que pueden presentarse a la hora de diseñar un modelo semántico para la búsqueda y recuperación semántica de información que se ajuste a GuidesIT

Consecutivamente se establecen los diferentes objetivos como así también la metodología a utilizar.

2.2 PLANTEAMIENTO DEL PROBLEMA

2.2.1 Área Temática

Nuestra Investigación está enfocada en el campo de la inteligencia artificial y el uso de herramientas de la web semántica hacia información relacionada en el área de auditoría de sistemas de información en las organizaciones modernas.

2.2.2 Formulación del Problema

Gran parte del éxito de las organizaciones modernas depende de la adecuada gestión de la tecnología informática. Por ello los expertos en administración empresarial o auditores recomiendan la necesidad de implantar procesos de TI que ayuden a la seguridad de la información. Esto es lo que se denomina Gobierno de TI. Partiendo de lo anterior en (Montaño,2010)¹ presentaron la creación de unas guías llamadas GuidesIT para la estructuración de un modelo de gobierno de TI basado en el modelo COBIT y la norma ISO 27000.

El modelo COBIT y La Norma ISO 27001 constituyen independientemente las mejores prácticas para cada uno de los aspectos que dieron origen a su concepción y desarrollo, pero logrando la articulación de ellos se logrará conformar un modelo para gobierno de TI robusto y de total satisfacción para los requerimientos de las organizaciones modernas.

(1) Montaño, V. (2010) "Guías para la implantación de un efectivo gobierno de TI" Educosta, 730 p.17.

Sin embargo, Algunas desventajas vista relacionada a estas guías son las siguientes:

1. Se encuentran plasmadas solo en libro de forma manual.
2. El número de guías es de 256 lo cual hace que la búsqueda sea exhaustiva.
3. Su búsqueda debe realizarse de forma escalada por encontrarse los contenidos alineados a normativas y modelos específicos.

De acuerdo al Conocimiento derivado del Auditor de la investigación y el conocimiento plasmado en los modelos surge la creación de un modelo de búsqueda y recuperación semántica de información apoyándose en las bases de conocimiento facilite a los auditores la búsqueda de la guía más apropiada de acuerdo a las necesidades que tienen en su área y a la vez también facilite al experto del conocimiento actualizar la base de conocimiento.

De esta necesidad surge una pregunta general:

¿Cómo la creación de un modelo de búsqueda y recuperación de información facilitaría la búsqueda de las guías?

Y de estas se deriva las siguientes preguntas:

¿Qué tipo de herramientas se deben considerar para poder construir una base de conocimientos a partir de una ontología de dominio específica?

¿Qué mecanismos de inferencia podremos utilizar para mejorar las búsquedas en nuestra base de conocimientos y encontrar los contenidos solicitados?

¿Qué diferentes herramientas de aplicación se deben utilizar para contribuir en el diseño y desarrollo de un sistema de búsqueda que permita acceder a la base de conocimiento?

2.2.3 Identificación y Delimitación del Problema

2.2.3.1 Delimitación Temporal

El proyecto se desarrolló durante el segundo semestre de 2012, primer semestre y parte del segundo de 2013.

2.2.3.2 Delimitación Espacial

El Proyecto se realizó en la universidad de la costa, de la ciudad de barranquilla – Colombia.

2.3 FINES DEL PROYECTO

2.3.1 Justificación

De acuerdo al Impacto que ha tenido el libro se concluyó que estas deberían ser Utilizadas en forma de consulta debido a la estructura que tiene el libro.

Conforme a lo anterior, se analiza que por ser estas guías una fuente de conocimiento tan relevante para el ámbito de la seguridad de la información, se hizo imprescindible llevar este resultado a un entorno Web, inclinándose a las nuevas tecnologías por medio de herramientas de búsqueda.

La idea es permitir al usuario por medio de una aplicación web y junto a una base de conocimiento que contenga la información asociada a las guías, pueda escoger el tipo de foco clave que quisiera dar una auditoria, el tipo de proceso del foco y subprocesos asociados, y según lo escogido, la aplicación le desglosara las guías respectiva para poder conocer los diversos pasos a tener en cuenta con el fin de realizar una efectiva auditoria y establecer un buen dominio de ti en el área o foco clave de la organización que escogió.

2.3.1.1 Beneficios tangibles

El beneficio más importante es el acceso óptimo de la información en una forma más oportuna, ya que al encontrarse en forma sistematizada las disminuciones en el tiempo ayudan a cumplir las tareas específicas para el interesado.

2.3.1.2 Beneficios Intangibles

El resultado de ofrecer estas guías en un formato de búsqueda digital ofrece a los expertos en administración empresarial la necesidad de implantar procesos coordinados entre la alta gerencia y el área de las TI para movilizar sus recursos de la forma más eficiente en respuesta a requisitos regulatorios, operativos o del negocio apoyándose en la información que se encuentra en la herramienta de búsqueda e incrementar la exactitud en las búsquedas.

2.4 OBJETIVOS

2.4.1 OBJETIVOS GENERALES

- El objetivo del trabajo es desarrollar un software denominado “Modelo de búsqueda y recuperación semántica de información para GuidesIT”

2.4.2 OBJETIVOS ESPECIFICOS

- Proporcionar una herramienta eficaz al usuario en la búsqueda de GuidesIT.
- Identificar los mecanismos más adecuados para lograr realizar mejores inferencias para la búsqueda de información.
- Proporcionar una estructura de conocimiento adecuada para la incorporación de los contenidos de GuidesIT.
- Evaluar el comportamiento consulta del software propuesto mediante casos de prueba.

2.5 METODOLOGIA USADA

La metodología a aplicar en el proyecto es producto de la combinación de algunas de las fases de Methodology² con otras del proceso unificado para ajustarse a las diferentes actividades asociadas a la construcción del prototipo de GuidesIT y su integración con la ontología asociada. La metodología de Methontology fue seleccionada por ser la más acogida por la comunidad de desarrolladores de la Web semántica para desarrollar ontologías y, Así mismo el marco de desarrollo de software, proceso unificado por la visibilidad de sus fases y flujos de trabajo contenidos en él mismo. (Abreo, Padilla, 2012)³

(2) O. Corcho. (2001, Oct.), Methodologies tools and languages for building ontologies. Where is their meeting point?, Universidad Politécnica de Madrid, España. [En línea], Recuperado en: http://www.aegean.gr/culturaltec/Kavakli/MIS/papers/Corcho_2003.pdf

(3) Abreo, C., Padilla, C. (2012) “Prototipo para recuperación semántica de información de proyectos de ingeniería de sistemas con base en un repositorio ontológico” Vol.3, núm. 1, Enero- Julio, pp 109-118.

2.5.1 Methontology⁴

Esta metodología fue desarrollada por el grupo de Ingeniería Ontológica de la Universidad Politécnica de Madrid. Methontology permite la construcción de ontologías a nivel de conocimiento.

Esta metodología permite construir ontologías en el nivel de conocimientos, y tiene sus raíces en las actividades identificadas por el proceso de desarrollo de software propuesto por la organización IEEE⁵ en este caso la IEEE – 1074 (Corcho, 2001)⁶ y en otras metodologías de ingeniería de conocimientos⁷. Es una metodología que engloba un conjunto de actividades con guías específicas para llevar a cabo el desarrollo de una ontología a nivel de conocimiento

METHONTOLOGY proporciona guías sobre cómo llevar a cabo el desarrollo de la ontología a través de las actividades de especificación, conceptualización, formalización, implementación y mantenimiento, gestión (planificación, control y aseguramiento de la calidad) y soporte (adquisición de conocimientos, integración, evaluación, documentación y gestión de la configuración).

2.5.2 Proceso Unificado⁸

Este nombre es usado para describir el proceso genérico que incluye aquellos elementos que son comunes a la mayoría de los refinamientos existentes. También permite evitar problemas de copyright con RUP, ya que es marca registrada por IBM. Proceso unificado incorpora buenas prácticas de desarrollo, para ser adaptable a un amplio rango de proyectos.

(4) Gómez-Pérez A., Fernández-López M. y Corcho M. (2004), *Ontological Engineering*, Springer Verlag, London, Mayo.

(5) IEEE (1996) IEEE Standard for Developing Software Life Cycle Processes.

(6) O. Corcho. (2001, Oct.), *Methodologies tools and languages for building ontologies. Where is their meeting point?*, Universidad Politécnica de Madrid, España. [En línea], consultado en Agosto 12 de 2011, disponible en: http://www.aegean.gr/culturaltec/Kavakli/MIS/papers/Corcho_2003.pdf

(7) Gómez-Pérez A, Juristo N, Montes C, Pazos J (1997) *Ingeniería del Conocimiento: Diseño y Construcción de Sistemas Expertos*. Ceura, Madrid, Spain

(8) Absolute astronomy, *Unified process, absolute astronomy*. [En línea], recuperado en: http://www.absoluteastronomy.com/topics/Unified_Process

Para una mejor organización, el proceso unificado agrupa las iteraciones en fases que facilitan la administración del proyecto las cuales son inicio o planificación, elaboración de la arquitectura, construcción y fase de transición o cierre, y las principales iteraciones contenidas son: modelamiento del negocio, requisitos, análisis y diseño, implementación, pruebas y despliegue.

2.5.3 Fases Del Proyecto

De Methontology fueron tomadas 2 fases en particular: La gestión y la planificación, especificación. Del proceso unificado se siguieron todas las fases. A continuación como se muestra en el gráfico N°1 se presenta los ciclos del proceso unificado y su interacción con las demás actividades de Methontology tomando la gestión del proyecto como eje central (Abreo, Padilla, 2012)⁹

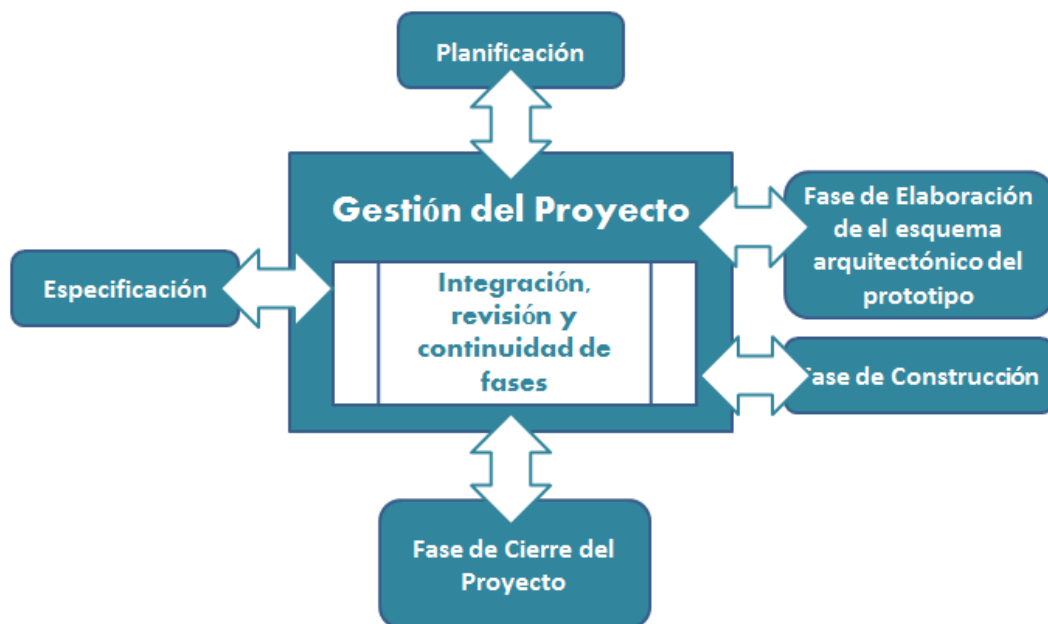


Gráfico N°1

Consecuente de lo anterior, se describe brevemente en la siguiente el resumen de las actividades de las fases del proyecto expuestas en el gráfico anterior.

(9) Abreo, C., Padilla, C. (2012) "Prototipo para recuperación semántica de información de proyectos de ingeniería de sistemas con base en un repositorio ontológico" Vol.3, num 1, Enero- Julio, pp 109-118.

Fase	Descripción
Gestión y Planificación →	La idea es llevar a cabo controles sobre las actividades progresivamente y validar que se fueran realizadas dentro del tiempo determinado.
Especificación →	Permite determinar porque se construye la ontología, cuál será su uso, y quienes serán sus usuarios finales
Elaboración del esquema arquitectónico →	Se definió la arquitectura del prototipo con el fin de construirlo de manera iterativa e incremental
Construcción del prototipo →	Se precede a la generación del código fuente del prototipo y su integración con la ontología, tomando como base la arquitectura diseñada y los escenarios de consulta definidos.
Transición y/o cierre del proyecto →	Se procedió al chequeo correspondiente de cumplimiento de actividades para dar cierre al proyecto.

Tabla N° 1

2.5.3.1 Gestión

Las metodologías de desarrollo Ontológico comprenden un conjunto establecido de principios, procesos, prácticas, métodos y actividades usados para el diseño, construcción, evaluación y puesta en producción de las mismas (Gašević, Dejuric, Devežvic, 2009.)¹⁰ . A continuación se presenta las actividades de gestión en la siguiente tabla:

(10) Gašević D., Dejuric D., Devežvic V., Model Driving Engineering and Ontology Development, Springer Verlag, Berlin, Febrero, 2009.

Actividades de Administración del proyecto	
Actividades	Objetivos
Planificación →	<ul style="list-style-type: none"> • Abstracción del conocimiento del libro: “Guías para la implantación de un efectivo gobierno de TI” • Identificación de cada uno de los elementos ontológicos de GuidesIT como lo son: las clases, instancias, propiedades y atributos para el diseño de la ontología. • Creación de la base de conocimientos de la aplicación. • Creación de la estructura base del aplicativo de consulta • Incorporación de herramientas de PLN y Preferencias del usuario. • Creación del componente para la interacción del usuario con el sistema.
Control →	<ul style="list-style-type: none"> • Se hizo el control mediante el uso de cronogramas de actividades (ver Gráfico N°2)
Calidad →	<ul style="list-style-type: none"> • Se hicieron revisiones estratégicas para mostrar avances y mejoras. • Investigaciones de tecnologías o reutilización de herramientas para garantizar un óptimo desarrollo.

Tabla N° 2

A continuación se puede ver el desarrollo de las actividades por medio de un cronograma de actividades:

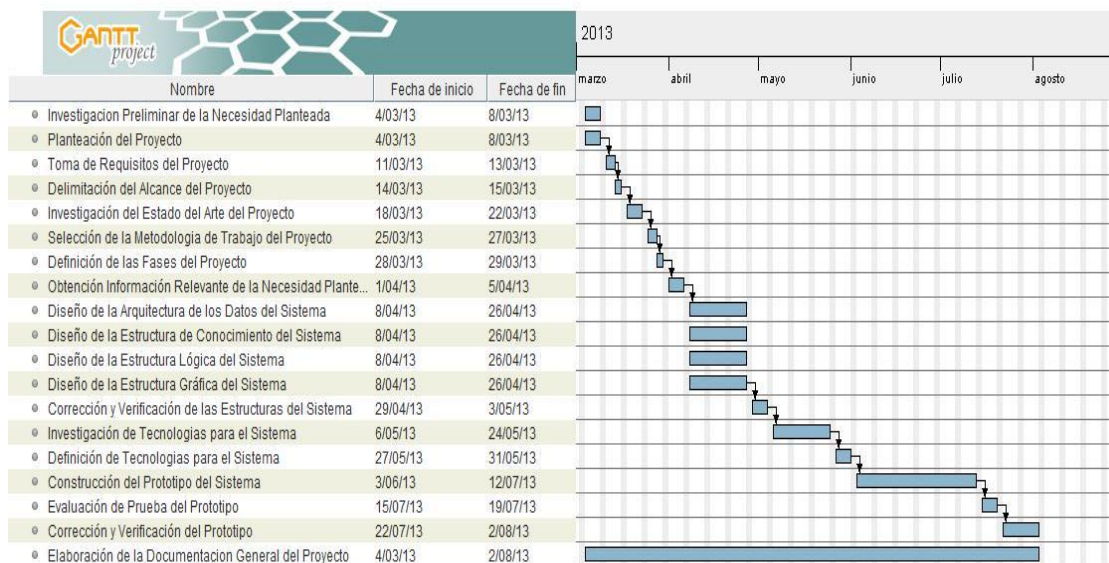


Gráfico N°2

2.5.3.2 Especificación

Para la especificación de la Ontología se crea una plantilla con el nombre Documento de Especificación de Requerimientos. En la siguiente tabla se muestra dicha plantilla para esta Ontología.

Documento de especificación de requerimientos	
Dominio →	GuidesIT
Fecha →	04/03/2013
Desarrollador(es) →	Juan C. Martínez Lindado, Katherine L. Lora Garizabalo
Propósito →	Construir un modelo de búsqueda y recuperación semántica de información para GuidesIT con el fin facilitar la búsqueda de contenidos asociados a un buen gobierno de TI para las organizaciones modernas.
Nivel de formalidad →	Formal
A quien va dirigido →	Audidores y consultores
Fuentes de conocimiento →	Expertos en auditoria de sistemas de información de la Universidad de la costa, CUC. Montaño, 2010., "Guías para la implantación de un efectivo gobierno de TI" Educosta, 730 p.17.

Tabla N° 3

2.6 ESTADO DEL ARTE

En la actualidad aunque no se ha masificado el uso de buscadores semánticos ya se tienen algunos funcionando y sirven de referencia para el futuro de la búsqueda de información (Moreno, Sánchez, 2012)¹¹.

En (González, Conesa, 2011)¹², encontramos una propuesta de modelo de ontologías para cada apartado de gestión de los procesos de control relacionados con ITIL v3, los cuales engloban la gestión de la configuración, la gestión de los cambios y la gestión de la entrega y despliegue por dichas normas. Se apoya en semantic mediawiki y smw como herramientas para implementar las ontologías. El conocimiento de dominio es ITIL¹³ junto a sus procesos de control. Se habla entonces de ontologías para ITIL. Maneja también la estructura ontológica RDF/XML en las propiedades y las entidades de su ontología, además de SPARQL¹⁴, para consultar los contenidos que se quieren tener conocimiento.

En (Isaza, Castillo, López, 2009)¹⁵, propone diseñar e implementar un modelo ontológico de detección y prevención de intrusiones basado en sistemas multi-agente e inteligencia computacional, campo referido al aseguramiento de la información. Su ontología utilizó una lógica descriptiva para representar conocimiento a partir de terminologías de dominio de una manera estructurada y con un formalismo entendido.

(11) Moreno C. and Sanchez Y. (2012) "prototipo de buscador semántico aplicado a la búsqueda de libros de ingeniería de sistemas y computación en la biblioteca Jorge Roa Martínez de la Universidad tecnológica de Pereira", p.20.

(12) González, Conesa. (2011) "AgilITIL: Creación de una Wiki Semántica de soporte a los procesos de control", Universidad Oberta de Catalunya.

(13) ITIL – Gestión de Servicios TI (s.f). Recuperado en: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/que_es_ITIL/que_es_ITIL.php

(14) SPARQL, SPARQL Query Language for RDF, Recuperado en: <http://www.w3.org/TR/rdf-sparql-query/>

(15) Isaza, G., A. Castillo et al. (2008). *Modelo de Detección de Intrusos basado en Sistemas Multi-agente, Inteligencia Computacional y representaciones Ontológicas*. X RECSI - X Reunión Española sobre Criptología y Seguridad de la Información. Salamanca (España), Salamanca. PP 271-281.

En (Rojas, Montilva, Barrios 2009)¹⁶ presenta la aplicación de un método para desarrollar ontologías durante el modelado del dominio en el contexto del servicio de soporte en tecnologías de información. Propone el diseño de una estructura de conocimiento que permita almacenar toda la experiencia e información necesaria para ofrecer una línea de soporte ágil y eficaz al cliente en el soporte post venta. La aplicación del método planteado facilita la construcción de un modelo conceptual que identifica y define los conceptos y relaciones bases del Servicio de Soporte en TI. Aquí exponen la definición del alcance del dominio, la lista de aspectos, la lista de conceptos y un modelo conceptual que relaciona dichos conceptos.

En (Lasheras, 2011)¹⁷ traza un marco de representación de catálogo de requisitos de seguridad de la información con el objetivo de mejorar la seguridad en los sistemas de información detectando inconsistencias y consiguiendo procesamiento semántico en el análisis de requisitos. El marco de representación consistía en una ontología que representa el conocimiento sobre el catálogo de requisitos de seguridad, con su meta información asociada, relaciones, restricciones, axiomas y reglas de inferencia. Especificaron la combinación del lenguaje para diseñar ontologías usando la herramienta de diseño ontológico Protégé¹⁸.

(16) Rojas, M., Montilva, J.y Barrios, J. (2009). "OntosMD: Un método para hacer ontologías durante el modelado de negocios". Revista Colombiana de Tecnologías de Avanzada, Vol.1, Número 13,2009.

(17) Lasheras (2011). "Marco de Representación y reúso de requisitos de seguridad", Tesis Doctoral, Universidad de Murcia, Departamento de Informática y Sistemas.

(18) Protégé, open source ontology editor and knowledge-base framework, Recuperado en <http://protege.stanford.edu/>

2.7 PROPUESTA DE SOLUCIÓN

No orientamos hacia un modelo de representación y búsqueda semántica para GuidesIT, que haga uso de una ontología específica, ya que por ser éstas herramientas para formalizar, conceptualizar y compartir el conocimiento en dominios complejos, optimiza la búsqueda. Junto con el procesamiento del lenguaje natural, una base de conocimiento y un Preferencias del usuario (PU), se busca recuperarlas de la manera más eficiente posible. Por consiguiente, se piensa desarrollar un prototipo en el cual albergue los conceptos mencionados.

Para la consecución del proyecto se utilizaron ciertas herramientas de software, así mismo se combinaron marcos de trabajo definidos en las metodologías de Proceso Unificado y Methontongy con el fin de tomar de cada una de las fases que más se ajustarán a la construcción del prototipo. (Abreo, Padilla, 2012)¹⁹

CAPITULO II

ESPECIFICACION DE REQUISITOS

3.1 INTRODUCCIÓN

El presente documento buscare definir las especificaciones funcionales y no funcionales del sistema, así mismo el alcance que puede tener este. La aplicación será usada por el usuario con conocimientos expertos y el usuario normal que tenga conocimiento de gobierno de TI y aseguramiento de la información. (GUI).

3.1.1 Alcance y Propósito

Desarrollo e implementación de un modelo de búsqueda y recuperación semántica de información para guías de un buen gobierno de TI.

(19) Abreo, C., Padilla, C. (2012) "Prototipo para recuperación semántica de información de proyectos de ingeniería de sistemas con base en un repositorio ontológico" Vol.3, num 1, Enero- Julio, pp 109-118.

El sistema será una aplicación desarrollada en el lenguaje JAVA, que permitirá acceder y obtener información de una ontología creada mediante ciertos parámetros. La Aplicación constara con los siguientes procesos:

1. Creación y modificación de usuario
2. Búsqueda de información en una base de conocimiento
3. Recopilación de la información

Mediante este sistema No se podrá:

1. Modificar los parámetros básicos de la ontología
2. No puede eliminar
3. No puede hacer cambios en la base de conocimiento

El sistema dará acceso a la información contenida mediante un servidor Local por el momento.

3.1.2 Ámbito

El desarrollo de este software está orientado a ofrecer un contenido de búsqueda funcional que pueda ayudar al interesado a obtener la información de GuidesIT de una forma rápida y sistematizada. En la aplicación sea diferenciadas dos partes para acceder a la información: de una forma pública y otra privada. La forma pública es cuando el usuario solo busca la información que quiera, y la forma privada incluye las preferencias del usuario, será para uso exclusivo de usuarios registrados, la ventaja de esto es que el usuario puede elegir lo que desea buscar y elegir la exactitud de la búsqueda lo que significa que contendrá información más específica.

3.1.3 Definiciones, Acrónimos y Abreviaturas

Para el desarrollo de la aplicación, se hace necesario tener información de las más utilizadas:

GuidesIT: GuidesIT es un conjunto de guías diseñadas para la adecuada planificación de un buen gobierno de TI, basado en el alineamiento del modelo COBIT²⁰ y norma ISO 27000.

(20) COBIT 4.1. ISACA. Marco de Trabajo, objetivos de control, directrices gerenciales y modelos de madurez.

Interfaz: Parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario.

GUI: Interfaz gráfica de usuarios, es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

HTML: HyperText Markup Language

JSP: JavaServer Pages

JSON: JavaScript Object Notation

PLN: Procesamiento del Lenguaje natural

PU: Preferencias del Usuario

KB: Knowledgebase (Base de conocimiento)

OWL: Ontología Web Lenguaje (Lenguaje web para ontología)

3.1.4 Referencias

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830- 1998.
- COBIT 4.1: Framework for IT Governance and Control
- METHONTOLOGY: From Ontological Art Towards Ontological Engineering
- The Rational Unified Process (RUP)
- ISO/IEC 27000-series

3.2 DESCRIPCIÓN GENERAL

3.2.1 Perspectiva del Producto

El sistema para este caso trabajara en forma Cliente/Servidor y de manera independiente.

3.2.2 Funciones del Producto

A continuación se describen las funciones mencionadas en el alcance del producto:

- **Creación y modificación de usuario:** Esta función permite la recopilación de la información básica de un usuario y así mismo las preferencias de búsqueda.
- **Búsqueda de información en una base de conocimiento:** Esta función permite recorrer la ontología creada por medio de propiedades de las clases que la conforman. Así mismo compara el texto ingresado con los conceptos recopilados de la KB.
- **Recopilación de la información:** Esta función permite mostrar los resultados obtenidos sobre la búsqueda de la información realizada

3.2.3 Características del Usuario

El sistema constara de dos tipos de usuarios:

1. Usuario experto
2. Usuario Registrado
3. Usuario No registrado

3.2.3.1 Perfil del Usuario

- **Usuario experto:** Este usuario es un auditor, consultor o experto en el manejo de GuidesIT.
- **Usuario Registrado:** Es aquel usuario que posee dentro del sistema una cuenta, que puede modificar y puede establecer las preferencias al momento de hacer la búsqueda.
- **Usuario No Registrado:** Es aquel usuario que accede al sistema y puede consultar el sistema de búsqueda lo que él quiera.

3.2.3.2 Jerarquía de Usuarios



Gráfico N° 3

3.2.4 Restricciones generales

3.2.4.1 Políticas Regulatoras

Este software será desarrollado utilizando un software de licencia libre, por lo tanto no se deberá pagar económicamente por el uso de: Servidor Glassfish 3.1.2.2, sistema de gestión de base de conocimiento: Protégé 4.1, y el lenguaje de programación JAVA. Por tanto, la utilización de estos programas se encuentra sujetos a políticas establecidas por el licenciamiento correspondiente.

3.2.4.2 Limitaciones de Hardware

Para la utilización de la aplicación, se encuentra necesario un computador servidor, lo cual tendrá instalado lo siguiente: Servidor Glassfish, Protégé, Netbeans 7.3.1, JDK1.7 y un navegador: Google Chrome o Internet Explorer 9.

Este sistema no tiene relación con ningún otro sistema, por tanto no se desarrollan interfaces con otras aplicaciones.

3.2.4.3 Funciones de Control

Se tendrá un control adecuado para la validación de datos que ingrese el usuario.

3.2.4.4 Requisitos del Lenguaje

Se hace necesario para este caso, que el documento para el usuario y el programa se encuentren en el idioma español.

3.2.4.5 Credibilidad de la Aplicación

El sistema deberá como requisito primordial ser sometido a evaluación y prueba para asegurarnos que se encuentre acorde a los requisitos del documento, el rendimiento y la respuesta en el tiempo.

3.2.4.6 Consideraciones de Seguridad

Para mantener la seguridad del sistema y su información, los usuarios luego de ingresar su nombre de usuario y contraseña, se debe verificar su acceso al sistema y las claves deberán estar seguros.

3.2.4.7 Dependencias

Para el adecuado manejo del sistema, cada usuario deberá estar capacitado para lo que va realizar, ya que por ser este un software de búsqueda específico, el usuario debe tener algún conocimiento de lo que va a buscar.

3.3 REQUERIMIENTOS ESPECÍFICOS

3.3.1 Requerimientos Funcionales

3.3.1.1 Ingreso a la Aplicación

Requerimiento No. 1				
Código	Nombre	Grado de Necesidad		
ERF_01	Ingreso a la aplicación	Esencial		
Descripción	La página puede ser accedida a través de una computadora que soporte Servidor Glassfish, Protégé, Netbeans 7.3.1, JDK1.7 y un navegador: Google Chrome o Internet Explorer 9.			
Entradas	Fuente	Salida	Destino	Restricciones
Usuario registrado	Formulario de ingreso a la cuenta	Accede a la cuenta	Muestra la pantalla	Es imprescindible que el usuario posea una cuenta.
Proceso	El sistema deberá verificar que los datos hayan sido ingresado correctamente por el usuario, si no es así se le debe permitir al usuario realizar las operaciones requeridas por el sistema.			
Efecto colateral	No puede ingresar a la cuenta.			

Tabla N° 4

3.3.1.2 Gestión del Usuario y la KB

Requerimiento No. 2				
Código	Nombre	Grado de Necesidad		
ERF_02	Creación de usuario	Esencial		
Descripción	Se requiere recopilar todos los datos referentes al usuario que necesite entrar al sistema. Datos relacionados con su alias y contraseña para ingresar al sistema, su nombre, apellido y fecha de nacimiento como también las preferencias de búsqueda de la ontología.			
Entradas	Fuente	Salida	Destino	Restricciones
	Formulario de creación de usuario	Mensaje de inserción exitosa	Muestra la pantalla de login	El usuario ingresado debe poseer un alias diferente de los demás
Proceso	Se valida que todos los campos del formulario del usuario estén llenos y que tengan valores acordes a los campos. Se envían estos mismos a un archivo XML para su almacenamiento.			
Efecto colateral	No se puede adicionar nuevos usuarios al sistema			

Tabla N° 5

Requerimiento No. 3				
Código	Nombre	Grado de Necesidad		
ERF_03	Modificación de usuario	Esencial		
Descripción	Se requiere recopilar la actualización todos los datos referentes al usuario. Datos relacionados con su alias y contraseña para ingresar al sistema, su nombre, apellido y fecha de nacimiento como también las preferencias de búsqueda de la ontología.			
Entradas	Fuente	Salida	Destino	Restricciones
	Formulario de modificación de los datos usuario	Mensaje de Actualización exitosa	Pantalla de búsqueda	El usuario debe haber iniciado sesión
Proceso	Se valida que todos los campos del formulario del usuario estén llenos y que tengan valores acordes a los campos. Se envían estos mismos a un archivo XML para su almacenamiento.			
Efecto colateral	Ninguna			

Tabla N° 6

Requerimiento No. 4				
Código	Nombre	Grado de Necesidad		
ERF_04	Búsqueda de información de la ontología	Esencial		
Descripción	Se realiza el recorrido en forma general de cada uno de los individuos que conforma la base de conocimiento, abstrayendo sus atributos básicos y por ende la información deseada.			
Entradas	Fuente	Salida	Destino	Restricciones
	Caja de texto de búsqueda diligenciado	Lista de guías asociada a la cadena	Llamado al método de visualización de resultados	Escribir una cadena valida
Proceso	Se realiza el procesamiento del lenguaje natural la cadena ingresada, un recorrido general de la ontología, la abstracción de las palabras claves de los individuos de la misma, la comparación de palabras y obtención de la información.			
Efecto colateral	Ninguno			

Tabla N° 7

Requerimiento No. 5				
Código	Nombre	Grado de Necesidad		
ERF_05	Recopilación de información de la ontología	Esencial		
Descripción	Se muestran los resultados de las guías deseadas por el usuario			
Entradas	Fuente	Salida	Destino	Restricciones
	Lista de guías obtenidas del proceso de búsqueda de la ontología	Guías contenidas en un paginador hecho en html y jquery	Pantalla de búsqueda	Ninguna
Proceso	Mostrar los contenidos de las guías en la pantalla con un paginador de contenidos.			
Efecto colateral	Ninguna			

Tabla N° 8

3.3.1.3 Requisitos de Desarrollo

Requerimiento No. 6				
Código	Nombre	Grado de Necesidad		
ERF_06	Cantidad de usuario concurrentes	Esencial		
Descripción	Para este caso, el sistema soporta la cantidad de usuarios cualesquiera.			
Entradas	Fuente	Salida	Destino	Restricciones
Infraestructura	Arquitectura del sistema	Número de usuarios	Aplicación	Ninguna
Proceso	El usuario podrá ingresar al sistema cuantas veces él lo requiera.			
Efecto colateral	Ninguno			

Tabla N° 9

Requerimiento No. 7				
Código	Nombre	Grado de Necesidad		
ERF_07	Cantidad de Información Almacenada	Esencial		
Descripción	Para este caso, el sistema soporta la cantidad de información a registrar que se requiera.			
Entradas	Fuente	Salida	Destino	Restricciones
Infraestructura	Arquitectura del sistema	Número de registros	Aplicación	Ninguna
Proceso	El sistema debe permitir realizar n inserciones de registros por parte de los usuarios cuando sea necesario.			
Efecto colateral	Ninguno			

Tabla N° 10

Requerimiento No. 8				
Código	Nombre	Grado de Necesidad		
ERF_08	Disponibilidad del sistema	Esencial		
Descripción	El sistema debe permitir al usuario hacer las búsquedas y modificaciones necesarias de su perfil			
Entradas	Fuente	Salida	Destino	Restricciones
Arquitectura de diseño	Ninguno	Ninguno	Ninguno	Depende de factores externos
Proceso	El sistema deberá estar disponible a menos que sucedan causas externas Como: perdida de fluido eléctrico.			
Efecto colateral	Ninguno			

Tabla N° 11

3.3.1.4 Atributos del software del sistema

Requerimiento No. 9				
Código	Nombre	Grado de Necesidad		
ERF_09	Seguridad de la contraseña	Esencial		
Descripción	El sistema debe permitir al usuario modificar su contraseña			
Entradas	Fuente	Salida	Destino	Restricciones
Contraseña de usuario	Formulario de acceso	Contraseña	Archivo XML	Ninguno
Proceso	Al momento que un usuario requiera ser validado en el sistema, este le presentara una pantalla d autenticación de usuario para que el usuario ingrese su nombre y contraseña			
Efecto colateral	No se le permitirá el acceso			

Tabla N° 12

3.3.2 Obligaciones del Diseño

3.3.2.1 Estándares Cumplidos

Se han intentado cumplir los estándares de cualquier web, creando un sistema de autenticación, manejo de usuarios y una plataforma de consulta básica.

3.3.3 Requisitos no Funcionales

Se han determinado los siguientes requisitos no funcionales para el sistema, pero si para el usuario.

- Paginación de los resultados de búsqueda.
- Recopilación de la fuente encontrada de la información solicitada.
- Apariencia gráfica del sistema.
- Ayudas didácticas en los componentes de los formularios de creación de usuario y búsqueda.

CAPITULO III

ANÁLISIS

4.1 INTRODUCCIÓN

Como se dijo anteriormente en la metodología a emplear, para realizar el análisis de esta aplicación, se ha optado por seguir las pautas definidas por el lenguaje unificado de modelado UML (por sus siglas en ingles), este modelo dispone de multitud de diagramas que ayudan a comprender en un futuro la complejidad del sistema, por eso mismo es tan importante la especificación de los requisitos, ya este permite plasmar en un lenguaje las funcionalidades y requerimientos que se ha detectado en el sistema.

Aunque UML define una gran cantidad de diagramas para representar los distintos aspectos del desarrollo de la aplicación, en nuestro caso nos hemos centrado en tres tipos de diagramas: el diagrama de clases y el diagrama de casos de usos y el diagrama entidad-relación.

4.2 DIAGRAMA DE CLASES

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

Para la elaboración del diagrama de clases se ha utilizado una herramienta llamada ArgoUML, permitiendo crear diagramas de una forma sencilla y óptima.

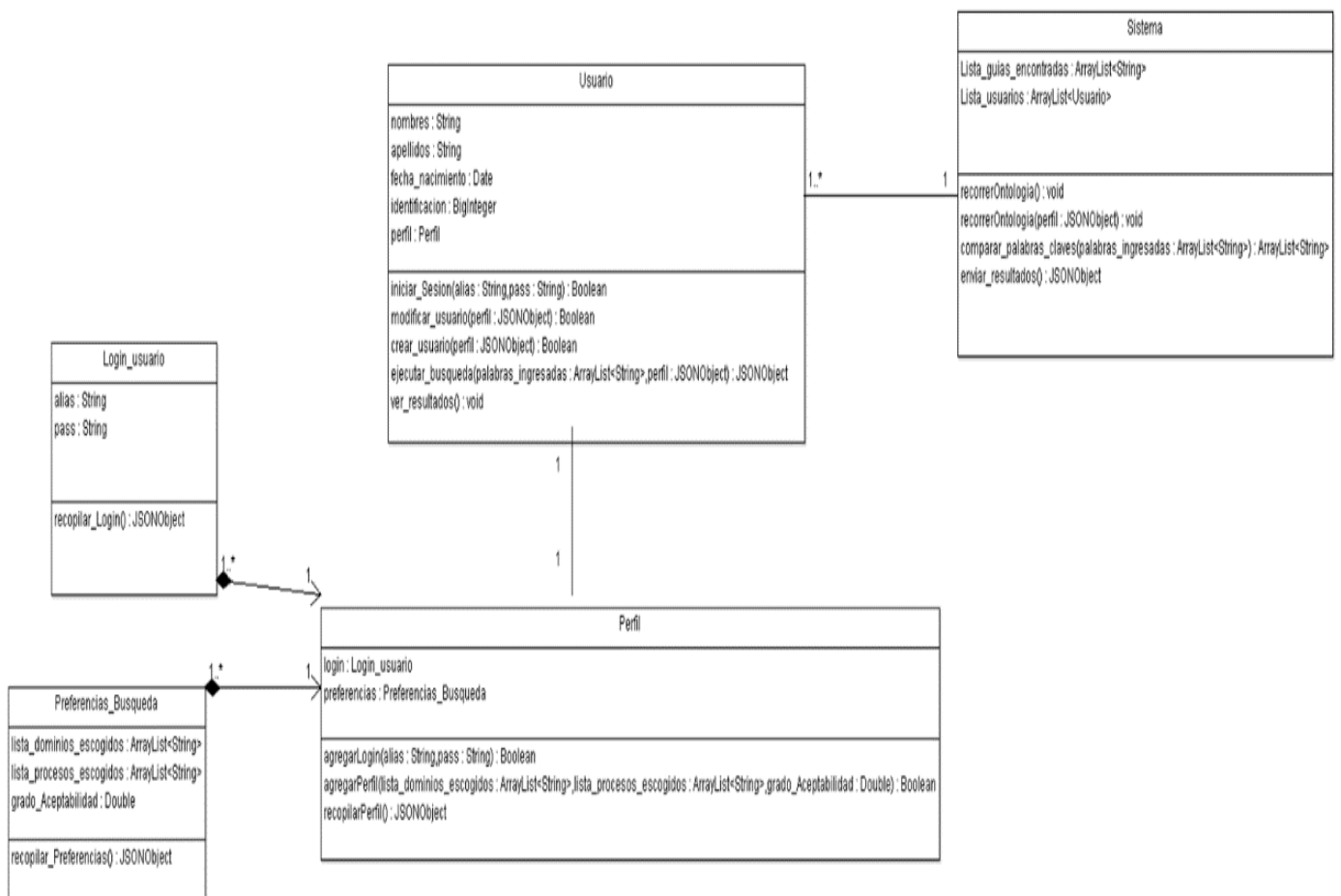


Gráfico N° 4

4.2.1 Diccionario de Entidades

Clase GestionOntologia	
Atributos:	Tipo:
Manager	OWLOntologyManager
rutaOntologia	Rutas
Ontología	OWLOntology
documentIRI	IRI
OntologyValue	IRI
ontologyFormat	PrefixOWLOntologyFormat
Factory	OWLDataFactory
Perfil	HashMap
resultadosBusqueda	HashMap
nombreEntidadesConsultadas	ArrayList<String>
palabras_claves_ingresadas	ArrayList<String>
obtencion_palabras_ontologia	String
rutaObjetivo	String
palabras_claves_ontologia	ArrayList<String>
Procesamiento	ProcesamientoPalabras
porcentajeAceptabilidad	Double
Descripción de la clase	
<p>Esta clase realiza el proceso de recorrer la ontología de guidesIT dependiendo del perfil del usuario, obtener las palabras clave que se encuentran dentro de ella y así mismo de gestionar y recopilar todas las guías asociadas a la cadena ingresada por usuario.</p>	

Tabla N° 13

Método # 1 GestionOntologia	
Nombre del método:	
recorrerOntologia	
Tipo de Método:	
Procedimiento	
Valor de Retorno:	
Ninguno	
Parámetros de Entrada:	Tipo:
Individual	OWLIndividual
Descripción del Método	
<p>Este método realiza el recorrido general de la ontología sobre el archivo OWL previamente cargado. Toma las propiedades de cada individuo encontrados en ella para establecer sus hijos y poder recorrer cada uno de los mismos en forma jerárquica hasta llegar al final de la ontología.</p> <p>Dependiendo del perfil omitirá o no el recorrido de ciertos individuos por razones del usuario. Este método accede a la información de cada uno de los individuos con el fin de extraer las palabras claves asociadas para establecer la comparación que se requiere para extraer la guía encontrada.</p>	

Tabla N° 14

Método # 2 GestionOntologia	
Nombre del método:	
obtenerValorAtributo	
Parámetros de Entrada:	Tipo:
Individual	OWLIndividual
Atributo	OWLDataProperty
Tipo de Método:	
Función	
Valor de Retorno:	
String	
Descripción del Método	
Retorna el valor de cualquier tipo de atributo de un individuo de la ontología.	

Tabla N° 15

Método # 3 GestionOntologia	
Nombre del método:	
obtener_guia	
Parámetros de Entrada:	Tipo:
Individual	OWLIndividual
AtributoPalabrasClave	String
AtributoContenidoGuia	String
Tipo de Método:	
Procedimiento	
Valor de Retorno:	
Ninguno	
Descripción del Método	
Este método accede a la propiedad para la obtención de las palabras clave de los individuos asociados a la clase guías de aseguramiento y guía de implementación, ejecuta el procedimiento comparativo y determina si guía pertenece a los conceptos descritos por el usuario.	

Tabla N° 16

Método # 4 GestionOntologia	
Nombre del método:	
getResultadosBusqueda	
Parámetros de Entrada:	Tipo:
Ninguno	
Tipo de Método:	
Función	
Valor de Retorno:	
HashMap	
Descripción del Método	
Este método retorna el contenido general de cada individuo de las clase guía de aseguramiento y guías de implementación en un tipo de dato hashmap, luego de haber pasado el proceso comparativo de forma individual.	

Tabla N° 17

Clase ProcesamientoPalabras	
Atributos:	Tipo:
rutaldentConceptos	Rutas
Descripción de la clase	
Esta clase realiza el proceso de recorrer la ontología de guidesIT dependiendo del perfil del usuario, obtener las palabras clave que se encuentran dentro de ella y así mismo de gestionar y recopilar todas las guías asociadas a la cadena ingresada por usuario.	

Tabla N° 18

Método # 1 ProcesamientoPalabras	
Nombre del método:	
Tagging	
Tipo de Método:	
Función	
Valor de Retorno:	
ArrayList<String>	
Parámetros de Entrada:	Tipo:
palabras_ingresadas	ArrayList<String>
Descripción del Método	
<p>Este método realizara un filtrado de las palabras suministradas, descartando y seleccionando palabras claves según una clasificación establecida. Se clasificarán según el etiquetado de palabras llamado Part-Of-Speech Tagger en idioma español, el cual identificará cual palabra del conjunto suministrado es un sustantivo, verbo en infinitivo o adjetivo para su clasificación de las palabras clave del usuario.</p>	

Tabla N° 19

Método # 2 ProcesamientoPalabras	
Nombre del método:	
GradodeAceptabilidadConceptosClaves	
Tipo de Método:	
Función	
Valor de Retorno:	
Double	
Parámetros de Entrada:	Tipo:
palabras_ingresadas	ArrayList<String>
Descripción del Método	
<p>Método para determinar la coincidencia entre las palabras ingresadas por el usuario con las palabras obtenidas de una guía cualquiera de la ontología. Establece un grado de similitud entre cadenas mediante algoritmos de similitud de palabras y determina que tan parecidas son para establecer coincidencias, otorgando así un porcentaje de aceptabilidad que indica si la guía cumple con los conceptos suministrados.</p>	

Tabla N° 20

Método # 3 ProcesamientoPalabras	
Nombre del método:	
eliminarRepetidos	
Tipo de Método:	
Función	
Valor de Retorno:	
ArrayList<String>	
Parámetros de Entrada:	Tipo:
palabras_ingresadas	ArrayList<String>
Descripción del Método	
Este método permite eliminar las palabras que se encuentren repetidas en un arreglo.	

Tabla N° 21

Método # 4 ProcesamientoPalabras	
Nombre del método:	
remover_acentos	
Tipo de Método:	
Función	
Valor de Retorno:	
ArrayList<String>	
Parámetros de Entrada:	Tipo:
palabras_ingresadas	ArrayList<String>
Descripción del Método	
Método para remover caracteres especiales de las palabras ingresadas del usuario como lo son puntos, comas, tildes, entre otros.	

Tabla N° 22

Clase GestionUsuarios	
Atributos:	Tipo:
Archivo	JespXML
rutaPerfiles	Rutas
Descripción de la clase	
Esta clase gestiona las preferencias y los datos básicos de cualquier usuario. Permite crear, modificar y obtener la información asociada de los mismos.	

Tabla N° 23

Método # 1 GestionUsuarios	
Nombre del método:	
validarUsuario	
Tipo de Método:	
Función	
Valor de Retorno:	
Boolean	
Parámetros de Entrada:	Tipo:
Perfil	JSONObject
Descripción del Método	
Este método valida el nombre de usuario y pass para determinar si se encuentra registrado en el sistema para iniciar sesión.	

Tabla N° 24

Método # 2 GestionUsuarios	
Nombre del método:	
crearUsuario	
Tipo de Método:	
Función	
Valor de Retorno:	
Boolean	
Parámetros de Entrada:	Tipo:
Perfil	JSONObject
Descripción del Método	
Este método realiza la inserción de la información del usuario hacia el archivo XML que contiene todos los usuarios del sistema.	

Tabla N° 25

Método # 3 GestionUsuarios	
Nombre del método:	
modificarInfo	
Tipo de Método:	
Función	
Valor de Retorno:	
Boolean	
Parámetros de Entrada:	Tipo:
Perfil	JSONObject
Descripción del Método	
Este método realiza la actualización de la información del usuario hacia el archivo XML que contiene todos los usuarios del sistema.	

Tabla N° 26

Método # 4 GestionUsuarios	
Nombre del método:	
consultarPreferenciasUsuario	
Tipo de Método:	
Función	
Valor de Retorno:	
JSONObject	
Parámetros de Entrada:	Tipo:
Login	JSONObject
Descripción del Método	
Método para consultar las preferencias de búsqueda del usuario relacionadas a la ontología.	

Tabla N° 27

4.3 CASOS DE USO

4.3.1 Resumen Actores

Los diagramas de casos de uso permiten diferenciar los actores que interactúan con la aplicación, las relaciones entre ellos y las acciones que puede realizar cada uno dentro del sistema. Este tipo de diagramas son fácilmente comprensibles tanto por clientes como por usuarios, representan los requisitos funcionales del sistema y se utilizan como base para un desarrollo iterativo e incremental. Los diagramas de casos de uso tienen tres elementos:

- **Actores:** Son los usuarios del sistema. Un actor puede ser una persona, un conjunto de personas, un sistema hardware o un sistema software. Los actores representan un rol, que puede desempeñar alguien que necesita intercambiar información con el sistema.
- **Casos de uso:** Un caso de uso describe una forma concreta de utilizar parte de la funcionalidad de un sistema. La colección de todos los casos de uso describe toda la funcionalidad del sistema.
- **Comunicación entre actores y casos de uso:** Cada actor ejecuta un número específico de casos de uso en la aplicación. Por eso decimos que hay comunicación entre actores y casos de uso.

4.3.1.1 Actores



Gráfico N° 5

4.3.2 Casos de uso del usuario en general

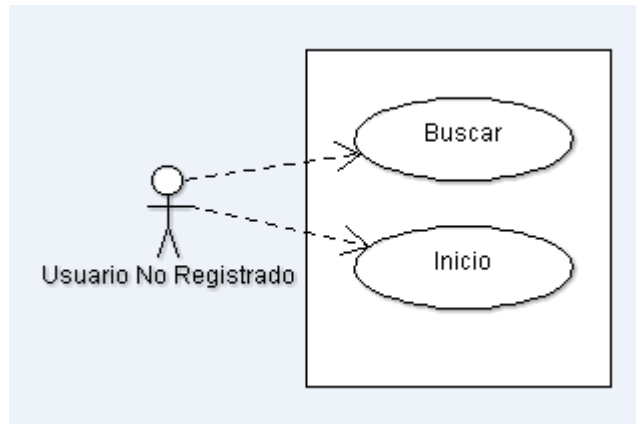


Gráfico N° 6

Este tipo de actor, que también podría denominarse usuario anónimo o visitante, no se ha identificado ante el sistema, las acciones que podrá realizar serán muy limitadas, centrándose en la obtención de información general del buscador.

4.3.3 Casos de uso del usuario registrado

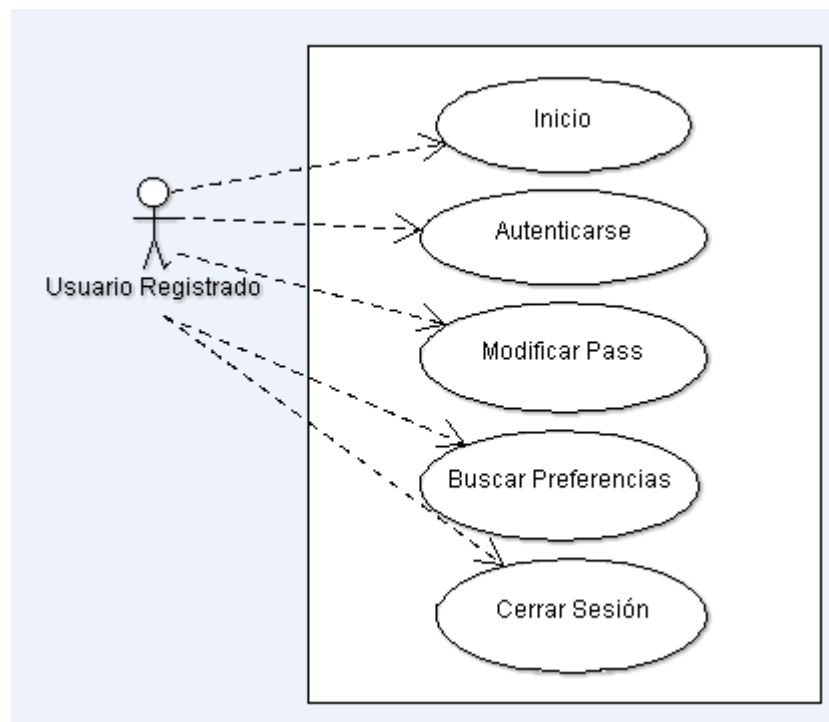


Gráfico N° 7

CAPITULO IV

DISEÑO DEL SISTEMA

5.1 INTRODUCCIÓN

El diseño es el proceso que extiende, refina y reorganiza los aspectos detectados en el proceso de modelado conceptual para generar una especificación rigurosa del sistema de información siempre orientada a la obtención de la solución del sistema software.

Para el diseño del sitio web se ha escogido una arquitectura de tres capas (presentación, negocio y conocimiento).

5.2 ARQUITECTURA DEL SISTEMA

Se basa en una estructura de tres capas, pues distribuye la carga de trabajo en cada una, y estas son:

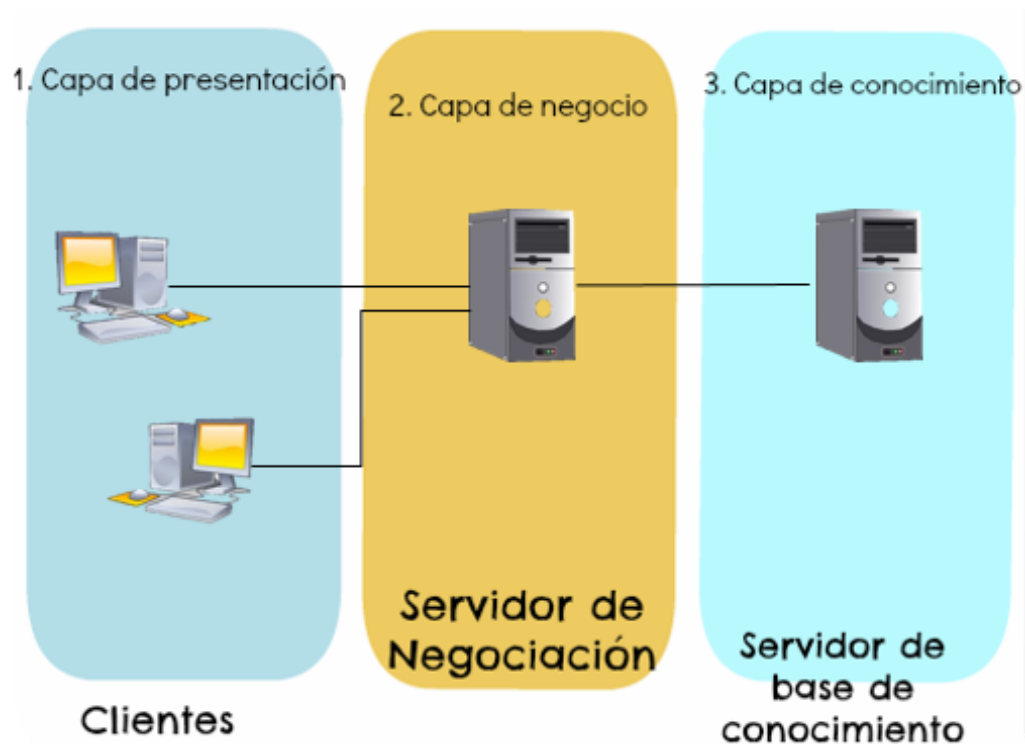


Gráfico N° 8

5.3 CAPAS DE LA APLICACIÓN

5.3.1 Descripción de la arquitectura

- **Capa de presentación:** Es la que ve el usuario, le comunica la información y captura la información recibida del mismo. Únicamente tiene comunicación con la capa de negocio.
- **Capa de negocio:** Esta se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados; también lo hace con la capa de conocimiento, para solicitar al gestor de base de conocimiento recuperar datos de él. Es donde reside el servidor Glassfich 3.1.1, que es el encargado de recibir las peticiones del usuario y las envía mediante una función de AJAX hacia un Servlet.
- **Capa de conocimiento:** Es donde residen los datos mediante el gestor de bases de conocimientos utilizando OWLAPI y recibe solicitudes de recuperación de información desde la capa de negocio.

5.3.2 Diagrama entidad-relación

A continuación se muestra el diagrama entidad-relación utilizado para la implementación de la base de conocimiento del proyecto.

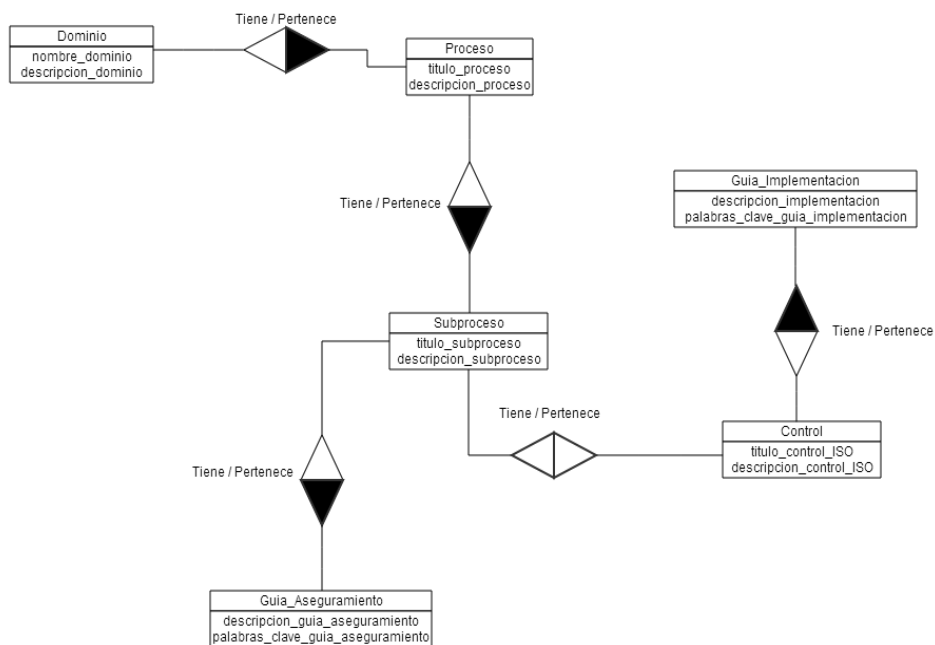


Gráfico N° 9

5.4 Diagrama de Interfaz

5.4.1 Interfaz Principal



Gráfico N° 10

5.4.2 Interfaz Para Buscar



Gráfico N° 11

5.4.2 Interfaz para las Preferencias del Usuario



Gráfico N° 12

CAPITULO V

IMPLEMENTACIÓN Y DESARROLLO

6.1 INTRODUCCIÓN

La implementación se encarga de concretar el diseño teniendo en cuenta todos estos factores.

6.2 CONCEPTOS UTILIZADOS

6.2.1 Ingeniería del Conocimiento

La ingeniería del conocimiento es una rama más de la Inteligencia Artificial, siendo una de las disciplinas emergentes que nació en la era del conocimiento. Este tipo de sistemas empezaron a desarrollarse en la década de los setenta con los sistemas basados en conocimiento y sistemas expertos.

La Ingeniería del Conocimiento tradicionalmente se ha relacionado con sistemas de software en donde el conocimiento y razonamiento juegan un papel muy importante. Sin embargo recientemente la Ingeniería del Conocimiento se ha ampliado para ser usado en la Gestión del Conocimiento, la modelación de empresas y los procesos de reingeniería del Negocio. (Reyes. 2005) ²¹

6.2.2 Base De Conocimientos

La Base de Conocimientos almacena y representa el conocimiento del dominio del Sistema Experto, facilita su acceso, manipulación y actualización. En la siguiente figura se aprecia que la base de conocimientos se encuentra relacionada con otros módulos de un Sistema Experto.

Existen diversas técnicas para la representación del conocimiento, encaminada a:

- Proveer un formato compatible con la computadora.
- Mantener una correspondencia entre el formato de representación y el conocimiento.
- Establecer una representación que puede ser organizada, recuperada, modificada y actualizada eficientemente. El conocimiento inicial se genera a partir de los hechos aprendidos y representados, que al ser manipulados por los procesos de búsqueda e inferencia producen nuevos elementos y actualizan otros, generando una nueva versión.

6.2.3 Inteligencia Artificial (IA)

La Inteligencia Artificial trata de conseguir que los ordenadores simulen en cierta manera la inteligencia humana. Se acude a sus técnicas cuando es necesario incorporar en un sistema informático, conocimiento o características propias del ser humano.

(21) Reyes., (2005) "Análisis de la relación entre la ingeniería del conocimiento y la gestión del conocimiento en base al modelo de Nonaka y Takeuchi" © Intangible Capital - Nº 9 - Vol. 1, Julio-Septiembre - ISSN: 1697-9818 (Cod:0033)

Las definición es de Inteligencia Artificial ha evolucionado tanto que autores como (Rich, 1994) ²², la definen en forma general como la capacidad que tienen las máquinas para realizar tareas que en el momento son realizadas por seres humanos. Sin embargo otros autores como, (Haugeland, 1988) ²³, (Blake, 2004) ²⁴, , (Martin, Peñas 2006) ²⁵ la definen La inteligencia artificial o IA como parte de los sistemas inteligentes que pertenece mayormente a la rama de la informática o computación pero también a los sistemas que hacen posible el funcionamiento del mismo es decir es el estudio tanto del hardware como del software que posibilitan la simulación de una inteligencia similar a la de los humanos generando sistemas capaces de aprender, resolver problemas y tomar decisiones.

6.2.4 Web Semántica

La web, tal y como la conocemos, es un gigantesco repositorio de hiperdocumentos cuyo diseño únicamente permite su comprensión por humanos. Estos materiales están confeccionados usando lenguajes de etiquetado que expresan la forma en que los navegadores deben presentar su contenido (colores, maquetación, fuentes, etc.) y no su significado o semántica.

Dado el gigantesco y creciente número de estos recursos, los actuales motores de búsqueda son incapaces de ofrecer tasas de precisión mínimamente adecuadas en sus resultados, evidenciando que las técnicas léxico estadísticas no pueden solucionar por sí solas la problemática de la recuperación de información. (Peis, Herrera – Viedma, Hassan, Herrera, 2003) ²⁶.

(22) E. Rich., (1994) "Inteligencia Artificial", McGraw-Hill / Interamericana de España, segunda edición.

(23) J. Haugeland, La inteligencia artificial. Siglo XXI, 1988.

(24) R. Blake, Sistemas electrónicos de comunicaciones. Cengage Learning Latin America, 2004.

(25) G. P. Martin sanz and M. S. Peñas, Inteligencia artificial e ingeniería del conocimiento, México: Alfaomega, 2006.

(26) Peis, E.; Herrera-Viedma, E.; Hassan, Y.; Herrera, J. C. (2003) "Análisis de la web semántica: estado actual y requisitos futuros". En: El profesional de la información, septiembre-octubre, v. 12, n. 5, pp. 368-376

La propuesta inicial de world wide web que (Berners – Lee, 1998) ²⁷ ideó, ya apuntaba a lo que posteriormente se ha denominado web semántica (Berners – Lee, Hendler, Lassila, 2001) ²⁸, una extensión de la actual web cuyo objetivo es que no sólo los humanos, sino también las máquinas, sean capaces de “comprender” el contenido de los documentos.

De esta forma, en el proceso de recuperación de información, el usuario interrogaría a un agente software que realizaría tareas complejas de asociación e inferencia de conocimiento, devolviendo al usuario resultados precisos y contextualizados. Para conseguirlo, sería necesario proporcionar semántica a la web y, esta tarea, podría realizarse mediante la elaboración de ontologías y etiquetados descriptivos.

Para cumplir los objetivos, Berners-Lee considera que son necesarias varias capas o niveles, entre las que destacamos: un modelo básico para establecer asertos (propiedades sobre los recursos), para el que se empleará RDF (resource description framework) (Lassila, Swick, 1999) ²⁹; un modelo para definir relaciones entre las fuentes a través de clases y objetos expresado mediante rdf schema (Brickley, Guha, 2002) ³⁰; una capa lógica que permita realizar consultas e inferir conocimiento, donde entrarían en juego las ontologías y los agentes software; y una capa de seguridad que permita asignar niveles de fiabilidad a determinados recursos de forma comprobable posteriormente por los agentes, para lo que se usarán firmas digitales y redes de “confianza”.

(27) T. Berners-Lee, (1989) Information management: a proposal, Recuperado en <http://www.w3.org/History/1989/proposal.html>

(28) T. Berners-Lee, J. Hendler, O. Lassila, (2001) “The semantic web: a newform of web content that is meaningful to computers will unleash a revolution of new possibilities”. Scientific American.

(29) O. Lassila, R. Swick., (1999) Resource description framework (RDF) model and syntax specification: W3C recommendation. Recuperado en. <http://www.w3.org/TR/REC-rdf-syntax>

(30) Brickley, R. V. Guha, (2002) Resource description framework (RDF) schema specification 1.0: W3C working draft,. Consultado en: 20-07-13. <http://www.w3.org/TR/rdf-schema>

6.2.5 Ontologías, OWL y Protégé

Las Ontologías ³¹ han sido tradicionalmente usadas como modelo de representación. Para construir la Web semántica se necesita poder representar el conocimiento de forma que sea legible por los ordenadores, esté consensuado, y sea reutilizable. Las ontologías proporcionan la vía para representar este conocimiento.

Las ontologías permiten trabajar con conceptos, en lugar de palabras clave, en los sistemas de recuperación de información. Desde el punto de vista de las fuentes de información, éstas describen el contenido de los repositorios de datos independientemente de la representación sintáctica de los mismos, posibilitando su integración semántica.

“Una ontología es una especificación explícita de una conceptualización”, según Thomas Gruber. Una conceptualización es una abstracción, una vista simplificada del mundo que queremos representar. Una ontología no es una base de datos ni un programa (porque tienen sus propios formatos internos), no es una conceptualización (porque no es una especificación, es sólo una vista) ni una tabla de contenidos (aunque una taxonomía sí es una ontología). Las ontologías son acuerdos, en un contexto social, para cubrir una serie de objetivos. Se crean, entre otras cosas, para:

- Permitir el intercambio de datos entre programas.
- Simplificar la unificación (o traducción) de distintas representaciones.
- Facilitar la comunicación entre personas.

Las ontologías tienen los siguientes componentes que servirán para representar el conocimiento de algún dominio

- Conceptos: son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.

(31) Choque Aspiazu, Guillermo Ontología para la Ingeniería del Software . Carrera de Informática, Universidad Mayor de San Andrés, La Paz, Bolivia, recuperado de: http://www.espe.edu.ec/portal/files/sitio_congreso_2011/papers/C2.pdf

- Relaciones: representan la interacción y enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.
- Funciones: son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, pueden parecer funciones como categorizar-clase, asignar-fecha, etc.
- Instancias: se utilizan para representar objetos determinados de un concepto.
- Axiomas: son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: “Si A y B son de la clase C, entonces A no es subclase de B”, “Para todo A que cumpla la condición C1, A es B”, etc.

Uno de los lenguajes con gran capacidad expresiva que se ha convertido en estándar para realizar anotaciones de ontologías en la Web es OWL (Ontology Web Language).

OWL³² es un lenguaje para crear ontologías en la Web. Si bien existían lenguajes ontológicos previos, especialmente destinados a temas científicos y aplicaciones de comercio electrónico, no fueron definidos para ser compatibles con la arquitectura de la World Wide Web en general, y de la Web Semántica en particular.

Para que la Web Semántica funcione, los ordenadores deben tener acceso a bases de conocimiento y conjuntos de reglas de inferencia que puedan utilizar para llevar a cabo un razonamiento automatizado. Sin embargo, la representación del conocimiento se encuentra actualmente en un estado comparable al del HTML antes del nacimiento de la Web: es claramente una buena idea, y existen algunas demostraciones prácticas prometedoras, pero todavía no se ha extendido ni ha cambiado el mundo. Para desarrollar todo su potencial debe existir un estándar y estar conectado a un sistema global único. El lenguaje OWL es ese estándar, ya que sin él, otros lenguajes como RDF³³ perderían su potencia al representar conocimiento de manera aislada.

(32) W3C, OWL Web Ontology Language Recuperado en: <http://blog.classora.com/2012/12/18/owl-el-lenguaje-de-ontologias-web-del-w3c/>

(33) W3C, Resource Description Framework (RDF), Recuperado en: <http://www.w3.org/RDF/>

Protégé³⁴ es un editor libre de código abierto y un sistema de adquisición de conocimiento. Al igual que Eclipse, Protégé es un framework para el cual otros proyectos sugieren plugins. La aplicación está escrita en Java y usa fuertemente Swing para crear su compleja interfaz.

Protégé es una herramienta integrada de software para desarrollar sistemas basados en el conocimiento. Los usos desarrollados con Protégé-2000 se utilizan para la solución de problemas y la toma de decisiones en un dominio particular. Esta herramienta facilita trabajar simultáneamente con las clases y los casos. Protégé permite:

- Modelar una ontología de las clases que describen un tema particular.
- Creación de una herramienta de adquisición de conocimiento para recoger conocimiento.
- Entrar en casos específicos de datos y de la creación de una base de conocimiento.
- La ejecución de usos.

6.2.6 Recuperación de la información (RI)

La Recuperación de Información (RI) consiste en la representación, almacenamiento, organización y acceso a ítems de información (Salton, 1983)³⁵. La representación y organización de estos ítems debería proporcionar al usuario un fácil acceso a la información en la cual está interesado. Pero hoy en día esto no es así debido a varios inconvenientes como la sobrecarga de información (Marcano, Talavera, 2006)³⁶, la heterogeneidad semántica (Mitra, Noy, Jaiswal, 2005)³⁷, y el uso inapropiado de la meta-información (D. Avello, 2002)³⁸, entre otros.

(34) Protégé, open source ontology editor and knowledge-base framework, Recuperado en (<http://personales.upv.es/ccarrasc/doc/2001-2002/Protege2000/PROTEGE.htm>)

(35) G. Salton (1983) Introduction to modern information retrieval, McGraw-Hill, New York, p. 448. Modelo Semántico de Expansión de Consultas para la Búsqueda Web (MSEC)

(36) Y Marcano and R. Talavera, (2006) "Gestión de la información a través de la Web Semántica: Iniciativas y dificultades," Revista Venezolana de Gerencia (RVG), vol. 11, No. 36, October, p. 36.

(37) P. Mitra, N. Noy, and A. Jaiswal, (2005) "Ontology Mapping Discovery with Uncertainty," Fourth International Conference on the Semantic Web, Galway Ireland 6th – 10th November, p. 15.

(38) D. Gayo-Avello, A Meta-Analysis of State-of-the-Art Electoral Prediction From Twitter Data, Social Science Computer Review 0894439313493979, first published on August 23, 2013.

Los usuarios de Internet continúan empleando motores de búsqueda tradicionales, los cuales ofrecen una visión de la Web que sólo se limita al tratamiento léxico de los documentos sin tener en cuenta el significado que éstos representan para el usuario. Es decir su contenido semántico. Esto dificulta la búsqueda de información útil y obliga a los usuarios a pasar un mayor tiempo examinando cientos de documentos hasta encontrar los adecuados.

6.2.7 Procesamiento del lenguaje Natural (PLN)

El ingreso de texto en lenguaje natural ha resultado en una de las principales actividades masivas en el ingreso de datos de las últimas décadas; impulsado recientemente por la conectividad masiva y la creciente interacción de los usuarios en la web 2.0, foros, blogs, redes sociales, chat y mensajes cortos móviles; coincidiendo con electrónica de consumo cada vez más sofisticada. Los sistemas de PLN³⁹ (Procesamiento de Lenguaje Natural) llamado NLP (en inglés) procesan ese texto “plano” generalmente producido por humanos.

Hoy más que nunca con la globalización e informatización este texto ingresado posee una extensa terminología no acotada y suele en consecuencia contener innumerables errores de todo tipo y etiología. A la vez la constante miniaturización de la electrónica, acompañada por su baja de costos y el aumento paulatino de su potencia de cómputo (ley de Moore) permitieron a la industria crear nuevos dispositivos electrónicos cada vez más inteligentes para toda clase de usos, requiriendo texto como entrada preferida.

Un creciente sinnúmero de nuevas funciones de estos dispositivos electrónico informáticos requiere del ingreso de órdenes cada vez más complejas, muchas de las cuales se realizan hoy en forma mecánica y usando sensores especializados (mouse/trackball/tacto/luz), mientras que el habla directa con conversión voz a texto aún no es lo suficientemente robusto para ser usado masivamente, queda por último el teclado alfanumérico como el elemento más usual y robusto para el ingreso de datos y posibles órdenes precisas y complejas en forma de texto.

(39) Andrés T. Hohendahl “Procesamiento de Lenguaje Natural Robusto” Laboratorio de Estereología y Mecánica Inteligente, Facultad de Ingeniería, UBA. Recuperado en: http://web.fi.uba.ar/~ahohenda/docs/prolen_robust.pdf

La difusión de estas nuevas tecnologías y el compromiso que presentan por su reducido tamaño, el ingreso de texto en estos dispositivos resulta cada vez más engorroso y problemático, multiplicando el número de errores que aparecen asociados. Por esto se requiere de nuevas estrategias y métodos para abordar los complejos problemas derivados de los errores de escritura. Esto se hace más destacado cuando se procesa lenguaje natural, pues torna potencialmente problemáticos los errores de escritura, como un ejemplo simple: los comandos de sistemas operativos y lenguajes de computación tradicionales, no toleran el más mínimo error.

Sin duda habrá una mayor necesidad de estas técnicas cuando la electrónica de consumo pueda dialogar plenamente con el usuario en lenguaje natural, habiendo numerosas y marcadas evidencias de esta tendencia en la industria.

El propósito de los sistemas de PLN es obtener alguna información específica o datos a partir de texto plano (sin características de formato). Ese texto básicamente es esperado ser ingresado desde un humano, he aquí la situación actual: los humanos se comunican mediante palabras ensambladas en estructuras, existiendo dentro de éstas una cantidad de elementos de ligadura como la sintaxis, coexistiendo con una fuerte semántica subyacente para dar sentido a nuestros actos ilocutorios en forma de texto escrito.

Si bien el estado del arte del PLN permite hoy procesar un texto en forma decente (no preciso ni profundizo pues daría para un libro completo), éste estado del arte choca contra un patrón muy frecuente culturalmente cuando se trata de escribir texto: los errores de escritura de los que estamos hablando al amparo de la enorme variedad de texto viable, con inclusiones multiculturales, acrónimos, fórmulas y palabras prestadas de múltiples idiomas ('loan words') además de una sinnúmero de términos nuevos o parasintéticos de uso muy frecuente.

Estos errores son muy difíciles de detectar y pueden ser de muchos tipos y de hecho los más frecuentes son los involuntarios y los que ocurren por falta de conocimiento o base cultural, como ser el saber si un determinado nombre propio se escribe con cierta combinación extraña de letras no frecuentes en nuestro idioma habitual.

Otros errores se centran en fenómenos como la mecánica de escritura, por ejemplo teclados demasiado chicos como para el dedo 'demasiado grande para la teclita' de un usuario promedio. Algunos fabricantes están usando mecanismos tan sutiles como los teclados 'táctiles' capacitivos que procesan por válido un mero roce imperceptible por el desafortunado escritor.

6.2.8 Algoritmos Genéricos

6.2.8.1 Levenstein ⁴⁰

Levenshtein es una distancia ideada por Vladimir Levenshtein en 1965 que nos permite medir la distancia entre dos secuencias, por ejemplo de caracteres.

En concreto, dadas dos secuencias A y B, la distancia Levenshtein nos informa del número mínimo de cambios (inserción, eliminación o sustitución de elementos) que deben hacerse a una cadena para obtener la otra.

Por ejemplo, la distancia Levenshtein entre “juan” y “jjuan” es de una única operación (eliminar/insertar una “j”).

La función Levenshtein es realmente sencilla y no requiere explicación, ésta podría ser una implementación en “JavaScript”:

```
// Levenshtein(String, String) -> Integer

Function Levenshtein(a, b) {
  var n = a.length;
  var m = b.length;
  // matriz de cambios mínimos
  var d = [];
  // si una de las dos está vacía, la distancia
  // es insertar todas las otras
  if(n == 0)
    return m;
  if(m == 0)
    return n;
  // inicializamos el peor caso (insertar todas)
  for(var i = 0; i <= n; i )
    (d[i] = [])[0] = i;
  for(var j = 0; j <= m; j )
    d[0][j] = j;
  // cada elemento de la matriz será la transición con menor coste
  for(var i = 1, I = 0; i <= n; i , I )
    for(var j = 1, J = 0; j <= m; j , J )
      if(b[J] == a[I])
        d[i][j] = d[I][J];
      else
        d[i][j] = Math.min(d[I][j], d[i][J], d[I][J]) + 1;
  // el menor número de operaciones
  return d[n][m];
}
```

(40) Levenshtein Algorithm. [En línea], Recuperado en: <http://www.levenshtein.net/>

6.2.8.2 LetterpairSimilarity⁴¹

Disecciona dos cadenas de caracteres en pares y calcula la similitud las mismas mediante la división del número de pares comunes por la suma de los pares de las dos cadenas.

6.2.9 Preferencias del Usuario

Debido a la evolución de las tecnologías de comunicaciones y a la gran diversidad de dispositivos móviles que han emergido en los últimos años, que respaldan cada vez más mejores servicios, en la actualidad se evidencia un uso masivo de terminales distribuidos, configurables dinámicamente y en proximidades a los usuarios que permiten un acceso permanente a la información. Ello abre paso a la visión introducida por (Weiser, 1991)⁴² con el nombre de computación ubicua. Según este autor, la computación ubicua se describe como la existencia de pequeños computadores, con capacidades de comunicación y computación, embebidos de forma casi invisible en cualquier tipo de dispositivo cotidiano, que se integran amigablemente con los humanos. Es decir, las personas interactúan con ellos de forma inconsciente (Almenárez, 2005; Weiser, 1991)⁴³.

Dado que en estos ambientes las personas están centradas en las tareas que deben cumplir, más que en el dispositivo que deben utilizar para ejecutarlas, ya que los equipos pasan inadvertidos, uno de los objetivos de la ubicuidad se enfoca en ayudar a los usuarios a identificar, en cualquier momento y en cualquier lugar, las tareas que se van a realizar, por medio del descubrimiento automático de los servicios ofrecidos en la red ubicua (Ben Mokhtar et ál., 2007)⁴⁴.

(41) LetterPairSimility Algorithm. [En línea], recuperado en: http://infocenter.pentaho.com/help/index.jsp?topic=%2Fpdi_user_guide%2Freference_step_fuzzy_matcher.html

(42) WEISER, M. The computer for the 21st century. Scientific American. 1991, vol. 265, núm. 3, pp. 94-104.

(43) ALMENÁREZ, F. Arquitectura de seguridad para entornos de computación ubicua abiertos y dinámicos. Tesis doctoral. Escuela Politécnica Superior, 2005.

(44) BEN MOKHTAR, S.; GEORGANTAS, N. e ISSARNY, V. COCOA: COntversation-based service COmposition in pervAsive computing environments with QoS support. Journal of Systems and Software. 2007, vol. 80, núm. 12, pp. 1941-1955.

Sin embargo, encontrar un servicio que cumpla exactamente con estas peticiones se convierte en un caso excepcional, ya que la mayoría requieren una adaptación de los servicios disponibles en la red o un proceso de composición de diversas capacidades para ejecutar tareas complejas. De este modo, en contextos tan dinámicos y heterogéneos como los ambientes ubicuos, es imprescindible contar con mecanismos de descubrimiento de servicios que consideren las preferencias del usuario, especificaciones de dispositivos y contexto de entrega, que ofrezcan la suficiente flexibilidad para reconfigurar los servicios de acuerdo con las variaciones del ambiente.

El descubrimiento de servicios deberá incorporar las características necesarias para ser tan dinámico y autónomo como las condiciones de la red ubicua lo requieran. Por ello la personalización es una de característica muy importante dentro de este proceso.

El paradigma de la personalización se enfoca en adaptar aplicaciones tanto como sea posible a las preferencias y al contexto del usuario. Hace referencia al conjunto de técnicas que permiten proveer al usuario información relevante. La necesidad de tener en cuenta la movilidad y la omnipresencia ha impuesto nuevas consideraciones, como la localización de usuario, el medio utilizado para la interacción y muchas otras características agrupadas en el concepto de contexto de usuario.

6.3 TECNOLOGIAS UTILIZADAS

6.3.1 Interfaz De Usuario

6.3.1.1 XML⁴⁵ Y HTML⁴⁶

XML es un sistema estándar de codificación de información. Los programas que utilizan el formato XML pueden intercambiar fácilmente sus datos, ya que responden a una misma lógica interna.

(45) W3C, XML Extensible Markup Language (XML) . Recuperado en: <http://www.w3.org/XML/>

(46) W3C, HyperText Markup Language (HTML) . Recuperado en: <http://www.w3.org/community/webed/wiki/HTML>

Los documentos XML son ficheros de texto que contienen la información organizada en forma de árbol: cada rama puede tener unos atributos propios y servir de base para otras ramas. Además, los documentos XML se pueden transformar (por ejemplo, a formato HTML, para mostrar la información en una página web), o combinar: un tronco con todas sus ramas puede pasar a ser una rama de otro árbol mayor.

Varias son las características que ofrece XML: Aunque hoy día XML aún no está tan extendido como HTML, su uso futuro en la Web mejorará la eficiencia de las búsquedas, al proporcionar cada documento XML metadatos sobre sí mismo.

Permite proporcionar diferentes vistas sobre los datos (HTML, PDF, voz, etc.), dependiendo de quién sea el cliente.

Facilita la integración desde fuentes de datos heterogéneas, por ejemplo, páginas Web, distintas bases de datos, ...

Los documentos tienen una estructura que los hace legibles e inteligibles no sólo para los ordenadores, sino también para los humanos.

Las aplicaciones de XML son fácilmente extensibles mediante definiciones de nuevos tipos de documento

HTML son las siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C.

Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Por convención, los archivos de formato HTML usan la extensión .htm o .html.

6.3.1.1.1 CSS⁴⁷

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML.

En este último podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Control centralizado de la presentación de un sitio web completo, con lo que se agiliza de forma considerable la actualización del mismo.
- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web remoto, con lo que aumenta considerablemente la accesibilidad.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

(47) W3C, Cascading Style Sheets (CSS). Recuperado en: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>

6.3.1.2 Controles

6.3.1.2.1 JavaScript⁴⁸

JavaScript es un lenguaje interpretado utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java. Sin embargo, al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia. Es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM (Modelo de Objetos del Documento). JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

6.3.1.2.2 JQuery⁴⁹

jQuery es una biblioteca o framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

6.3.1.2.3 Ajax⁵⁰

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

(48) W3C, JavaScript Language. Recuperado en: <http://www.pergaminovirtual.com.ar/definicion/JavaScript.html>

(49) W3C, JQuery JavaScript framework . Recuperado en: <http://www.actualidadjquery.es/acerca-de/>

(50) W3C, AJAX de Asynchronous JavaScript And XML. Recuperado en: <http://www.etnassoft.com/biblioteca/aspects-of-ajax/>

De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

6.3.1.2.4 Servlet⁵¹

Entre las muchas particularidades que distinguen a la Web de los restantes medios de comunicación, está la capacidad de interacción. Interacción que inicialmente era bastante reducida y se conseguía a través de código HTML y algunos elementos embebidos de JavaScript. Cuando se mencionaba el nombre de Java en este entorno, algunos lo asociaban a JavaScript, y otros, un poco más conocedores, lo asociaban con pequeños programas que se ejecutaban en el cliente y se llamaban Applets. Esto realmente fue el inicio de Java en la Web y surgió para darle más vida y dinamismo a la misma.

Los Servlet son la respuesta de la tecnología en Java a la programación de la Interfaz de Compuerta Común (CGI). Son programas que se ejecutan en el servidor, realizando la función de una capa intermedia entre una petición proveniente de un navegador Web u otro cliente HTTP, y las aplicaciones del servidor, pudiendo utilizar toda la paquetería y potencialidades del lenguaje. Su función principal es proveer páginas web dinámicas y personalizadas, utilizando para este objetivo el accesos a bases de datos, flujos de trabajo y otros recursos.

6.3.1.2.5 JSP⁵²

JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas web con programación en Java.

Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma.

(51) Ibáñez J 20, Extracción de datos web usando técnicas de screen-scraping, Universidad Politécnica de valencia. p. 11-15. Recuperado en: http://riunet.upv.es/bitstream/handle/10251/10104/PFC_Javier_Iba%C3%B1ez.pdf

(52) Ejemplos de Servlet y JSP Web Application Development Recuperado en: http://webap.wikispaces.com/file/view/JSP_Servlet.pdf

Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual.

El motor de las páginas JSP está basado en los servlets de Java -programas en Java destinados a ejecutarse en el servidor-, aunque el número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor, dado que resulta mucho más sencillo aprender que los servlets.

En JSP creamos páginas de manera parecida a como se crean en ASP o PHP -otras dos tecnologías de servidor-. Generamos archivos con extensión .jsp que incluyen, dentro de la estructura de etiquetas HTML, las sentencias Java a ejecutar en el servidor. Antes de que sean funcionales los archivos, el motor JSP lleva a cabo una fase de traducción de esa página en un Servlet, implementado en un archivo class (Byte codes de Java). Esta fase de traducción se lleva a cabo habitualmente cuando se recibe la primera solicitud de la página .jsp, aunque existe la opción de precompilar en código para evitar ese tiempo de espera la primera vez que un cliente solicita la página.

6.3.1.3 Modelo

6.3.1.3.1 Java⁵³

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

(53) Descripción y características del potente y moderno lenguaje de programación Java, Recuperado en: <http://www.desarrolloweb.com/articulos/497.php>

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

Java fue pensado originalmente para utilizarse en cualquier tipo de electrodoméstico pero la idea fracasó. Uno de los fundadores de Sun rescató la idea para utilizarla en el ámbito de Internet y convirtieron a Java en un lenguaje potente, seguro y universal gracias a que lo puede utilizar todo el mundo y es gratuito. Una de los primeros triunfos de Java fue que se integró en el navegador Netscape y permitía ejecutar programas dentro de una página web, hasta entonces impensable con el HTML.

Actualmente Java se utiliza en un amplio abanico de posibilidades y casi cualquier cosa que se puede hacer en cualquier lenguaje se puede hacer también en Java y muchas veces con grandes ventajas. Para lo que nos interesa a nosotros, con Java podemos programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que deseemos hacer con acceso a través web se puede hacer utilizando Java.

6.3.1.3.2 Librerías

6.3.1.3.3 OpenNLP 1.5.3⁵⁴

La biblioteca OpenNLP Apache es una máquina herramienta de aprendizaje basado en el procesamiento de texto en lenguaje natural. Es compatible con las tareas más comunes de PNL, como tokenización, segmentación oración, etiquetas de medio de voz, llamada extracción entidad, fragmentación, de análisis y resolución de la correferencia.

El OpenNLP 1.5.3 contiene un par de nuevas características, mejoras y correcciones de errores. La CLI se ha mejorado para una mejor consistencia. Ahora las herramientas soportan extensiones que se pueden configurar desde el modelo, incluyendo generadores de contexto personalizados y validadores.

(54) Biblioteca OpenNLP Apache, Recuperado en: <http://opennlp.apache.org/>

6.3.1.3.4 OWLAPI 3.4.4⁵⁵

OWLAPI es una Java API que permite crear, manipular y serializar ontologías OWL. La última versión de la API está enfocada a OWL 3

OWLAPI es de código abierto y está disponible en régimen de LGPL o Apache Licencias

La API de OWL incluye los siguientes componentes:

- Una API para OWL 2 y una implementación de referencia en memoria eficiente
- RDF / XML parser y escritor
- OWL / XML parser y escritor
- OWL funcional Sintaxis analizador y escritor
- Turtle analizador y escritor
- KRSS parser
- OBO Flat formato de archivo parser
- Interfaces Reasoner para trabajar con razonadores como hecho , ermitaño, Pellet y Racer

La versión original de la API para OWL 1.0 fue desarrollado como parte del proyecto WonderWeb . Versión 2.0.0 del API para OWL fue desarrollado como parte de la CO-ODE proyecto y la TONOS proyecto. Versión 3.4.4 es la última versión.

6.3.1.3.5 JESPXML⁵⁶

JespXML, es un conjunto de clases hecho en java para manipular fácilmente archivos XML. Está en un proyecto Netbeans. Es sin interfaz gráfica, solo son las clases necesarias para llevar a cabo el objetivo con archivos XML, tales como abrir o generar XML. JespXML está desarrollado en JDK v7.

(55) OWL Java API, Recuperado en: <http://owlapi.sourceforge.net/>

(56) JespXML, Recuperado en: <https://code.google.com/p/analizador-xml/>

Ejemplo de Estructura de un archivo XML (prueba.xml)

El siguiente archivo XML, llamado "prueba.xml" compone una estructura básica de un archivo XML que puede ser leído con la api JspXML contenido la raíz de nuestro proyecto:

```
<?XML versión="1.0" encoding="UTF-8"?>
```

```
<biblioteca cantidadDeLibros="2">
```

```
<!-- Comentario-->
```

```
<libro paginas="100">
```

```
<titulo>
```

```
Libro Accion 1
```

```
</titulo>
```

```
<autor>
```

```
Autor ACTION
```

```
</autor>
```

```
</libro>
```

```
</biblioteca>
```

6.3.1.3.6 JSON – lib⁵⁷

JSON-lib es una biblioteca Java para la transformación de los frijoles, mapas, colecciones, arrays de Java y XML a JSON y viceversa, Beans y DynaBeans. Está basada en la obra de Douglas Crockford en <http://www.json.org/java> .

Las siguientes tablas repasan los tipos de la conversión entre Java y JavaScript:

JSON		Java
cadena	<=>	java.lang.String, java.lang.Character, carbón de leña
número	<=>	java.lang.Number, byte, short, int, long, float, double
true false	<=>	java.lang.Boolean, booleano
nulo	<=>	Nulo
función	<=>	net.sf.json.JSONFunction
colección	<=>	net.sf.json.JSONArray (Object, String, Number, Boolean, función)
objeto	<=>	net.sf.json.JSONObject

Tabla N° 28

(57) JSON-lib java library, Recuperado en: <http://json-lib.sourceforge.net/>

Json-lib requiere (al menos) las siguientes dependencias en la ruta de clases:

- jakarta commons-lang 2.5
- jakarta Commons-BeanUtils 1.8.0
- jakarta commons-colecciones 3.2.1
- jakarta commons-logging 1.1.1
- ezmorph 1.0.6

6.3.2 Software

6.3.2.1 Netbeans 7.3.1⁵⁸

NB es un entorno integrado de desarrollo o IDE (Integrated Development Environment), en él podemos realizar todas las tareas asociadas a la programación:

- Editar el código
- Compilarlo
- Ejecutarlo
- Depurarlo

NetBeans es un entorno de desarrollo muy completo y profesional. Contiene muchas funcionalidades, para distintos tipos de aplicaciones y para facilitar al máximo la programación, la prueba y la depuración de las aplicaciones que se desarrollan. También incorpora un editor propio.

6.3.2.2 Protégé 4.1⁵⁹

Es un editor de código abierto usado para construir Ontologías y un marco general para representar el conocimiento. Está escrito en Java que es un lenguaje de programación orientado a objetos. Se usa para construir aplicaciones para la Web Semántica, para hacer una descripción semántica de la información. Sus archivos se hacen en el lenguaje OWL.

Protégé es un editor de Ontologías cuyo sitio oficial es la Universidad de Stanford. Fúe creado por Stanford Center for Biomedical Research.

(58) Netbeans IDE, Recuperado en: (<http://ocw.udl.cat/enginyeria-i-arquitectura/programacio-2/continguts-1/1-introduccioi81n-a-netbeans.pdf>)

(59) Protégé, open source ontology editor and knowledge-base framework, Recuperado en <http://protege.stanford.edu/>

6.3.2.3 Glassfish Server 3.1.2.2⁶⁰

GlassFish es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito, de código libre y se distribuye bajo un licenciamiento dual a través de la licencia CDDL y la GNU GPL. La versión comercial es denominada Oracle GlassFish Enterprise Server (antes Sun GlassFish Enterprise Server).

GlassFish está basado en el código fuente donado por Sun y Oracle Corporation; este último proporcionó el módulo de persistencia TopLink. GlassFish tiene como base al servidor Sun Java System Application Server de Oracle Corporation, un derivado de Apache Tomcat, y que usa un componente adicional llamado Grizzly que usa Java NIO para escalabilidad y velocidad.

6.3.3 Datos Y Metadatos⁶¹

Los metadatos son simplemente datos sobre datos, es decir, información estructurada que describe a otra información y que nos permite encontrarla, gestionarla, controlarla, entenderla y preservarla en el tiempo.

Los metadatos no son un concepto nuevo: la inclusión de metadatos es el método utilizado tradicionalmente en el mundo de las bibliotecas para gestionar libros y archivos. Sin embargo, el nacimiento de la informática y, más concretamente, las posibilidades de intercambio de información surgidas en torno a la Web Semántica, hizo que aumentara la popularidad del término. Del mismo modo se hizo patente la necesidad de describir estándares.

Los metadatos han estado con nosotros desde que el primer bibliotecario hizo una lista de los documentos que tenía en un estante, en un pergamino escrito a mano. El término "meta" viene de una palabra griega que significa "junto a, con, después, siguiente".

(60) GlassFish servidor de aplicaciones de software, Recuperado en: (<http://es.wikipedia.org/wiki/GlassFish>)

(61) Datos y Metadatos, Recuperado en <http://blog.classora.com/2013/02/28/metadatos-definicion-aplicaciones-y-estandares/>

El uso más reciente del latín y el inglés emplearía "meta" para indicar algo trascendental, o sobrenatural. Los metadatos, pues, pueden ser definidos como datos sobre otros datos. Es el término usado en la era de Internet para la información que los bibliotecarios tradicionalmente habían puesto en los catálogos, y más comúnmente se refiere a información descriptiva sobre recursos de la Web. Un registro de metadatos consiste en un conjunto de atributos, o elementos necesarios para describir la fuente en cuestión.

Actualmente hay un gran interés, y constituye una parte fundamental del desarrollo del Web semántico, por adoptar a gran escala estándares y prácticas descriptivas para los recursos electrónicos porque ello contribuirá a mejorar la recuperación de recursos relevantes en cualquier contexto.

6.4 DESCRIPCIÓN DEL PROYECTO

6.4.1 Implementación de la Ontología

- Ontología De Dominio Específico

Para que una ontología resulte de utilidad en la Web Semántica tiene que ser aceptada universalmente, y para que esto se cumpla se debe definir colaborativamente por la comunidad a la que se destina (García & Sicilia, s/f)⁶². Actualmente no existe una ontología para manejar GuidesIT, por lo que para comprobar las ventajas de este enfoque ha sido necesario construir una ontología parcial cuya única pretensión es la de permitir anotar algunos recursos de nuevas tecnologías y probar un prototipo de recuperación de información por medio del uso de una metodología para representar su conocimiento específico de la forma más adecuada.

La ontología aplicada para gestionar el conocimiento del sistema se conforma básicamente en un dominio de conocimiento, la cual consta de varias jerarquías distribuidas de la siguiente manera:

(62) García, E., Sicilia, M.A. (s/f) "Una propuesta para la búsqueda semántica de recursos web de nuevas tecnologías aplicadas a la educación". En línea <http://tecnologiaedu.us.es/ticsxxi/comunic/egb-masu.htm>

Existe una clase principal llamada dominio, compuesto de 4 dominios el cual contiene 34 procesos, éstos dominios se dividen en (Planificar y Organizar), (Adquirir e Implementar), (Entrega y Soporte) y (Monitorear y Evaluar), Como se muestra a continuación:

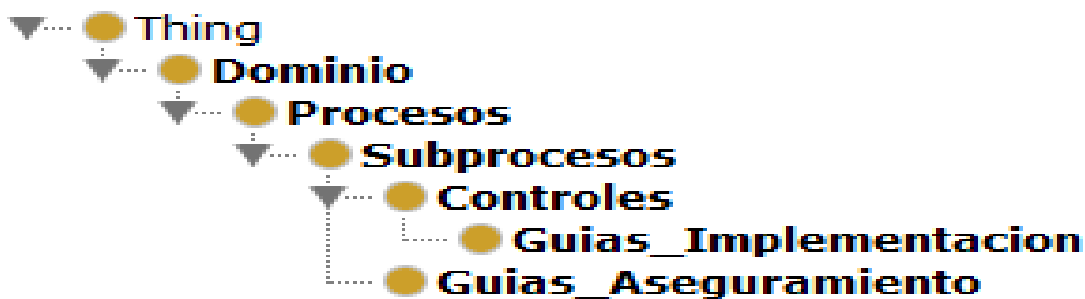


Gráfico N°13

Cada dominio está conformado por instancias de la clase “Procesos”, los cuales son subclases de los dominios. Un dominio puede tener de 1 a muchos procesos pero cada proceso es único en su dominio específico. Los procesos son objetivos que se deben tener en cuenta para llevar un buen manejo aspecto de gestión de TI. Sin embargo cada proceso cuenta además de instancias de la clase “Subprocesos”, siendo estas subclases de los procesos. Los subprocesos son actividades que pueden encontrarse en cualquier tipo de proceso junto a un dominio específico. Cada una guías puede estar presente a cualquier subproceso y estas mismas son las que desea el usuario tener conocimiento.

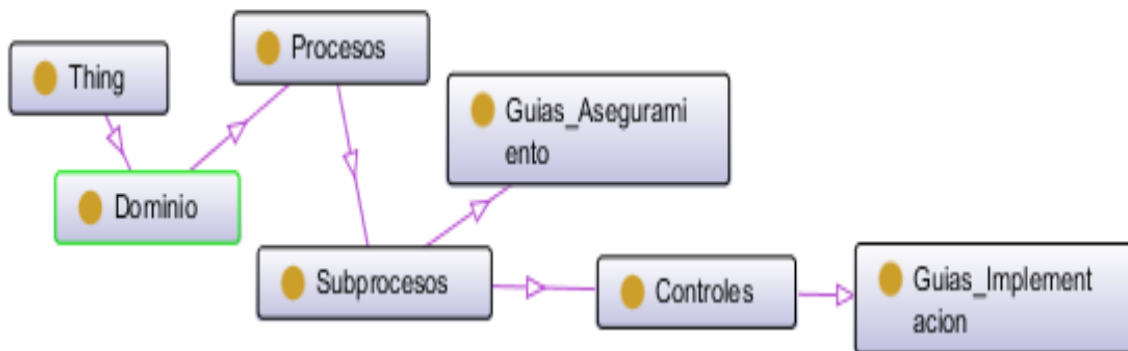


Gráfico N°14

6.4.2 Modelo De Búsqueda Semántica

En esta sección se describe más detalladamente cada uno de los distintos componentes que conforman el buscador ontológico para GuidesIT. A continuación se expone un esquema arquitectónico para la búsqueda de información, el cual engloba a cada componente, sus interacciones y resultados dentro de sus procesos:

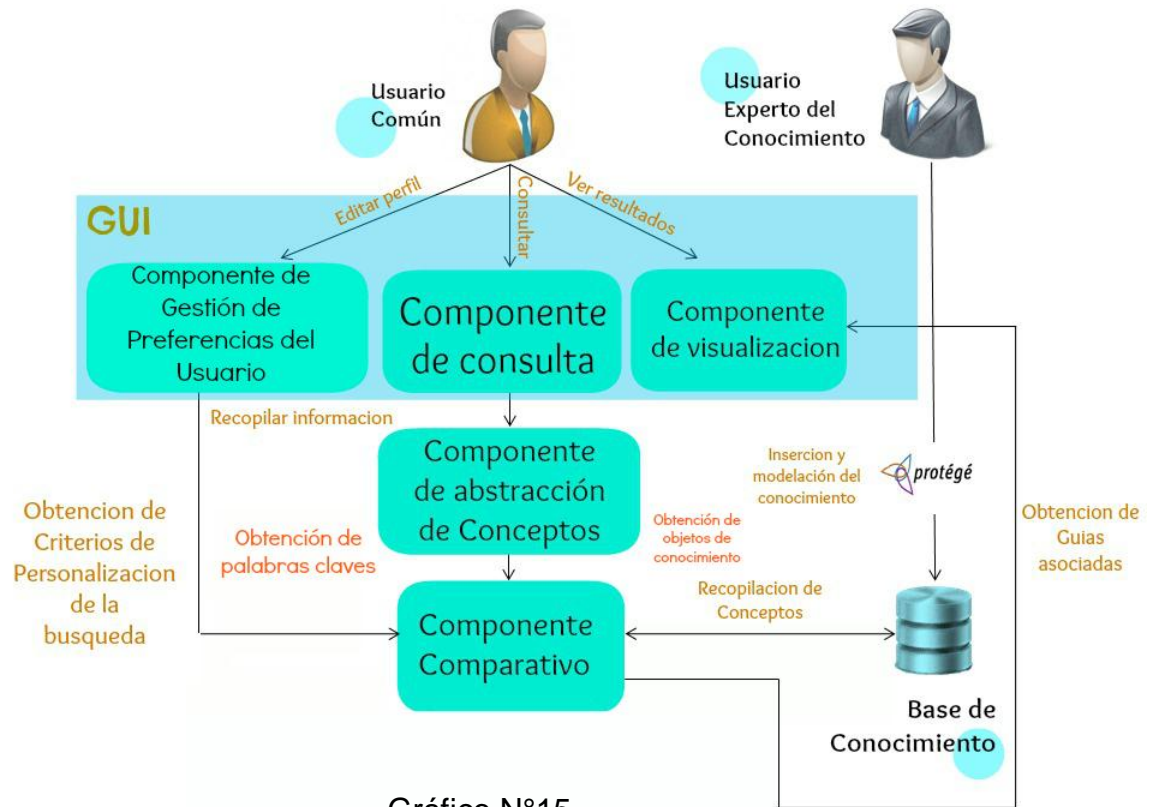


Gráfico N°15

6.4.3 Componente de Consulta

Su razón de ser es proporcionarle al usuario una interfaz gráfica principal la cual consta de una caja de texto el cual podrá escribir la información asociada a las guías que quiere tener conocimiento, y también de un botón de envío que recopilará los datos suministrados para efectuar la operación de búsqueda como se muestra en la figura.



Gráfico N°16

6.4.4 Componente de Abstracción de Conceptos

Este componente del sistema efectúa una identificación de las palabras más relevantes ingresadas que permitan compararse y establecer guías que contengan dichas palabras en la ontología.

Mediante el procesamiento del lenguaje natural (PLN) se emplea una forma de interpretación sobre estas palabras con el fin de hacer un filtrado únicamente de las palabras con mejor significado, y esto se realiza mediante la siguiente secuencia:

- 1) El tokenizar⁶³ y distribuir mediante un arreglo las palabras que se obtuvieron en el paso anterior.
- 2) La eliminación de las palabras que se encuentren repetidas.
- 3) La Estandarización las palabras mediante la eliminación de caracteres especiales y la transformación de las palabras en minúsculas.
- 4) La aplicación del marcado gramatical "POST TAGGER" que se basa en la identificación de palabras de una oración escrita, indicando cual palabra del texto suministrado corresponde a un verbo, adjetivo, sustantivo, entre otros. Con el fin de realizar un filtrado general para aquellas palabras con un mejor significado para efectuar la búsqueda deseada.

(63) Méndez J, Iglesias E, Riverola F, Díaz F., and Corchado J., (2005) "Tokenising, Stemming and Stopword Removal on Anti-spam Filtering Domain", ;in Proc. CAEPIA, pp.449-458.

Cabe destacar que este procedimiento debe realizarse cada vez que el usuario efectúe una consulta tras otra lo cual siempre se abstraerán los palabras más importantes de la cadena ingresada para efectuar la búsqueda de las guías que concuerden con estas.

6.4.5 Componente Comparativo de Conceptos

Una vez identificadas cada una de las palabras clave como resultado del PLN de la consulta realizada por el usuario, se realizará una comparación especializada para determinar que guías contienen las mismas.

La base de conocimiento conformada por la ontología descrita tiene una jerarquía de clases organizada por dominios, procesos, subprocesos y controles respectivamente, los cuales contienen instancias que permiten diferenciarse entre ellas, las instancias de los subprocesos tienen guías de aseguramiento y cada control guías de implementación, las cuales éstas últimas son las que se quieren consultar.

La ontología se debe recorrer para obtener las instancias de las guías y es por ello que se recomienda el uso de OWLAPI que efectúa el uso de propiedades de relación para recuperarlas. Cada clase en la base de conocimiento tiene instancias y una instancia puede contener otras instancias que posiblemente sean de otra clase.

OWLAPI recorrerá jerárquicamente la ontología teniendo encuenta las preferencias de usuario (en donde se seleccionaron los tipos de instancias de dominio y de los procesos que se desean consultar) y las diferentes propiedades de relación de las instancias para poder así extraer el atributo que contendrán las palabras clave asociadas de cada guía encontrada de la base de conocimiento.

A continuación se muestra el algoritmo de recorrido de la ontología:

```
/* Método del recorrido general de la ontología*/
recorrerOntologia (individuo)
Si (individuo = nulo) entonces
  clasedominio = ClasePrincipalOntologia.obtenerSubclases(ObjetoOntologia);
  individuosClaseDominio = filtrarSegunPerfil(clasedominio.obtenerIndividuos(ObjetoOntologia));
  Para (Entero i= 0 hasta i < individuosClaseDominio.tamaño incremento i++) Hacer
    recorrerOntologia(individuosClaseDominio.obtener[i])
  FinPara
Sino
  Claseindividuo = individuo.clase;
  Si (claseindividuo= ("Guias_Aseguramiento")) Entonces
    obtener_guia(individuo, atributo_palabras_clave_guia_aseguramiento,
    atributo_descripcion_guia_aseguramiento, palabras_clave_usuario)
  Sino
    Si (claseindividuo= ("Guias_Implementacion")) Entonces
      obtener_guia(individuo, atributo_palabras_clave_guia_implementacion,
      atributo_descripcion_guia_inplementacion, palabras_clave_usuario)
    Sino
      HijosIndividuo= filtrarSegunPerfil(individuo.obtenerhijos(propiedad_de_relacion))
      Para (Entero i = 0 hasta i < HijosIndividuo.tamaño incremento i++) Hacer
        recorrerOntologia (HijosIndividuo.obtener[i])
      FinPara
    finSi
  finSi
finSi
```

Gráfico N°17

El algoritmo anterior identifica sistémicamente las instancias asociadas a las clases Guia_de_Aseguramiento y Guia_de_Implementacion, las cuales contienen palabras claves que deben ser comparadas con las palabras procesadas por el PLN que fueron introducidas por el usuario.

La comparación respectiva se realiza guía por guía y mediante el algoritmo comparativo del sistema procede a medir la distancia o grado de similitud entre las palabras de ambas partes mediante el uso de algoritmos de similitud y distancia de cadenas tales como LetterPairSimility y Levenshtein los cuales permiten encontrar posibles patrones erróneos de digitación de las palabras ingresadas y poder con esto tomar inferencias a palabras claves de instancias de guías que pueden asociarse con ellas.

```

/* Método Comparativo de palabras del usuario y guías encontradas*/
Decimal MetodoComparativo( Lista palabras_ingresadas, lista palabras_base_conocimiento)
LetterPairSimilarity metodoComparativo1 = nuevo LetterPairSimilarity();
Levenshtein metodoComparativo2 = nuevo Levenshtein() ;
Para (Entero i = 0 hasta i < palabras_ingresadas.tamaño incremento i++) Hacer
    valorPrevio = 0.0d;
    Para (Entero j = 0 hasta j < palabras_base_conocimiento.tamaño incremento j++) Hacer
        valorLetterPairSilarity = metodoComparativo1.compararCadenas(palabras_ingresadas.obtener(i), palabras_base_conocimiento.obtener(j));
        valorLevenstein = metodoComparativo2.compararCadenas(palabras_ingresadas.obtener(i), palabras_base_conocimiento.obtener(j));
        Si (valorLevenstein > valorLetterPairSilarity) Entonces
            Si ((esAceptable(valorLevenstein)) y (valorPrevio < valorLevenstein)) Entonces
                valorPrevio = valorLevenstein;
            FinSi
        Sino
            Si ((esAceptable(valorLetterPairSilarity)) y (valorPrevio < valorLetterPairSilarity)) Entonces
                valorPrevio = valorLetterPairSilarity;
            FinSi
        Fin Para
    Si (valorPrevio > 0.0d) Entonces
        numPalabrasAceptadas++;
    FinSi
Fin Para
Retorne (numPalabrasAceptadas / palabras_ingresadas.tamaño) * 100;

```

Gráfico N° 18

Por ejemplo: Si el usuario digita “aseguramiento” y alguna de las palabras clave de una guía tiene la palabra “aseguramientos” se podría inferir mediante el uso de estos algoritmos la cadena es aceptable y se podrá inferir que dicha palabra coincide con la palabras clave establecida en la guía.

Estos algoritmos trabajan en forma simultánea realizando las diferentes comparaciones que acarrear las palabras claves introducidas vs las encontradas por las instancias de guías y el sistema solo tomará el mejor grado o distancia del algoritmo asociado.

La información de las guías que se desean obtener, se obtendrán únicamente si la mayoría de las palabras procesadas que ingresó el usuario son aceptables con relación a las obtenidas de la guía, es decir, si cada una de las palabras clave ingresadas (en su mayoría) son inferidas por las que se encuentran en una instancia de guía se prosigue a obtener la información de la misma por medio de una consulta de atributos realizada por OWLAPI. Esa información es guardada y acumulada en un arreglo de resultados.

Cabe destacar que de la misma manera se realizan las demás comparaciones restantes según el recorrido de las instancias de guías establecidos por el sistema, dando así un posible incremento de información obtenida de guías en el arreglo descrito.

Una vez finalizado el recorrido se procede a enviar el arreglo hacia el componente de visualización de la información para mostrar los contenidos de las guías que cumplen con los parámetros de búsqueda.

6.4.6 Componente de Visualización de la Información

Este componente del sistema toma el arreglo de resultados y lo recorre en cada una de sus posiciones, obtiene los valores y los ingresa a un paginador de resultados en la misma interfaz del componente de consulta. El paginador muestra las diferentes guías una tras otra, de forma organizada y con la información referente a la fuente donde se encontró la guía, las palabras que cumplieron el criterio de búsqueda y contenido total de la misma relacionando a todas guías que cumplieron los criterios de búsqueda con las palabras claves establecidas en las instancias de guías de la base de conocimientos según lo muestra el siguiente gráfico.

Fuente: [Dominio_DS->](#) [Proceso_DS5->](#) [Subproceso_DS5_1->](#) [Control_ISO6_1_2->](#)
[GuíaImplementación6_1_2](#)

Palabras inferidas: [coordinacion, seguridad, informacion]

Comúnmente la **coordinación** de la **seguridad** de la **información** involucra la cooperación y colaboración de directores, usuarios, administradores, diseñadores de aplicación, auditores y personal de **seguridad**, así como habilidades especializadas en áreas como seguros, temas legales, recursos humanos, tecnología de la **información** o gestión de riesgo.

Esta actividad debería:

- a) Garantizar que las actividades de **seguridad** se efectúan en cumplimiento de la política de **seguridad** de la **información**.
- b) Identificar la forma de manejar los incumplimientos.
- c) Aprobar metodologías y procesos para la **seguridad** de la **información**, como la evaluación de riesgos y la clasificación de la **información**.
- d) Identificar cambios significativos en las amenazas y la exposición de la **información** y de los servicios de procesamiento de la **información** a las amenazas.
- e) Evaluar la idoneidad y coordinar la implementación de los controles de **seguridad** de la **información**.
- f) Promover eficazmente la educación, la formación y la concientización de la **seguridad** de la **información** en toda la organización.
- g) Valorar la **información** recibida del monitoreo y la revisión de los incidentes de **seguridad** de la **información** y recomendar las acciones apropiadas para responder a los incidentes identificados de la **seguridad** de la **información**.

Si la organización no emplea grupos con funciones separadas, por ejemplo debido a que dichos grupos no son apropiados para el tamaño de la organización, las acciones descritas anteriormente las debería llevar a cabo otros organismos de la dirección o un solo director.

Gráfico N°19

Así mismo el usuario tiene la opción de visualizar todo el contenido de cada guía, seleccionado la fuente en donde se encontró, el cual el sistema automáticamente realizara una consulta especifica únicamente de las instancias que comprenden la fuente generando una descripción más detallada de la guía especifica tal como se muestra en la siguiente imagen.

ENTREGAR Y SOPORTE (DS)

Cubre la entrega de los servicios requeridos. Incluye la prestación del servicio, la administración de la seguridad y de la continuidad, el soporte del servicio a los usuarios, la administración de los datos y de las instalaciones operacionales. Recibe las soluciones y las hace utilizables por los usuarios finales.

DS5 Garantizar la Seguridad de los Sistemas

La necesidad de mantener la integridad de la información y de proteger los activos de TI, requiere de un proceso de administración de la seguridad. Este proceso incluye el establecimiento y mantenimiento de roles y responsabilidades de seguridad, políticas, estándares y procedimientos de TI.

La administración de la seguridad también incluye realizar monitoreos de seguridad y pruebas periódicas así como realizar acciones correctivas sobre las debilidades o incidentes de seguridad identificados. Una efectiva administración de la seguridad protege todos los activos de TI para minimizar el impacto en el negocio causado por vulnerabilidades o incidentes de seguridad.

DS5.1 Administración de la Seguridad de TI

Administrar la seguridad de TI al nivel más alto apropiado dentro de la organización, de manera que las acciones de administración de la seguridad estén en línea con los requerimientos del negocio.

6.1.2 Coordinación de la seguridad de la información

Las actividades de la seguridad de la información deberían ser coordinadas por los representantes de todas las partes de la organización con roles y funciones laborales pertinentes.

GuiImplementacion6_1_2

Comúnmente la coordinación de la seguridad de la información involucra la cooperación y colaboración de directores, usuarios, administradores, diseñadores de aplicación, auditores y personal de seguridad, así como habilidades especializadas en áreas como seguros, temas legales, recursos humanos, tecnología de la información o gestión de riesgo.

Esta actividad debería:

- a) Garantizar que las actividades de seguridad se efectúan en cumplimiento de la política de seguridad de la información.
- b) Identificar la forma de manejar los incumplimientos.
- c) Aprobar metodologías y procesos para la seguridad de la información, como la evaluación de riesgos y la clasificación de la información.
- d) Identificar cambios significativos en las amenazas y la exposición de la información y de los servicios de procesamiento de la información a las amenazas.
- e) Evaluar la idoneidad y coordinar la implementación de los controles de seguridad de la información.
- f) Promover eficazmente la educación, la formación y la concientización de la seguridad de la información en toda la organización.
- g) Valorar la información recibida del monitoreo y la revisión de los incidentes de seguridad de la información y recomendar las acciones apropiadas para responder a los incidentes identificados de la seguridad de la información.

Si la organización no emplea grupos con funciones separadas, por ejemplo debido a que dichos grupos no son apropiados para el tamaño de la organización, las acciones descritas anteriormente las debería llevar a cabo otros organismos de la dirección o un solo director.

Gráfico N° 20

6.4.7 Componente de Gestión de Preferencias del usuario

En el siguiente gráfico se muestra el componente de Gestión de Preferencias del usuario tiene la capacidad de ofrecerle al usuario una interfaz gráfica que permite la visualización y edición sus datos personales y preferencias, donde este último provee criterios relacionados a la búsqueda necesita establecer para ejecutar una consulta en la base de conocimientos de guías de gobierno de TI de manera personalizada, dentro de los cuales se destacan

- 1) La Selección del tipo de dominio de gobierno de TI específico en la ontología que desea consultar.
- 2) La Selección los diferentes tipos de procesos pertenecientes a los dominios de gobierno de TI.
- 3) Porcentaje mínimo de palabras claves aceptadas para la extracción de guías.
- 4) Porcentaje de similitud para la determinación de palabras claves.

The screenshot displays the 'Preferencias del Usuario' (User Preferences) interface. At the top, there is a navigation bar with 'Buscador de Guías' and 'Preferencias del Usuario' buttons. The user is logged in as 'jcml300' and can click 'cerrar sesión'. The main content area is divided into two sections: 'Datos Personales' and 'Preferencias de Búsqueda'.

Datos Personales

Usuario	jcml300
Pass
Nombres	Juan Carlos
Apellidos	Martinez Lindado
Identificación	1140835428
Fecha Nacimiento	1990-11-03

Preferencias de Búsqueda

Grado de Aceptabilidad de Palabras Ingresadas	100 %
Grado de Similitud de Palabras	100 %

Dominios y Procesos

<input checked="" type="checkbox"/> Dominio_AI	<input checked="" type="checkbox"/> Dominio_DS	<input checked="" type="checkbox"/> Dominio_ME	<input checked="" type="checkbox"/> Dominio_PO
<input checked="" type="checkbox"/> Proceso_AI1	<input checked="" type="checkbox"/> Proceso_DS1	<input checked="" type="checkbox"/> Proceso_ME1	<input checked="" type="checkbox"/> Proceso_PO2
<input checked="" type="checkbox"/> Proceso_AI2	<input checked="" type="checkbox"/> Proceso_DS10	<input checked="" type="checkbox"/> Proceso_ME2	<input checked="" type="checkbox"/> Proceso_PO3
<input checked="" type="checkbox"/> Proceso_AI3	<input checked="" type="checkbox"/> Proceso_DS11	<input checked="" type="checkbox"/> Proceso_ME3	<input checked="" type="checkbox"/> Proceso_PO4

6.5 Escenario Ilustrativo

El sistema básicamente utiliza los datos suministrados para poder ejecutar búsquedas en este repositorio y poder obtener resultados según lo ingresado. En una búsqueda tradicional si se desean encontrar guías relacionadas a la cadena “Coordinación de la Seguridad de la Información”, consultaría todas y cada una de las instancias de la base de conocimiento con la misma cadena ingresada, comparando si alguna guía contiene exactamente igual dicha cadena, que por obvias razones ocasiona resultados poco satisfactorios, lo cual no aseguraría el éxito de la consulta.

Conforme al anterior, se desarrolló entonces, un modelo de búsqueda que aplica técnicas de procesamiento de lenguaje natural, para el reconocimiento de palabras claves que puedan inferir cualquier tipo de guía como también las preferencias del usuario para limitar las instancias que se desean consultar en la base de conocimiento.

Siguiendo el ejemplo anterior, se quiere encontrar guías que traten de “Coordinación de la Seguridad de la información”, lo cual se proporcionan tales palabras en el cuadro de búsqueda como lo muestra la figura 16, Se aplica en primer lugar un proceso de identificación de palabras claves sobre la cadena ingresada, el cual parte de la Tokenización de todas las palabras ingresadas removiendo espacios en blanco y comas (,), que acorde al ejemplo quedaría de la forma: [“Coordinación”, “de”, “la”, “Seguridad”, “de”, “la”, “Información”]. Seguidamente se eliminan las palabras que se repiten quedando únicamente las palabras únicas: [“Coordinación”, “de”, “la”, “Seguridad”, “Información”].

Luego las palabras se les deben remover sus acentos para que no tengan caracteres especiales y estar en mayúsculas, dando como resultado [“coordinacion”, “de”, “la”, “seguridad”, “información”].

Por último se aplica la utilización del marcado gramatical post tagger para identificar si estas palabras corresponden a verbos en infinitivo, adjetivos y sustantivos propios y comunes para clasificarlos como palabras claves para buscar en las instancias de la base de conocimiento.

El marcado gramatical no tiene en cuenta las palabras clasificadas como artículos, números, verbos en gerundio, posesivos, lo cual elimina palabras que no significan mucha para la búsqueda que se piensa realizar. Dando como resultado [“coordinacion”, “seguridad”, “información”] como las palabras clave de búsqueda.

Cabe destacar que la base de conocimiento el cual se va a obtener la información, cuenta con palabras clave o conceptos predefinidos por parte del experto del conocimiento en cada una de las instancias sobre las clases que conforman específicamente la información asociada de las guías de GuidesIT.

El usuario tiene opción de escoger o no las preferencias de búsqueda que le permitirán recorrer de una forma personalizada las instancias de la base de conocimiento determinando qué tipo de dominio y procesos de los mismos desea recorrer, la personalización de los porcentajes asociados a la cantidad de palabras claves mínimas para extraer las guías como también el grado de similitud para establecer comparaciones entre las mismas, tal como lo muestra el gráfico 21.

Las palabras [“coordinacion”, “seguridad”, “información”] deben compararse con las otras palabras clave encontradas durante el recorrido de la base de conocimiento con ayuda de los atributos de instancias de (Guías de Implementación y Guías de Aseguramiento) que fueron creadas y almacenadas previamente en la misma.

La comparación tiene el propósito de comprobar que tan similares son las palabras [“coordinacion”, “seguridad”, “información”] junto con las palabras encontradas de cada instancia de guías y por eso que se utiliza algoritmos de similitud y distancia durante el recorrido de la base de conocimientos.

[“coordinacion”, “seguridad”, “información”], son 3 palabras clave encontradas como resultado del procesamiento de palabras del usuario, el usuario mediante las preferencias puede indicar un porcentaje de las palabras mínimas aceptadas por los algoritmos para poder variar el número de guías a obtener; en pocas palabras si el porcentaje indica 100%, solo las guías que contengan esas mismas palabras clave serán consultadas. Dando el caso que el usuario no utilice preferencias solo se traerán guías que cumplan con la mayoría de palabras aceptadas.

Los algoritmos de similitud otorgan un grado de similitud a cada palabra clave del usuario, si el grado es alto, la palabra es contada como aceptada y si la mayoría son aceptadas, se extrae la descripción de la guía determinada a las palabras clave asociadas de ella.

El proceso anterior se realiza iterativamente hasta terminar el recorrido total de la base de conocimiento y al final se despliegan los resultados así como lo muestra el gráfico N° 19.

CAPITULO VII

EVALUACIÓN Y PRUEBAS

7.1 INTRODUCCIÓN

La fase de evaluación y pruebas concluye lo que es el ciclo de vida del presente proyecto. Esta fase mide el nivel de calidad que ofrece al usuario la aplicación creada.

7.2 VALIDACIONES

7.2.1 Validación de CSS

Dado que en el proyecto se utilizaron diversas hojas de estilo para para todas y cada una de las interfaces de la aplicación, se hizo relevante realizar una prueba de validación de CSS. Para ello con la ayuda de la página web <http://jigsaw.w3.org/cssvalidator/> se analizaron los diferentes códigos que contienen las mismas con el fin de determinar errores de sintaxis relacionados a sus propiedades, dando los siguientes resultados:

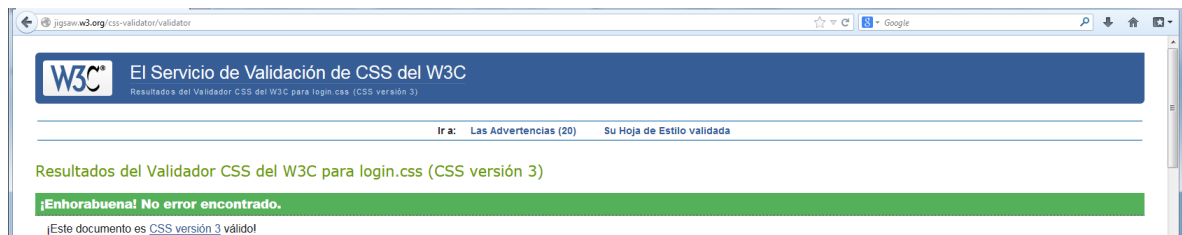


Gráfico N°22

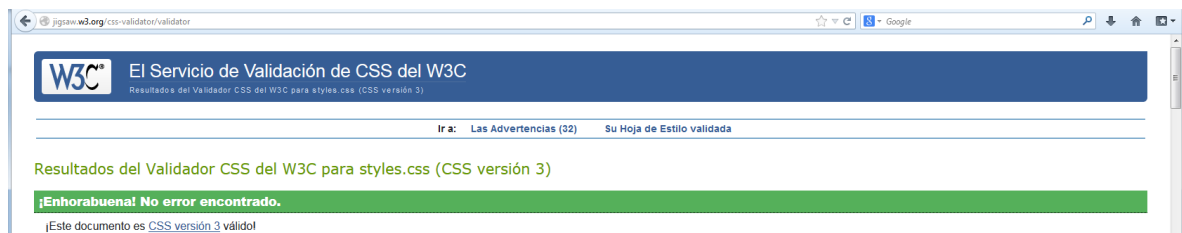


Gráfico N°23

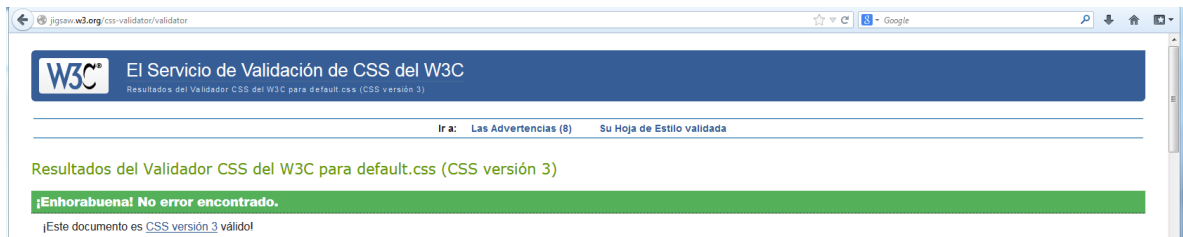


Gráfico N°24

Cabe destacar que las advertencias que describen las imágenes son ocasionadas por propiedades de las hojas de estilo que solo son reconocibles por navegadores específicos como Internet Explorer, Google Chrome o Mozilla FireFox, lo cual permiten el reconocimiento de las mismas evitando cambios visuales en las interfaces gráficas de la aplicación.

7.2.2 Validación de enlaces

Dado que el proyecto realiza muchos redireccionamientos hacia diversas páginas para poder acceder a los diferentes contenidos de la aplicación es necesario contar con una propiedad en los validadores que permitan distinguir que tipo de navegador se está utilizando y determinar qué tipo de acción utilizar para redirigir las diferentes paginas según el navegador correspondiente.

```
if (BrowserDetect.browser == 'Explorer') {  
    window.open('', '_parent', '');  
}  
if (BrowserDetect.browser == 'Mozilla') {  
    window.open('', '_parent', '');  
}  
    if (BrowserDetect.browser == 'Explorer' && BrowserDetect.version == '8') {  
        window.open("newuser.jsp");  
        window.close();  
    } else {  
        window.location.href = "newuser.jsp";  
    }  
}
```

Gráfico N°25

La propiedad para distinguir el tipo de navegador, fue testada y probada en Internet Explorer 10, Google Chrome v28 y Mozilla Firefox v22 y redirigieron todas las páginas de la aplicación sin problemas.

7.2.3 Validación de navegadores

La aplicación fue testada y validada en los navegadores de Internet Explorer v10, Google Chrome v28 y Mozilla Firefox v22 produciendo resultados favorables por el hecho de no tener cambios visuales en los componentes gráficos y también en su funcionalidad para realizar consultas, iniciar secciones de usuario y la configuración de preferencias de búsqueda. A continuación se mostraran imágenes de la aplicación ejecutante en los navegadores descritos: Nota: Para poder realizar estas pruebas se debe ingresar a la web (nombre del servidor)/SGIT/



Gráfico N°26



Gráfico N° 27



Gráfico N° 28

7.3 CASOS DE PRUEBA

Para la satisfactoria culminación del proceso del software se han diseñado un conjunto de pruebas, están hechas por un usuario No registrado y por uno Registrado con conocimientos de GuidesIT

CASOS DE PRUEBA I							
Usuario No registrado							
ID caso de prueba	Modulo a probar	Descripción	Pre-Requisito	Funcionamiento		Resultado obtenido /Observaciones	Estado
				Si	No		
CO_001	Realizar Búsqueda	En este punto el usuario no registrado debe realizar en la barra de búsqueda el texto que desea buscar	El usuario debe hacer iniciado la aplicación	x		El usuario no registrado ingresó el texto "Conocimiento" en los anexos se puede ver la imagen del CO_0001 del resultado.	Exitoso
CO_002	Autocompletar	En este punto el usuario no registrado al momento de ingresar texto en la barra de búsqueda esta debe ayudarlo a autocompletar	El usuario debe hacer iniciado la aplicación		X	El usuario no registrado ingresó el texto "Conocimiento" al ir ingresándolo el sistema no le mostró la función autocompletar	Fallido

Tabla N°29

CASOS DE PRUEBA II							
Usuario Registrado (Preferencias del Usuario)							
ID caso de prueba	Modulo a probar	Descripción	Pre-Requisito	Funcionamiento		Resultado obtenido	Estado
				Si	No		
CO_003	Registrarse	En este punto el usuario va a registrarse para acceder a la búsqueda personalizada, al entrar debe aparecer un formulario llamado "Datos personales" (Usuario, Pass, Nombres, Apellidos, identificación y fecha de nacimiento) que este debe llenar, luego "Preferencias de Búsqueda" que es el grado de exactitud de la búsqueda y "Dominios y Procesos" donde el usuario elige libremente en que parte de los dominios y procesos desea buscar.	El usuario debe hacer iniciado la aplicación	x		El usuario a registrar digita sus datos personales, elige el grado de exactitud de la búsqueda y elige como el prefiera los dominios y procesos sin ningún problema y la información se guardó. La imagen respectiva se encuentra en los anexos.	Exitoso

Tabla N° 30

CASOS DE PRUEBA II							
Usuario Registrado (Preferencias del Usuario)							
ID caso de prueba	Modulo a probar	Descripción	Pre-Requisito	Funcionamiento		Resultado obtenido	Estado
				Si	No		
CO_004	Autenticarse	En este punto el usuario deberá ingresar su usuario y contraseña	El usuario se ha debido registrar para realizar este modulo	x		El usuario entra al sistema. La imagen respectiva se encuentra en los anexos.	Exitoso
CO_005	Modificar Datos Personales	En este punto el usuario podrá cambiar todos sus datos personales a excepción de su nombre de usuario	El usuario se ha debido registrar para realizar este modulo	x		El usuario cambia exitosamente todos sus datos personales a excepción de su nombre de usuario	Exitoso

Tabla N°31

CASOS DE PRUEBA II							
Usuario Registrado (Preferencias del Usuario)							
ID caso de prueba	Modulo a probar	Descripción	Pre-Requisito	Funcionamiento		Resultado obtenido	Estado
				Si	No		
CO_006	Realizar Búsqueda	En este punto el usuario realiza su búsqueda específica. Para este caso, el usuario realizará la búsqueda con un grado de exactitud del 75%	1. El usuario debe haber iniciado sesión. 2. El usuario se ha debido registrar para realizar este modulo	x		El usuario registrado digitó “ procedimientos de escalamiento de excepciones de control” arrojándole el resultado deseado. La imagen de este caso se encuentra en Anexos	Exitoso
CO_007	Modificar Grado de exactitud de Búsqueda	En este punto el usuario registrado modifica el grado de exactitud de búsqueda hasta un 100%	El usuario se ha debido registrar para realizar este modulo	x		El usuario registrado digitó “ normas de la tecnología de la empresa” arrojándole el resultado deseado. Las imágenes de este caso se encuentra en Anexos	Exitoso

Tabla N°32

CONCLUSIONES

8.1 Conclusiones y Trabajo Futuro

Esta investigación tuvo como objetivo, diseñar e implementar un prototipo de consulta a una base de conocimiento en el área de Auditorías de Gobierno de Tecnología Informática mediante las herramientas y métodos que proporciona la ingeniería del conocimiento para la construcción del sistema informático.

La propuesta se basa en el uso del procesamiento del lenguaje natural (PLN) y el uso de las preferencias de usuario dentro del modelo semántico. El resultado, un sistema de búsqueda para GuidesIT llamado SGIT. El conocimiento se representa mediante las técnicas de la Ingeniería del Conocimientos, el diseño de la solución y la validación del conocimiento, hasta que el conocimiento acumulado de un dominio sea traducido en un código probado y refinado, para conformar la base de conocimiento del sistema. La base de conocimiento del estudio realizado por los expertos se refiere a la articulación del Modelo COBIT con las Norma ISO 2000 llamado GuidesIT para la exitosa implantación de un Gobierno de TI.

A nivel de la información suministrada para el usuario final en un futuro es preciso el uso de otros modelos como ITIL o estándares que se puedan alinear con el Modelo COBIT teniendo así un buscador para guías de TI más amplio.

Y a nivel técnico es preciso considerar los siguientes aspectos como:

- 1.) La Investigación de otros tipos de mecanismos acceso y consulta eficientes para la base de conocimiento del sistema
- 2.) La implementación de la funciones como el autocompletado para facilitar al usuario el ingreso de términos de consulta.
- 3.) La Investigación de otros tipos de algoritmos comparativos que sean más precisos en realizar inferencias para conseguir mejores resultados en lo que se busca.
- 4.) La Implementación de un usuario maestro que gestione el registro de los usuarios en el sistema y otorgue ciertos permisos de acceso y consulta de los contenidos almacenados en la base de conocimiento.

BIBLIOGRAFIA

- (1) Montaña, V. (2010) "Guías para la implantación de un efectivo gobierno de TI" Educosta, 730 p.17.
- (2) O. Corcho. (2001, Oct.), Methodologies tools and languages for building ontologies. Where is their meeting point?, Universidad Politécnica de Madrid, España. [En línea], Recuperado en: http://www.aegean.gr/culturaltec/Kavakli/MIS/papers/Corcho_2003.pdf
- (3) Abreo, C., Padilla, C. (2012) "Prototipo para recuperación semántica de información de proyectos de ingeniería de sistemas con base en un repositorio ontológico" Vol.3, num 1, Enero- Julio, pp 109-118.
- (4) Gómez-Pérez A., Fernández-López M. y Corcho M. (2004), Ontological Engineering, Springer Verlag, London, Mayo.
- (5) IEEE (1996) IEEE Standard for Developing Software Life Cycle Processes.
- (6) O. Corcho. (2001, Oct.), Methodologies tools and languages for building ontologies. Where is their meeting point?, Universidad Politécnica de Madrid, España. [En línea], consultado en Agosto 12 de 2011, disponible en: http://www.aegean.gr/culturaltec/Kavakli/MIS/papers/Corcho_2003.pdf
- (7) Gómez-Pérez A, Juristo N, Montes C, Pazos J (1997) Ingeniería del Conocimiento: Diseño y Construcción de Sistemas Expertos. Ceura, Madrid, Spain
- (8) Absolute astronomy, Unified process, absolute astronomy. [En línea], recuperado en : http://www.absoluteastronomy.com/topics/Unified_Process
- (9) Abreo, C., Padilla, C. (2012) "Prototipo para recuperación semántica de información de proyectos de ingeniería de sistemas con base en un repositorio ontológico" Vol.3, num 1, Enero- Julio, pp 109-118.
- (10) Gašević D., Dejuric D., Devežvic V., Model Driving Engineering and Ontology Development, Springer Verlag, Berlin, Febrero, 2009.
- (11) Moreno C. and Sanchez Y. (2012) "prototipo de buscador semántico aplicado a la búsqueda de libros de ingeniería de sistemas y computación en la biblioteca Jorge Roa Martínez de la Universidad tecnológica de Pereira", p.20.
- (12) González, Conesa. (2011) "AgillTIL: Creación de una Wiki Semántica de soporte a los procesos de control", Universidad Oberta de Catalunya.

- (13) ITIL – Gestión de Servicios TI (s.f). Recuperado en: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/que_es_ITIL/que_es_ITIL.php
- (14) SPARQL, SPARQL Query Language for RDF, Recuperado en: <http://www.w3.org/TR/rdf-sparql-query/>
- (15) Isaza, G., A. Castillo et al. (2008). *Modelo de Detección de Intrusos basado en Sistemas Multi-agente, Inteligencia Computacional y representaciones Ontológicas*. X RECSI - X Reunión Española sobre Criptología y Seguridad de la Información. Salamanca (España), Salamanca. PP 271-281.
- (16) Rojas, M., Montilva, J.y Barrios, J. (2009)."OntosMD: Un método para hacer ontologías durante el modelado de negocios". Revista Colombiana de Tecnologías de Avanzada, Vol.1, Número 13,2009.
- (17) Lasheras (2011)."Marco de Representación y reúso de requisitos de seguridad", Tesis Doctoral, Universidad de Murcia, Departamento de Informática y Sistemas.
- (18) Protégé, open source ontology editor and knowledge-base framework, Recuperado en <http://protege.stanford.edu/>
- (19) Abreo, C., Padilla, C. (2012) "Prototipo para recuperación semántica de información de proyectos de ingeniería de sistemas con base en un repositorio ontológico" Vol.3, num 1, Enero- Julio, pp 109-118.
- (20) COBIT 4.1. ISACA. Marco de Trabajo, objetivos de control, directrices gerenciales y modelos de madurez.
- (21) Reyes., (2005) "Análisis de la relación entre la ingeniería del conocimiento y la gestión del conocimiento en base al modelo de Nonaka y Takeuchi" © Intangible Capital - Nº 9 - Vol. 1, Julio-Septiembre - ISSN: 1697-9818 (Cod:0033)
- (22) E. Rich., (1994) "Inteligencia Artificial", McGraw-Hill / Interamericana de España, segunda edición.
- (23) J. Haugeland, La inteligencia articula. Siglo XXI,1988.
- (24) R. Blake, Sistemas electrónicos de comunicaciones. Cengage LearningLatin America, 2004.

- (25) G. P. Martin sanz and M. S. Peñas, Inteligencia artificial e ingeniería del conocimiento, México: Alfaomega, 2006.
- (26) Peis, E.; Herrera-Viedma, E.; Hassan, Y.; Herrera, J. C. (2003) “Análisis de la web semántica: estado actual y requisitos futuros”. En: El profesional de la información, septiembre-octubre, v. 12, n. 5, pp. 368-376
- (27) T. Berners-Lee. , (1989) Information management: a proposal, Consultado en: 20-07-03. Recuperado en <http://www.w3.org/History/1989/proposal.html>
- (28) T. Berners-Lee, J.,Hendler, O., Lassila, (2001) “The semantic web: a newform of web content that is meaningful to computers will unleash a revolution of new possibilities”. Scientific American.
- (29) O. Lassila, R. Swick., (1999) Resource description framework (RDF) model and syntax specification: W3C recommendation. Recuperado en. <http://www.w3.org/TR/REC-rdf-syntax>
- (30) Brickley, R. V. Guha, (2002) Resource description framework (RDF) schema specification 1.0: W3C working draft, Consultado en: 20-07-13. <http://www.w3.org/TR/rdf-schema>
- (31) Choque Aspiazu, Guillermo Ontología para la Ingeniería del Software. Carrera de Informática, Universidad Mayor de San Andrés, La Paz, Bolivia, recuperado de: http://www.espe.edu.ec/portal/files/sitio_congreso_2011/papers/C2.pdf
- (32) W3C, OWL Web Ontology Language Recuperado en: <http://blog.classora.com/2012/12/18/owl-el-lenguaje-de-ontologias-web-del-w3c/>
- (33) W3C, Resource Description Framework (RDF), Recuperado en: <http://www.w3.org/RDF/>
- (34) Protégé, open source ontology editor and knowledge-base framework, Recuperado en (<http://personales.upv.es/ccarrasc/doc/2001-2002/Protege2000/PROTEGE.htm>)
- (35) G. Salton (1983) Introduction to modern information retrieval, McGraw-Hill, New York, p. 448. Modelo Semántico de Expansión de Consultas para la Búsqueda Web (MSEC)
- (36) Y Marcano and R. Talavera, (2006) “Gestión de la información a través de la Web Semántica: Iniciativas y dificultades,” Revista Venezolana de Gerencia (RVG), vol. 11, No. 36, October, p. 36.

(37) P. Mitra, N. Noy, and A. Jaiswal, (2005) "Ontology Mapping Discovery with Uncertainty," Fourth International Conference on the Semantic Web, Galway Ireland 6th – 10th November, p. 15.

(38) D. Gayo-Avello, A Meta-Analysis of State-of-the-Art Electoral Prediction From Twitter Data, Social Science Computer Review 0894439313493979, first published on August 23, 2013. doi:10.1177/0894439313493979

(39) Andrés T. Hohendahl "Procesamiento de Lenguaje Natural Robusto" Laboratorio de Estereología y Mecánica Inteligente, Facultad de Ingeniería, UBA. Recuperado en: http://web.fi.uba.ar/~ahohenda/docs/prolen_robust.pdf

(40) Levenshtein Algorithm. [En línea], Recuperado en: <http://www.levenshtein.net/>

(41) LetterPairSilarity Algorithm. [En línea], recuperado en: http://infocenter.pentaho.com/help/index.jsp?topic=%2Fpdi_user_guide%2Freference_step_fuzzy_match.html

(42) WEISER, M. The computer for the 21st century. Scientific American. 1991, vol. 265, núm. 3, pp. 94-104.

(43) ALMENÁREZ, F. Arquitectura de seguridad para entornos de computación ubicua abiertos y dinámicos. Tesis doctoral. Escuela Politécnica Superior, 2005.

(44) BEN MOKHTAR, S.; GEORGANTAS, N. e ISSARNY, V. COCOA: CONversation-based service COMposition in pervAsive computing environments with QoS support. Journal of Systems and Software. 2007, vol. 80, núm. 12, pp. 1941-1955.

(45) W3C, XML Extensible Markup Language (XML). Recuperado en: <http://www.w3.org/XML/>

(46) W3C, HyperText Markup Language (HTML). Recuperado en: <http://www.w3.org/community/webed/wiki/HTML>

(47) W3C, Cascading Style Sheets (CSS). Recuperado en: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>

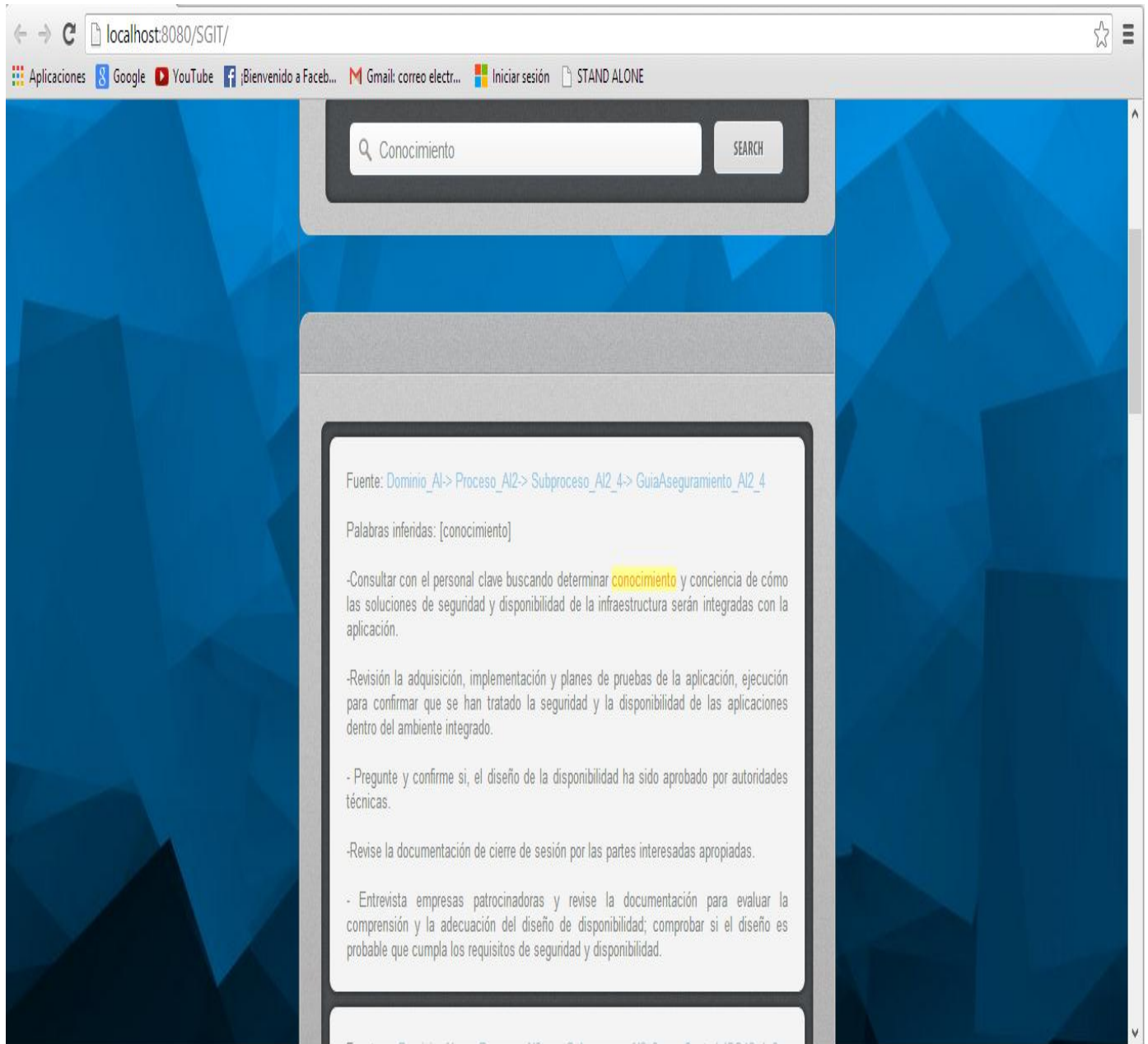
(48) W3C, JavaScript Language. Recuperado en: <http://www.pergaminovirtual.com.ar/definicion/JavaScript.html>

(49) W3C, JQuery JavaScript framework . Recuperado en: <http://www.actualidadjquery.es/acerca-de/>

- (50) W3C, AJAX de Asynchronous JavaScript And XML . Recuperado en: <http://www.etnassoft.com/biblioteca/aspects-of-ajax/>
- (51) Ibáñez J 20, Extracción de datos web usando técnicas de screen-scraping, Universidad Politécnica de Valencia. p. 11-15. Recuperado en: http://riunet.upv.es/bitstream/handle/10251/10104/PFC_Javier_Iba%C3%B1ez.pdf
- (52) Ejemplos de Servlet y JSP Web Application Development Recuperado en: http://webap.wikispaces.com/file/view/JSP_Servlet.pdf
- (53) Descripción y características del potente y moderno lenguaje de programación Java Recuperado en: <http://www.desarrolloweb.com/articulos/497.php>
- (54) Biblioteca OpenNLP Apache, Recuperado en: <http://opennlp.apache.org/>
- (55) OWL Java API, Recuperado en: <http://owlapi.sourceforge.net/>
- (56) JespXML, Recuperado en: <https://code.google.com/p/analizador-xml/>
- (57) JSON-lib java library, Recuperado en: <http://json-lib.sourceforge.net/>
- (58) Netbeans IDE, Recuperado en: (<http://ocw.udl.cat/enginyeria-i-arquitectura/programacio-2/continguts-1/1-introduccioi81n-a-netbeans.pdf>)
- (59) Protégé, open source ontology editor and knowledge-base framework, Recuperado en <http://protege.stanford.edu/>
- (60) GlassFish servidor de aplicaciones de software, Recuperado en: (<http://es.wikipedia.org/wiki/GlassFish>)
- (61) Datos y Metadatos, Recuperado en <http://blog.classora.com/2013/02/28/metadatos-definicion-aplicaciones-y-estandares/>
- (62) García, E., Sicilia, M.A. (s/f) “Una propuesta para la búsqueda semántica de recursos web de nuevas tecnologías aplicadas a la educación”. En línea <http://tecnologiaedu.us.es/ticsxxi/comunic/egb-masu.htm>
- (63) Méndez J, Iglesias E, Riverola F, Díaz F., and Corchado J., (2005) "Tokenising, Stemming and Stopword Removal on Anti-spam Filtering Domain", ;in Proc. CAEPIA, pp.449-458.

ANEXOS

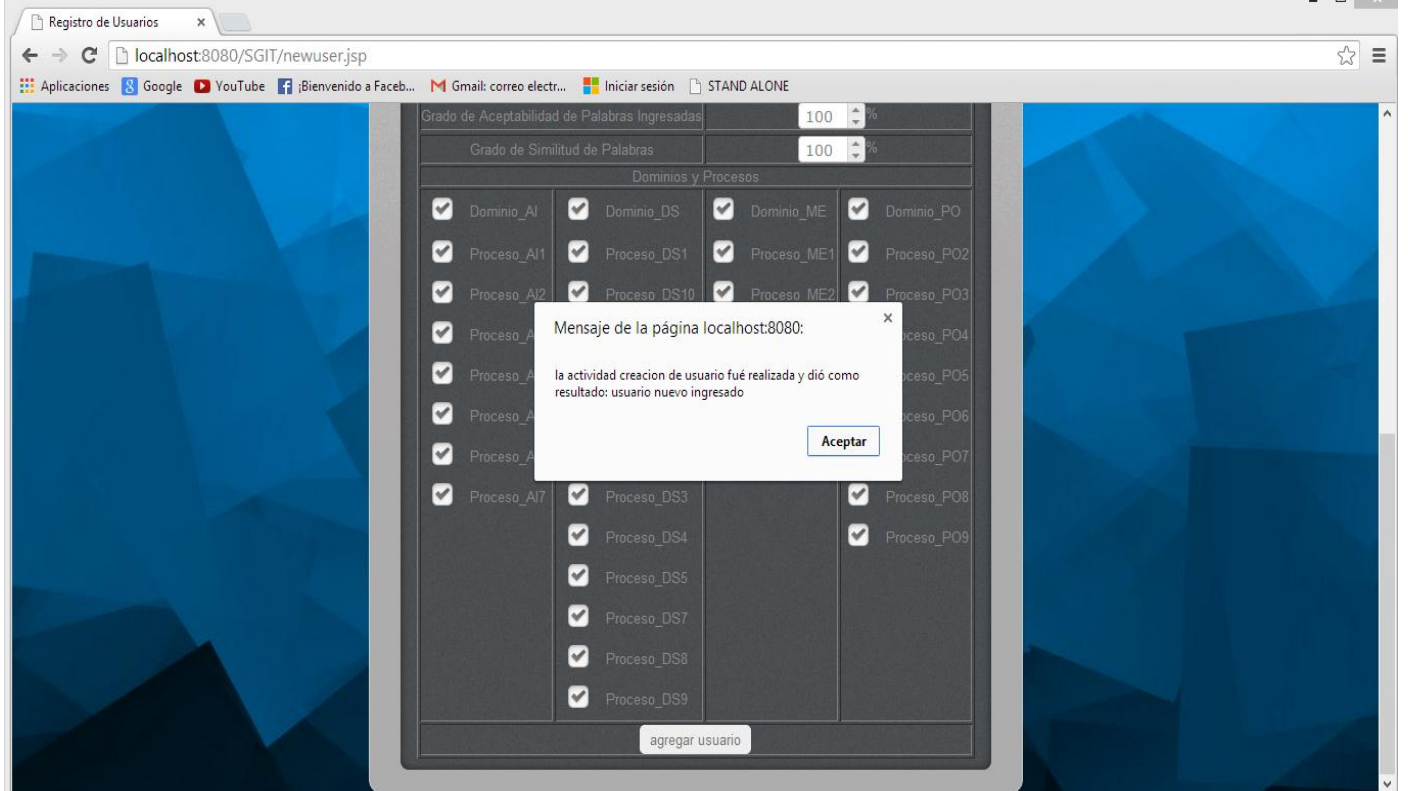
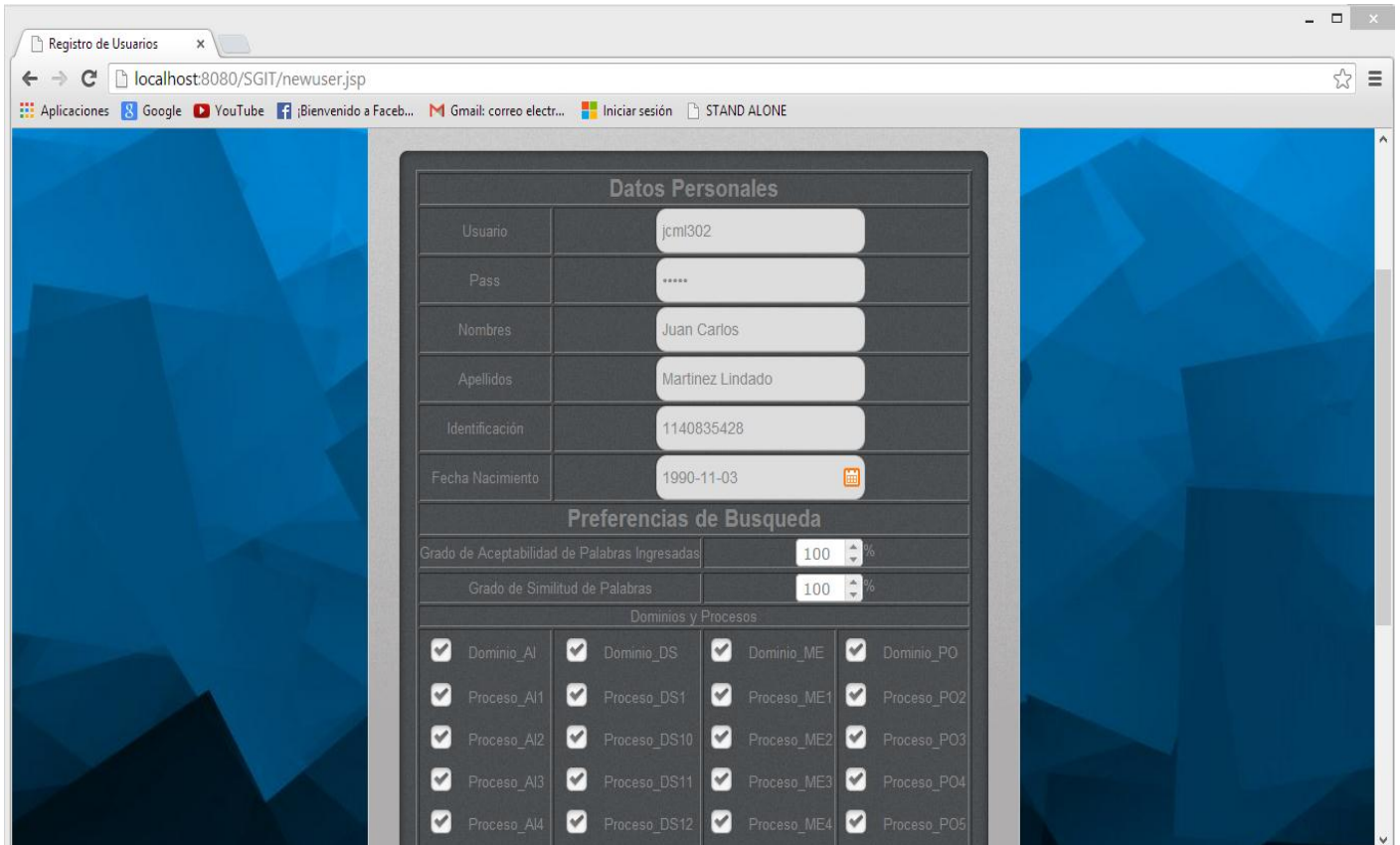
Resultados Casos de Prueba



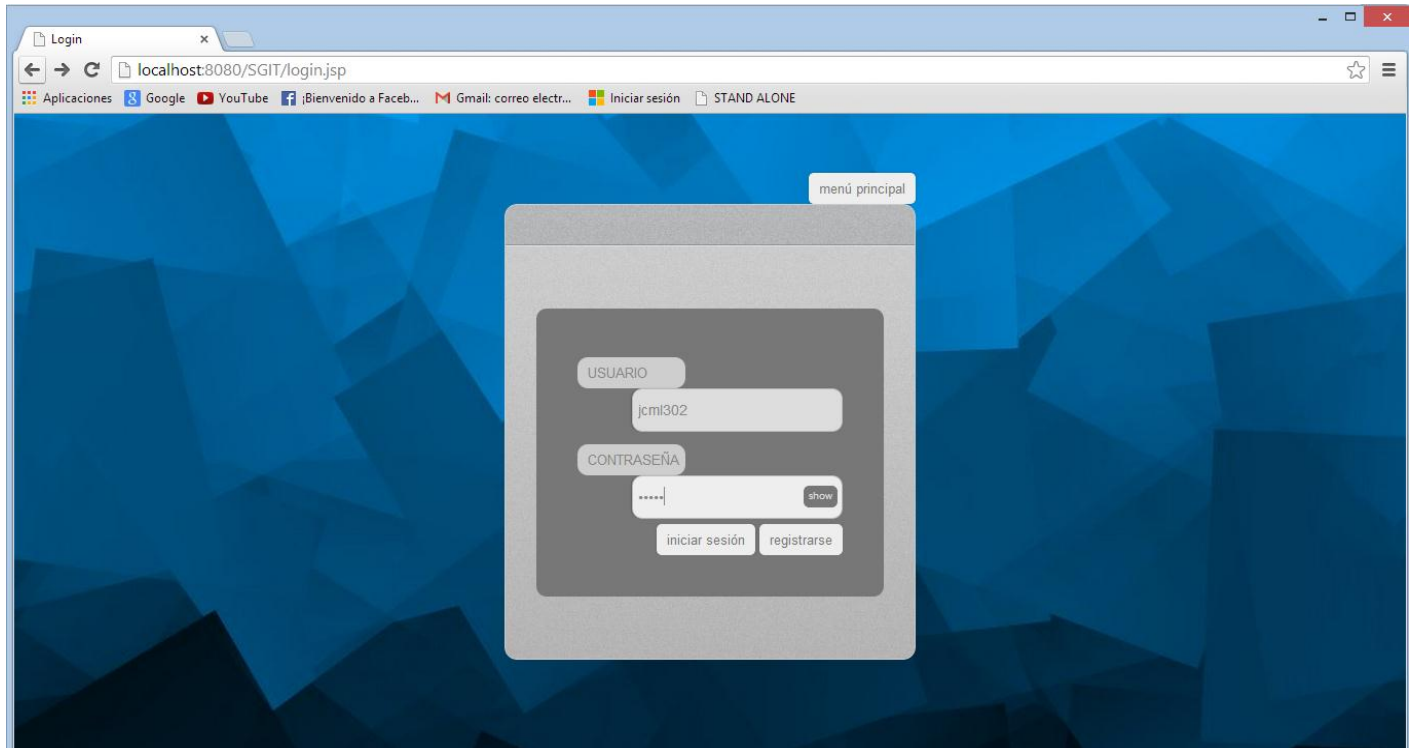
The screenshot shows a web browser window with the address bar displaying 'localhost:8080/SGIT/'. The browser's taskbar includes icons for 'Aplicaciones', 'Google', 'YouTube', 'Bienvenido a Faceb...', 'Gmail: correo electr...', 'Iniciar sesión', and 'STAND ALONE'. The main content area features a search bar with the text 'Conocimiento' and a 'SEARCH' button. Below the search bar, the source of the information is listed as 'Fuente: Dominio_AI-> Proceso_AI2-> Subproceso_AI2_4-> GuiaAseguramiento_AI2_4'. The search results are titled 'Palabras inferidas: [conocimiento]' and consist of a list of five bullet points:

- Consultar con el personal clave buscando determinar **conocimiento** y conciencia de cómo las soluciones de seguridad y disponibilidad de la infraestructura serán integradas con la aplicación.
- Revisión la adquisición, implementación y planes de pruebas de la aplicación, ejecución para confirmar que se han tratado la seguridad y la disponibilidad de las aplicaciones dentro del ambiente integrado.
- Pregunte y confirme si, el diseño de la disponibilidad ha sido aprobado por autoridades técnicas.
- Revise la documentación de cierre de sesión por las partes interesadas apropiadas.
- Entrevista empresas patrocinadoras y revise la documentación para evaluar la comprensión y la adecuación del diseño de disponibilidad; comprobar si el diseño es probable que cumpla los requisitos de seguridad y disponibilidad.

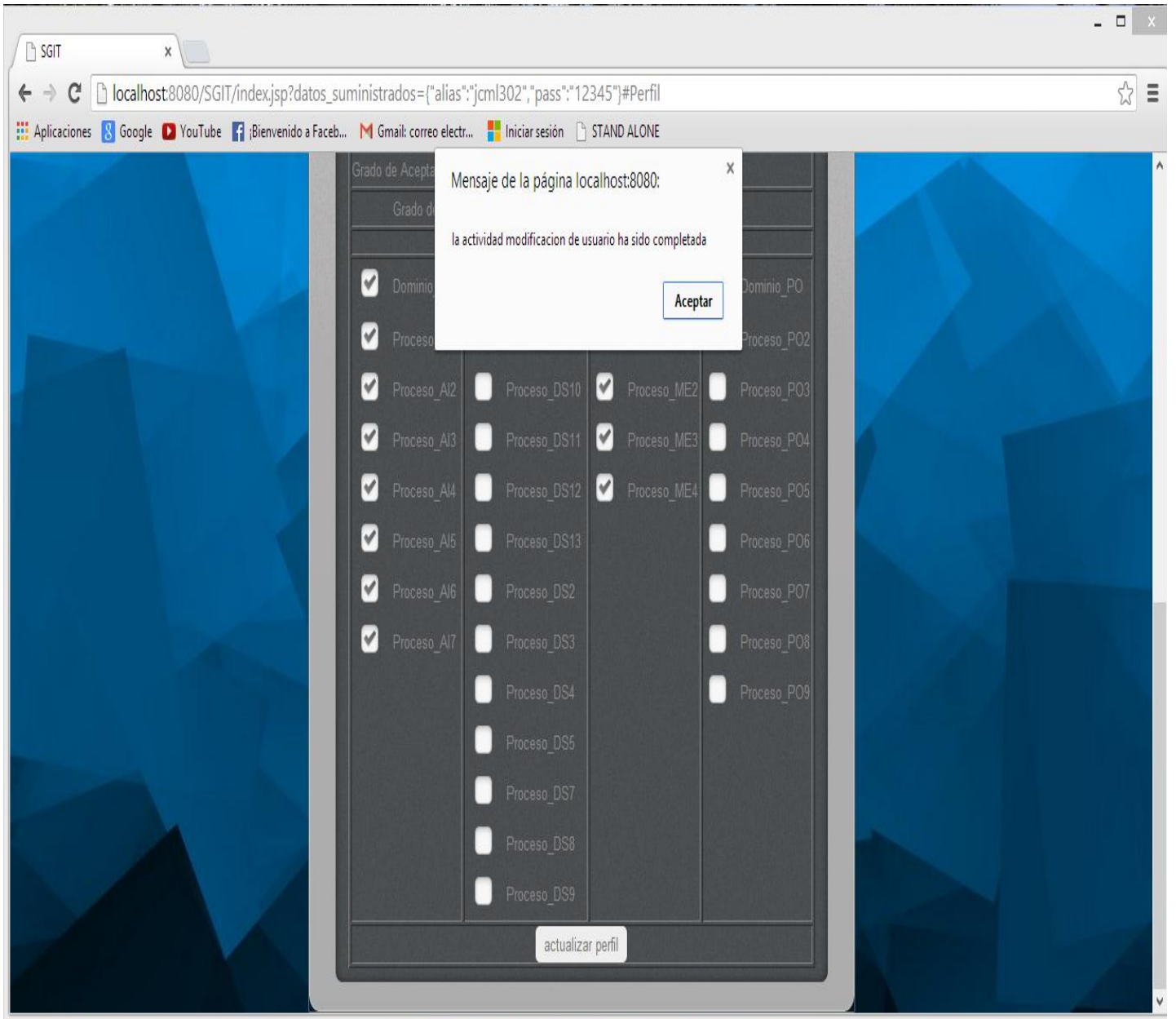
Resultado Caso CO_001




Resultado Caso CO_003



Resultado Caso CO_004



Resultado Caso CO_005

Datos Personales	
Usuario	<input type="text" value="jcml302"/>
Pass	<input type="password" value="....."/>
Nombres	<input type="text" value="Juan Carlos"/>
Apellidos	<input type="text" value="Martinez Lindado"/>
Identificación	<input type="text" value="1140835428"/>
Fecha Nacimiento	<input type="text" value="1990-11-03"/> 
Datos Personales	
Grado de Aceptabilidad de Palabras Ingresadas	<input type="text" value="75"/> %
Grado de Similitud de Palabras	<input type="text" value="100"/> %

Fuente: [Dominio_ME->](#) [Proceso_ME2->](#) [Subproceso_ME2_3->](#)
[GuiaAseguramiento_ME2_3](#)

Palabras inferidas: [procedimientos, escalamiento, excepciones, control]

- Confirmar que las políticas son los umbrales que se establece para los niveles aceptables de **control** de **excepciones** y de **control** de averías.

- Confirmar que los **procedimientos** de **escalamiento** de **excepciones** de **control** se han comunicado e informó a las empresas de TI y las partes interesadas (por ejemplo, a través de la intranet, copia impresa **procedimientos**). Los **procedimientos** de escalada deben incluir criterios o umbrales para la escalada (por ejemplo, las **excepciones** de **control** a menos de una cantidad específica de impacto no es necesario se intensificó, las **excepciones** de **control** mayor que una cantidad específica de impacto deben informar inmediatamente a CIO, y las **excepciones** de **control** mayor que una cantidad específica de impacto exigirá la notificación inmediata a la junta de directores). Entrevista de gestión para evaluar el conocimiento y el conocimiento de los **procedimientos** de escalada, así como análisis de causa raíz y presentación de informes.

- Confirme que los individuos se les ha asignado la responsabilidad para el análisis de causa raíz y presentación de informes, así como la resolución de excepción.

Fuente: Dominio_PO->
GuiaAseguramiento_PO3_4

Proceso_PO3->

Subproceso_PO3_4->

Palabras inferidas: [normas, tecnología, empresa]

Verificar que las **normas** de la **tecnología** de la **empresa** están siendo aprobadas por el Consejo de la arquitectura de TI. Evaluar la eficacia del proceso de comunicación de las **normas** técnicas para los miembros del personal de TI (por ejemplo, gestores de proyectos, arquitectos de información). Entrevista pertinentes al personal de TI para determinar su conocimiento de las **normas** técnicas.

Determinar a partir de la administración de TI que el seguimiento y la evaluación comparativa de los procesos se ponen en marcha para confirmar el cumplimiento de los estándares de **tecnología** y directrices establecidas.

Evaluar la documentación técnica de análisis de viabilidad de proyectos seleccionados para evaluar el cumplimiento de estándares de la **tecnología** empresarial.

Resultado Caso CO_007