



Universidad de las Ciencias Informáticas

Facultad # 9

Título

**SERVICIO WEB DE NOMENCLADORES
PARA LA OFICINA NACIONAL DE
RECURSOS MINERALES**

**Trabajo para optar por el Título de Ingeniero en
Ciencias Informáticas**

Autores: Yaricel Alcantara Laurencio

Reinier Corzo Utra

Tutor: Ing. Neysis Hernández Díaz

Co-tutor: Ing. Armando Ortiz Cabrera

Ciudad de la Habana, Julio 2008

“Año 50 de la Revolución”

UCi



La ciencia es universal y tenemos que aprender del mundo, de la misma manera que debemos estar siempre dispuestos a mostrar el aporte que nosotros podemos obtener para ellos.

Fidel Castro

*A mis padres y hermana con todo mi amor,
Por haberse dedicado a mi formación,
Por confiar plenamente en mí,
y no haber escatimado ni un momento, para ver realizado este
gran sueño...*

Yaricel

Este trabajo es dedicado a aquellas personas que pusieron su granito de arena, y de una forma u otra siempre estuvieron ahí presente cuando más lo necesité, pero en especial:

A mis padres (Euna Caridad y Miguel Corzo) y hermano (Doiunys Corzo). A mi abuelita, a mi novia Danais y a mis amigos de la vocacional, en especial a Yoandy Pérez y Yoandy Campo.

Reinier

De Yaricel

En estos momentos me gustaría agradecerles a tantas personas que de una forma u otro me han apoyado, pero sería imposible poderlas mencionar a todas, por lo que en especial les agradezco de todo corazón a:

- ◆ A mi hermana que siempre ha estado ahí para apoyarme y ayudarme a cumplir este maravilloso sueño.
- ◆ A mis padres que sin su confianza y apoyo en todos los momentos no hubiese sido posible llegar a ser una profesional.
- ◆ A mi novio Yerandy Vázquez Arencibia que me ha ayudado con todo su amor y que siempre me dio fuerzas para seguir adelante hasta lograr todo lo que he alcanzado.
- ◆ A mi mejor amigo Lenier Jiménez Domínguez que siempre ha estado a mi lado brindándome su confianza y cariño.
- ◆ A mis compañeros de aula que han aportado su granito de arena en la realización de este trabajo en especial a Yulaine Arias Guerra, a Yulien Figueredo Guzmán y a Jose A. Fernández.
- ◆ A mis amistades que estuvieron en todo momento a mi disposición, en especial a los profesores Yaresis Pérez Heredia, a Jorge infante Osorio y a Armando Ortiz Cabrera.
- ◆ A mi grupo de profesores que me han brindado sus conocimientos en estos cinco años de mi carrera.
- ◆ A Clara Conill González que siempre me ha dado su ayuda y apoyo incondicional.
- ◆ Agradecer el apoyo que me ha brindado mi tutora Neysis Hernández Díaz.
- ◆ Gracias a la revolución y a mi comandante en jefe por haberme dado la posibilidad de hacer realidad el sueño de prepararme profesionalmente logrando alcanzar el título de Ingeniero en Ciencias Informáticas.

Muchas gracias a todos por ayudarme a convertir este sueño realidad...

De Corzo

Le agradezco a todas aquellas personas que me han apoyado y a mis amistades de las diferentes aulas por la que he pasado, todos me han ayudado a ser mejor persona y compañero, pero en especial:

- ◆ A mis padres y hermano que me estuvieron apoyando desde que fui el primer día a la escuela hasta hoy, sin ellos no sería esto que soy hoy, y a los que le debo la vida.
- ◆ A mi abuelita que aunque ya no se encuentra físicamente, me dio su apoyo para seguir adelante, y me enseñó lo que es luchar por la vida.
- ◆ A mi novia Danais por su apoyo y ayuda incondicional durante todo este curso, y por la paciencia que ha tenido conmigo, gracias chinita.
- ◆ Geordanis Baños Vega por contar siempre con él para lo que fuera necesario.
- ◆ David Rodríguez por su carisma y amistad desinteresada que siempre obtuve desde primer año.
- ◆ José Luis Muñoz por ayudarme cada vez que lo necesite, que no fueron pocas las veces.
- ◆ Otros que por algún motivo u otro ya no se encuentran en la escuela pero que también pusieron su granito de arena en su momento.
- ◆ Neisys mi tutora que nos ayudo muchísimo desde el principio del proyecto.

Muchas gracias a todos por su gran apoyo.

DEDICATORIA DE LA AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año_____.

Yaricel Alcantara Laurencio

Autor

Reinier Corzo Utra

Autor

Neysis Hernández Díaz

Tutor

OPINIÓN DE USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado < Título del Trabajo de Diploma >, fue realizado en < Nombre de Lugar >. Esta Entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface.

- Totalmente
- Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma le reportan a esta Entidad los beneficios siguientes (cuantificar):

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a _____ < valor en MN o USD del efectivo económico >. Y para que así conste, se firma la presente a los _____ Días del Mes de _____ del Año _____.

Representante de la Entidad

Cargo

Firma

Cuño

OPINIÓN DE LA TUTORA DEL TRABAJO DE DIPLOMA

Título: Servicio Web de Nomencladores para la ONRM

Autores: Yaricel Alcantara Laurencio

Tutora: Ing. Neysis Hernandez Diaz

Reinier Corzo Utra

Co-tutor: Ing. Armando Ortiz

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

El trabajo realizado se desarrolló con muy alta independencia y responsabilidad por parte de los diplomantes, los cuales demostraron en todo momento estar preparados para asumir y acometer correctamente el trabajo propuesto.

La labor de ambos fue desempeñada de forma ardua y constante, lo que les permitió desenvolverse fácilmente y avanzar sin grandes contratiempos. La creatividad se vio demostrada por la versatilidad del software desarrollado y el tiempo dedicado. Además demuestran su laboriosidad y responsabilidad a la hora de asumir tareas.

La alta calidad científico-técnica, se evidencia en la investigación realizada, demostrando un alto dominio de los conceptos actuales del desarrollo de software. Los mismos, apoyándose en una gran variedad de bibliografía, fueron capaces de realizar profundos análisis y llegar a conclusiones que proveen al trabajo de un alto valor científico.

El documento presentado tiene una estructura adecuada, hace un uso correcto del lenguaje y refleja de manera clara y concisa todas las etapas desarrolladas en la investigación. El trabajo contiene resultados que poseen valor para ser presentados en eventos y talleres científicos relacionados.

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros Informáticos; y propongo que se le otorgue al Trabajo de Diploma la calificación de 5 puntos.

Ing: Neysis Hernández Díaz

Fecha

RESUMEN

En la actualidad la información es el eslabón fundamental en las instituciones cubanas, ya sean Empresas, universidades u otras organizaciones. En la Oficina Nacional de Recursos Minerales existe un desarrollo constante y acelerado de las TIC, dicho volumen de información ha ido aumentando considerablemente al paso de los años, llegando a convertirse en un obstáculo para la empresa.

Con el objetivo de facilitar la búsqueda y acceso a los recursos que existen en la oficina, se han diversificado los esfuerzos, y como respuesta se está construyendo un conjunto de aplicaciones que en cierto modo garantizará la organización.

Sin embargo todavía no se podrá decir que toda la información estará agrupada y disponible para los trabajadores de la empresa pues existen factores como la diversidad, descentralización y duplicación que atentan contra un desarrollo exitoso de las aplicaciones que se están creando. Por lo que se hace necesario priorizar la concentración de los datos. Este trabajo se propone estudiar la propuesta de un servicio web que una y organice toda la información referente a las aplicaciones y al mismo tiempo la brinde de forma confiable.

Por tanto este trabajo comprende como objetivo concreto: Desarrollar un Servicio Web, el cual proporcione mecanismos de comunicación estándares entre las diferentes aplicaciones web que interactúan, así como ofrecer la información de manera estandarizada y centralizada.

ABSTRACT

Nowadays, the information is the main part of the Cuban institutions, either in enterprises, universities, or some other organizations. In the National Office of Mineral Resources there is a constant and accelerated development of the Information and Communication Technologies, such volume of information has been increasing considerably with the time, becoming in an obstacle for the company.

In order to make easier the search and access to the existing resources in the office, the efforts have been multiplied, and as a response it is being built a set of applications that somehow will ensure the organization.

However, it can not be said yet that all the information will be pooled and available for the workers of the enterprise as there are factors such as diversity, decentralization, and duplication that attempt on a successful development of the applications that are being created. That's why it is necessary to focus on the concentration of data. This work aims at studying the proposal of a web service that joins and organizes all the information related to applications and at the same time provides it in a reliable way.

Therefore this work includes as specific objective: to develop a web service, which provides standard communication mechanisms between the different web applications that interact, as well as to offer the information standardized and centralized.

RESUMEN.....	X
ABSTRACT.....	XI
INTRODUCCION.....	I
CAPÍTULO 1 FUNDAMENTACIÓN DEL TEMA.....	5
Introducción.....	5
1.1 Servicios.....	6
1.1.1 Servicio Informático	6
1.1.2 Servicio Web	6
1.2 Aplicación Informática	7
1.2.1 Aplicación Web	8
1.3 Base de datos.....	8
1.3.1 Bases de Datos Dinámicas	9
1.4 Conceptos asociados al dominio del problema	9
1.4.1 Dominio de datos geológico.....	9
1.4.2 Listas Código.....	9
1.4.3 Variables de la Geología.....	10
1.4.4 Nomencladores	10
1.5 Estado Actual del Negocio.....	11
1.5.1 Objeto de Estudio	11
1.5.2 Descripción actual del dominio del problema.....	11
1.5.3 Situación Problemática	15
1.5.4 Fundamentación del objetivo	16
1.5.5 Objeto de automatización.....	16
1.5.6 Objetivo general propuesto	17

1.5.7 Propuesta de solución	17
1.6 Análisis de otras soluciones existentes	17
1.7 Conclusiones parciales	18
CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS A UTILIZAR.....	19
Introducción.....	19
2.1 Herramientas, metodologías y lenguaje para el modelado de la Capa de Servicios Web	20
2.1.1 El Lenguaje Unificado de Modelado (UML) como soporte de lenguaje orientado a objetos para el modelado de aplicaciones Web.....	20
2.1.2 Metodología de Desarrollo de Software	22
2.1.2.1 Programación Extrema (XP)	22
2.1.2.2 Desarrollo Guiado por Funcionalidad (FDD).....	23
2.1.2.3 Proceso Unificado de Desarrollo de Software (RUP).....	23
2.1.2.4 Selección de la Metodología a utilizar	27
2.1.3 Herramienta CASE de Desarrollo de Software.....	27
2.1.3.1 Rational Rose.....	27
2.1.3.2 Visual Paradigm.....	28
2.1.3.3 Selección de la herramienta CASE a utilizar	28
2.2 Lenguajes y Herramientas a utilizar para la creación y funcionamiento de la Capa de Servicios Web.	29
2.2.1 Lenguajes de programación para la Web.....	29
2.2.1.1 Personal Home Page (PHP).....	29
2.2.1.2 ASP.NET	30
2.2.1.3 Processing Extraction Report Language (PERL)	30
2.2.1.4 Fundamentación del lenguaje a utilizar.....	31

2.2.2 Servidores Web	32
2.2.2.1 Apache.....	32
2.2.2.2 AOLserver	32
2.2.2.3 Roxen webservice.....	33
2.2.3 Sistema Gestor de Base de Datos (SGBD)	33
2.2.3.1 Oracle	34
2.2.3.2 PostgreSQL.....	35
2.2.3.3 MySQL.....	35
2.2.3.4 Fundamentación del Sistema Gestor de Base de Datos a utilizar (PostgreSQL).....	36
2.3 Fundamentación del IDE de programación a utilizar	37
2.3.1 Eclipse.....	37
2.4 Fundamentación de otras herramientas a utilizar para el desarrollo del trabajo.	37
2.4.1 Gestor de referencia: EndNote 9	37
2.4.2 Planner (Gestor de Proyectos).....	38
2.5 Conclusiones Parciales	39
CAPÍTULO 3 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	40
Introducción.....	40
3.1 Entorno donde trabajará el sistema	41
3.1.1 Conceptos principales del entorno.....	41
3.1.1.1 Balance	41
3.1.1.2 Metadatos	41
3.1.1.3 Nomencladores	41
3.1.1.4 Base de Datos Referativa	41

3.1.1.5 Concesionario	41
3.1.1.6 Geodatos.....	42
3.1.2 Eventos principales del entorno.....	42
3.1.3 Diagrama de clases del Modelo de Dominio	43
3.2.1 Requisitos Funcionales	44
3.2.2 Requisitos No Funcionales.....	46
3.4 Descripción del Sistema Propuesto	47
3.4.1 Descripción de los actores.....	47
3.4.2 Modelo de Caso de Uso del Sistema	48
3.4.3 Descripción de los Casos de Uso del Sistema.....	48
3.5 Conclusiones parciales	56
CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA	57
Introducción.....	57
4.1 Diagrama de Clases del Diseño.....	58
4.1.1 Modelo de Diseño.....	58
4.1.2 Principios de Diseño	62
4.1.3 Concepción general de ayuda.....	64
4.2 Diseño de la Base de Datos.....	64
4.2.1 Descripción de las tablas de la Base de datos.	65
4.2.2 Diagrama de Clases Persistentes	68
4.3 Modelo de Despliegue	68
4.3.1 Descripción de los componentes del Diagrama de Despliegue.....	69
4.4 Modelo de Implementación	70
4.4.1 Diagrama de Componentes	70

4.4.1.2 Diagrama de Componente del Caso de Uso Buscar elementos de las Listas Código.....	72
4.4.1.3 Diagrama de Componente Caso de Uso Buscar los sinónimos de los elementos de las Listas Código	73
4.5 Prueba del sistema propuesto	73
4.6. Conclusiones Parciales	75
CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD.....	76
Introducción.....	76
5.1 Estimación de costos	77
5.2 Puntos de función sin ajustar (COCOMO II)	77
5.2.1 Propósito	78
5.3 Beneficios tangibles e intangibles	83
5.4 Análisis de costo y beneficio	84
5.5 Conclusiones parciales	84
CONCLUSIONES GENERALES.....	85
RECOMENDACIONES	86
REFERENCIAS BIBLIOGRAFICAS.....	87
GLOSARIO DE TÉRMINOS.....	89
ANEXOS.....	94

INTRODUCCION

Al igual que en otras naciones o países en Cuba existe una institución dedicada al control de los recursos minerales, con el objetivo de garantizar el aprovechamiento racional de dichos recursos en el país. Esta institución es la Oficina Nacional de Recursos Minerales la cual surge a partir de la promulgación de la ley 76 “Ley de Minas” que fue concebida el 23 de enero de 1995.

La Oficina Nacional de Recursos Minerales es la autoridad minero petrolera de la República de Cuba, es la encargada de administrar el conocimiento geológico y toda la información geológica, minera y petrolera de la nación. Con el fin de que todos los datos se encuentren almacenados de manera organizada y tener conocimiento de cada unas de las actividades que se realicen en el sector de la geología y de la minería, a nivel nacional.

Actualmente la ONRM se encuentra sumergida en un importante proceso de informatización, la cual esta desarrollando un grupo de proyectos que forman parte de un programa integral denominado PNICG (Proyecto Nacional de Informatización del Conocimiento Geológico). Este se desarrolla en cooperación conjunta entre la ONRM y la Universidad de las Ciencias Informáticas (UCI), con el fin de adentrarse en un proceso de informatización de sus diferentes actividades (control de concesionario, archivo técnico, balance de recursos y reservas del país, los meta datos y la información de las investigaciones geológicas). Automatizando con esto el flujo de trabajo que se realiza en las mismas, ya que actualmente poseen un gran número de información en copia dura (papel), que con el paso del tiempo gran parte de esta información se ha perdido o deteriorado.

La Universidad de las Ciencias Informáticas (UCI) es una institución de educación superior dedicada a la preparación de ingenieros en ciencias informáticas y al desarrollo de software tanto internamente en el país como fuera del mismo.

En el grupo de proyectos que se desarrolla en la UCI surgen un gran número de aplicaciones, las cuales usan alguna información en común llamada nomencladores que no es más que descripciones de la información que utiliza cada aplicación. Estos nomencladores se usan por separado y en algunas ocasiones están repetidos en varias aplicaciones o incluso el mismo descriptor de información con nombres diferentes. Provocando que se dificulte el trabajo al ser utilizados estos descriptores de la información y que exista un descentralización de ellos.

Según lo descrito anteriormente, la ONRM conjuntamente con la UCI ha decidido crear y brindar un servicio que estandarice y centralice los nomencladores para el uso de las aplicaciones

que se desarrollaran en el grupo de proyectos PNICG. Con el fin de tener mejor control de ellos, porque toda la información estaría almacenada, segura y disponible para aquellos usuarios que tengan acceso a la misma.

Dada la explicación expuesta en la situación problemática surge el **problema a resolver** que queda formulado de la siguiente forma:

¿Cómo perfeccionar el uso y la disponibilidad de los nomencladores para las aplicaciones que se desarrollaran en el proyecto PNICG para la ONRM?

Luego de haber analizado la situación existente en la ONRM surge la necesidad de encontrar una solución para dicha situación teniendo como **objeto de estudio** de la investigación: proceso de gestión de los descriptores de la información (nomencladores), usados por las aplicaciones que se desarrollan en el proyecto PNICG para la ONRM.

Para concretar de forma consciente los resultados esperados y determinar en esencia que se persigue con la investigación, se define que el **objetivo general** de este trabajo es desarrollar un servicio web (Web Service) que permita centralizar, estandarizar y ofrecer información detallada de los nomencladores a las diferentes aplicaciones que se desarrollan en el proyecto PNICG.

Se puede definir a partir del objetivo general de este trabajo los objetivos específicos que se persiguen.

1. Adquirir conocimientos acerca de toda la información referente a los nomencladores utilizados a nivel nacional por la ONRM.
2. Realizar un previo estudio de las herramientas y tecnologías para poder diseñar e implementar el Servicio Web.

Constituye el **campo de acción** para precisar objeto de estudio de la investigación: El proceso de gestión de información de la ONRM.

Para cumplir el objetivo general y los objetivos específicos propuestos y darle solución a la situación problemática planteada, se proponen las tareas siguientes:

- Realizar entrevistas y encuestas a los a los trabajadores de la ONRM
- Comprobar la validez del problema existente.

- Realizar un estudio exhaustivo de los fundamentos teóricos de los nomencladores nacionales, específicamente los usados por la ONRM.
- Analizar las aplicaciones de los nomencladores en la rama de la geología.
- Definir cada uno de los descriptores de la información con lo que se trabaja en la ONRM.
- Realizar un estudio sobre las tecnologías a utilizar para el desarrollo de las Aplicaciones Web.
- Estudio de la arquitectura SOA y sus aplicaciones para los servicios web.
- Estudiar la planificación de los recursos humanos y materiales en el tiempo basado en dos métodos; punto de función y COCOMO II.
- Análisis y estudio del funcionamiento de los servicios Web.
- Desarrollar la aplicación de forma modular haciendo uso de la metodología RUP como proceso iterativo e incremental y el modelado a través del lenguaje de UML.
- Desarrollar un Servicio Web para centralizar, estandarizar y ofrecer información detallada de los nomencladores a las diferentes aplicaciones del proyecto PNICG teniendo en cuenta el estudio preliminar realizado sobre las tecnologías a utilizar para el desarrollo de las Aplicaciones Web.
- Prueba y validación del sistema implementado.

La **idea a defender** que se plantea es: con el desarrollo de un Servicio Web, el cual proporciona mecanismos de comunicación estándares entre las diferentes aplicaciones web que interactúan; es posible presentarle a las demás aplicaciones que se desarrollan para la ONRM en el proyecto PNICG, la información de los nomencladores de manera estandarizada, dinámica y centralizada.

Para lograr un mejor entendimiento y obtener una mayor comprensión de la situación existente, se pondrán en práctica los siguientes métodos científicos:

Como Métodos Empíricos se realizaron entrevistas a profesionales de la ONRM para recoger toda la información necesaria sobre los nomencladores, se desarrollaron encuestas a directivos y trabajadores de la ONRM sobre cómo se desarrolla el trabajo con los nomencladores.

Además en cuanto a el Método Teórico se uso el “Método Histórico Lógico”, que sirve de apoyo para investigar si existían proyectos informáticos de este tipo implementados en el país, y en caso de existir, conocer cómo es su funcionamiento.

Se utilizó el analítico_ sintético, que permitió el estudio de la bibliografía utilizada así como las tesis y los sitios web que abarcan el tema a investigar. Además permitió que se realizara una recapitulación de la misma.

También dentro del Método Teórico se utilizará la Modelación, que ayudará a modelar los diagramas para lograr un mejor entendimiento de lo que se va a implementar.

Estructura de la investigación:

La presente investigación está estructurada por 5 capítulos, a continuación se expone brevemente una descripción de cada uno.

Capítulo 1: Se tratan conceptos específicos del dominio del problema que originan el estudio de la historia del arte.

Capítulo 2: Se abordan temas relacionados con las herramientas, lenguaje y gestor de base de datos a utilizar, así como el lenguaje de modelado y la metodología a emplear. Además se describen temas asociados con la tecnología a utilizar para el desarrollo de Servicio Web y temas vinculados al sistema operativo a utilizar.

Capítulo 3: Se comprenden las características del sistema, presentación de la solución propuesta. Descripción de la información que se maneja, de la aplicación que se propone así como de todos los requisitos funcionales (RF) y no funcionales (RNF) con los cuales se debe cumplir.

Capítulo 4: Construcción de la solución propuesta. Incluye la definición del modelo de análisis del sistema y del modelo de clases. Describe los diagramas de secuencia del modelo del diseño para cada realización de los casos de uso. Muestra el diagrama de clases del diseño, modelo de datos y diagramas de implementación (Componentes y Despliegue).

Capítulo 5: Se analiza el estudio de la factibilidad, la planificación, los costos y los beneficios que reporta la investigación.

1

CAPÍTULO 1 FUNDAMENTACIÓN DEL TEMA

Introducción

En el presente capítulo se explica de manera precisa los conceptos de servicios y de aplicaciones, entre otros conceptos importantes, que serán necesarios para el conocimiento de los diferentes temas en que se fundamenta este trabajo y los conceptos asociados al dominio del problema. Por otro lado se hace una descripción más general del objeto de estudio, se identifica los principales problemas que motivan la investigación y se realiza un análisis de la situación en las Oficina Nacional de Recursos Minerales (ONRM), que determina la necesidad de crear un Servicio Web. Además de realizar una investigación de las soluciones existentes entre ellas las que existen en Cuba que ayuden a prevenir o disminuir el problema.

1.1 Servicios

Según Lamb, Hair y McDaniel, "un servicio es el resultado de la aplicación de esfuerzos humanos o mecánicos a personas u objetos. Los servicios se refieren a un hecho, un desempeño o un esfuerzo que no es posible poseer físicamente. [1]

Los servicios son actividades identificables, intangibles y percederas que son el resultado de esfuerzos humanos o mecánicos que producen un hecho, un desempeño o un esfuerzo que implican generalmente la partición del cliente y que no es posible poseer físicamente, ni transportarlos o almacenarlos, pero que pueden ser ofrecidos en renta o la venta; por tanto pueden ser el objeto principal de una transacción ideada para satisfacer las necesidades o deseos de los clientes. [2]

Teniendo en cuenta las anteriores definiciones de servicio, se plantea a modo de resumen que un servicio es:

Un servicio es el resultado de llevar a cabo necesariamente al menos una actividad que al ser concluida proporcione algún beneficio a terceros y generalmente esta actividad es intangible.

1.1.1 Servicio Informático

Conjunto de actividades (planeamiento, análisis, diseño, programación, operación, entrada de datos, autoedición, bases de datos.) asociadas al manejo automatizado de la información que satisfacen las necesidades de los usuarios de este recurso. [3]

1.1.2 Servicio Web

Un Servicio Web (en inglés Web Service) es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma.[4]

Los Servicios Web permiten a las aplicaciones compartir información e incluso invocar funciones de otras aplicaciones independientes, sin importar el sistema operativo, plataforma en la que se ejecuten o incluso el dispositivo que se utilice para acceder a ellos. Los Servicios Web aunque son independientes pueden tener comunicación entre ellas y formar grupos de colaboración para poder realizar una tarea específica.[4]

Los Servicios Web son fáciles de publicar, invocar y utilizar; los usuarios no necesitan un estudio bien detallado de los mismos, solo es necesario tener un conocimiento básico de su manipulación. El funcionamiento de los servicios Web está basado en un conjunto de estándares que permiten su creación, registro, ubicación y utilización. Además pueden comunicarse entre sí, ya que están basados en XML¹ que puede ser interpretado por cualquier aplicación.[5]

Para resumir los diferentes conceptos que se exponen de Servicio web, se concluye que es un sistema de información distribuido que permite acceder a la mayoría de los recursos que ofrece Internet, a través de un medio ameno y sencillo con entorno gráfico que son los navegadores.

1.2 Aplicación Informática

Son los programas con los cuales el usuario final interactúa, es decir, son aquellos programas que permiten la interacción entre el usuario y la computadora. Esta comunicación se lleva a cabo cuando el usuario elige entre las diferentes opciones o realiza actividades que le ofrece el programa.

Juan Salvador Castejón Garrido, Secretario del CIIRM²; define aplicación informática mediante la figura que se muestra a continuación. [6]

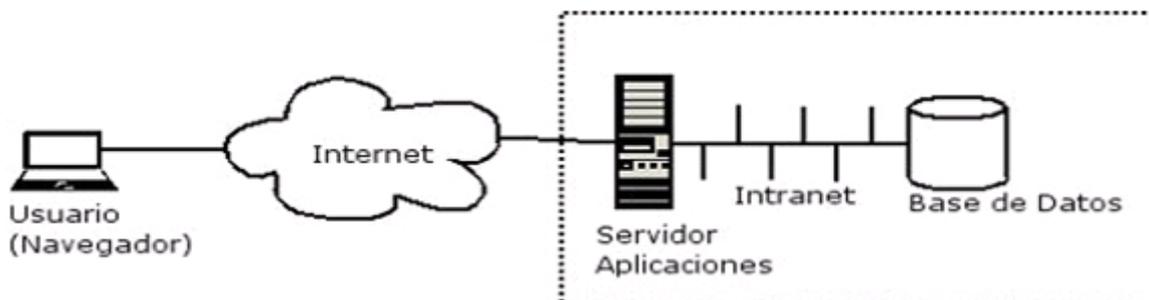


Figura 1: Descripción gráfica de aplicación informática

Es un programa de ordenador que se compra ya realizado y listo para usar. Las hay de muy diversos tipos, según el propósito con que se hayan diseñado: procesadores de texto, bases de datos, programas de contabilidad, de facturación, entre otras. [7]

¹ Extensible Markup Language (Lenguaje de Marcas Extensibles).

² Colegio de Ingenieros en Informática de la Región de Murcia.

Las aplicaciones por su uso se pueden dividir en dos grandes tipos:

- Aplicaciones de Escritorio (almacenan tradicionalmente sus datos en un archivo, aunque algunos tienen la capacidad de colaborar con otros usuarios, o almacenar sus datos en una base de datos central.)
- Aplicaciones Web

A lo largo de la historia de las aplicaciones se ha demostrado con grandes resultados que las aplicaciones informáticas, ya sean de escritorio o Web han tenido un impacto crucial en la economía de un país, pues estas han jugado un gran papel en el progreso de la humanidad en todas las esferas (educación, salud, cultura, deporte, economía, entre otras).

1.2.1 Aplicación Web

Con el surgimiento de Internet surgió la necesidad de publicar la información, diversos son los métodos que hoy día existen con este fin. Las aplicaciones Web son las encargadas de manejar el estado del negocio y de gestionar datos almacenados con ayuda de algún Sistema Gestor de Base de Datos (SGBD) con el fin de poder ser observadas por disímiles usuarios.

Finalmente, se puede resumir en que una aplicación Web es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Es un sitio que se encuentra en la Red, al cual los usuarios acceden con el fin de beneficiarse con el uso de la información que posea. Aborda un determinado tema. Se dice que los usuarios se benefician con la información que este les brinda porque una vez que acceden a él, pueden interactuar con la aplicación, actualizarla o modificarla según los privilegios de seguridad de la misma que estos tengan.

1.3 Base de datos

Una base de datos o banco de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática.

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación:

Según la variabilidad de los datos almacenados

1. Bases de datos bibliográficas
2. Bases de datos de texto completo
3. Bases de datos estáticas
4. Bases de datos dinámicas

1.3.1 Bases de Datos Dinámicas

Estas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, etc.

1.4 Conceptos asociados al dominio del problema

Nuestro dominio del problema tiene asociado un grupo de conceptos que se pasara inmediatamente a describir, entre los que se encuentran: dominio de datos geológicos, las listas código, variables de la geología y los nomencladores.

1.4.1 Dominio de datos geológico

El dominio de datos geológico es el conjunto de datos de la geología donde hay variables cuyo dominio de definición es una lista de palabras codificadas.

1.4.2 Listas Código

Las listas códigos es el conjunto de palabras que constituye el dominio de una variable y que dan valores a las variables de la geología que son codificadas, constituidas por un identificador y un nombre. Esos dominios o listas de palabras que dan valores a esos tipos de variables son privativos de la geología, aparecen en muchos campos del conocimiento y a

menudo se les domina tesoro, que no es más que un listado de palabras o términos empleados para representar concepto.

Se llaman Listas Códigos el conjunto de palabra de un mismo tema que contiene todos los términos utilizados a través del tiempo y según han sido utilizados por los diferentes autores, muchos de esos términos son equivalentes y por tanto debe ser indicada esa equivalencia para ser recuperados independientemente del término que se utilice en la realización de búsquedas.

Cada una de la lista código esta formada por varios elementos, algunos de estos elementos poseen un grupo de sinónimos que no es mas que palabras que tienen un significado similar o idéntico, pero tienen distinta escritura y ortografía, aunque se refieren a las mismas cosas. Todos los sinónimos de una lista son el universo total de valores que puede tomar una variable.

1.4.3 Variables de la Geología

Las variables de la geología son los atributos de los rasgos, son las que definen en si cada elemento de la geología, es decir no son más que aquellos elementos por lo que esta conformado la naturaleza, ejemplo:

Rocas en sus mas diversas formas (sedimentarias, calizas, acidas entre otras) .Minerales metálicos y no metálicos (zeolita, serpentina, cuarzo entre otras).

1.4.4 Nomencladores

Existen Listas Códigos para las cuales se han declarado como oficiales mediante algún instrumento legal un determinado subconjunto de palabras que forman parte de ellas, ese subconjunto de términos oficialmente asumidos constituye un nomenclador. Un ejemplo ilustrativo de esto es la lista código de provincias, donde aparecen los nombres de las antiguas provincias y las vigentes, las cuales constituyen el nomenclador provincias de la división política administrativa vigente .

Los nomencladores son las listas de código que están sobre un mandato oficial, no son más que un caso particular de una lista de código cuyas definiciones están aprobadas a un nivel determinado.

1.5 Estado Actual del Negocio

1.5.1 Objeto de Estudio

Después de haber realizado un estudio profundo de la situación existente en la ONRM, el cual está plasmado en la introducción de este trabajo investigativo. Se hace imprescindible determinar la necesidad de encontrar una solución para dicha situación teniendo como objeto de estudio de la investigación: proceso de gestión de los descriptores de la información (nomencladores), usados por las aplicaciones que se desarrollan en el proyecto PNICG para la ONRM.

1.5.2 Descripción actual del dominio del problema

EL Ministerio de la Industria Básica” (MINBAS), el cual fue creado en 1980, es el encargado de dirigir, ejecutar y controlar la política del Estado y del Gobierno, en cuanto a las actividades de:

1. Generación, transmisión, distribución y comercialización de la energía eléctrica.
2. Producción de la industria del combustible y los lubricantes, y su distribución.
3. Búsqueda, exploración y extracción de petróleo y gas.
4. Búsqueda, exploración, extracción y beneficio de minerales sólidos.
5. Producción de fertilizantes, fibras químicas, gases industriales, artículos de plásticos, plaguicidas, pinturas, colorantes, barnices y otros productos químicos.
6. Producción de neumáticos, cámaras y productos del caucho.
7. Producción y conversión de papel, cartón y pulpa celulosa.
8. Producción de vidrio y sus artículos.

Este ministerio tiene una estructura de 8 uniones que comprenden las ramas de generación y distribución de electricidad; extracción, refinación y distribución de petróleo; extracción y procesamiento de Níquel; producción de sal, extracción y procesamiento de minerales; producción de pinturas, fertilizantes, gases industriales y medicamentos; industria de neumáticos, de papel, cemento y vidrio. En este organismo no existen delegaciones territoriales, solo existe en cada provincia un Consejo de Cooperación formado por las empresas de ese territorio, con un presidente que es el director de una de estas empresas y que su función es de coordinación e información.

Existen además cuatro entidades independientes que realizan las siguientes funciones:

- Oficina Nacional de Recursos Minerales (ONRM) es la rectora nacional para garantizar la racional explotación y utilización de los recursos minerales e implementar el marco jurídico para el desarrollo y control de la geología, la minería y el petróleo.
- La Escuela Superior es la encargada del sistema de superación e información de jefes y trabajadores sobre temas de carácter técnico, político y económico.
- El Sanatorio Nacional Obrero brinda servicios médicos a los trabajadores de la Industria Básica.
- La Empresa de Computación ejerce un doble papel, al asumir la función estatal rectora de la Computación, Automatización y Comunicaciones en el MINBAS, y a la vez, una función empresarial, ambas de carácter nacional dentro del Ministerio.

Nuestro trabajo esta dirigido específicamente al análisis y solución de los problemas existente en una de estas cuatros entidades, en la ONRM. Como bien se explicó anteriormente la ONRM es la autoridad minero petrolera de la República de Cuba. Es la encargada de administrar el conocimiento geológico y toda la información geológica, minera y petrolera de la nación; con el fin de que todos lo datos se encuentren almacenados de manera organizada y tener conocimiento de cada unas de las actividades que se realicen en el sector de la geología y de la minería, a nivel nacional.

En esta oficina en el año 2001 el Comité Ejecutivo del Consejo de Ministros de Cuba aprueba las funciones que debe realizar la ORNM (acuerdo 3985), adscripta al Ministerio de la Industria Básica y son:

- Fiscalizar y controlar la actividad minera y el uso racional de los recursos minerales, según lo dispuesto en la legislación vigente.
- Fiscalizar y controlar las actividades de exploración-producción de hidrocarburos líquidos y gaseosos, según las disposiciones legales vigentes.
- Asesorar al Ministerio de la Industria Básica y demás organismos de la administración central del estado sobre las actividades mineras de exploración producción de hidrocarburos, sin perjuicio de sus debidas competencias.
- Responder por el registro minero y el registro petrolero y mantener actualizadas las anotaciones sobre las concesiones mineras, áreas mineras reservadas, los contratos de exploración-producción de petróleo, yacimientos, manifestaciones minerales y de hidrocarburos, áreas en investigación, minas y pozos de hidrocarburos en explotación o abandonados.

- Ejercer la inspección estatal sobre las personas naturales y jurídicas que ejecuten actividades mineras y de exploración producción de hidrocarburos, en el territorio nacional o en la zona económica exclusiva, para controlar y comprobar el cumplimiento de las disposiciones legales vigentes que rigen estas actividades.
- Aprobar los cálculos de reservas minerales sólidos, líquidos y gaseosos; además de registrar y mantener actualizado el inventario de dichas reservas, así como certificar el grado de aprobación de dichas reservas para su asimilación industrial u otros fines.
- Ser el depositario de la información geológica, minera y petrolera de la nación, recibir, organizar y conservar la información, así como brindar servicios de información técnica.
- Emitir dictámenes técnicos sobre el otorgamiento, anulabilidad y extinción de las concesiones mineras.
- Fiscalizar el cumplimiento de las condiciones bajo las cuales se otorgó la concesión.
- Proponer y en su caso dictar las disposiciones de carácter técnico que regulen la investigación, la explotación, el procesamiento y el uso de los recursos minerales.
- Proponer y en su caso dictar las disposiciones técnicas que regulen los trabajos de exploración-producción de petróleo, basadas en el óptimo conocimiento de las características geológicas y el potencial petrolífero de las zonas objeto de exploración, así como en la protección y el uso racional de las reservas de las reservas de hidrocarburos.
- Aprobar los proyectos de explotación y procesamiento, según corresponda, de minas, aguas minerales, fangos medicinales, salinas e hidrocarburos, de acuerdo con las disposiciones legales vigentes.
- Controlar la ejecución de los planes de preservación del medio ambiente y de las medidas para mitigar el impacto ambiental establecidas por los organismos competentes a las entidades que en el territorio nacional o en la zona económica exclusiva de Cuba realicen actividades mineras y petroleras.
- Mantener actualizadas las estadísticas mineras y de exploración-producción de petróleo y gas del país.
- Controlar el estado del fondo de pozos de petróleo y gas natural del país y dictar las disposiciones que permitan establecer un efectivo control de los pozos, como parte principal del sistema de explotación de los yacimientos.
- Participar en el cierre de minas, yacimientos y pozos de petróleo, certificar dicho cierre y controlar las medidas del programa de cierre que se ejecuten.
- Proponer y elaborar la documentación para la declaración de las áreas mineras reservadas.
- Rectorar la organización y división del territorio nacional y la zona económica exclusiva de Cuba en bloques o parcelas para la realización de trabajos de exploración-producción de petróleo y gas, certificando sus dimensiones y límites a los efectos de considerar los

mismos como el objeto de los contratos petroleros que se negocien con personas extranjeras.

- Revisar y elaborar dictámenes para el nivel superior correspondiente, sobre los proyectos de contratos petroleros que se negocien con personas extranjeras en los aspectos de su competencia. Asimismo, elaborar dictámenes en los casos de paralización, anulación y extinción de dichos contratos.
- Controlar la ejecución de las disposiciones vigentes sobre la seguridad minera y petrolera en las actividades de exploración-producción y dictar las medidas que correspondan cuando se detecte la posibilidad de ocurrencia de un accidente de trabajo o peligro para la seguridad humana.
- Procesar y generalizar la información especializada conservada en su archivo técnico.
- Brindar servicios de consultoría y asesoría técnica en la esfera de su competencia.
- Asesorar a los organismos y entidades competentes en la evaluación de planes, programas de desarrollo ramales y estudios de factibilidad, de los proyectos de explotación de los recursos minerales y los hidrocarburos.
- Aprobar y controlar los objetivos que se ejecuten financiados por el presupuesto estatal para la actividad geológica de minerales y de hidrocarburos.

La Oficina Nacional de Recursos Minerales tiene una estructura de un solo nivel, su oficina central, que tiene las direcciones que a continuación se relacionan:

- Dirección General.
- Dirección de Documentación e Informática.
- Dirección de Control Económico.
- Dirección de Registro Minero.
- Dirección de Evaluación Económica.
- Dirección de Minerales.
- Dirección de Níquel.
- Dirección de Hidrocarburos.

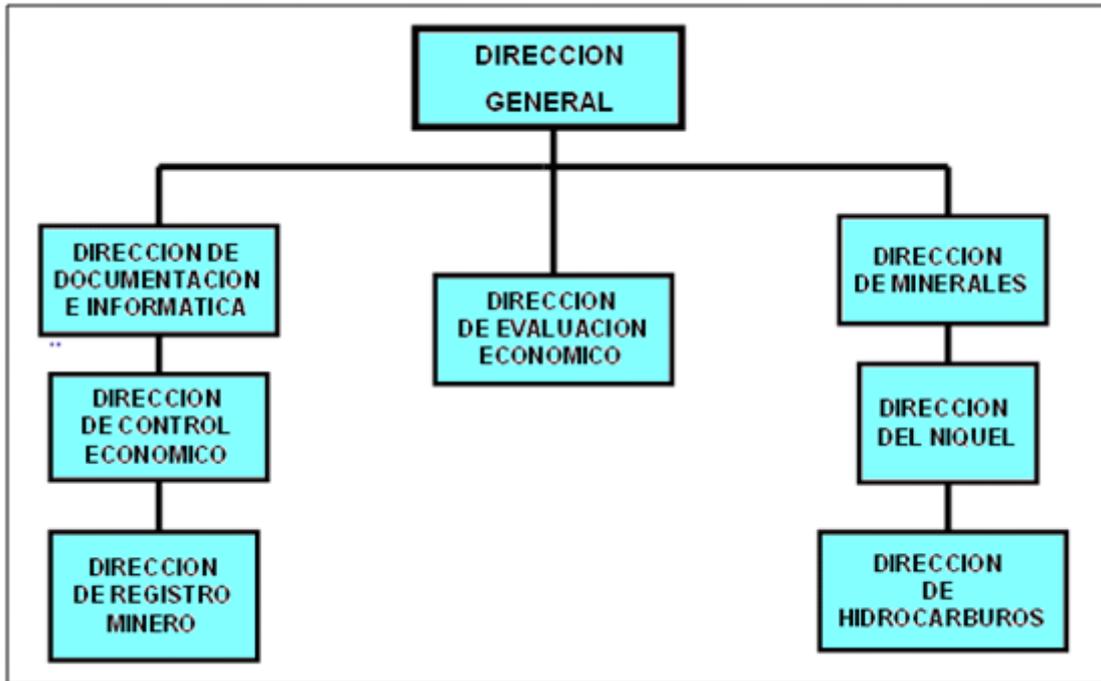


Figura 2: Estructura de la ONRM.

1.5.3 Situación Problemática

Actualmente la Oficina Nacional de Recursos Minerales (ONRM) como se explicó en la introducción del trabajo se encuentra sumergida en un importante proceso de informatización. Con el fin de adentrarse en la automatización de sus diferentes actividades como: el control de concesionario, el archivo técnico, el balance de recursos, las reservas del país, los meta datos y la información de las investigaciones geológicas. Informatizando con esto el flujo de informaciones que se realiza en las mismas para lograr una mayor eficiencia y rapidez, ya que actualmente poseen un gran número de información en copia dura (papel), que con el paso del tiempo gran parte de esta información se ha perdido o deteriorado, provocando que el flujo de información en la oficina sea lento e ineficiente. Además el cumplimiento de cada una de las actividades mencionadas anteriormente, al realizarse de forma manual, se torna complejo y agotador para las personas responsables de llevar a cabo dichas tareas.

Debido a la situación que tiene la ONRM, surgen un gran número de aplicaciones, las cuales usan alguna información común llamada nomencladores que no es más que descripciones de la información que utiliza cada aplicación. Estos nomencladores se usan por separado y en algunas ocasiones están repetidos en varias aplicaciones o incluso el mismo descriptor de información con nombres diferentes, provocando que se dificulte el trabajo al ser utilizados, y que

exista una descentralización de ellos. Además cada una de estas aplicaciones no tiene forma de interactuar entre sí.

Según lo descrito anteriormente la ONRM conjuntamente con la UCI ha decidido crear y brindar un servicio que permita la interacción entre las aplicaciones y que de manera eficiente estandarice y centralice los nomencladores para el uso de las aplicaciones que se desarrollarán en el grupo de proyectos PNICG. Con el fin de tener mejor control de ellos porque así toda la información estaría almacenada, segura y disponible para aquellos usuarios que tengan acceso a la misma.

1.5.4 Fundamentación del objetivo

Con la implementación de un Servicio Web para el uso y disponibilidad de los nomencladores asociados al campo de la geología y definidos por la ONRM se facilitará la seguridad y confidencialidad en la base de datos que guarda toda la información de los descriptores. Además se gestionará de forma eficiente los pedidos y solicitudes que efectúen las aplicaciones, proporcionándoles con la mayor rapidez las respuestas de dicha solicitud. Los logros que se esperan con el desarrollo de esta aplicación son:

- Primero: Garantizar la disponibilidad de los nomencladores para que puedan ser de uso colectivo para el resto de las aplicaciones.
- Segundo: Resolver el problema del uso de los nomencladores por parte de las aplicaciones implantadas para la informatización de la ONRM que no es más que la descentralización de dichos nomencladores.

1.5.5 Objeto de automatización

Es preciso para la ONRM el perfeccionamiento de cada una de las actividades que se llevan a cabo en la oficina, pues se hace imprescindible la informatización del flujo de informaciones para mejorar y tener un control de la gestión de cada una de las actividades. Una vez informatizadas, se hace necesario la creación de algún sistema para tener centralizada toda la información común que será utilizada por las aplicaciones que se desarrollen y así lograr una buena organización y centralización de la información.

Para la creación del sistema hay que tener en cuenta que las aplicaciones desarrolladas estarán ejecutando alguna de sus funcionalidades y en algún momento determinado le será necesario usar o consultar alguna información referente a los nomencladores. Por lo que se tiene

que brindar de forma eficiente la respuesta a la solicitud o pedido realizado por cualquier aplicación y es de vital importancia tener la seguridad de la veracidad de los datos brindados.

1.5.6 Objetivo general propuesto

El objetivo general que se persigue con este trabajo investigativo es desarrollar un Servicio Web que permita centralizar, estandarizar y ofrecer información detallada de los nomencladores a las diferentes aplicaciones del proyecto PNICG, además de permitir la interacción entre ellas.

1.5.7 Propuesta de solución

Después de realizar un análisis y determinar la situación actual sobre el objeto de estudio que tiene este trabajo, se aprecia la necesidad de implementar un Servicio Web que permita centralizar y estandarizar información detallada de los nomencladores e interrelacionar todas las aplicaciones desarrolladas en el proyecto PNICG; porque continuamente las aplicaciones tendrán la necesidad de conocer todos los datos referente a los nomencladores y estos deben estar disponibles en todo momento. Además que los usuarios de las aplicaciones deben de tener la seguridad de que la información que estará disponible tiene un alto grado de veracidad e integridad.

1.6 Análisis de otras soluciones existentes

En Cuba se han desarrollado un gran número de servicios web, en las principales instituciones que podemos encontrar muchos de estos servicios son las universidades; específicamente aquellas que incluyen en su plan de estudio alguna carrera vinculada a la esfera de la informática, a parte de las empresas que se dedican de alguna manera al desarrollo de software. Unas de las universidades pilares en el desarrollo de estos servicios son la CUJAE y la Universidad de las Ciencias Informáticas (UCI).

En la UCI se brindan servicios como el Servicio web del sistema Akademos, destinado a la gestión académica de los estudiantes. El Servicio web de Trabajadores que brinda información sobre trabajadores que son plantilla de la Universidad y Tercerizados. Además del sistema Identificación, sistema orientado a la búsqueda de personas utilizando mecanismos de identificación como el solapín, el identificador de la persona. También se desarrolla la guía telefónica que mantiene información sobre los teléfonos asignados a apartamentos, docentes y personas.

Son muchos los servicios que se pueden encontrar actualmente funcionando, pero hasta el momento no se ha desarrollado en Cuba ningún servicio web para brindar información de los nomencladores en la ONRM, que actualmente son manejados mediante el Microsoft Office Excel herramienta del paquete de Office.

1.7 Conclusiones parciales

En este capítulo se detallaron las condiciones y problemas actuales que rodean al objeto de estudio enmarcado en este trabajo, para obtener una mejor comprensión y visión del problema permitiendo formar la base para una solución eficiente. Se abordaron conceptos generales propios del tema central de la investigación, importantes para el futuro desarrollo del proyecto y significativos para el proceso de comprensión de las características que tiene el sistema.

2

CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS A UTILIZAR

Introducción

En la actualidad, con el rápido incremento de las tecnologías informáticas casi todas las empresas están conectadas a la WWW³, sea para la prestación de servicios o para recibirlos de ella. Los sistemas operativos no son lo suficientemente confiables, pues los usuarios exponen información sensible a grandes riesgos. Los Servicios Web cambian a diario la forma de ver la información por todos los problemas de seguridad existentes en la red. En este capítulo se relacionan aspectos sobre las herramientas que se han escogido para la implementación y documentación del sistema a desarrollar, teniendo en cuenta el auge desenfrenado que han causado en el mundo actual. Apoyándose en comparaciones en cuanto a ventajas y desventajas de las mismas, se podrá determinar las propuestas para el funcionamiento del Servicio Web.

³ World Wide Web

2.1 Herramientas, metodologías y lenguaje para el modelado de la Capa de Servicios Web

Para la creación de cualquier software o para el desarrollo de proyectos se hace imprescindible tener una visión de lo que desea construir, por lo que es de vital importancia contar con las herramientas y lenguajes necesarios para hacer un modelado o trazar una arquitectura a desarrollar.

A continuación se realiza un breve estudio de el lenguaje, la Metodología de Desarrollo de Software y de la herramienta CASE necesaria para crear un modelo de la Capa de Servicios Web que se implementará.

2.1.1 El Lenguaje Unificado de Modelado (UML) como soporte de lenguaje orientado a objetos para el modelado de aplicaciones Web.

En todas las disciplinas de la ingeniería se hace evidente la importancia de los modelos ya que describen el aspecto y la conducta de “algo”. Ese “algo”, puede existir, estar en un estado de desarrollo o estar, todavía, en un estado de planeación; de ahí la vital importancia del uso del Lenguaje Unificado de Modelado (UML) en el desarrollo de software ya que permite detallar los artefactos en el sistema, documentar y construir, en fin, es el lenguaje en el que está descrito el modelo.

UML es, probablemente, una de las innovaciones conceptuales en el mundo tecnológico del desarrollo de software que más expectativa ha generado a lo largo de muchos años, comparable con la aparición e implantación de los lenguajes COBOL⁴, Basic⁵, Pascal⁶, C++⁷. UML es un estándar de la industria del software, utilizado para la construcción de modelos como condición previa para el diseño de cualquier empresa y posteriormente para la de prototipos. [8]

⁴ Common Business Oriented Language

⁵ Beginners All-purpose Symbolic Instruction Code (código de instrucciones simbólicas de propósito general para principiantes)

⁶ lenguaje de programación desarrollado por el profesor suizo Niklaus Wirth

⁷ Lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup

UML ha nacido como un lenguaje, pero es mucho más que un lenguaje de programación. En realidad se ha diseñado y construido con una madurez sólida, en comparación con los últimos desarrollos de HTML, C#, Java, Ajax, XML, los lenguajes por excelencia del mundo de la Internet. [9]

UML ayuda a los usuarios a entender la realidad desde un punto de vista de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades de dinero en proyectos que no estén seguros en su desarrollo, reduciendo el costo y el tiempo empleado en la construcción de los módulos que construirán el software. [10]

Tiene propiedades que han sido realmente las que han contribuido a hacer de UML el estándar de la industria en el análisis y diseño de sistemas de cómputo en la actualidad.

Algunas de las propiedades de UML como lenguaje de modelado son:

- Es un lenguaje distribuido y adecuado a las necesidades actuales y futuras, ampliamente utilizado por la industria del software.
- Reemplaza a decenas de notaciones empleadas por otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soporta tienen su fundamento en la tecnología orientada a objeto, tales como objetos, clases, componentes y nodos.
- Comportamiento del sistema: casos de usos, diagramas de secuencia, de colaboración, que sirve para evaluar el estado de las máquinas.

Entre más complejo es el sistema que se desea crear más beneficios presenta el uso de UML.[10]

- Diseño y documentación.
- Código reutilizable.
- Descubrimiento de fallas.
- Ahorro de tiempo en el desarrollo del software.
- Mucho más fáciles las modificaciones.
- Más fácil comunicación entre programadores.

Se utilizará como notación el Lenguaje Unificado de Modelado para lograr un mayor entendimiento ya que se puede modelar y describir secuencialmente por pasos todos los procesos que se lleva a cabo según la problemática planteada. Sirve porque es un lenguaje gráfico para

visualizar, especificar, construir y documentar los artefactos de un sistema de software y nos facilita un vocabulario controlado, con reglas y símbolos, para que todos los agentes de un proyecto eviten ambigüedades y dispersión conceptual.

2.1.2 Metodología de Desarrollo de Software

Uno de los principales problemas en la actualidad en el desarrollo de software es seleccionar la metodología más adecuada que posibilite obtener los resultados óptimos que se desean; o sea, cómo trabajar eficientemente evitando desarrollar software de baja calidad con los cuales los clientes y los desarrolladores queden insatisfechos. Una metodología tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Se analizará seguidamente tres de las más conocidas y sus características.

2.1.2.1 Programación Extrema (XP)

Es una metodología ágil, intenta reducir la complejidad del software por medio de un trabajo orientado al objeto, basado en las relaciones interpersonales y la velocidad de reacción. Intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. Este representante debe estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones.[11]

Estas historias las escribe el cliente y describe escenarios sobre el funcionamiento del software, que no solo se limitan a las Interfaces Gráficas de Usuarios (GUI, siglas en inglés) sino también puede describir el modelo y el dominio. A partir de los UserStories⁸ y de la arquitectura perseguida se crea un plan de liberaciones (releases) entre el equipo de desarrollo y el cliente. Para cada liberación se discutirán los objetivos de la misma con el representante del cliente y se definirán las iteraciones (de pocas semanas de duración) necesarias para cumplir con los objetivos de la release. El resultado de cada iteración es un programa que se transmite al cliente para que lo juzgue. [11]

En base a su opinión se definen las siguientes iteraciones del proyecto y si el cliente no está satisfecho se adaptará el plan de liberaciones e iteraciones hasta que el cliente de su aprobación y el software este a su agrado. UserStories y casos de pruebas son la base sobre la que se

⁸ XP define *UserStories* como base del software a desarrollar

asienta el trabajo del desarrollador. Se sigue un diseño evolutivo con la siguiente premisa: conseguir la funcionalidad deseada de la forma más sencilla posible. Este diseño evolutivo hace que no se le dé apenas importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento. [11]

2.1.2.2 Desarrollo Guiado por Funcionalidad (FDD)

Se puede considerar a medio camino entre RUP y XP, aunque al seguir siendo un proceso ligero, se asemeja más a este último. Está pensado para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se basa en un proceso iterativo con iteraciones cortas (2 semanas). Las iteraciones se deciden en base a funcionalidades (features⁹), que son pequeñas partes del software con significado para el cliente. [11]

Un proyecto con metodologías FDD consta de las siguientes fases:

- 1) Desarrollo de un modelo general.
- 2) Construcción de una lista de funcionalidades.
- 3) Plan de liberaciones en base a las funcionalidades a implementar.
- 4) Diseñar en la base a las funcionalidades.
- 5) Implementar en base a las funcionalidades.

Las primeras tres fases ocupan gran parte del tiempo en las primeras iteraciones, siendo las dos últimas las que absorben la mayor parte del tiempo según va avanzando el proyecto, limitándose las primeras a un proceso de refinamiento. Las funcionalidades a implementar en un release¹⁰ se dividen entre los distintos subgrupos del equipo, y se procede a implementarlas. [11]

2.1.2.3 Proceso Unificado de Desarrollo de Software (RUP)

Actualmente el proceso unificado lleva a la construcción de sistemas de software mucho más complejos y grandes. Debido al auge de las computadoras y a la vez el aumento del rigor de los usuarios, esto ha traído un rápido incremento en Internet para el intercambio de todo tipo de información. James Rumbaugh, Grady Booch e Ivar Jacobson, autores de “El proceso unificado de desarrollo de software”, opinan que “El problema del software se reduce a la dificultad que

⁹ Pequeñas partes del software con significado para el cliente

¹⁰ Término bajo el cual se reconoce una aproximación

afrontan los desarrolladores para coordinar las múltiples cadenas de trabajo de un gran proyecto de software” [10, 12]

El proceso unificado de desarrollo, RUP, es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. Permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual. Cuenta con las mejoras prácticas del modelo de desarrollo de un software en particular.[10, 12]

- Desarrollo de software de forma iterativa.
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software de forma visual, usando UML.
- Verifica la calidad del software.
- Controla los cambios.
- Dirige las tareas de cada desarrollador por separado y del equipo como un todo.
- Especifica los artefactos que deben desarrollarse en cada fase de desarrollo del software.

RUP consta de cuatro etapas o fases por donde el software tiene que transitar para que obtenga la calidad requerida. [13]

- 1) Fase de Inicio
- 2) Fase de Elaboración
- 3) Fase de Construcción
- 4) Fase de Transición

Además se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Flujos de trabajo:

- Modelamiento del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

- Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- Prueba (Testeo): Busca los defectos a lo largo del ciclo de vida.
- Instalación: Produce una liberación (release) del producto y realiza actividades (empaquete, instalación y asistencia a usuarios) para entregar el software a los usuarios finales.
- Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

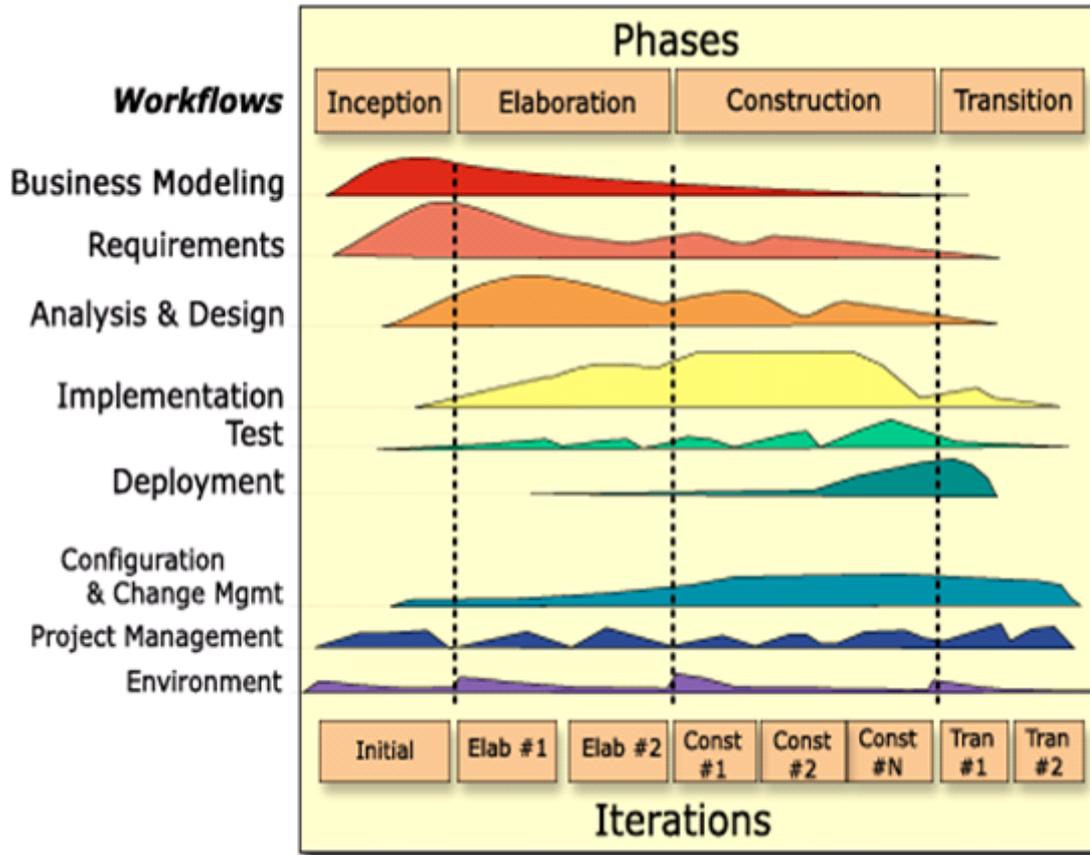


Figura 3: RUP en Dos Dimensiones

En el caso específico de las aplicaciones Web, las actividades establecidas por dicho proceso son suficientes para garantizar cubrir todos los aspectos de los entornos de este tipo de aplicación.[10]

El proceso unificado de desarrollo de software, se caracteriza principalmente por ser un proceso:

- Dirigido por Casos de Uso
- Centrado en la Arquitectura
- Iterativo e Incremental

Esta metodología utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema de software. Es una parte esencial del Proceso Unificado de desarrollo de software y fueron desarrollados paralelamente por las mismas personas, haciendo que su integración sea un éxito.

2.1.2.4 Selección de la Metodología a utilizar

Después de hacer un estudio de estas tres metodologías, se ha llegado a la conclusión que la metodología a utilizar para el desarrollo del Servicio Web es RUP, por ser la más completa y abarcadora. También porque permite la mitigación temprana de posibles riesgos, progreso visible en las primeras etapas, gestión de la complejidad y el conocimiento adquirido en una iteración puede aplicarse de iteración a iteración.

Además, como señalan algunos autores, las otras metodologías son casos particulares de RUP. La metodología XP y FDD presentan algunas debilidades, lo que representa riesgos considerables, a la hora de una buena obtención de requisitos para el sistema.

2.1.3 Herramienta CASE de Desarrollo de Software

Las **Herramientas case** es la mejor base para el proceso de análisis y desarrollo de software, se ha hecho habitual que cada vez que se desarrolle un software por pequeño o grande que se sea, se utilizan las herramientas CASE¹¹, con el fin de automatizar los aspectos clave de todo el proceso de desarrollo de un sistema, desde el principio hasta el final incrementando su calidad.

2.1.3.1 Rational Rose

Es la herramienta CASE que comercializan los desarrolladores de UML (Booch, Rumbaugh y Jacobson) y que soporta de forma completa la especificación del UML. Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática, otra dinámica de los modelos del sistema, una lógica y otra física; que permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. [9]

A continuación los tipos de modelos:

1. Desarrollo Iterativo.
2. Generador de Código.
3. Ingeniería Inversa.
4. Trabajo en Grupo.

¹¹ Ingeniería de Software Asistida por Computadora (**Computer Aided Software Engineering**)

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases.[14]

2.1.3.2 Visual Paradigm

Es una herramienta CASE que facilita a los ingenieros de software diseñar, integrar y modelar visualmente los distintos diagramas que se generan a lo largo del desarrollo del software. Presenta un generador de código que soporta más de 10 lenguajes y proporciona la ingeniería inversa: [15]

1. Java.
2. C++.
3. CORBA.
4. IDL.
5. PHP.
6. Esquema de XML.

Esta herramienta está especializada en la ingeniería del software de bases de datos. Permite gestionar proyectos muy complejos con gran sencillez. Incluye herramientas muy interesantes para ingeniería inversa de bases de datos. Básicamente se trata de una herramienta para trabajar con UML.

2.1.3.3 Selección de la herramienta CASE a utilizar

Después de hacer un estudio de estas dos herramientas, se llegó a la conclusión de que la herramienta a utilizar para un buen desarrollo ingenieril de esta investigación es el Visual Paradigm. Se realiza esta selección porque es una herramienta que entre sus grandes ventajas unifica el formato de todos los diagramas. Ofrece navegación intuitiva entre el modelo visual y el código, además de permitir la sincronización entre el código fuente y el modelo en tiempo real o bajo demanda. Brinda soporte para toda la notación UML, sofisticados y automáticos diagramas de capas y el análisis de textos. No se puede dejar de mencionar que es una poderosa herramienta de generación de PDF/HTML a partir de diagramas UML.

2.2 Lenguajes y Herramientas a utilizar para la creación y funcionamiento de la Capa de Servicios Web.

Para la creación y posterior funcionamiento de una Capa de Servicios Web es necesario definir una serie de elementos como son lenguajes y herramientas a utilizar para su correcto funcionamiento. Primeramente se debe tener en cuenta el lenguaje de programación sobre el cual se va a implementar los Servicios Web.

Posteriormente se debe seleccionar el Servidor de Aplicaciones Web sobre el cual estará la aplicación y el Gestor de Bases de Datos a utilizar para manipular la base de datos de nuestra aplicación. Para la selección de todos estos elementos se tiene en cuenta aspectos como funcionalidades y plataforma sobre la que funciona. A continuación se hace un estudio de estos aspectos antes mencionados, para definir cual es el más adecuado para la creación y funcionamiento de la Capa de Servicios Web.

2.2.1 Lenguajes de programación para la Web

Existen diferentes lenguajes de programación para las aplicaciones Web, cada uno posee sus características específicas. Entre los más sobresalientes se encuentran PERL, ASP, PHP, Java, JSP, los módulos CGI, entre otros. Del lado del cliente se encuentran principalmente el Java Script y el Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores. A continuación se presenta una serie de características de los diferentes lenguajes de programación en la Web de más auge en la actualidad.

2.2.1.1 Personal Home Page (PHP)

Es un lenguaje de programación usado frecuentemente para la creación de contenidos para sitios Web con los cuales se puede programar las páginas HTML y los códigos de fuente. Es un lenguaje de programación del lado del servidor, gratuito e independiente de plataforma, rápido, con una gran librería de funciones, mucha documentación y está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente se encuentra en su versión 5.2.5 (9 de noviembre de 2007).

Con este lenguaje se pueden realizar múltiples tipos de aplicaciones Web debido a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos. Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes como MySQL, Oracle, Informix, PostgreSQL.

2.2.1.2 ASP.NET

Es la última versión de la tecnología ASP pero no es una versión aumentada de esta, sino que es enteramente una nueva tecnología para el lado del servidor. ASP.NET es la parte principal del Framework .NET de Microsoft. También contiene un nuevo sistema de entradas de controles orientado a objetos, como las cajas de lista programables y los controles de la validación.

Incluye algunas cosas nuevas como:

- Soporta los mejores lenguajes
- Controles programables.
- Componentes basados en XML
- Autenticación de usuarios con cuentas y roles.
- Alta escalabilidad
- Código compilado
- Fácil configuración y despliegue.
- Utiliza ADO.Net, soporta C#, C++ y Visual Basic (VB) pero no Visual Basic Script (VBS).

2.2.1.3 Processing Extraction Report Language (PERL)

Es un lenguaje de programación interpretado, muy utilizado para construir aplicaciones CGI para la Web, se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

Es un lenguaje de libre uso, es decir gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Permite efectuar búsquedas de secuencias de caracteres. Perl se utiliza con fuerza en la escritura de procesos CGI (Common Gateway Interface) instalados en un servidor Web, o para el desarrollo de procesos de mantenimiento de las actividades de un servidor.

No es muy recomendado para desarrollar procesos de puro cálculo científico o programas que necesitan una gran velocidad y precisión de cálculo o elaboraciones numéricas complejas, pero se convierte en una necesidad para quien tenga que manejar un sitio Web que no esté compuesto sólo de texto e imagen.

2.2.1.4 Fundamentación del lenguaje a utilizar

Luego de haber realizado un estudio detallado y profundo de los diferentes lenguajes de programación y las comparaciones realizadas en cuanto a las características principales de cada uno, se decide que el lenguaje a utilizar es Personal Home Page (PHP). Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierte en la herramienta ideal para la creación de páginas Web dinámicas: [16]

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que se realiza con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Además de todas sus características, el uso de este lenguaje proporciona unas series de ventajas que aumenta en gran medida que sea la herramienta ideal para la creación de Aplicaciones Web.

Ventajas que proporciona la utilización del lenguaje

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, se destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos llamados ext's o extensiones.
- Es libre, se presenta como una alternativa de fácil acceso para todos.

- Posee una amplia documentación de ayuda para un mayor entendimiento del lenguaje.
- Soporte sólido para Programación Orientada a Objetos (OOP) con PHP Data Objects.
- Mejoras de rendimiento.

2.2.2 Servidores Web

Un servidor web es un programa que se ejecuta en el servidor e implementa el protocolo HTTP, éste escucha las peticiones del cliente y dependiendo de ellas buscará una página web o ejecutará un programa en el servidor. El servidor responde al cliente enviándole el código HTML de la página, cuando éste lo recibe, lo interpreta y lo muestra en pantalla. Existen un gran número de servidores web que se utilizan y a continuación se nombran algunos ellos.

2.2.2.1 Apache

El servidor HTTP Apache es un software (libre), de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido

Es un servidor que tiene una amplia aceptación en la red: en 2005, Apache fue el servidor HTTP más usado, siendo el servidor empleado en el 48% de los sitios web en el mundo y en los últimos años se ha convertido en el más popular con un 70% de la cuota de mercado. Puede instalarse en Linux, Novell, Unix y Windows.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

2.2.2.2 AOLserver

AOLserver es el servidor web de código abierto creado en 1994 por Navisoft (Naviserver) y actualmente empleado, lógicamente, en todos los sitios web de América Online y en otros muchos de internet. AOLserver tiene procesamiento multihilo, tiene soporte para Tcl, y se usa para sitios web dinámicos de gran tamaño.[17]

Una de las ventajas que posee AOLServer es que incluye un interpretador **Tcl** embebido en su estructura, debido a esto es posible mantener "Scripts" en **Tcl** que se ejecutaran internamente. A diferencia de Apache, en donde el ejecutarse un "script" (CGI "Common Gateway Interface") existe la necesidad de iniciar un proceso nuevo. Cabe mencionar que AOLServer también tiene soporte para CGI, pero debido a la ejecución superior de "Scripts" (internos) en **Tcl** el uso de CGI (con Perl) es de muy poco uso.[17]

AOLServer mantiene grupos de conexiones latentes ("Pools") hacia bases de Datos, estas conexiones ("Pools") permiten a AOLServer realizar búsquedas más rápidas (no hay necesidad de continuamente abrir y cerrar la conexión con la base de Datos), actualmente existen drivers para las siguientes Bases de Datos: SOLID, PostgreSQL, Oracle, Sybase, Informix, Interbase y MySQL.[17]

2.2.2.3 Roxen webserver

Roxen webserver: Servidor Web de software libre producido por Roxen Internet Software que fue especialmente diseñado para la gestión de páginas activas. Su diseño modular y la facilidad para extenderlo con módulos en una gran variedad de lenguajes lo hacen atractivo para servicios web activos.[18]

Es un servidor multiplataforma, seguro, cuenta con un potente sistema de codificación para reforzar la seguridad. Es gratuito, que resulta muy fácil tanto de instalar como de utilizar. Funciona en cualquier sistema operativo, y entre sus características principales decir que integra una base de datos MySQL.[18]

2.2.3 Sistema Gestor de Base de Datos (SGBD)

Ante la notable demanda de soluciones informáticas para la progresiva informatización de todas las organizaciones con la necesidad de optimizar servicios y productos, han surgido diferentes gestores de bases de datos, siendo estos programas que permiten manejar un lenguaje de manipulación de datos, un lenguaje de consulta y el manejo de bases de datos. Se conocen con el término SGBD¹² y DBMS¹³ siendo ambos equivalentes y acrónimos respectivamente. [19]

¹² Sistema Gestor de Base de Datos

¹³ Data Base Management System

SGBD ofrecen un control centralizado de la información teniendo como objetivos evitar la redundancia de los datos, mejorar los mecanismos de seguridad de los mismos y la privacidad, mantener la integridad de los datos realizando las validaciones necesarias y mejorar la eficacia de acceso a los datos.

Ventajas:

- Facilidad de manejo de grandes volúmenes de información.
- Gran velocidad en muy poco tiempo.
- Independencia del tratamiento de información.
- Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
- No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
- Integridad referencial al terminar los registros.

Desventajas:

- El costo de actualización del hardware y software son muy elevados.
- Salario del administrador de la base de datos es costoso.
- El mal diseño de esta puede originar problemas futuros.
- Un mal adiestramiento a los usuarios puede originar problemas futuros.
- Si no se encuentra un manual del sistema no se podrán hacer relaciones con facilidad.
- Generan campos vacíos en exceso.
- El mal diseño de seguridad genera problemas.

Es necesario conocer las características, ventajas y desventajas de cada sistema gestor de base de datos, es por ello que se debe analizar los productos que más se destacan y de acuerdo a la necesidad, inclinarnos por la mejor opción. Algunas de estas alternativas son Oracle, PostgreSQL, MySQL, que comercialmente son más fuertes en el mundo del software libre.

2.2.3.1 Oracle

Es básicamente una herramienta cliente/servidor para la gestión de bases de datos. Posee escalabilidad para soportar un gran número de usuarios y cargas de trabajo de alto volumen de transacciones, el optimizador de consultas determina dinámicamente los caminos de acceso y métodos de enlace más eficientes para cada consulta. Además soporta efectivamente ambientes

mixtos de carga de trabajo caracterizados por actividades simultáneas de consulta y actualización.[16]

Posee lectura de multiversión sin bloqueo, siempre proporciona a los usuarios resultados consistentes, al mismo tiempo que no penaliza al rendimiento en la actividad concurrente de actualización. También garantiza la autenticidad apropiada de los usuarios y la privacidad e integridad de los datos, y permiten manejar la asignación de privilegios y monitorear las operaciones de la base de datos a lo largo de toda la empresa.[16]

2.2.3.2 PostgreeSQL

Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado Ingres en la Universidad Berkeley de California. Ingres fue más tarde desarrollado comercialmente por la Relational Technologies/Ingres Corporation.

PostgreeSQL es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2. Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreeSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.[20]

PostgreeSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta. Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido.

Además provee soporte para: números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones IP, claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (foreign keys), vistas, herencia de tablas, tipos de datos y operaciones geométricas.

2.2.3.3 MySQL

Es un sistema de gestión de base de datos, multihilo y multiusuario, posee código fuente abierto y es muy usado en el mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. Posee recuperación automática ante fallas. MySQL es muy utilizado en aplicaciones Web como MediaWiki o Drupal. MySQL es una base de datos muy rápida en la

lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. [21]

Funciona sobre múltiples plataformas, además las opciones de índices de texto completo han sido movidas al archivo de configuración de MySQL, así que solamente se tienen que hacer las adecuaciones necesarias y reiniciar MySQL para que los cambios tengan efecto. Muchos fallos en las búsquedas de texto completo han sido corregidos también. En cambio, las tablas usan bloqueo a nivel de filas para mejorar de manera impresionante el rendimiento.[21]

Teniendo en cuenta que MySQL es usado en un gran número de sistemas Web como PostNuke y Slashcode, los desarrolladores de MySQL implementaron una caché de consultas para acelerar las que son ejecutadas comúnmente. Esta caché simplemente almacena las consultas SELECT ejecutadas recientemente y sus resultados en memoria. Integridad referencial. Utilizando MySQL se pueden definir llaves foráneas entre tablas relacionadas para asegurarse de que un registro no puede ser eliminado de una tabla si aún está siendo referenciado por otra tabla.[21]

2.2.3.4 Fundamentación del Sistema Gestor de Base de Datos a utilizar (PostgreSQL)

El gestor de base de datos que se utilizará es PostgreSQL, ya que es considerado como una de las alternativas de sistema de bases de datos de código abierto. PostgreSQL posee una estabilidad y confiabilidad legendaria, nunca ha presentado caídas en varios años de operación de alta actividad; es extensible ya que el código fuente está disponible para todos sin costo. Es un gestor que está diseñado para ambientes de alto volumen ya que usa una estrategia de almacenamiento de filas llamada MVCC¹⁴ para conseguir una mejor respuesta en ambientes de grandes volúmenes. Además es Multiplataforma, está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.[22]

¹⁴ Acceso concurrente multiversión (*Multiversion Concurrency Control*)

2.3 Fundamentación del IDE de programación a utilizar

2.3.1 Eclipse

Eclipse es una plataforma de software de código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE).

Eclipse fue desarrollado originalmente por IBM como el sucesor de VisualAge. Ahora es desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

La versión actual de Eclipse dispone de las siguientes características:

- Editor de texto.
- Resaltado de sintaxis.
- Compilación en tiempo real.
- Pruebas unitarias con JUnit.
- Control de versiones con CVS.
- Integración con Ant.
- Asistentes (*wizards*): para creación de proyectos, clases, tests, etc.
- Refactorización.

2.4 Fundamentación de otras herramientas a utilizar para el desarrollo del trabajo.

2.4.1 Gestor de referencia: EndNote 9

El EndNote 9 permite crear y organizar bibliografías o referencias utilizadas durante un proceso de investigación, esta herramienta de apoyo permite:

- Realizar la referencia bibliográfica adecuadamente.
- Insertar citas, referencias después de tener agrupados todos los datos de la bibliografía consultada.

- Permite exportar desde Internet o bases de datos consultadas hacia el propio gestor de referencias.

EndNote 9 amplía la posibilidad de recoger datos desde gran cantidad de fuentes online, es también muy rápido, añade compatibilidad con las nuevas versiones de Microsoft Word, contiene nuevos registros sintácticos y codificaciones de texto.

2.4.2 Planner (Gestor de Proyectos)

Es normal, que las personas hagan proyectos en sus vidas, desde el hacer la tarea de la escuela, hasta revisar en que etapa va la construcción de un edificio, todo necesita estar organizado, más que nada por la comodidad de tener todo bien estructurado, y saber exactamente donde estamos situados en un proyecto, que nos falta y que hemos hecho hasta el momento.

Software para gestionar hay muchos y para diversas plataformas, entre los mas conocidos esta, Microsoft Project para Windows y aunque no es muy conocido también existe el Planner para GNU/Linux-Gnome, que se utilizará en la planificación de este proyecto.

Planner es una herramienta para planear, programar y seguir proyectos para el escritorio GNOME. Además es un proyecto de código abierto destinado a ser una alternativa mejor que las herramientas propietarias disponibles. Entre sus características podemos encontrar:

- Gestión de Calendarios.
- Gestión de Recursos.
- Seguimiento del avance del proyecto.
- Enlazar tareas.
- Exportación a diferentes formatos.

También utiliza un formato de archivo basado en XML o una base de datos PostgreSQL para almacenar los detalles del proyecto y cuenta con diagramas de Gantt y gestión de tareas y recursos.

2.5 Conclusiones Parciales

Luego de haber realizado un estudio de las tecnologías y de haber plasmado las características fundamentales de cada una de ellas y teniendo en cuenta las estructuras de la Universidad de las Ciencias Informática en cuanto las tecnologías que se utilizan, se puede afirmar que la selección realizada es la más óptima:

- El lenguaje de programación PHP es viable para la implementación del Servicio Web
- PostgreSQL constituye una herramienta insuperable como Gestor de Bases de Datos para el sistema a realizar.
- La metodología RUP representa la guía más óptima para el desarrollo de la aplicación propuesta.

3

CAPÍTULO 3 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Introducción

UML, como se ha comentado, ha generado y sigue generando un enorme entusiasmo comparable al nacimiento del desarrollo orientado a objetos a principio de los 90 y posteriormente el desarrollo de componentes en la segunda mitad de la década. Este entusiasmo se ha hecho una gran realidad y UML se ha convertido ya en uno de los mejores lenguajes para el diseño y desarrollo de software fiable, eficiente y de calidad.

En este capítulo se describe el sistema propuesto utilizando el Lenguaje Unificado de Modelado (UML) para lograr la modelación (generación) de los distintos artefactos que involucran el desarrollo del trabajo, haciendo uso de la herramienta Case Visual Paradigm.

También se enumeran los requisitos tanto Funcionales como No funcionales que debe cumplir el sistema que se propone, permitiendo así hacer una mejor concepción de forma general del sistema.

3.1 Entorno donde trabajará el sistema

El Servicio Web funcionará en la Oficina Nacional de Recursos Minerales, pues será la oficina la encargada de trabajar con los nomencladores. Además el conjunto de aplicaciones que usarán el servicio web estarán funcionando en esta área.

La ONRM necesita este Servicio Web debido a que los nomencladores son usados por separado y en algunas ocasiones están repetidos en varias aplicaciones o incluso el mismo descriptor de información con nombres diferentes, provocando que se dificulte el trabajo y que exista descentralización entre los mismos.

3.1.1 Conceptos principales del entorno

3.1.1.1 Balance

Este módulo es el encargado de actualizar la información referente a la explotación y estudio de las diferentes fuentes de minerales sólidos, petróleo y aguas minerales. Además del almacenamiento y mantenimiento de la misma, para consultas y registros de informaciones.

3.1.1.2 Metadatos

Este módulo es el encargado de la creación y manipulación de los Metadatos Generados por los Datos Geológicos que maneja la ONRM.

3.1.1.3 Nomencladores

Este módulo tiene como propósito la gestión integral de Nomencladores utilizados en la ONRM, atendiendo a las regulaciones existentes y facilitar su relación con el resto de la información geológica.

3.1.1.4 Base de Datos Referativa

Este módulo es el encargado de la gestión y consulta de la información en el archivo técnico de la ONRM y de los pozos de petróleo de la República de Cuba.

3.1.1.5 Concesionario

Este módulo es el encargado de otorgar cualquiera de los cinco permisos establecidos a los usuarios que hagan la solicitud, una vez que el usuario se convierte en Titular o Concesionario.

Después de obtenido los permisos, los usuarios pueden solicitar una prórroga para continuar con la explotación o estudio de la zona.

3.1.1.6 Geodatos

Este módulo es el encargado de la creación de Bases de Datos basadas en la Tecnología de Base de Datos Relacional con Atributos y Registros Variables (BDARV). Geodatos es realizado a solicitud del Instituto de Geología y Paleontología (IGP) de la Unión Geominera del Ministerio de la Industria Básica (MINBAS). Este módulo se divide en dos submódulos.

Geodato IC

Sistema diseñado para Pocket PC, su objetivo es el de almacenar información que un geólogo recolecta cuando realiza una investigación geológica en el campo.

Geodato RC

Sistema Web, que se encargará de generar reportes y búsquedas de la información sobre las investigaciones geológicas.

3.1.2 Eventos principales del entorno

Búsqueda de información: Es el proceso mediante el cual las aplicaciones de cada uno de los módulos como resultado de sus necesidades, se le hace imprescindible revisar u obtener la información de los nomencladores para desempeñar su trabajo.

3.1.3 Diagrama de clases del Modelo de Dominio

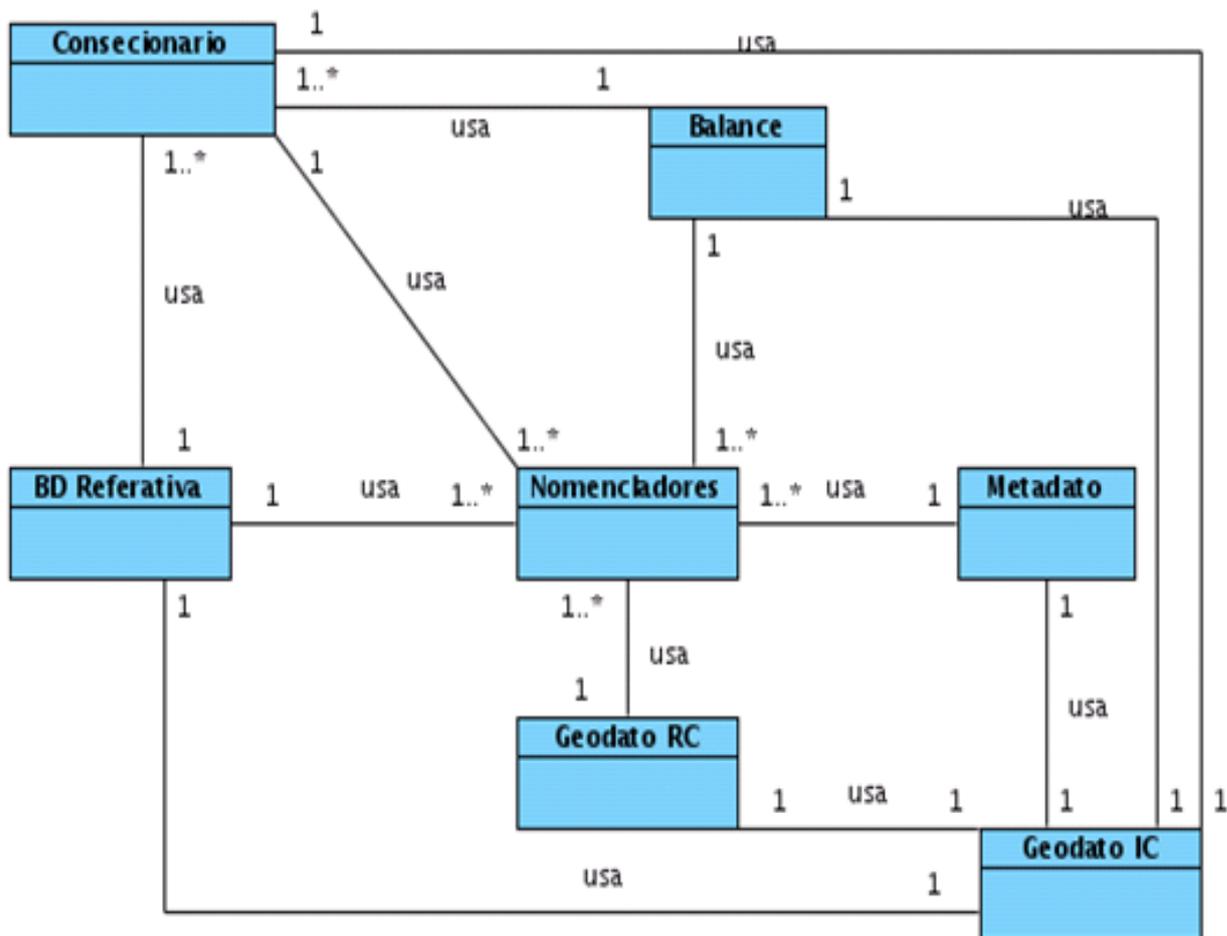


Figura 4: Modelo de Dominio

3.2 Requisitos

El paso desde la determinación de las necesidades del cliente hasta la implementación no es trivial. Las necesidades del cliente no son fáciles de discernir pues se debe diseñar una implementación funcional que se ajuste a esas necesidades.

De acuerdo a lo expresado en el párrafo anterior se presentan los acápites 3.2.1 y 3.2.2 con un listado consecutivo de los requerimientos tanto funcionales como no funcionales de la aplicación a desarrollar; resultado de los métodos descritos en la Fundamentación Teórica y en la fase de Comienzo o Inicio del Proceso Unificado de Rational.

3.2.1 Requisitos Funcionales

1. La capa de servicios web debe ser capaz de permitir realizar búsquedas de Listas Código por diferentes criterios:
 - 1.1 Obtener la información de una lista código dado el id de la lista.
 - 1.2 Obtener la información de las listas código dado el nombre.
 - 1.3 Obtener la información de las listas código dada la descripción.
 - 1.4 Obtener la información de las listas código dado el nombre y la descripción.

2. La capa de servicios web debe ser capaz de permitir realizar búsquedas de los elementos de las Listas Código por diferentes criterios:
 - 2.1 Obtener la información de los elementos de una lista código dado el id de la lista.
 - 2.2 Obtener la información de un elemento de la lista código dado el id del elemento
 - 2.3 Obtener la información de los elementos de la lista código dado el nombre del elemento.
 - 2.4 Obtener la información de los elementos de la lista código dada la descripción del elemento.
 - 2.5 Obtener la información de los elementos de la lista código dado si es un nomenclador o no.
 - 2.6 Obtener la información de los elementos de la lista código dado el id de la lista y el nombre del elemento
 - 2.7 Obtener la información de los elementos de la lista código dado el id de la lista y la descripción del elemento
 - 2.8 Obtener la información de los elementos de la lista código dado el id del lista y si es nomenclador o no.
 - 2.9 Obtener la información de los elementos de la lista código dado el nombre del elemento y la descripción del elemento.

2.10 Obtener la información de los elementos de la lista código dado el nombre del elemento y si es nomenclador o no.

2.11 Obtener la información de los elementos de la lista código dada la descripción del elemento y si es nomenclador o no.

2.12 Obtener la información de los elementos que tienen unidad de medida de la lista código dada la pendiente y término independiente del elemento.

2.13 Obtener la información de los elementos que tienen unidad de medida de la lista código dada la pendiente del elemento.

2.14 Obtener la información de los elementos que tienen unidad de medida de la lista código dado el término independiente del elemento.

2.15 Obtener la información de los elementos que tienen unidad de medida de la lista código dado el id del elemento.

3. La capa de servicios web debe ser capaz de permitir realizar búsquedas por diferentes criterios de los sinónimos de cada elemento que compone a la Listas Código:

3.1 Obtener la información de los sinónimos de un elemento de la lista código dado el id del elemento

3.2 Obtener la información de los sinónimos de un elemento de la lista código dado el id del sinónimo

3.3 Obtener la información de los sinónimos de un elemento de la lista código dado el nombre del sinónimo

3.4 Obtener la información de los sinónimos de un elemento de la lista código dada la descripción del sinónimo

3. Obtener la información de los sinónimos de un elemento de la lista código dado el id del elemento y el nombre del sinónimo

3.6 Obtener la información de los sinónimos de un elemento de la lista código dado el Id del elemento y la descripción del sinónimo

3.7 Obtener la información de los sinónimos de un elemento de la lista código dado el id del elemento, el nombre y la descripción del sinónimo

3.2.2 Requisitos No Funcionales

Usabilidad

- Facilidad de uso.
- Garantizar una búsqueda y conexión rápida con la base de datos que contendrá la información.

Rendimiento

- El sistema debe ser capaz de brindar y mostrar la información con el criterio de búsqueda seleccionado por el usuario.
- La eficiencia del producto estará determinada por la velocidad de transmisión de la red y su nivel de congestión.

Utilización de recursos

- Capacidad de respuesta, obtener los resultados a mostrar en el menor tiempo posible.

Disponibilidad

- El sistema estará disponible las 24 horas del día.

Soporte

- Las pruebas del sistema se realizarán en la UCI, las mismas permitirán evaluar en la práctica las funcionalidades y las ventajas de este nuevo producto, así como los errores.
- La información deberá estar disponible a las aplicaciones en todo momento, limitada solamente por algún fallo del sistema.

Software

- Se usará Linux en cualquier versión (Ubuntu, Debian, Suse, etc.).
- PostgreSQL 8.2 como gestor de Base de Datos.
- PHP como lenguaje de programación.

Hardware

- Procesador: Intel Pentium class, 90 MHz o mayor.
- Memoria RAM: Es aconsejable más de 256 megabytes (MB).
- Disco duro: 40 GB de espacio de disco duro.
- Buen funcionamiento de la red soportando hasta 100 Mbps de velocidad

3.4 Descripción del Sistema Propuesto

3.4.1 Descripción de los actores

Tabla 1: Descripción de los Actores del sistema a automatizar

Actores	Justificación
Aplicación	Son las aplicaciones de lo diferentes módulos que necesitan constantemente de la información de los nomencladores.

3.4.2 Modelo de Caso de Uso del Sistema

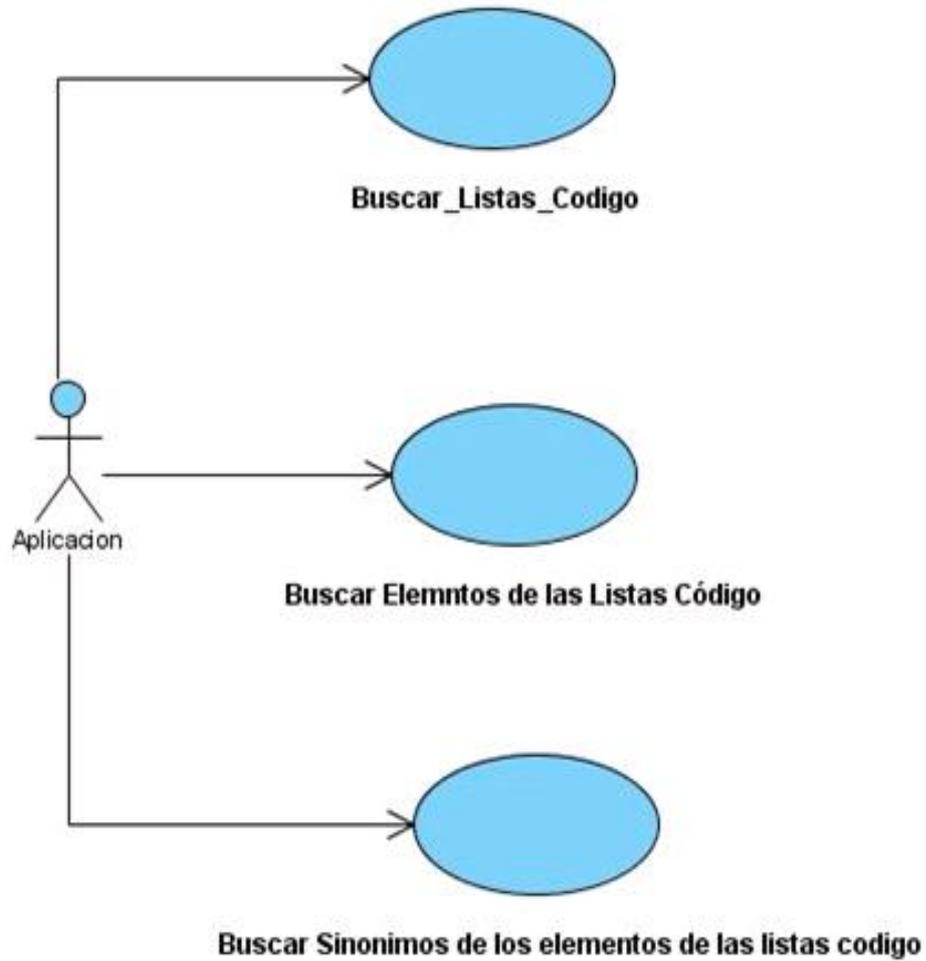


Figura 5: Diagrama de Casos de Uso del Sistema

3.4.3 Descripción de los Casos de Uso del Sistema

Tabla 2: Descripción Caso de Uso Buscar Listas Código

Caso de Uso	Buscar Listas Código
Actores	Aplicación
Propósito	Mostrar la información de la Listas Código solicitada
Resumen	El Caso de Uso se inicia cuando la aplicación realiza el pedido de informaciones de las Listas Código y termina con el envío de la información que desea.
Referencias	R1.1, R1.2, R1.3, R1.4
Precondiciones	

Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Realizar el pedido de las Listas Código según los criterios de búsquedas. Si realiza el pedido de búsqueda por id de la lista ver CA1, Si realiza el pedido de búsqueda por nombre ver CA2, Si realiza el pedido de búsqueda por la descripción ver CA3, Si realiza el pedido de búsqueda por nombre del elemento y la descripción ver CA4.	2. Realizar la búsqueda de las Listas Código según el criterio de búsqueda. 3. Enviar la información encontrada y el caso de uso termina
Curso Alterno CA1	
	2. Realizar la búsqueda de las Listas Código por el id. 3. Enviar la información encontrada y el caso de uso termina
Curso Alterno CA2	
	2. Realizar la búsqueda de las Listas Código por el nombre. 3. Enviar la información encontrada y el caso de uso termina
Curso Alterno CA3	
	2. Realizar la búsqueda de las Listas Código por la descripción. 3. Enviar la información encontrada y el caso de uso termina
Curso Alterno CA4	
	2. Realizar la búsqueda de las Listas Código por el nombre y la descripción. 3. Enviar la información encontrada y el caso de uso termina
Pos condiciones	Se envía la información solicitada por la aplicación.

Requisitos Especiales	
Prioridad	Crítico
Interfaz	

Tabla 3: Descripción Caso de Uso Buscar elementos de las Listas Código

Caso de Uso	Buscar elementos de las Listas Código	
Actores	Aplicación	
Propósito	Mostrar la información de los elementos de las Listas Código	
Resumen	El Caso de Uso se inicia cuando la aplicación realiza el pedido de información de los elementos de las Listas Código y termina con el envío de la información que desea.	
Referencias	R2.1, R2.2, R2.3, R2.4, R2.5, R2.6, R2.7, R2.8, R2.9, R2.10, R2.11, R2.12, R2.13, R2.14, R2.15.	
Precondiciones		
Curso Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Realizar el pedido de los elementos o del elemento de las listas código según los criterios de búsquedas. Si realiza el pedido de búsqueda por el id de la lista ver CA1, Si realiza el pedido de búsqueda por el id del elemento ver CA2, Si realiza el pedido de búsqueda por el nombre ver CA3, Si realiza el pedido de búsqueda por la descripción ver CA4, Si realiza el pedido de búsqueda por si el elemento es un nomenclador o no, ver CA5, Si realiza el pedido de búsqueda por el id de de la lista y el nombre del elemento ver CA6, Si realiza el pedido de búsqueda por id de la lista y la descripción ver CA7, Si realiza el pedido de búsqueda	2. Realizar la búsqueda de los elementos según el criterio de búsqueda. 3. Enviar la información encontrada y el caso de uso termina	

<p>por el id de la lista y si el elemento es un nomenclador o no ver CA8, Si realiza el pedido de búsqueda por el nombre del elemento y la descripción ver CA9, Si realiza el pedido de búsqueda por el nombre del elemento y si es nomenclador o no ver CA10, Si realiza el pedido de búsqueda por la descripción del elemento y si es nomenclador o no ver CA11, Si realiza el pedido de búsqueda por la pendiente y el término independiente del elemento ver CA12, Si realiza el pedido de búsqueda por la pendiente ver CA13, Si realiza el pedido de búsqueda por el termino independiente, ver CA14, Si realiza el pedido de búsqueda por el id del elemento ver CA15.</p>	
Curso Alterno CA1	
	<ol style="list-style-type: none"> 2. Realizar la búsqueda de los elementos por el id de la lista 3. Enviar la información encontrada y el caso de uso termina
Curso Alterno CA2	
	<ol style="list-style-type: none"> 2. Realizar la búsqueda del elemento por el id del elemento. 3. Enviar la información encontrada y el caso de uso termina
Curso Alterno CA3	
	<ol style="list-style-type: none"> 2. Realizar la búsqueda de los elementos por el nombre. 3. Enviar la información encontrada y el caso de uso termina

Curso Alterno CA4	
	<p>2. Realizar la búsqueda de los elementos por la descripción.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA5	
	<p>2. Realizar la búsqueda de los elementos por si es un nomenclador o no.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA6	
	<p>2. Realizar la búsqueda de los elementos por el id de la lista y el nombre del elemento.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA7	
	<p>2. Realizar la búsqueda de los elementos por el id de la lista y la descripción.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA8	
	<p>2. Realizar la búsqueda del elemento por el id de la lista y si el elemento es un nomenclador.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA9	
	<p>2. Realizar la búsqueda del elemento por el nombre y la descripción</p> <p>3. Enviar la información encontrada y el caso</p>

	de uso termina
Curso Alterno CA10	
	<p>2. Realizar la búsqueda del elemento por el nombre del elemento y si es un nomenclador o no</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA11	
	<p>2. Realizar la búsqueda del elemento por la descripción del elemento y si es un nomenclador o no.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA12	
	<p>2. Realizar la búsqueda del elemento por la pendiente y el término independiente.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA13	
	<p>2. Realizar la búsqueda de los elementos por la pendiente.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA14	
	<p>2. Realizar la búsqueda de los elementos por el término independiente.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA15	
	<p>2. Realizar la búsqueda del elemento por el</p>

	id del elemento 3. Enviar la información encontrada y el caso de uso termina
Pos condiciones	Se envía la información solicitada por la aplicación.
Requisitos Especiales	
Prioridad	Crítico
Interfaz	

Tabla 4: Descripción Caso de Uso Buscar los sinónimos de los elementos de las Listas Código

Caso de Uso	Buscar los sinónimos de los elementos de las Listas Código	
Actores	Aplicación	
Propósito	Mostrar la información de los sinónimos de los elementos de las Listas Código.	
Resumen	El Caso de Uso se inicia cuando la aplicación realiza el pedido de información de los sinónimos de los elementos de las Listas Código y termina con el envío de la información que desea.	
Referencias	R3.1, R3.2, R3.3. R3.4, R3.5, R3.6, R3.7.	
Precondiciones		
Curso Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Realizar el pedido de los sinónimos de los elementos o del elemento de las Listas Código según los criterios de búsquedas. Si realiza el pedido de búsqueda por el id del elemento ver CA1, Si realiza el pedido de búsqueda por el id del sinónimo ver CA2, Si realiza el pedido de búsqueda por el nombre del sinónimo ver CA3, Si	4. Realizar la búsqueda de los elementos según el criterio de búsqueda. 5. Enviar la información encontrada y el caso de uso termina	

<p>realiza el pedido de búsqueda por la descripción del sinónimo ver CA4, Si realiza el pedido de búsqueda por el id del elemento y el nombre del sinónimo ver CA5, Si realiza el pedido de búsqueda por el id del elemento y la descripción del sinónimo ver CA6, Si realiza el pedido de búsqueda por el id del elemento , nombre del sinónimo y la descripción ver CA7.</p>	
Curso Alterno CA1	
	<ol style="list-style-type: none"> 2. Realizar la búsqueda de los sinónimos de los elementos por el id del elemento. 3. Enviar la información encontrada y el caso de uso termina
Curso Alterno CA2	
	<ol style="list-style-type: none"> 2. Realizar la búsqueda de los sinónimos de los elementos por el id del sinónimo. 3. Enviar la información encontrada y el caso de uso termina
Curso Alterno CA3	
	<ol style="list-style-type: none"> 2. Realizar la búsqueda de los sinónimos de los elementos por el nombre. 3. Enviar la información encontrada y el caso de uso termina
Curso Alterno CA4	
	<ol style="list-style-type: none"> 2. Realizar la búsqueda de los sinónimos de los elementos por la descripción. 3. Enviar la información encontrada y el caso de uso termina
Curso Alterno CA5	
	<ol style="list-style-type: none"> 2. Realizar la búsqueda de los sinónimos de

	<p>los elementos por el id del elemento y el nombre del sinónimo.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA6	
	<p>2. Realizar la búsqueda de los sinónimos de los elementos por el id del elemento y la descripción del sinónimo.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Curso Alterno CA7	
	<p>2. Realizar la búsqueda de los sinónimos de los elementos por el id del elemento, nombre del sinónimo y la descripción.</p> <p>3. Enviar la información encontrada y el caso de uso termina</p>
Pos condiciones	Se envía la información solicitada por la aplicación.
Requisitos Especiales	
Prioridad	Crítico
Interfaz	

3.5 Conclusiones parciales

El estudio realizado en este capítulo ha facilitado un entendimiento de la dinámica y estructura de la organización, contribuyendo a una mejor comprensión del problema que el software debe resolver; logrando así una obtención eficiente del producto final.

De esta forma se logró de manera clara obtener las funcionalidades que le conciernen conllevando a un mejor desempeño del sistema tanto por los clientes como por el equipo de desarrollo. Además se ganó claridad en cuanto a la concesión del sistema a construir y se sentaron las bases para las restante fases del proceso de diseño e implementación del mismo.

4

CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

Introducción

El análisis de un sistema se centra en la investigación del problema y no en la manera de definir una solución, mientras que el diseño pone de relieve una solución lógica de cómo el sistema debe satisfacer los requerimientos funcionales, requerimientos de calidad y las restricciones, es decir, en esta parte del proceso de desarrollo del software se decide como se va a llevar a cabo el mismo.

En este capítulo se realiza el diseño de la propuesta de solución, a través de los flujos de Análisis y Diseño e Implementación, se presenta el diagrama de clases del diseño de los diferentes casos de usos definidos en el capítulo anterior, también se muestran los diagramas de interacción correspondientes a cada uno de ellos, se muestra el modelo lógico y físico de datos, se especifican los principios de diseño así como una concepción general del tratamiento de los errores que pueden aparecer. Finalmente se realiza el modelo de despliegue donde se representan los nodos en los que se distribuye la aplicación y el de componentes para una mejor descripción de la solución propuesta.

4.1 Diagrama de Clases del Diseño

4.1.1 Modelo de Diseño

En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales, creándose así una entrada apropiada para las actividades de implementación.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de implementación.

En este modelo, los casos de uso son realizados por las clases del diseño y sus objetos, lo cual se denota por la realización de casos de uso del diseño que describe cómo se realizan estos en particular. Para un mejor entendimiento ver las figuras 6, 7 y 8.

Caso de Uso: Buscar Listas Código.

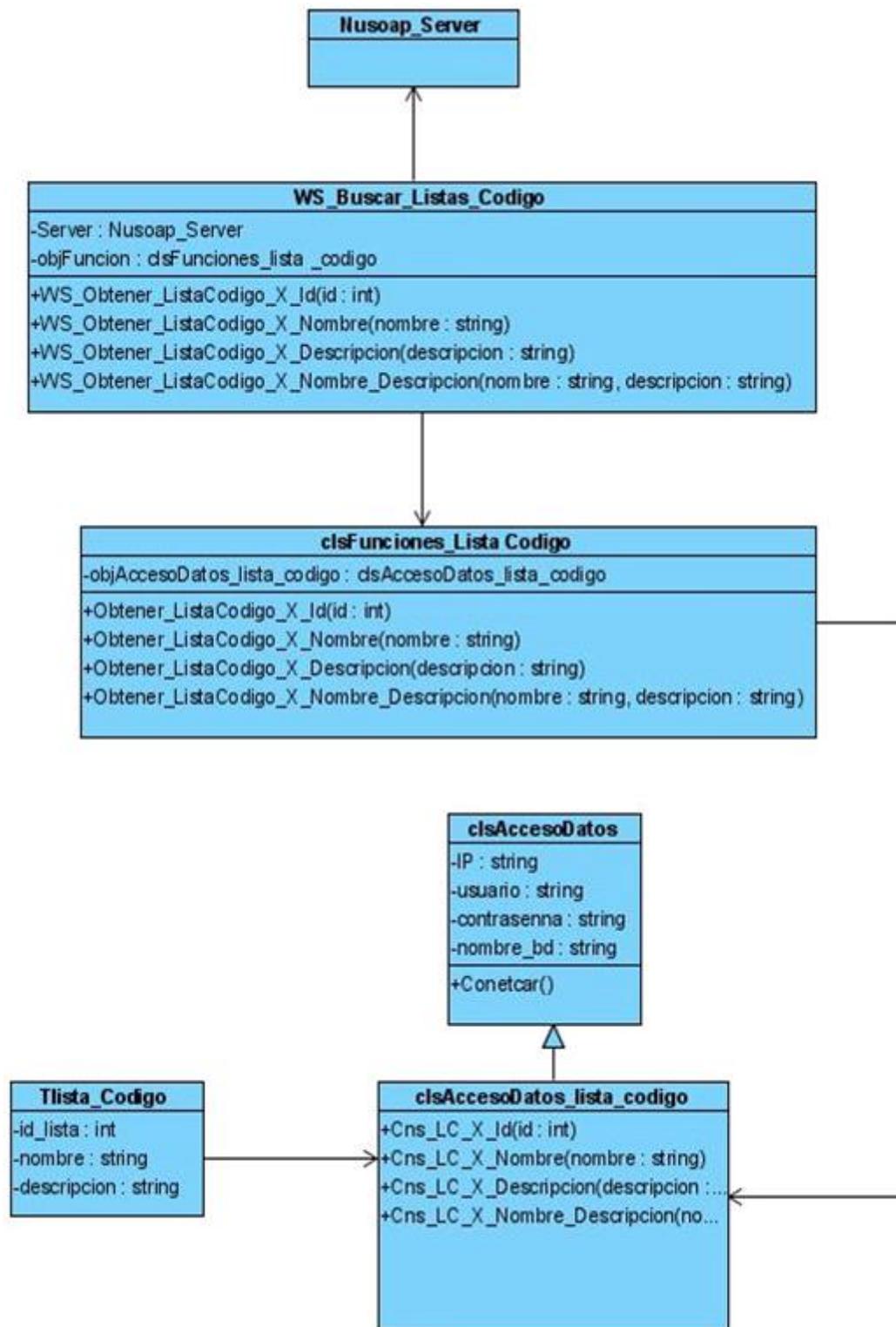


Figura 6: Diagrama de Clases de diseño Buscar Listas Código.

Caso de Uso: Buscar elementos de las Listas Código.

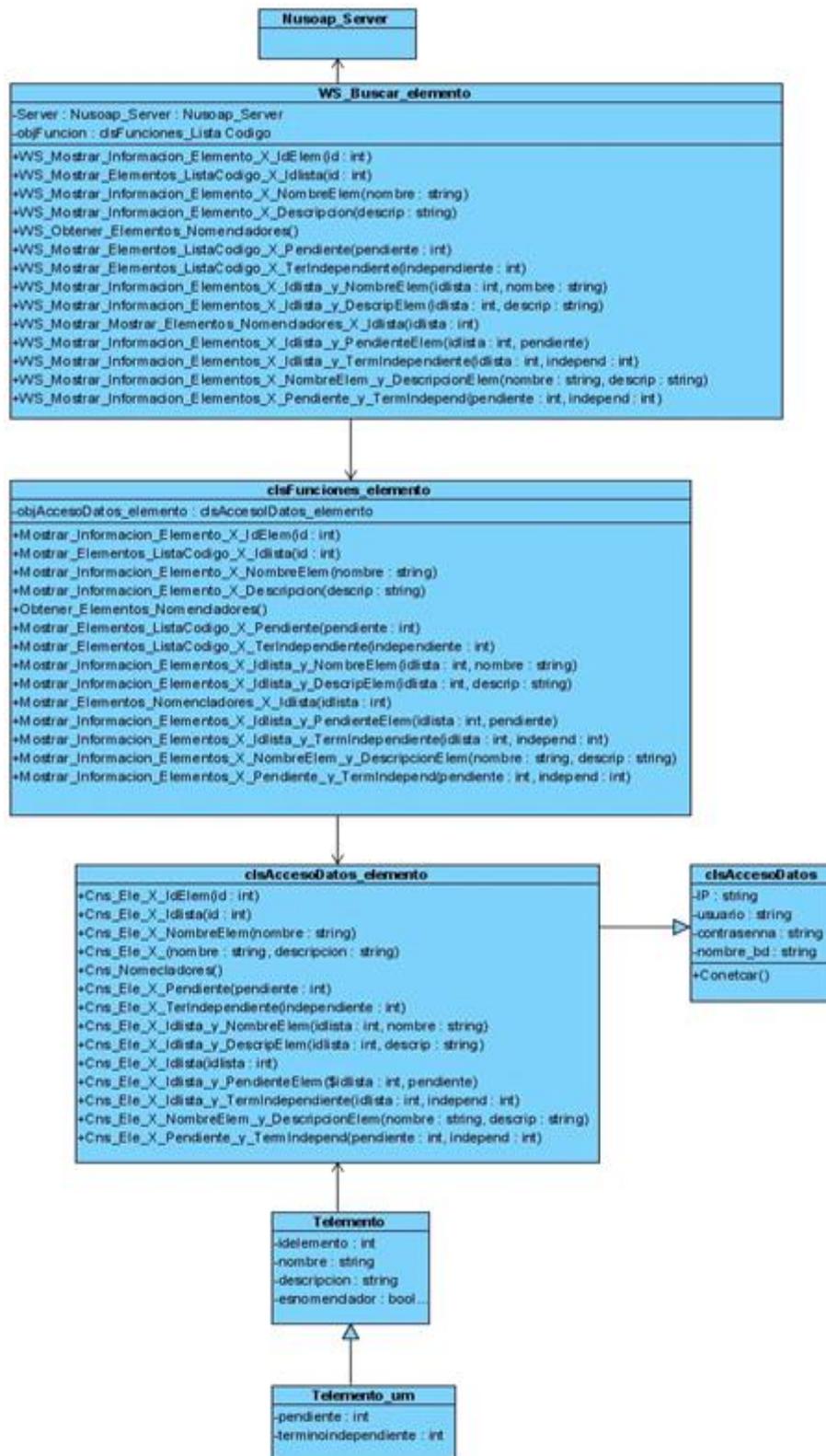


Figura 7: Diagrama de Clases de diseño Buscar elementos de las Listas Código.

Caso de Uso: Buscar los sinónimos de los elementos de las Listas Código

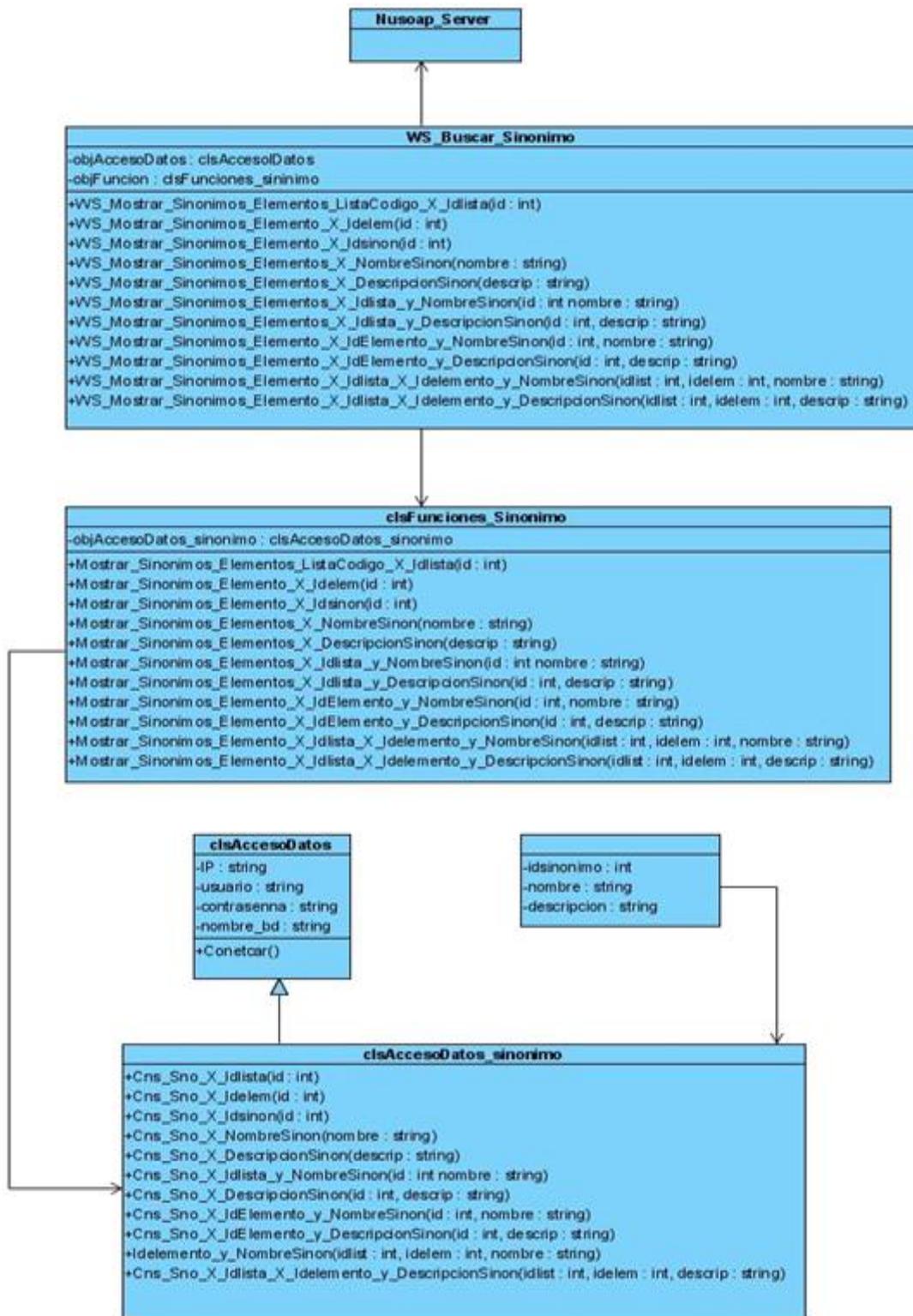


Figura 8: Diagrama de Clases de diseño Buscar los sinónimos de los elementos de las Listas Código

4.1.2 Principios de Diseño

El diseño tiene que satisfacer los requerimientos del usuario, y en este caso será para un personal que tenga algún conocimiento informático que le posibilite consumir los servicios web. El diseño Web no se limita a la apariencia estética, la combinación de colores, o a un logo acertado. De él depende que la información sea útil, que los servicios se puedan usar, es decir, el diseño convierte a una aplicación Web en algo atractivo para el usuario, por su estética y su utilidad.

Una aplicación para que adquiera un gran valor para el usuario debe tener un buen diseño gráfico, una buena navegabilidad, usabilidad y distribución de los contenidos. Para ello, este sistema utiliza ciertos principios, específicamente los del diseño orientado a objetos; los cuales para aplicarlos al campo de los servicios Web, es necesario comprender los "cuatro principios" de la orientación a servicios tan frecuentemente citados y a partir de ellos determinar los que se usarán para los Servicios Web que se implementarán.

Principio 1: Los límites son explícitos.

“Los servicios interactúan a través de límites explícitos de paso de mensajes bien definidos. Pasar los límites de los servicios puede resultar costoso, en función de los factores geográficos, de confianza o ejecución. Un límite representa el paso entre la interfaz pública del servicio y su implementación interna y privada. El límite de un servicio se publica a través de WSDL y puede incluir aserciones que indican las expectativas de un servicio dado”.

1. El consumo de los servicios debe resultar fácil. Al diseñar un servicio, los desarrolladores deben procurar que su consumo resulte sencillo para otros desarrolladores.
2. La interfaz (contrato) del servicio se debe diseñar también de modo que su evolución no implique la ruptura de contratos con los usuarios existentes.
3. Evitar interfaces RPC. Pasar mensajes de forma explícita constituye una técnica más adecuada que el uso de modelos RPC. Este enfoque aísla al usuario del mecanismo interno de la implementación del servicio, lo que evita que los desarrolladores de servicios deban, además de desarrollar los servicios, minimizar los efectos en los usuarios del servicio (encapsulación por mensajes públicos en lugar de métodos públicos disponibles).
4. El área de la superficie del servicio se debe mantener pequeña.
5. Proporcionar pocas interfaces públicas bien definidas al servicio, cuanto mayor sea el número de interfaces públicas expuestas por un servicio, más difícil resultará su consumo y mantenimiento. Además estas interfaces deben ser relativamente simples, y estar

diseñadas para aceptar un mensaje de entrada bien definido y responder con un mensaje de salida igualmente bien definido y una vez diseñadas, deben permanecer estáticas.

6. Los detalles de la implementación interna (privada) no deben salir de los límites de un servicio.

Principio 2: Los servicios son autónomos.

“Los servicios son entidades que se implementan, versionan y administran de forma independiente. Los desarrolladores deben evitar hacer suposiciones en cuanto al espacio entre los límites del servicio, ya que es más probable que éste cambie a que lo hagan los propios límites.”

7. Los servicios se deben implementar y versionar de forma independiente al sistema en el que se implementan y consumen.
8. Los contratos se deben diseñar con la asunción de que una vez que se publiquen, no se podrán modificar. Este enfoque obliga a los desarrolladores a elaborar los diseños de sus esquemas de forma flexible.
9. Se debe garantizar la ausencia de errores en los servicios. Para ello, es necesario adoptar un punto de vista pesimista. Desde la perspectiva del usuario, se deben planear niveles no confiables de disponibilidad y rendimiento del servicio. Asimismo, desde el punto de vista del proveedor, se debe esperar que el usuario hará un mal uso del servicio (deliberado o no) y que cometerá errores, tal vez sin notificar al servicio.

Principio 3: Los servicios comparten esquema y contrato, pero no clase.

“La interacción del servicio se debe basar únicamente en políticas, esquemas y comportamientos basados en contratos. El contrato de un servicio se define generalmente a través de WSDL, mientras que los contratos de las agregaciones de servicios se pueden definir a través de BPEL (que, a su vez, utilizan WSDL para cada uno de los servicios agregados).”

10. Garantizar que el servicio permanece estable con el fin de minimizar los efectos en los usuarios del mismo.
11. Los contratos se deben diseñar para que sean lo más explícitos posible con el fin de minimizar la tergiversación. Asimismo, los contratos se deben diseñar para admitir futuras versiones del servicio a través de la extensibilidad tanto de la sintaxis XML como del modelo de procesamiento SOAP.

12. Mantener siempre definida la línea entre las representaciones de datos públicos y privados.
El formato de los datos internos de un servicio determinado se debe ocultar a los usuarios al tiempo que el esquema de datos públicos debe permanecer inmutable.
13. Versionar los servicios cuando los cambios en el contrato de los mismos sean inevitables.
Este enfoque minimiza la ruptura de las implementaciones de los usuarios existentes

Principio 4: La compatibilidad de los servicios se basa en una directiva.

“No es posible comunicar determinados requisitos para la interacción de servicios sólo en WSDL. Las expresiones de directiva se pueden utilizar para separar la compatibilidad estructural (lo que se comunica) de la semántica (cómo y a quién se comunica un mensaje).”

14. Se debe garantizar que sus aserciones de directivas sean lo más explícitas posible en cuanto a las expectativas y las capacidades de semántica del servicio.

4.1.3 Concepción general de ayuda

La capa de Servicios Web va dirigida a usuarios que poseen conocimiento en el tema de la informática. Por este motivo se consideró que una ayuda formal, independiente de las acciones habituales del usuario, sería muy funcional. La ayuda será un sitio web que estará a tiempo completo a disposición de los usuarios, donde se describirá de manera muy detallada cada uno de los servicios que se brindarán, con el fin de orientar y guiar al usuario con lo que necesite.

4.2 Diseño de la Base de Datos

Según Jeffrey Ullman: “Un modelo de datos es un sistema formal y abstracto que permite describir los datos de acuerdo con reglas y convenios predefinidos. Es formal pues los objetos del sistema se manipulan siguiendo reglas perfectamente definidas y utilizando exclusivamente los operadores definidos en el sistema, independientemente de lo que estos objetos y operadores puedan significar.”

Es la descripción de la organización de una base de datos, constituyéndose en una representación gráfica orientada a la obtención de la estructura de datos mediante métodos. Para una interpretación de lo descrito puede verse la figura 9 y en las tablas 5, 6, 7, 8 y 9 en las cuales se hace una descripción de las tablas de la base de datos.

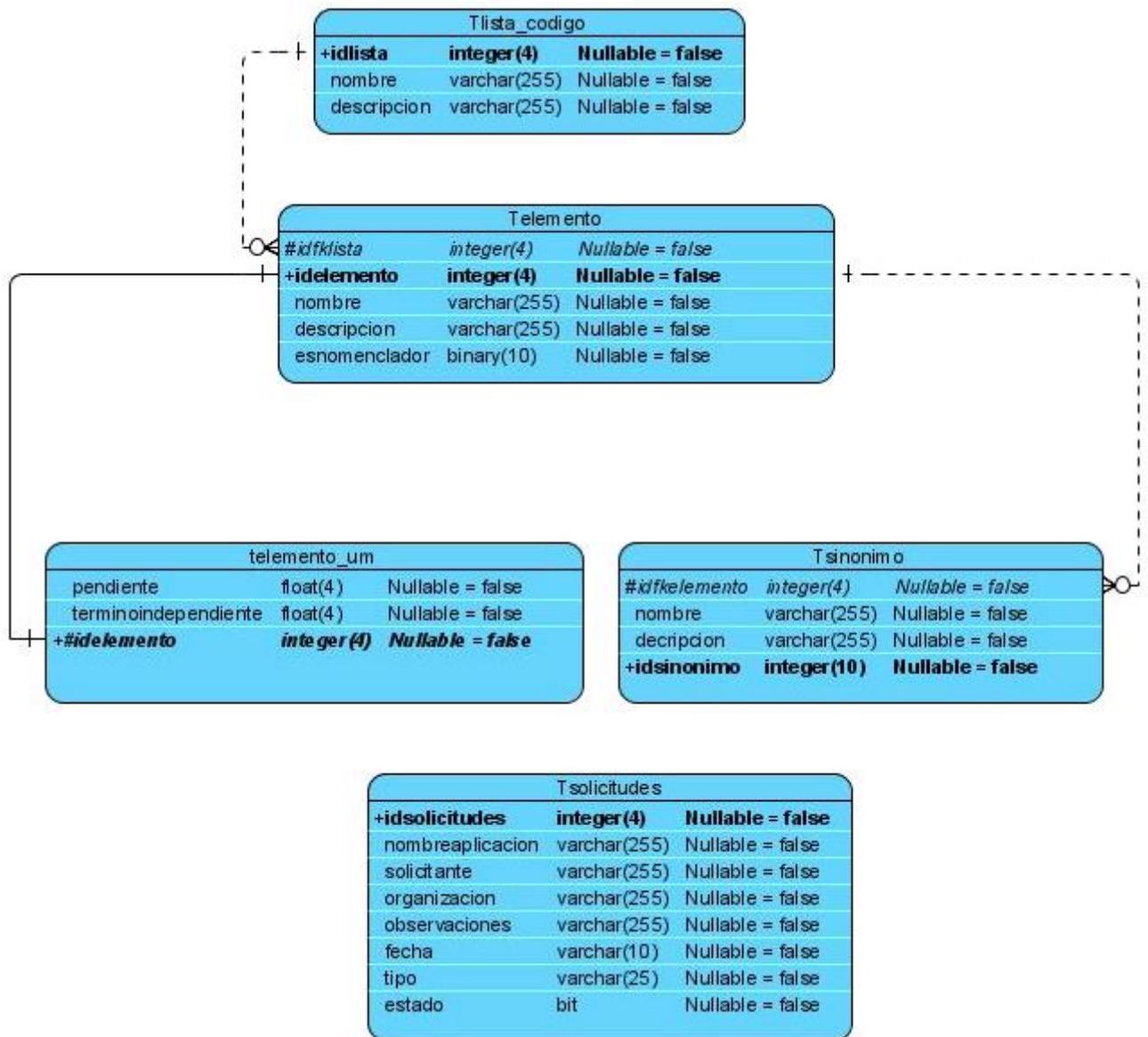


Figura 9: Diseño de la Base de datos “Modelo de datos”

4.2.1 Descripción de las tablas de la Base de datos.

Tabla 5: Descripción de la tabla “Tlista_código”

Nombre: Tlista_código		
Descripción: En esta tabla se almacenan los datos de cada una de las listas código que están definida hasta la actualidad en la rama de la geología.		
Atributo	Tipo	Descripción
idlista	INTEGER	Identificador de las listas código

nombre	VARCHAR	Nombre de la lista código
descripción	VARCHAR	Breve descripción de cada una de las listas código.

Tabla 6: Descripción de la tabla “Telemento”.

Nombre: Telemento		
Descripción: En esta tabla se almacenan los datos de cada uno de los elemento por los que están compuestas las listas código que están definida hasta la actualidad en la rama de la geología.		
Atributo	Tipo	Descripción
idfklista	INTEGER	Identificador de las listas código.
idelemento	INTEGER	Identificador de los elementos de las listas código.
nombre	VARCHAR	Nombre de los elementos por los que esta compuesto las listas código.
descripción	VARCHAR	Breve descripción de cada uno de los elementos de las listas código.
esnomenclador	BINARY	Información de si un elemento es un nomenclador o no

Tabla7: Descripción de la tabla “Telemento_um”.

Nombre: Telemento_um		
Descripción: En esta tabla se almacenan los datos de cada uno de los elemento que poseen unidad de medida por los que están compuestas las listas código que están definida hasta la actualidad en la rama de la geología.		
Atributo	Tipo	Descripción
idelemento	INTEGER	Identificador de los elementos de las listas código.
terminoindependiente	INTEGER	Valor predeterminado que varía en dependencia del tipo de conversión que se vaya a hacer.

pendiente	INTEGER	Valor predeterminado que varía en dependencia del tipo de conversión que se vaya a hacer.
-----------	---------	---

Tabla 8: Descripción de la tabla “Tsolicitudes”

Nombre: Tsolicitudes		
Descripción: En esta tabla se almacenan los datos de cada una de las solicitudes que se realizan ya sea por otra aplicación o por un usuario.		
Atributo	Tipo	Descripción
idsolicitudes	INTEGER	Identificador de las solicitudes,
nombreakaplicacion	VARCHAR	Nombre de la aplicación que solicita una inserción o una modificación.
organización	VARCHAR	Nombre de la organización que solicita una inserción o una modificación.
observaciones	VARCHAR	Breve descripción del por qué de cada una de las solicitudes de inserción o de modificación.
fecha	VARCHAR	Fecha en que solicita una inserción o una modificación.
tipo	VARCHAR	Identificador para saber el tipo de solicitud realizada (Puede ser de cambio o de inserción).
estado	BIT	Identificador para saber el estado de la solicitud realizada (Puede estar pendiente o resuelto).

Tabla 9: Descripción de la tabla “Tsinonimo”.

Nombre: Telemento_um		
Descripción: En esta tabla se almacenan todos los sinónimos de los elementos que tiene cada lista código		
Atributo	Tipo	Descripción

idsinonimo	INTEGER	Identificador de los sinónimos.
idfkelemento	INTEGER	Identificador del elemento al que le corresponde el sinónimo.
nombre	VARCHAR	Nombre del elemento al cual pertenece el sinónimo.
descripción	VARCHAR	Breve descripción del elemento al que le corresponde el sinónimo.

4.2.2 Diagrama de Clases Persistentes

A continuación se muestra el diagrama de clases persistentes del sistema, donde aparecen todas las entidades que se manejan en él.

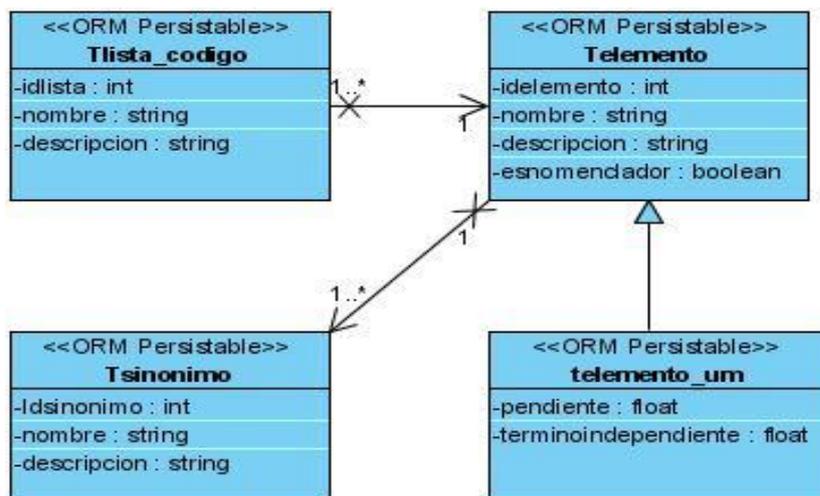


Figura 10: Diagrama de clases persistentes.

4.3 Modelo de Despliegue

El modelo de despliegue describe la distribución física del sistema, muestra la distribución de los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware

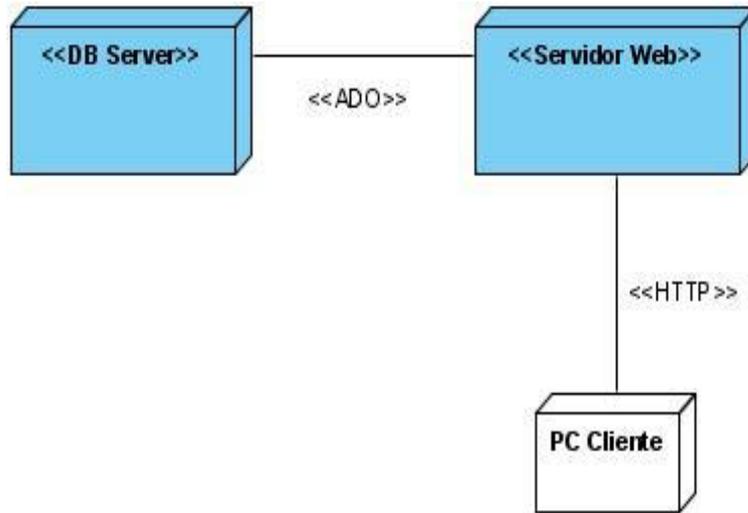


Figura 11: Diagrama de Despliegue.

4.3.1 Descripción de los componentes del Diagrama de Despliegue.

Tabla 8: Descripción de los componentes del Diagrama de Despliegue.

Componentes	Descripción
DB Server	Servidor donde se encuentra ubicada la base de dato de donde nuestros servicios se nutren de información.
PC Cliente	Ordenador desde donde los usuarios o las aplicaciones consumen los servicios.
Server	Servidor Web que contiene las aplicaciones web y los servicios web.
HTTP	Protocolo utilizado para la conexión entre las PC Cliente y el servidor de aplicaciones y servicios web.
ADO	Librería utilizada para establecer la conexión entre el servidor de aplicaciones y servicios web y el servidor de base de dato.

4.4 Modelo de Implementación

4.4.1 Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Estos representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, entre otros. Las relaciones de dependencia se utilizan en estos diagramas para indicar que un componente utiliza los servicios ofrecidos por otro.

Un diagrama de componentes representa las dependencias entre componentes software, incluyendo código fuente, código binario, y ejecutables. Un módulo de software se puede representar como componente. Algunos existen en tiempo de compilación, algunos en tiempo de enlace y algunos en tiempo de ejecución, otros en varias de éstas.

Es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. Para visualizar los componentes ver las siguientes figuras.

4.4.1.1 Diagrama de Componente del Caso de Uso Buscar Listas Código.

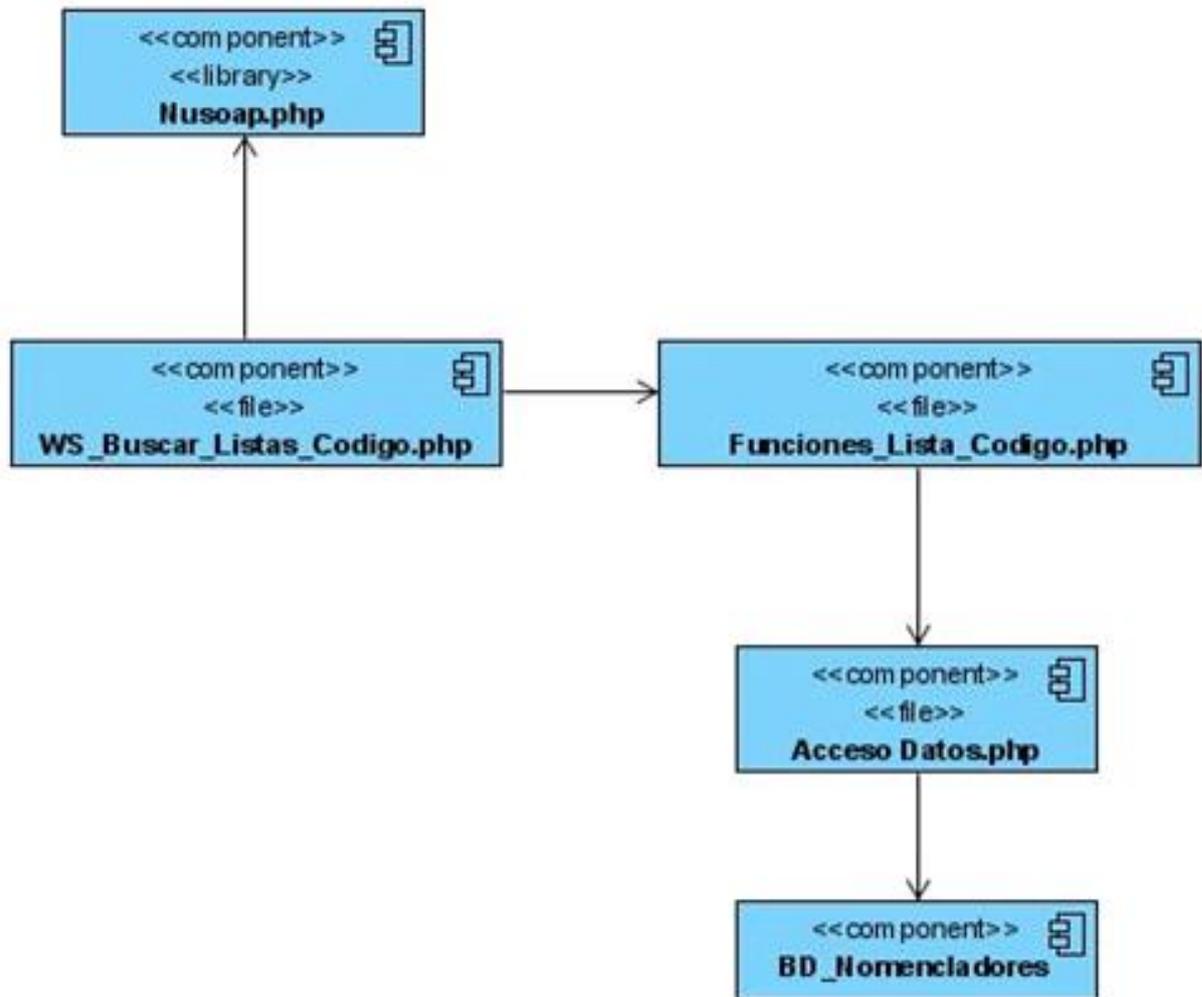


Figura 12: Diagrama Componente Buscar Listas Código.

4.4.1.2 Diagrama de Componente del Caso de Uso Buscar elementos de las Listas Código

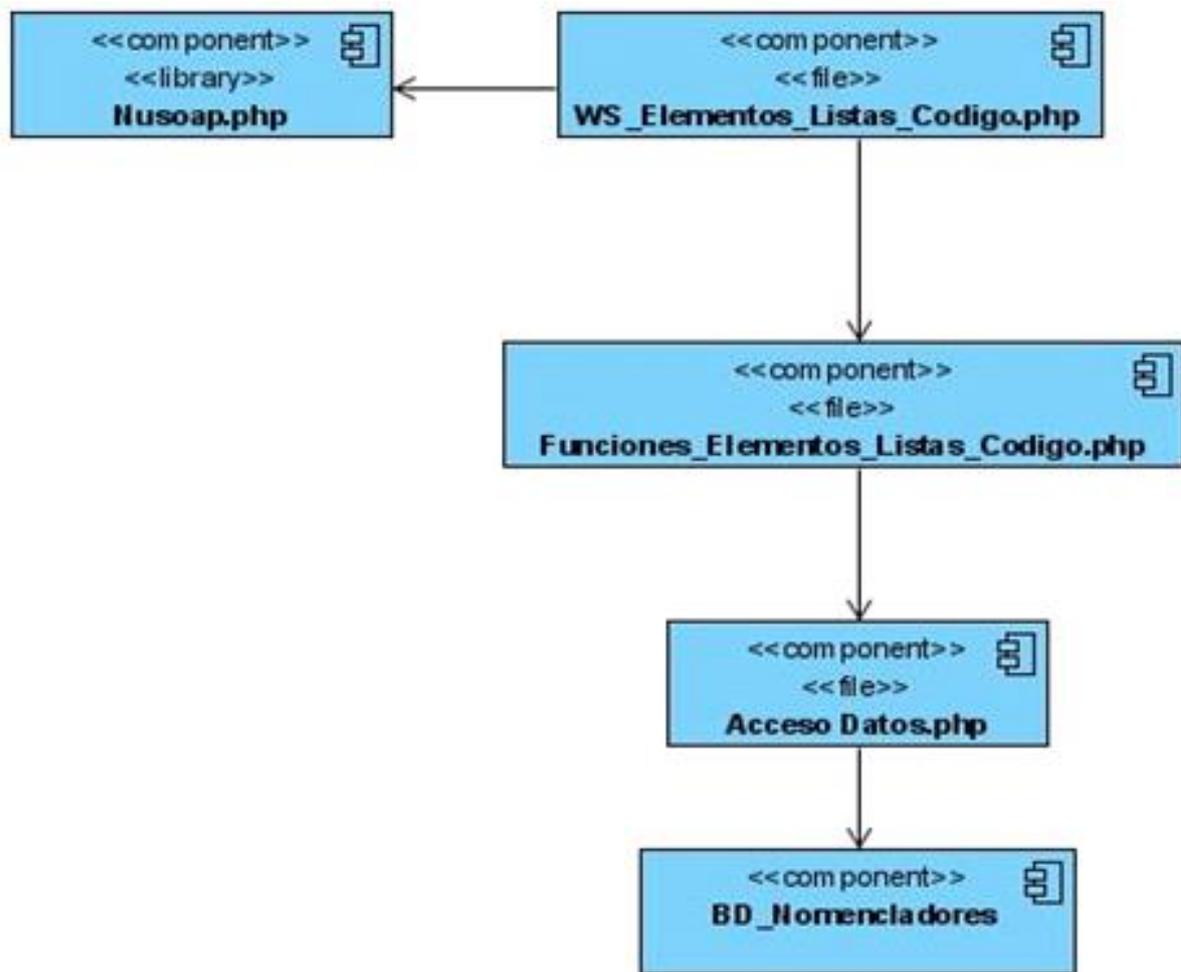


Figura 13: Diagrama Componente del Buscar elementos de las Listas Código.

4.4.1.3 Diagrama de Componente Caso de Uso Buscar los sinónimos de los elementos de las Listas Código

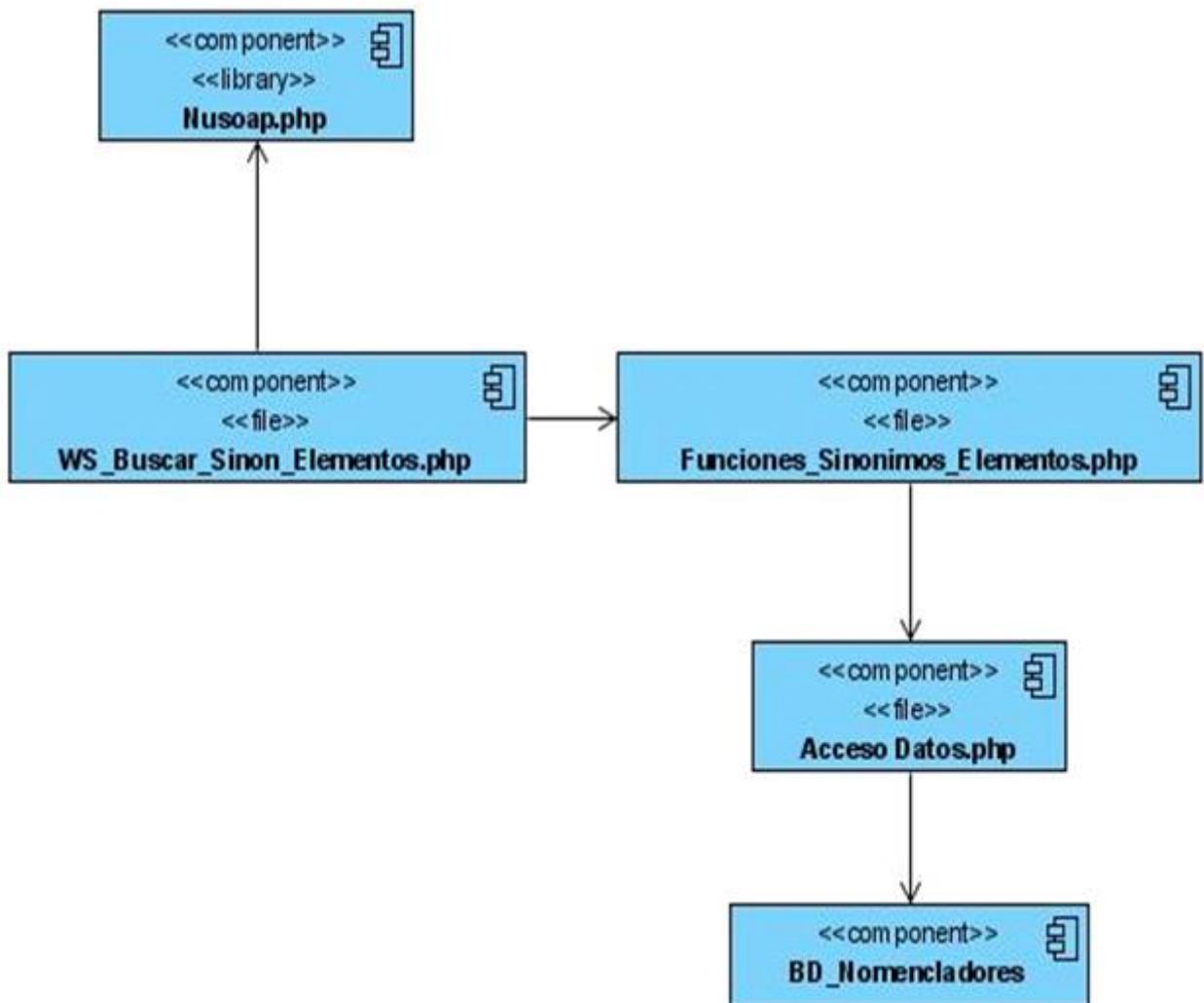


Figura 14: Diagrama Componente Buscar los sinónimos de los elementos.

4.5 Prueba del sistema propuesto

Las pruebas se realizan a lo largo de todo el desarrollo del proyecto ya que pueden arrojar resultados que alertarían a los desarrolladores respecto a la calidad del producto a terminar.

Existen dos tipos de prueba la de caja blanca y caja negra. En la de caja negra se realiza una comprobación de los datos de entrada y los de salida, comprobando si los datos de salida que devuelve el sistema son los esperados de acuerdo a los datos que se le introdujeron.

En la prueba de caja blanca se realiza el traseo de un pedazo de código sentencia a sentencia asignándole valores numéricos a cada una de dichas sentencias. Posteriormente se construye un grafo con dichos valores para seguir la secuencia de pasos consecutivos por los que transcurre la ejecución del mismo.

Por último al grafo se le calcula la complejidad ciclomática la cual mide el número de decisiones lógicas en un segmento de código. A continuación se le realizará la prueba de caja blanca a unas de las funciones implementadas en la creación de los Servicios Web:

```
public function Obtener_ListaCodigo_X_Id($id) { ①
    parent::Conectar(); ②
    $consulta = "SELECT * FROM Tlista_código WHERE idlista = '$id'"; ③
    $resultado = pg_query($consulta); ④
    $datos = array(); ⑤
    while($valores = pg_fetch_array($resultado, null, PGSQL_ASSOC)) ⑥
    {
        $datos = array('Identificador'=>$valores['idlista'], 'Nombre'=> `
        $valores['nombre'], 'Descripcion'=>$valores['descripcion']); ⑦
    }
    return $datos; ⑧
} ⑨
```

Figura 15: Código de función

Cálculo de complejidad para el segmento de código anterior:

$$\text{Complejidad} = \text{NumAristas} - \text{NumNodos} + 2$$

$$\text{NumAristas} = 9$$

$$\text{NumNodos} = 9$$

$$\text{Complejidad} = 9 - 9 + 2 = 2$$

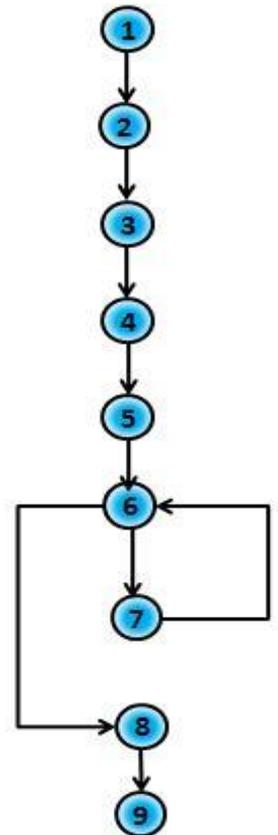


Figura 16: Grafo de Flujo Prueba

4.6. Conclusiones Parciales

Una vez concluido este capítulo se tienen todos los elementos necesarios para la implementación del software. Se describen las clases y sus relaciones mediante los diagramas de clase, se define el diagrama de clases persistentes y el modelo de datos, para la construcción de la base de datos y por último se determinan los nodos y componentes mediante los diagramas de despliegue e implementación para poder mostrar como queda estructurada la aplicación físicamente.

5

CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD

Introducción

En este capítulo se realiza un análisis de los costos y beneficios que concebirá el desarrollo de la Capa de Servicios Web y la proporción entre ellos, con el objetivo de demostrar la factibilidad de la propuesta, así como la conveniencia de su explotación en la Oficina Nacional de Recursos Minerales.

5.1 Estimación de costos

La estimación de costos es de gran importancia en el campo empresarial, puesto que un buen estimado abre las posibilidades de éxito en el cumplimiento del proyecto, ya sea este grande o pequeño, ambos requieren de estimaciones de costos confiables durante las fases conceptuales, diseño y construcción para un desarrollo exitoso del producto final.

Una de las técnicas más difundidas en la estimación es el Análisis de Puntos de Función. Esta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación, las metodologías, plataformas o tecnologías utilizadas.

Sin embargo debido a la complejidad de los proyectos software el SEI¹⁵ (del inglés, Software Engineering Institute) propone el método de estimación COCOMO II(creado por Barry Boehm), debido a que permite determinar el esfuerzo y tiempo de un proyecto de software a partir de ecuaciones matemáticas que permiten calcular el esfuerzo a teniendo en cuenta ciertas métricas de tamaño estimado, y los puntos de función sin ajustar. Lo cual supone una gran ventaja, dado que en la mayoría de los casos es difícil determinar el número de líneas de código de que constará un nuevo desarrollo, en especial cuando se tiene poca o ninguna experiencia previa en proyectos de software.

5.2 Puntos de función sin ajustar (COCOMO II)

Existe una relación natural entre los Puntos de Función y los Casos de Uso. Los Puntos de Función permiten estimar el tamaño del software a partir de sus requerimientos, mientras que los Casos de Uso permiten documentar los requerimientos del software. Ambos tratan de ser independientes de las tecnologías utilizadas para la implementación.

El método de Casos de Uso permite documentar los requerimientos de un sistema en términos de Actores y Casos de Uso, mientras que el Análisis de los puntos de función permite por sí mismos efectuar una estimación del tamaño que tendrá el sistema o del esfuerzo que tomaría implementarlo.

¹⁵ Software Engineering Institute (Instituto de Ingeniería de Software)

5.2.1 Propósito

Con la utilización de COCOMO II como técnica de estimación, que sin dudas es una de las más utilizadas a nivel mundial, y está entre las más efectivas en el cálculo de la estimación de un proyecto, podremos saber aproximadamente el tiempo de duración del proyecto y cuántas personas se necesitarán para su buen funcionamiento.

Paso 1. Identificar los Puntos de casos de uso sin ajustar

$$UUCP = UAW + UUCW$$

1ro- Calcular UAW

Factor de peso de los actores sin ajustar

Tipo	Descripción	Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3

$$UAW = 2 * 3 = 6.$$

Para calcular UUCW

Tipo	Descripción	Peso	Cantidad * Peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	3 * 5

Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	0 * 10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	0 * 1
Total			15

UUCW = 15.

Luego UUCP = 15 + 6

UUCP = 21.

Paso 2. Ajustar los Puntos de casos de uso

UCP = UUCP * TCF * EF

Para Calcular TCF

TCF = 0.6 + 0.01 * Σ (Peso * Valor) (Donde Valor es un número del 0 al 5)

Significado de los valores

- 0: No presente o sin influencia,
- 1: Influencia incidental o presencia incidental
- 2: Influencia moderada o presencia moderada
- 3: Influencia media o presencia media
- 4: Influencia significativa o presencia significativa
- 5: Fuerte influencia o fuerte presencia

Factor	Descripción	Peso	Valor	Comentario	Σ(Peso*Valor)
T1	Sistema distribuido	2	3	El sistema es centralizado	6
T2	Objetivos de performance o tiempo de	1	1	La velocidad es limitada por las entradas	1

	respuesta			provistas por el usuario	
T3	Eficiencia del usuario final	1	1	Escasas restricciones de eficiencia	1
T4	Procesamiento interno complejo	1	0	No hay cálculos complejos	0
T5	El código debe ser reutilizable	1	0	No se requiere que el código sea reutilizable	0
T6	Facilidad de instalación	0.5	1	Escasos requerimientos de facilidad de instalación	0.5
T7	Facilidad de uso	0.5	3	Normal	1.5
T8	Portabilidad	2	0	No se requiere que el sistema sea portable	0
T9	Facilidad de cambio	1	3	Se requiere un costo moderado de mantenimiento	3
T10	Concurrencia	1	0	No hay concurrencia	0
T11	Incluye objetivos especiales de seguridad	1	3	Seguridad normal	3
T12	Provee acceso directo a terceras partes	1	5	Los usuarios web tienen acceso directo	5
T13	Se requieren facilidades especiales de entrenamiento a los usuarios	1	1	Pocos usuarios internos, sistema fácil de usar.	1

Total		16
--------------	--	-----------

Por tanto:

$$TCF = 0.6 + 0.01 * 16$$

$$TCF = 0.76$$

2dº- Calcular EF

$$EF = 1.4 - 0.03 * \Sigma (\text{Peso} * \text{Valor}) \text{ (Donde Valor es un número del 0 al 5)}$$

Factor	Descripción	Peso	Valor	Comentario	$\Sigma(\text{Peso} * \text{Valor})$
E1	Familiaridad con el modelo de proyecto utilizado	1.5	4	El grupo está bastante familiarizado con el modelo	6
E2	Experiencia en la aplicación	0.5	3	La mayoría del grupo ha trabajado mucho tiempo en ésta aplicación	1.5
E3	Experiencia en orientación a objetos	1	4	La mayoría del grupo programa en Objetos	4
E4	Capacidad del analista líder	0.5	3	Somos diplomantes	1.5
E5	Motivación	1	5	El grupo está altamente motivado	5
E6	Estabilidad de	2	2	Se esperan	4

	los requerimientos			cambios	
E7	Personal part-time	-1	0	Todo el grupo es full-time	0
E8	Dificultad del lenguaje de programación	-1	4	Se usará lenguaje PHP	-4
Total					18

Por tanto:

$$EF = 1.4 - 0.03 * 18$$

$$EF = 0.86$$

$$\text{Luego UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

$$\text{UCP} = 21 * 0.76 * 0.86$$

$$\text{UCP} = 13.725.$$

$$\text{UCP} \approx 14.$$

Paso 3. Calcular esfuerzo de FT Implementación

$$E = \text{UCP} * \text{CF}$$

Donde:

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: factor de conversión.

1rº- Para calcular CF

CF = 20 horas-hombre (si $Total_{EF} \leq 2$)

CF = 28 horas-hombre (si $Total_{EF} = 3$ ó $Total_{EF} = 4$)

CF = abandonar o cambiar proyecto (si $Total_{EF} \geq 5$)

$Total_{EF} = Cant\ EF < 3$ (entre E1 –E6) + $Cant\ EF > 3$ (entre E7, E8)

$Total_{EF} = 2 + 0$

$Total_{EF} = 2$

Por tanto:

CF = 20 horas-hombre.

$E = UCP * CF$

$E = 14 * 2 = 28$ horas hombre

Paso 4. Calcular esfuerzo de todo el proyecto

Esfuerzo del proyecto

Actividad	% esfuerzo	Valor Esfuerzo
Diseño	45%	29 horas-hombre
Implementación	40%	31 horas-hombre
Prueba	15%	10 horas-hombre
Esfuerzo Total	100%	70 horas-hombre

5.3 Beneficios tangibles e intangibles

El Servicio Web de Nomencladores de la ONRM, es un sistema adaptable a cualquier otro sistema con características similares, sobre todo en instituciones que tengan que ver con la

geología en el país, lo que lo hace hasta cierto punto comerciable, aunque no fue diseñado para estos fines.

Por otra parte su establecimiento en la ONRM garantizará una mejor organización de los nomencladores a las distintas aplicaciones que se están desarrollando para la oficina, además de mejorar considerablemente la calidad del trabajo y el tiempo en la búsqueda de cualquier información, como por ejemplo un nomenclador para cualquier aplicación determinada, debido a que antes de la existencia del servicio web se hacía de forma manual y era un proceso muy engorroso.

Otros beneficios intangibles a obtener son:

- Eliminación de la redundancia de datos debido a la creación de una base de datos a la cual se conecta el servicio web.
- Centralización de toda la información necesaria para la empresa.

5.4 Análisis de costo y beneficio

El Servicio Web fue creado con un conjunto de herramientas de código abierto y libre, por lo que este no reclamó una gran inversión a la hora de realizar el producto. Esta aplicación está orientada a las distintas aplicaciones que usarán los nomencladores para su trabajo, por lo que no es difícil de familiarizarse con ella, y no reporta grandes costos para su uso.

Lo más costoso radica en el equipamiento a instalar en la oficina, como por ejemplo un servidor de bases de datos, así como la conectividad de la red local para el trabajo con el servicio web. Este sistema podría ser comercializado debido a que fue desarrollado con herramientas libres que no requieren pagos de licencias para vender productos creados en estas.

Por todo lo antes expuesto llegamos a la conclusión de que el sistema es factible.

5.5 Conclusiones parciales

Después de haber realizado un estudio exhaustivo del costo que tendrá la aplicación y utilizando diferentes herramientas como el COCOMO II para determinar con exactitud el costo que tendrá el sistema; se llega a la conclusión que el mismo será factible debido a los beneficios que trae (Centralización de los Nomencladores en la ONRM), además de que se realizó con un reducido esfuerzo horas- hombre de sus desarrolladores y un mínimo costo.

CONCLUSIONES GENERALES

Con el desarrollo de la capa de Servicios Web, a partir de la investigación realizada para la elaboración de la misma; guiándose por el Proceso Unificado de Desarrollo de Software (RUP) como metodología de desarrollo, para conquistar una mejor comprensión de los requisitos de la aplicación y formalización de dichos requisitos se llega a las conclusiones siguientes:

1. Se desarrolló la Capa de Servicios Web de Nomencladores que:
 - 1.1. Permite aportar interoperabilidad entre las aplicaciones de software, independientemente de sus propiedades o de las plataformas sobre las que se instalen.
 - 1.2. Fomenta los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
 - 1.3. Contiene una ayuda que estará disponible a toda hora y que facilita el consumo de cada uno de los servicios por los usuarios y que estará disponible a toda hora.
 - 1.4. Mejora la calidad del trabajo con los nomencladores para cada una de las aplicaciones que consumirán de los servicios prestados.
2. El empleo de los métodos científicos proporcionó conocer el estado del objeto de estudio.
3. La encuesta aplicada al personal de ONRM y a los desarrolladores de las aplicaciones para dicha empresa para apreciar el problema, confirmó la necesidad de perfeccionar el proceso de gestión de los descriptores de la información (nomencladores).

RECOMENDACIONES

- A partir de los resultados o beneficios que proporciona este trabajo de diploma, se proponen las siguientes recomendaciones:
- Que este material junto al manual de usuario sean tomados como materiales de consulta por el personal (dígase técnicos o profesionales) que vayan a desarrollar un sistema similar.
- Que consulten las referencias bibliográficas, toda aquellas personas que se dirijan a este trabajo para obtener algún conocimiento o tenerlo como apoyo para otras investigaciones.
- Poner en marcha la Capa de Servicios web construida para perfeccionar el trabajo con lo nomencladores en la Oficina Nacional de Recursos Minerales.
- Realizar un sistema que posibilite el proceso de mantenimiento y actualización de forma periódica y continua, logrando de esta forma que se mantenga la confidencialidad y funcionamiento insuperable de los servicios y de la información que se gestiona mediante el mismo.

REFERENCIAS BIBLIOGRAFICAS

1. Charles, L., H. Joseph, and M. Carl, *MARKETING INTERNACIONAL - SEPTIMA EDICIÓN*. Septiembre 2004. 666.
2. Thompson, I. *Definición de Servicios*. Agosto 2006. [cited Noviembre 2007; Available from: www.promonegocios.net].
3. Salazar Barriga, O. *Glosario de términos sobre Administración Pública*. [cited; Available from: <http://www.unmsm.edu.pe/ogp/ARCHIVOS/Glosario/inds.htm#11>].
4. Morales, J.H.G. *Web Services mediante WebServiceConnector de Flash*. 17 de Diciembre del 2004 [cited 2007; Available from: <http://www.cristalab.com/tutoriales/77/web-services-mediante-webserviceconnector-de-flash>].
5. Gaytán, I. *Introducción a Web Services con .NET y FLEX*. [cited 2007; Available from: <http://www.madeinflex.com/2006/10/30/introduccion-a-web-services-con-net-y-flex/>].
6. Castejón Garrido, J.S. *Arquitectura y diseño de sistemas web modernos* 2004 [cited; Available from: http://www.cii-murcia.es/informas/ene05/articulos/Arquitectura_y_diseño_de_sistemas_web_modernos.html].
7. Ignacio Cabanes, J. *Diccionario Básico de Informática*. [cited; Available from: <http://usuarios.lycos.es/Resve/diccioninform.htm>].
8. Hernández Orallo, E. *El Lenguaje Unificado de Modelado (UML)* 2001 [cited 2008; Available from: <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>].
9. Larman, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos* 1999 [cited 2008; Available from: <http://bibliodoc.uci.cu/pdf/reg00061.pdf>].
10. Rumbaugh, J., I. Jacobson, and G. Booch. *El Proceso Unificado de Desarrollo de Software*. 2000 [cited 2007; Available from: <http://bibliodoc.uci.cu/pdf/reg00060.pdf>].
11. Molpeceres, A. *Procesos de desarrollo: RUP, XP y FDD*. 2003 [cited 2007; Available from: <http://www.javahispano.org/articles.article.action?id=76>].
12. Pressman, R.S. *Ingeniería del software. Un enfoque práctico*. 2002 [cited; 5ta ed.: [Available from: <http://bibliodoc.uci.cu/pdf/reg02689.pdf>].
13. Rumbaugh, J., I. Jacobson, and G. Booch, *El Lenguaje Unificado de Modelado. Manual de referencia*. 2000, Madrid: Pearson Educación.
14. Diego, S. *Slideshare*. 2007 [cited 2008; Available from: http://www.slideshare.net/vivi_jocadi/rational-rose/].

15. Visual Paradigm. *Visual Paradigm for UML*. 2007 [cited 2007; Available from: <http://www.visual-paradigm.com/product/vpuml/>].
16. Rodas Hinostroza, R. *Características de PHP*. 2007 [cited; Available from: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>].
17. Lacaci, A.T. *Servidor HTTP AOLserver*. 24/09/1999 [cited 2008; Available from: <http://bulma.net/body.phtml?nIdNoticia=105>].
18. www.roxen.com. *Roxen WebServer* 2008 [cited 2008; Available from: <http://www.roxen.com/products/cms/webserver/>].
19. Alvarez, S. *Sistema gestor de base de datos*. 2006 [cited 2007; Available from: <http://www.desarrolloweb.com/faq/504.php>].
20. Dorantes González, V.H., F. Magariños Lamas, and J.N. Jury Fabre. *Curso de Bases de Datos y PostgreSQL*. 2001 [cited; Available from: <http://es.tldp.org/Tutoriales/NOTAS-CURSO-BBDD/notas-curso-BD/notas-curso-BD.html>].
21. Gilfillan, I., *La Biblia de MySQL* 2003, Madrid: Anaya Multimedia.
22. Serrano, A. *Ventajas de PostgreSQL*. 2005-12-26 [cited 2008; Available from: http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html].

GLOSARIO DE TÉRMINOS

MINBAS: Ministerio de la Industria Básica.

ONRM: Oficina Nacional de Recursos Minerales.

PINCG: Proyecto de Informatización Nacional del Conocimiento Geológico.

RUP: Proceso Unificado de Desarrollo.

UML: Lenguaje Unificado de Modelado.

RF: Requisitos Funcionales.

RNF: Requisitos no Funcionales.

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos, asociación estadounidense dedicada a la estandarización. Es una asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros de telecomunicaciones, ingenieros electrónicos, ingenieros en informática y otros (*The Institute of Electrical and Electronics Engineers*).

CU: Caso de Uso.

CUS: Caso de Uso del Sistema.

HTTP: Protocolo para transferir archivos o documentos hipertexto a través de la red. (Hyper Text Transmisión protocolo).

Identificación: Técnica utilizada en programación con el objetivo de mantener claridad en el código escrito por el programador y que resulte fácil su comprensión por otras personas.

UNIX: Sistema operativo portable, flexible, potente, con entorno programable, multiusuario y multitarea, muy difundido.

PHP: Es acrónimo de Hypertext Pre-processor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación (Personal Home Page).

PostgreSQL: Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD

MySQL: Sistema de administración de base de datos, es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto.

WWW: Es el universo de información accesible a través de Internet, una fuente inagotable del conocimiento humano. Es un sistema de información global, interactiva, dinámica, distribuida, gráfica, basada en hipertexto, con plataforma de enlaces cruzados, que se ejecutan en Internet (Word Wide Web)

COBOL: Lenguaje común Orientado a Negocios (COmmon Business- Oriented Language).

Basic: Es una familia de lenguajes de programación.

Pascal: Lenguaje de programación estructurado de alto nivel y propósito general, es un lenguaje orientado a procedimientos con usos varios, desarrollado por Niklaus Wirth hace ya más de 20 años; profesor del Instituto tecnológico de Zurich, Suiza.

Java: Lenguaje de programación orientado a objetos con el que se puede realizar cualquier tipo de programa, es un lenguaje muy extendido, es un lenguaje independiente de la plataforma, es compilado en un bytecode que es interpretado desarrollado por la compañía Sun Microsystems a principios de los 90.

C#: Lenguaje de programación orientado a objetos, evolución del lenguaje C++, desarrollado por Microsoft.

XML: Extensible Markup Language (Lenguaje extensible de etiquetas) Es un meta-lenguaje que permite definir lenguajes de marcado adecuado a usos determinados. Se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación) para intercambio de información estructurada entre diferentes plataformas.

HTML: Es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el estándar de las páginas Web. (Hyper Text Markup).

Ajax: Unión de varias tecnologías que juntas puede lograr cosas realmente impresionantes. Es el acrónimo para Asynchronous JavaScript + XML y su concepto es cargar y renderizar una página

XP: Programación Externa.

FDD: Desarrollo Guiado por Funcionalidad.

CASE: Ingeniería del software asistida por computadora.

CORBA: Es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos. (Common Object Request Broker Architecture).

IDL: Lenguaje de especificación de interfaces que se usa como parte de la tecnología CORBA. Ofrece la sintaxis necesaria para definir los métodos que queremos invocar remotamente. (Interface Definition Language).

Rational Rose: Herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto.

GNU: Conjunto de programas desarrollados por miembros de la Fundación por el Software Libre, son de uso gratuito (FSF- Free Software Foundation).

GNU/LINUX: Es un sistema operativo, es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo. Es multitarea, multiusuario, multiplataforma y multiprocesador.

GPL: Es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software Libre (General Public License).

Distribución: Es un conjunto de aplicaciones reunidas que permiten brindar mejoras para instalar fácilmente un sistema Linux.

UNIX: Sistema operativo portable, multitarea y multiusuario.

LAN: Red de Área Local.

WAN: Red de Área Amplia.

SGBD: Sistema Gestor de Bases de Datos.

BD: Base de Datos.

DBMS: Es un Sistema Gestor de Bases de Datos, un tipo de software muy específico, es equivalente al término SGBD (Data Base Management System).

ANSI: Principal organización encargada de promover el desarrollo de estándares tecnológicos en Estados Unidos (EL Instituto Nacional Estadounidense de Estándares), es miembro de la Organización Internacional para la Estandarización (ISO) y de la Comisión Electrotécnica Internacional (internacional Electrotechnical Commission, IEC).

SQL: Lenguaje declarativo de acceso a bases de datos relacionales que permiten especificar diversos tipos de operaciones sobre las mismas (Structured Query Language).

Oracle: Es un sistema de administración de base de datos, es una potente herramienta cliente/servidor para la gestión de bases de datos.

Apache: Proyecto nacido para crear un servidor Web estable, fiable y veloz para plataformas Unix.

PostgreSQL: Servidor de Base de Datos relacional libre, liberado bajo la licencia BSD, es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

OOP: Programación Orientada a Objetos.

Office: Es la suite ofimática por Microsoft y la más usada en la actualidad. Funciona bajo los sistemas operativos Microsoft Windows y Apple Mac OS.

RREE: Rational Rose Enterprise Edition.

COCOMO: Modelo Constructivo de Coste, permite realizar estimaciones y planificaciones de proyectos de sistemas de información.

WSDL: Lenguaje de Descripción de los Servicios Web. (Web Services Description Language).

BPEL: Lenguaje de Ejecución de Procesos de Negocio (Business Process Execution Language).

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

UCP: Puntos de Casos de Uso ajustados.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

E: esfuerzo estimado en horas-hombre.

CF: factor de conversión

ANEXOS

ANEXO 1 Resultados de los Métodos Teóricos utilizados para la investigación

Encuesta sobre el flujo de información referente a los nomencladores.

Oficina Nacional de Recursos Minerales y Universidad de las Ciencias Informáticas

Esta encuesta tiene como objetivo principal conocer cómo se realiza el manejo de información referente a los nomencladores con vistas a construir un Servicio Web que agilice dichos procesos.

Nombre de el Departamento: _____

Cargo que ocupa: _____

1. Tipos de nomencladores con lo se trabaja en el departamento _____
2. Describa, con sus palabras, las funciones que realizan con dicho Nomencladores.
3. ¿Quién le suministra la información de los nomencladores con la que trabaja?

4. ¿A quién usted envía la información con la que trabaja?
5. ¿En qué formato o aplicación maneja la información de los nomencladores?

____ Excel

____ Word

____ Correo electrónico

____ Otros. ¿Cuáles? _____

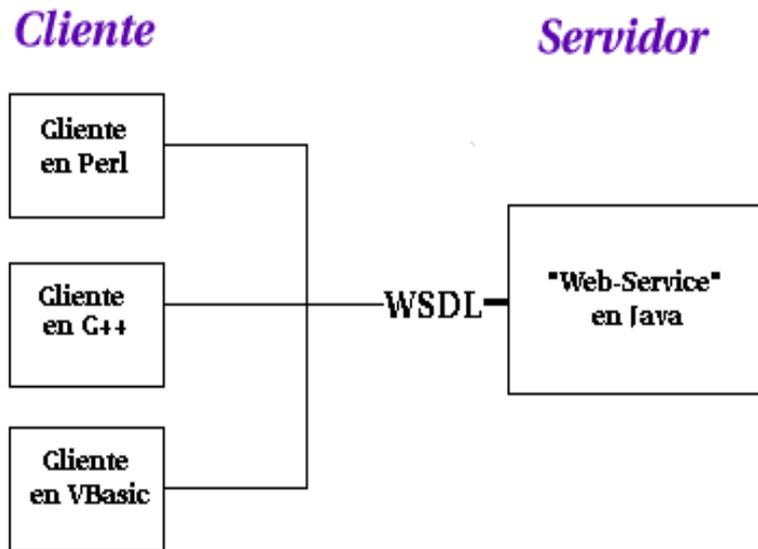
6. ¿Cree que es más fácil el trabajo con los nomencladores de manera manual o con una aplicación que le brinde toda la información necesaria?

Si____ No____

6.1. De marcar No argumente su respuesta.

7. ¿Tiene alguna recomendación para perfeccionar el flujo de información referente a los nomencladores

ANEXO 2 Lógica de los Servicios Web.



ANEXO 2 Representación de la Arquitectura SOA

