

**HERRAMIENTA INTERACTIVA PARA SIMULAR EL
PROCESO DEL DNS CON EL PROTOCOLO UDP**

**ADNERYS GALVIS MORENO
BLANCA MILENA MACHADO VALENZUELA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE CIENCIAS FISICOMECAICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
2004**

**HERRAMIENTA INTERACTIVA PARA SIMULAR EL
PROCESO DEL DNS CON EL PROTOCOLO UDP**

**ADNERYS GALVIS MORENO
BLANCA MILENA MACHADO VALENZUELA**

**Trabajo de grado presentado como
requisito para optar el título de Ingeniero
de Sistemas**

**HERNAN PORRAS DIAZ
Ing. Civil; M. Sc., Ph. D.**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE CIENCIAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
2004**

DEDICATORIA

A mis padres, hermanos y a todos aquellos que de una u otra forma han creído en mí, apoyándome a lo largo de este recorrido.

ADGAMO

A mis padres, a mi hermano y a toda mi familia que es el motor de mi vida.

Blanca Milena Machado Valenzuela

AGRADECIMIENTOS

Las autoras expresan sus agradecimientos a:

Dios por iluminar nuestros corazones y nuestra mente y permitirnos culminar exitosamente esta etapa de la vida.

Nuestras familias por el inmenso amor y apoyo que nos han ofrecido.

Nuestra amistad que nos permitió conformar un buen equipo de trabajo y lograr nuestra meta.

Ingeniera Judith Gómez Martínez por sus recomendaciones y su amistad incondicional.

Ingeniero Hernán Porras Díaz por su confianza y orientación.

CONTENIDO

	Pág.
INTRODUCCION	12
1. GENERALIDADES	14
1.1 ANTECEDENTES	14
1.2 OBJETIVOS DEL PROYECTO	15
1.2.1 Objetivo General	15
1.2.2 Objetivos Específicos	16
1.3 IMPACTO DEL PROYECTO	16
1.4 CONTENIDO DEL INFORME	17
2 AMBIENTE DE DESARROLLO DEL PROYECTO	18
2.1 SOFTWARE	18
2.2 HARDWARE	18
3 FUNDAMENTACION TEORICA	19
3.1 FUNDAMENTOS SISTEMA DE NOMBRES DE DOMINIO	19
3.1.1 Introducción	19
3.1.2 Definición	20
3.1.3 Estructura del DNS	21
3.1.3.1 Sintaxis del Nombre	21
3.1.3.2 Implementación de la Base de Datos	24
3.1.4 Transporte	35
3.1.5 Referencias	35
3.2 FUNDAMENTOS PROTOCOLO DATAGRAMAS DE USUARIO	36
3.2.1 Introducción	36
3.2.2 Definición	36
3.2.3 Formato de Datagrama UDP	37
3.2.4 Pseudocabecera IP	38
3.2.5 Interfaz IP	40

3.2.6 Encapsulación de UDP y Estratificación por Capas de Protocolos	40
3.2.7 Aplicaciones del UDP	41
3.2.8 Números de Puertos UDP Reservados y Disponibles	41
3.3 INFORMATICA EDUCATIVA	42
3.3.1 Definición de Software Educativo	42
3.3.1.1 Conceptualización	42
3.3.1.2 Características Esenciales de los Programas Educativos	43
3.3.2 Estructura Básica de los Programas Educativos	44
3.3.2.1 El Entorno de Comunicación o Interfaz	44
3.3.2.2 Las Bases de Datos	45
3.3.2.3 El Motor o Algoritmo	46
3.3.3 Clasificación de los Programas Educativos	47
3.3.3.1 Programas Tutoriales	49
3.3.3.2 Bases de Datos	51
3.3.3.3 Simuladores	52
3.3.3.4 Constructores	53
3.3.3.5 Programas Herramienta	54
3.3.4 Funciones del Software Educativo	57
3.4 INGENIERIA DEL SOFTWARE	61
3.4.1 El Proceso Unificado de Desarrollo de Software	62
3.4.1.1 Descripción del Proceso Unificado	62
3.4.1.2 Elementos del Proceso Unificado	64
3.4.2 Metodología de Desarrollo para la Herramienta DNS con UDP	67
3.4.3 El Lenguaje Unificado de Modelado	68
3.4.3.1 Elementos	69
3.4.3.2 Relaciones	71
3.4.3.3 Diagramas	71
3.5 FUNDAMENTACION TECNICA	73
3.5.1 Multimedia	73

3.5.1.1 Elementos Multimedia	74
3.5.1.2 Beneficios de una Presentación Multimedia	76
3.5.2 Bases de Datos	77
3.5.2.1 Ventajas de las Bases de Datos	77
3.5.2.2 Diseño de Base de Datos	78
3.6 SOFTWARE EN EL DESARROLLO DE LA HERRAMIENTA	79
3.6.1 Lenguaje de Programación JAVA	79
3.6.1.1 Introducción	79
3.6.1.2 Definición	80
3.6.1.3 Características de JAVA	81
3.6.1.4 Entorno de Desarrollo de JAVA	83
3.6.2 Interbase 6.02	84
3.6.3 Java DataBase Connectivity (JDBC)	84
3.6.3.1 Definición	84
3.6.3.2 Tipos de Drivers JDBC	86
3.6.3.3 Interclient	88
3.6.4 Lenguaje de Marcas de Hipertexto (HTML)	88
4 DESARROLLO DE LA HERRAMIENTA	91
4.1 ANALISIS DE REQUISITOS	91
4.1.1. Falencias en el proceso enseñanza-aprendizaje	91
4.1.2 Especificación de Requisitos	92
4.2 DISEÑO	93
4.2.1 Arquitectura del Sistema	94
4.2.2 Módulos del Sistema	94
4.2.2.1 Módulo de Estudio	95
4.2.2.2 Módulo de Guías	95
4.2.2.3 Módulo de Laboratorios	96
4.2.2.4 Módulo de Ayuda	105
4.3 IMPLEMENTACION	106

4.3.1 Descripción de la Base de Datos	106
4.3.2 Diagramas de Actividad	111
4.3.2.1 Modelo de Actividad. Módulo Laboratorios - Laboratorio 1	112
4.3.2.1 Modelo de Actividad. Módulo Laboratorios - Laboratorios 1 y 2	113
4.3.2.1 Modelo de Actividad. Módulo Laboratorios - Laboratorio 3	114
4.3.3 Funcionamiento de la Herramienta DNS con UDP	115
5 CONCLUSIONES	123
6 RECOMENDACIONES	125
BIBLIOGRAFIA	126

LISTA DE FIGURAS

	Pág.
Figura 1. Parte del espacio de dominios	23
Figura 2. Formato Mensaje DNS	28
Figura 3. Formato Encabezado-Campo Parámetros	29
Figura 4. Formato Sección "Question"	31
Figura 5. Dominio IN-ADDR.ARPA	34
Figura 6. Formato Datagrama UDP	37
Figura 7. Pseudocabecera IP	39
Figura 8. Lugar de UDP en las Capas de TCP/IP	40
Figura 9. Encapsulación del Datagrama UDP	41
Figura 10. Principios básicos – Casos de Uso	63
Figura 11. Principios básicos – Proceso iterativo e incremental	64
Figura 12. Elementos del proceso unificado- Flujos de trabajo en las cuatro etapas del ciclo	65
Figura 13. Elementos del proceso unificado – Hitos principales y secundarios	66
Figura 14. Arquitectura del Sistema	94
Figura 15. Diagrama Caso de Uso - Módulo de Estudio	95
Figura 16. Diagrama Caso de Uso - Módulo de Guías	96
Figura 17. Diagrama Caso de Uso – Módulo de Laboratorios	97
Figura 18. Diagrama de Caso de uso. Módulo Laboratorios- Laboratorio 1 – (Vista usuario)	99
Figura 19. Diagrama de Caso de uso. Módulo Laboratorios- Laboratorio 1 – (Vista software)	101
Figura 20. Diagrama de Caso de Uso. Módulo Laboratorios – Laboratorio 2- (Vista usuario)	102
Figura 21. Diagrama de Caso de Uso. Módulo Laboratorios –	

Laboratorio 2- (Vista software)	103
Figura 22. Diagrama de Caso de Uso. Módulo Laboratorios – Laboratorio 3- (Vista usuario)	104
Figura 23. Diagrama de Caso de Uso. Módulo Laboratorios – Laboratorio 3- (Vista software)	104
Figura 24. Diagrama Caso de Uso - Módulo de Guías	105
Figura 25. Modelo Entidad Relación	109
Figura 26. Diagrama de Actividad. Módulo Laboratorios- Laboratorio 1 – Ingreso	112
Figura 27. Diagrama de Actividad. Módulo Laboratorios- Laboratorios 1 y 2 – Desarrollo	113
Figura 28. Diagrama de Actividad. Módulo Laboratorios- Laboratorio 3 – Desarrollo	114
Figura 29. Pantalla inicial de la herramienta	115
Figura 30. Pantalla Principal – Laboratorio	116
Figura 31. Pantalla Iniciar Sesión – Laboratorio	117
Figura 32. Pantalla Usuario No Registrado – Laboratorio	118
Figura 33. Pantalla Inicio Laboratorio 2	119
Figura 34. Pantalla Inicio Laboratorio 3	120
Figura 35. Asistente para Respuesta - Laboratorio 3	120
Figura 36. Asistente para Respuesta – Laboratorio 3	121
Figura 37. Ayuda	122

LISTA DE TABLAS

	Pág.
Tabla 1. Tipos de registros de recursos y el contenido del campo datos según el tipo RR	26
Tabla 2. Números de puerto UDP Reservados y Disponibles	41

LISTA DE ANEXOS

	Pág.
Anexo A. Documentación pruebas.	128

TITLE*: **INTERACTIVE TOOL TO SIMULATE THE PROCESS OF THE DNS WITH THE UDP PROTOCOL**

AUTHORS: **ADNERYS GALVIS MORENO**
BLANCA MILENA MACHADO VALENZUELA **

KEY WORDS: **DNS, UDP, Database, Java, Interbase, Interclient, UML, Teaching and Learning.**

CONTENT:

With the development of this software, it pretends to support the process of teaching and learning, in the area of Telematics and Communications, offering to the student a better understanding and assimilation of the function of the Domain Name System with the User Datagram Protocol by practices included in the software. These practices refer to laboratories with its respective guides, that allow the user to interact with the tool. The methodology used for the elaboration of the tool was the Unified Process of Software Development.

The architecture of the tool includes the following work modules:

Study module: Facilitates information to the user about the topics related with the tool.

Guides module: Provides orientation to the user on the development of each one of the laboratories. In it finds specified factors like: objectives, procedure and range.

Laboratories module: Includes three different practices, each one of them specifies the activities that the user should carry out, which are validated through its development. This module is the most important because trough this lets the user interact with the tool in order to simulate processes related with the topic.

Help module: Provides information about the function of the tool and the development of the laboratories with examples detailed step by step.

The software was developed under the Java platform; the database (db) that emulates the DNS it was implemented in Interbase 6.02. For the connection of the bd with Java the driver JDBC Interclient 1.6 was used.

* Investigation work.

** Physical-mechanical sciences faculty, System and Informatics Engineering School, Director ing. Hernán Porras Díaz.

TITLE*: **INTERACTIVE TOOL TO SIMULATE THE PROCESS OF THE DNS WITH THE PROTOCOL UDP**

AUTHORS: **ADNERYS GALVIS MORENO**
BLANCA MILENA MACHADO VALENZUELA**

KEY WORDS: **DNS, UDP, Database, Java, Interbase, Interclient.**

CONTENT:

With the development of the software seeks to lean on the process teaching-learning, in the area of Telematic and Communications, offering the student a better understanding and assimilation of the operation of the Domain Name System with the User Datagram Protocol by means of practices included in the software. These practices refer to laboratories with their respective guides that allow to the user the interaction with the tool. The methodology used for the elaboration of the tool was the Unified Process of Development of Software.

The architecture of the tool understands the following work modules:

Study module, facilitates information to the user about the topics related with the tool.
Guides module, provides orientation to the user on the development of each one of the laboratories. In him factors are specified like: objectives, procedure and reach.
Laboratories module, understands three different practices, each one of them specifies the activities that the user should carry out, which are validated through his development. This module is since the most important because it let to the user the interaction with the tool in order to simulate processes related with the topic.
Help module, provides information on the operation of the tool and the development of the laboratories with examples detailed step to step.

The software was developed under the platform Java, the database (db) that emulates the DNS, it was implemented in Interbase 6.02. For the connection of the bd with Java the driver JDBC Interclient 1.6. was used.

* Investigation work.

** Physical-mechanical sciences faculty, System and Informatics Engineering School, Director ing. Hernán Porras Díaz.

INTRODUCCION

El aprendizaje es un proceso activo en el que el sujeto tiene que realizar una serie de actividades para asimilar los contenidos informativos que recibe. Según repita, reproduzca o relacione los conocimientos, realizará un aprendizaje repetitivo, reproductivo o significativo.

La elaboración de un software educativo siempre parte de una idea inicial que tiene como fin favorecer los procesos de enseñanza/aprendizaje y que va tomando forma poco a poco; una idea que configura unas actividades atractivas para el sujeto que potencialmente pueden facilitar la consecución de unos determinados objetivos educativos. Es conveniente que estas actividades estén en consonancia con las tendencias pedagógicas actuales, para que su uso en las aulas y demás entornos educativos provoque un cambio metodológico en este sentido.

La tendencia de las herramientas educativas es evitar la simple memorización y presentar *entornos heurísticos centrados en los sujetos* que tengan en cuenta las teorías *constructivistas* y los principios del *aprendizaje significativo* donde además de comprender los contenidos puedan investigar y buscar nuevas relaciones. Así el sujeto se sentirá constructor de su aprendizaje mediante la interacción con el entorno que le proporciona el software y a través de la reorganización de sus esquemas de conocimiento.

La herramienta DNS-UDP, ofrece al usuario la posibilidad de descubrir nuevos conceptos y afianzar conocimientos en el área de los sistemas distribuidos, mediante prácticas sencillas que representan procesos

importantes que aunque son transparentes al usuario, hacen parte de su diario interactuar con el mundo de las redes, específicamente Internet.

Para finalizar, se debe destacar que el proceso de elaboración del software educativo no es un proceso lineal, sino iterativo: en determinados momentos de la realización se comprueba el funcionamiento, el resultado, se evalúa el producto... y frecuentemente se detecta la conveniencia de introducir cambios a fin de cumplir a cabalidad con los objetivos propuestos.

1. GENERALIDADES

1.1 ANTECEDENTES

Hace ya tiempo que, en mayor o menor medida, los ordenadores han entrado en las escuelas, en los institutos, y desde luego en las universidades. En muchos casos con ellos se practican conocimientos específicamente informáticos, pero cada vez más se usan sobre todo como herramienta para enseñar otro tipo de disciplinas, o simplemente para permitir a los usuarios practicar con herramientas genéricas (procesadores de texto, hojas de cálculo, consulta Web, etc.).

El software educativo, utiliza entre otros, ejercicios y sesiones de preguntas y respuestas para presentar un tema y verificar su comprensión por parte del usuario, permitiéndole también estudiar a su propio ritmo, es por esto que los temas son diversos y la complejidad va de acuerdo al nivel del usuario.

En la medida en la que han ido apareciendo distintas expresiones para referirse a la utilización del ordenador en la enseñanza, se puede tratar de explicar la evolución que ha experimentado esta herramienta didáctica. Si el software educativo respondía en principio a un planteamiento rígido, que hacía que todos los usuarios tuviesen que seguir exactamente la misma secuencia de aprendizaje, hoy día existen diversos productos que permiten adaptarse a las necesidades educativas de los usuarios. Los primeros eran programas simples, con colecciones de preguntas elaboradas bajo el esquema *estímulo-respuesta*, las preguntas eran cerradas y la respuesta se daba como errónea siempre que no coincidiese con la que tenía archivada el propio programa, lo que daba pobres resultados en un aprendizaje que

pretendiese ser conceptual. Los programas actuales, elaborados en muchos casos utilizando lenguaje experto de programación, no sólo admiten respuestas más abiertas, sino que su secuenciación es tal que permiten distintos desarrollos en función del aprendizaje del usuario; en determinados momentos del programa se introducen elementos de evaluación cuyo resultado orienta al usuario a pasar a una lección siguiente o a repasar una anterior, proponiéndole, incluso, ejercicios de refuerzo. Actualmente una de las principales ventajas de las computadoras es la cantidad de información que ofrecen mediante la conexión en red a una gran variedad de bases de datos, esto ha fomentado el empleo del ordenador como fuente de conocimiento, muy superior a las tradicionales fuentes en papel. Los materiales que aportan la base documental suelen ser productos multimedia, cuyo soporte estará en elementos que permiten una gran capacidad de almacenamiento, como CD-ROM o DVD, o en servidores de Internet; esto último tiene también la ventaja de la inmediatez, con lo que se pueden utilizar como recursos didácticos materiales de gran actualidad que, en un planteamiento clásico de la enseñanza, no llegarían a los usuarios hasta varios años después de que el suceso se hubiese producido.

1.2 OBJETIVOS DEL PROYECTO

1.2.1 Objetivo General

Brindar al estudiante una herramienta de apoyo que le permita comprender el proceso de conexión al Sistema de Nombres de Dominio (DNS) y el mecanismo de resolución de nombres de dominio, utilizando el Protocolo de Datagramas de Usuario (UDP).

1.2.2 Objetivos Específicos

- Proveer información en línea al usuario acerca de los conceptos básicos correspondientes a UDP y DNS.
- Facilitar al usuario la comprensión del funcionamiento del DNS con UDP mediante la simulación de dicho proceso, considerando diferentes alternativas de solución¹.
- Proporcionar una guía² que incluya laboratorios, permitiendo al usuario la realización de prácticas, de manera que pueda asimilar positivamente la funcionalidad de la herramienta.

1.3 IMPACTO DEL PROYECTO

La herramienta facilitará la comprensión del funcionamiento del Sistema de Nombres de Dominio con el Protocolo de Datagramas de Usuario, mediante prácticas que estimulan al estudiante a explorar a fondo el tema, generando interés por conocer otros conceptos relacionados e indagar en nuevas áreas.

Por otro lado, la herramienta puede motivar a los estudiantes a realizar proyectos similares basados en otras plataformas que ayuden a mejorar el proceso de aprendizaje.

¹ Alternativas de solución: búsqueda recursiva y búsqueda iterativa.

² Guía de laboratorios: orienta al estudiante en la comprensión de conceptos y procesos (DNS con UDP), mediante prácticas específicas.

1.4 CONTENIDO DEL INFORME

El informe se encuentra organizado en los siguientes capítulos que facilitan el seguimiento del desarrollo del proyecto.

Capítulo 1. Hace referencia a las generalidades del proyecto, el cual describe sus antecedentes, objetivos e impacto.

Capítulo 2. En este capítulo se hace mención al ambiente de desarrollo del proyecto, describiendo los aspectos más importantes a nivel de hardware y software.

Capítulo 3. Muestra el contenido teórico general sobre el cual se tomaron las bases necesarias para el desarrollo del proyecto.

Capítulo 4. Describe la metodología utilizada para el desarrollo de las actividades a fin de alcanzar los objetivos del proyecto.

Capítulo 5. Presenta las conclusiones referentes a las actividades realizadas durante el proyecto.

Capítulo 6. Enuncia algunas recomendaciones y consideraciones de los autores para futuros esfuerzos de desarrollo.

2 AMBIENTE DE DESARROLLO DEL PROYECTO

2.1 SOFTWARE

- El sistema operativo requerido es Microsoft Windows 98.
- El software de desarrollo utilizado es Java 2 SDK Standard Edition v1.3.1.
- La herramienta para el desarrollo de bases de datos cliente-servidor es Interbase 6.02.
- El driver JDBC para conexión a bases de datos desde java es Interclient para Interbase 6.02.

2.2 HARDWARE

A nivel de hardware, el equipo utilizado para el desarrollo del proyecto tiene las siguientes características.

- Procesador Pentium IV de 3 Ghz o equivalente en otro fabricante.
- Memoria RAM 256 Mb
- Disco Duro 40 Gb
- Tarjeta de red

3 FUNDAMENTACIÓN TEÓRICA

3.1 FUNDAMENTOS SISTEMA DE NOMBRES DE DOMINIO

3.1.1 Introducción

En la actualidad pocas veces hacemos referencia a los hosts, direcciones de correo y otros recursos por su dirección IP, es por esto que llegado un tiempo en que Internet empezó a crecer desmesuradamente, las aplicaciones permitieron el uso de un sistema de nombres, en vez de utilizar las direcciones de los hosts. Esto introdujo problemas. En primer lugar, hacía falta asegurar que no hubiese dos computadores con el mismo nombre en Internet. Además era necesario proveer una forma de convertir los nombres en direcciones numéricas.

Como primera opción, en los tiempos de ARPANET, simplemente había un archivo, *hosts.txt*, en el que se listaban todos los hosts y sus direcciones IP correspondientes; cada noche, todos los hosts obtenían este archivo de la instalación en la que se mantenía. Pero este enfoque solo funcionaba en una red con unas cuantas máquinas, sin embargo, cuando la red empezó a crecer, la solución no resultaba tan efectiva. Por una parte, el archivo se hubiese vuelto demasiado grande, pero un problema aún más importante era que ocurrirían conflictos constantes con los nombres de los hosts a menos de que estos se administraran centralmente. Con la finalidad de resolver este problema se inventó DNS (Sistema de Nombres de Dominio), cuya función es convertir nombres a direcciones IP y viceversa, ofreciendo Ventajas como:

- *Desaparece la carga excesiva en la red y en los hosts:* ahora la información está distribuida por toda la red, al tratarse de una BD distribuida.
- *No hay Duplicidad de Nombres:* el problema se elimina debido a la existencia de dominios controlados por un único administrador. Puede haber nombres iguales pero en dominios diferentes.
- *Consistencia de la Información:* ahora la información que está distribuida es actualizada automáticamente sin intervención de ningún administrador.

DNS es un protocolo de aplicación y usa tanto UDP como TCP. Los clientes solicitan a los servidores de DNS sus consultas por medio de UDP para hacer más rápida la comunicación y utilizan TCP sólo en caso de que llegara a ocurrir una respuesta truncada.

3.1.2 Definición

El Sistema de Nombres de Dominios (DNS) es un conjunto de protocolos y servicios sobre una red TCP/IP, que permite a los usuarios de red utilizar nombres jerárquicos sencillos para comunicarse con otros equipos, en lugar de memorizar y usar sus direcciones IP. Este sistema es muy usado en Internet y en muchas de las redes privadas actuales.

El DNS es un sistema distribuido, jerárquico, replicado y tolerante a fallas. El punto central se basa en un árbol conocido como *espacio de nombres de dominio* que define la jerarquía entre los dominios y los subdominios. En un nombre de dominio, la jerarquía se lee de derecha a izquierda. Por ejemplo, en *dcc.uchile.cl*, el dominio más alto es *cl*. Para que exista una raíz del árbol, se

puede ver como si existiera un punto al final del nombre: *dcc.uchile.cl.*, y todos los dominios están bajo esa raíz (también llamada "punto").

DNS es un protocolo cliente-servidor que usa el puerto 53 UDP. El cliente, en este protocolo, no está muy bien definido, no es un programa concreto, sino que muchos programas hacen uso de la librería que se conecta al servidor. Por ejemplo, ping, whois, nslookup, el navegador... etc. para realizar consultas. En general, son llamados "*resolvers*" o "*resolvedores*".

3.1.3 Estructura del DNS

En el DNS participan dos elementos independientes: El primero especifica la sintaxis del nombre y el segundo especifica la implementación de la base de datos.

3.1.3.1 Sintaxis del Nombre. Un nombre de dominio está compuesto por etiquetas máximo de 63 caracteres separadas por puntos, deben empezar con una letra, terminar con una letra o dígito y tener como caracteres internos sólo letras dígitos o guiones. Cada dominio es un índice en la BD del DNS. En los nombres no se distinguen mayúsculas de minúsculas.

Un nombre de dominio está organizado en forma jerárquica o por niveles, comenzando el nivel superior por la derecha, por ejemplo: *robotica.uv.es* tiene 3 etiquetas, siendo el dominio de nivel superior "es.", dominio de segundo nivel "*uv.es.*" y dominio de nivel inferior "*robotica.uv.es.*". Este nombre además de un dominio puede representar un host. Los nombres de dominio pueden ser absolutos o relativos. Un nombre absoluto terminan con "." (por ejemplo: "*uv.es.*") , un nombre relativo no, necesitando saber el contexto del dominio superior para determinar de manera única su significado verdadero.

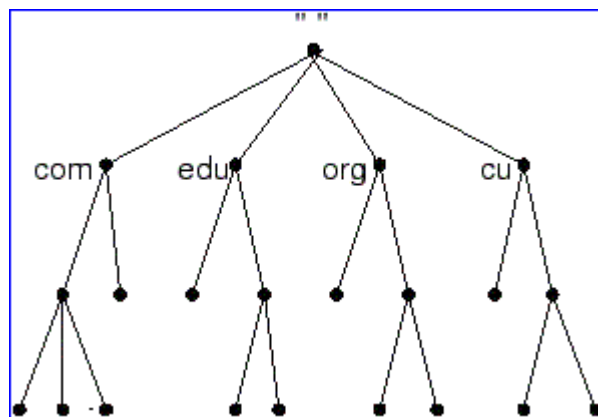
La estructura de la base de datos del DNS posee una forma jerárquica que permite administrar nombres dando diferentes grupos de responsabilidad para subconjuntos de nombres. Cada nivel en este sistema se llama dominio, este va separado por puntos: dom4.dom3.dom2.dom1. Un *dominio* es simplemente un subárbol del espacio de nombres. El nombre de un dominio es el nombre del nodo raíz correspondiente. Un dominio agrupa un conjunto de hosts y/o subdominios que se relacionan de acuerdo a cierto criterio, ya sea geográfico u organizacional. En el DNS cada dominio es administrado por una organización o empresa determinada. Esta puede decidir dividir él o los dominios que administra en subdominios, así como asignar la administración de éstos a otras entidades. Cada dominio puede contener tanto subdominios como hosts independientes y se define desde la hoja del árbol hasta la raíz, puede haber nombres duplicados en dominios diferentes. El número de dominios de una misma dirección es variable pero no suele sobrepasar los cinco, siendo frecuente usar menos.

El DNS en la actualidad sigue ciertos patrones en cuanto a su organización. Esta se basa en niveles de acuerdo a la posición del dominio. El nivel superior o primer nivel lo forman aquellos dominios descendientes del nodo raíz, estos están divididos en dos grupos:

- Los *dominios genéricos* (también denominados internacionales o globales) están organizados conceptualmente y son los dominios básicos en Internet. Estos dominios son los que más popularidad han adquirido. Los fundamentales se listan a continuación:
 - *com*: Agrupa a organizaciones comerciales.
 - *edu*: Reúne a organizaciones de propósitos educacionales.

- *net*: Agrupa a organizaciones dedicadas al desarrollo de las redes.
 - *org*: Reúne a organizaciones no comerciales.
 - *gov*: Agrupa a organizaciones gubernamentales.
 - *int*: Reúne organizaciones internacionales.
 - *mil*: Agrupa organizaciones militares.
 - *arpa*: Para organizaciones que se refieren específicamente a Internet.
-
- Los *dominios territoriales* (también denominados geográficos o de países) son los dominios mantenidos por cada país. Estos son utilizados por las organizaciones y empresas que desean establecerse en Internet o que quieren proteger la identidad de su marca o su nombre comercial en un país concreto. Los dominios territoriales tienen dos letras, existen aproximadamente 230 dominios territoriales, ejemplos de ellos son:
 - *co* para Colombia.
 - *es* para España.
 - *uk* para el Reino Unido.
 - *au* para Australia, etc.

Figura 1. Parte del espacio de dominios.



3.1.3.2 Implementación de la Base de Datos. El DNS usa el concepto de espacio de nombres distribuido. Los nombres de dominio se agrupan en zonas de autoridad, o más comúnmente, zonas. Una *zona* es un archivo físico para almacenar y administrar un conjunto de registros del espacio de nombres, en cada una de ellas, uno o más hosts tienen la tarea de mantener una base de datos de nombres de dominio y direcciones IP y de suministrar la función de servidor para los clientes que deseen traducir nombres de dominio a direcciones IP y viceversa. Estos servidores de nombres locales se interconectan lógicamente en un árbol jerárquico de dominios. Cada zona contiene una parte del árbol o subárbol y los nombres de esa zona se administran con independencia de los de otras zonas. La autoridad sobre zonas se delega en los servidores de nombres. Normalmente, los servidores de nombres que tienen autoridad en la zona tendrán nombres de dominio de la misma, aunque no es imprescindible. En los puntos en los que un dominio contiene un subárbol que cae en una zona diferente, se dice que el servidor o servidores de nombres con autoridad sobre el dominio superior delegan autoridad al servidor o servidores de nombres con autoridad sobre los subdominios. Los servidores de nombres también pueden delegar autoridad en sí mismos; en este caso, el espacio de nombres sigue dividido en zonas, pero la autoridad para ambas las ejerce el mismo servidor. La división por zonas se realiza utilizando registros de recurso guardados en el DNS: Registro SOA (define el inicio de una zona), Registro NS (Marca el fin de una zona iniciada por un SOA y apunta a un servidor de nombres con autoridad sobre la zona siguiente).

En este contexto, el comienzo de un dominio está más cerca a la raíz del árbol que a sus terminaciones. En la raíz, no puede haber servidores de nombres superiores para delegar autoridad: la autoridad para la raíz se deposita en un conjunto de servidores de nombres de la raíz. El resultado de

este esquema es: En lugar de tener un servidor central para la base de datos, el trabajo implicado en mantenerla se reparte entre los hosts a lo largo y ancho del espacio de nombres. La autoridad para crear y cambiar nombres de hosts y la responsabilidad de mantener una base de datos para ellos le corresponde a la organización propietaria de la zona que los contiene. Desde el punto de vista del usuario, hay una sola base de datos que trata la resolución de las direcciones.

- **Registros de Recursos del DNS.** Cada unidad de información del DNS se llama Registro de Recurso (RR). Cada registro tiene un tipo asociado que describe el dato que contiene, y una clase que especifica el tipo de red al que se aplica. Estos proporcionan un mapeado entre nombres de dominio y objetos de red. Los objetos de red más comunes son las direcciones de los hosts, pero el DNS está diseñado para acomodarse a una variada gama de distintos objetos. El formato general del registro de recurso es:

NOMBRE	TTL	CLASE	TIPO	DATOS
--------	-----	-------	------	-------

donde:

- *NOMBRE*: Es el nombre de dominio a definir.
- *TTL*: Es el "time-to-live" o tiempo en segundos que el registro será válido en la caché de un servidor de nombres. Se almacena en el DNS como un valor de 32 bits sin signo. 86400(un día) es un valor típico para registros que apuntan a una dirección IP.
- *CLASE*: Identifica la familia del protocolo. Valores habituales son: IN (Sistema de Internet) CH (Sistema Chaos).
- *TIPO*: Identifica el tipo de recurso del registro.

- **DATOS:** Su valor depende del tipo.

En la siguiente tabla se mencionan algunos de los tipos de registros de recursos existentes y el contenido del campo datos según el tipo RR:

Tabla 1. Tipos de registros de recursos y el contenido del campo datos según el tipo RR

Tipo de RR	Descripción	Valor del campo datos
A	Dirección IP	Dirección IP
NS	Servidor autorizado de nombres	Nombre de dominio del host
PTR	Puntero hacia un nombre de dominio	Nombre de dominio que actúa como un puntero hacia una ubicación
SOA	Comienzo de autoridad de una zona	Información sobre la zona
MG	Miembro del buzón	El nombre de dominio de un buzón
CNAME	Nombre alias canónico	Un alias para la máquina
HINFO	Información sobre el host	Una cadena que identifica el tipo de CPU y otra que identifica el SO
MINFO	Información sobre el buzón	Un buzón responsable para enviar las listas y otro para mensajes de error
MR	Dominio para renombre de correspondencia	Renombra la dirección de un buzón específico
NULL	Registro nulo del recurso	Se puede colocar cualquier cosa

- **Servidores de Nombre.** En un principio podríamos suponer que un solo servidor de nombres podría contener la base de datos DNS completa y responder a todas las consultas sobre ella. En la práctica, este servidor estaría tan sobrecargado que sería inservible. Además si llegara a caerse toda la Internet se vendría abajo. Para evitar los problemas asociados a tener una sola fuente de información, el espacio de nombres DNS se

divide en zonas. Cada zona contiene información autoritativa que le corresponda. Los servidores de nombres (NS) almacenan la información acerca del espacio de nombres de dominio y generalmente "tienen autoridad" para administrar una o mas zonas.

Cuando un servidor de nombres mantiene toda la información acerca de una zona se le llama autorizado para esa zona. Cualquier petición para esa zona será enviada a uno de esos servidores maestros. Para tener una representación coherente de la zona, sus servidores maestros deben estar sincronizados. Para ello, a uno de ellos se le nombra servidor primario, el cual obtiene la información de la zona a partir de unos ficheros locales, y a los demás se les nombra servidores secundarios. Estos últimos cargan la información de la zona pidiéndosela al primario cada cierto tiempo. El proceso de obtener la información del archivo de la base de datos de la zona por medio de la red, se conoce como: una transferencia de zona. Hay dos razones para tener NS secundarios. Estas razones son:

- *Redundancia.* Se necesitan al menos dos NS en cada zona, un primario y al menos un secundario por redundancia. Como cualquier sistema de tolerancia a fallas, las máquinas deben ser independientes, por ejemplo, diferentes redes.
- *Localidades Remotas.* Para reducir la cantidad de trabajo en el servidor primario.

También es posible tener servidores de nombres locales, estos no mantienen información autorizada de ningún dominio. Este tipo de servidores es útil, pues al mantener una caché con los nombres que resuelven, disminuyen la

carga de la red y de otros servidores. Como la información de cada zona es almacenada en archivos independientes, la designación de primario o secundario, está definida a un nivel de zona. En otras palabras, un NS puede ser primario para ciertas zonas y secundario para otras. Cuando se define una zona con un NS como secundario, se debe especificar un NS del cual obtener la información de la zona. A la fuente de información de la zona para un NS secundario se le llama: servidor de nombres maestro. Un NS maestro puede ser primario o secundario. Cuando un NS secundario se inicializa, este se comunica con el NS maestro e inicia una transferencia de zona con el servidor. El uso de servidores secundarios como servidores maestros ayuda mucho, cuando el servidor primario esta muy ocupado, o cuando la forma de comunicación es mas eficiente. Para resolver nombres de hosts externos se consulta a los servidores raíz, cuyas direcciones IP están presentes en un fichero de configuración del sistema y se cargan en la caché del DNS al iniciar el servidor. Estos proporcionan referencias directas a servidores de los dominios de primer nivel, tales como *COM*, *EDU*, *GOV*, etc.

- **Mensajes del DNS.** Los mensajes del DNS utilizan un único formato:

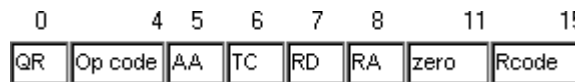
Figura 2. Formato Mensaje DNS.

0	15	31
1. Identification (ID)	2. Parameters	
3. QDcount	4. ANcount	
5. NScount	6. ARcount	
7. Question Section		
8. Answer Section		
9. Authority Section		
10. Additional Information Section		

El "resolver" envía la trama al servidor de nombres. Sólo el encabezado y la sección "question" se utilizan para la consulta. Las respuestas o retransmisiones de las consultas usan la misma trama, pero llenan más secciones de la misma (las secciones "Answer", "Authority" y "Additional").

- *Formato del encabezado.* La sección encabezado tiene una longitud fija de 12 bytes. Las otras secciones son de longitud variable. Comprende los siguientes campos:
 - *Identification (ID):* El campo identificación es un valor de 16 bits asignado por el programa. Este identificador se copia en la respuesta correspondiente del servidor de nombres y se puede usar para diferenciar respuestas cuando concurren múltiples consultas.
 - *Parameters:* El campo parámetros es un valor de 16 bits con el siguiente formato:

Figura 3. Formato Encabezado-Campo Parámetros.



donde:

- *QR:* Flag que indica consulta (0) o respuesta (1).
- *Op code:* Campo de 4 bits especificando el tipo de consulta:
 - 0 - Consulta estándar
 - 1- Consulta inversa

2 -Solicitud del estado del servidor

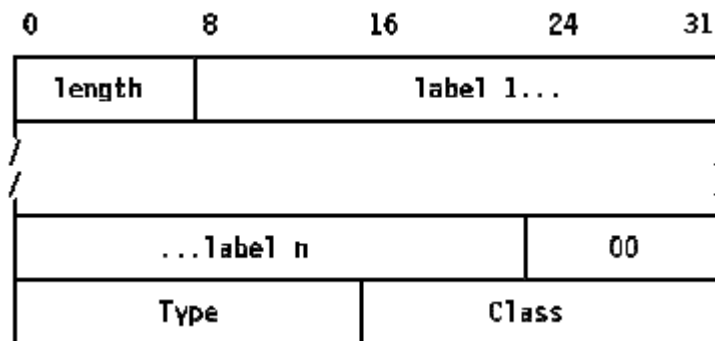
Se reservan los otros valores para su uso en el futuro.

- *AA*: Flag de respuesta autoritativa. Si está activo en una respuesta, especifica que el servidor de nombres que responde tiene autoridad para el nombre de dominio enviado en la consulta.
- *TC*: Flag de truncado. Activo si el mensaje es más largo de lo que permite el canal.
- *RD*: Flag de recursividad. Este bit indica al servidor de nombres que se pide resolución recursiva. El bit se copia en la respuesta.
- *RA*: Flag de recursividad disponible. Indica si el servidor de nombres soporta resolución recursiva.
- *Zero*: 3 bits reservados para uso futuro. Deben ser cero.
- *Rcode*: Código de respuesta de 4 bits. Posibles valores son:
 - 0 - Ningún error
 - 1 - Error de formato. El servidor fue incapaz de interpretar el mensaje.
 - 2 - Fallo en el servidor. El mensaje no fue procesado debido a un problema con el servidor.
 - 3 - Error e nombre. El nombre de dominio de la consulta no existe. Sólo válido si el bit *AA* está activo en la respuesta.
 - 4 - No implementado. El tipo solicitado de consulta no está implementado en el servidor de nombres.
 - 5 - Rechazado. El servidor rechaza responder por razones políticas.

Los demás valores se reservan para su usuario en el futuro.

- *QDcount*: Un entero sin signo de 16 bits que especifica el número de RRs en la sección "question".
 - *ANcount*: Un entero sin signo de 16 bits que especifica el número de RRs en la sección "answer".
 - *NScount*: Un entero sin signo de 16 bits que especifica el número de RRs en la sección "authority".
 - *ARcount*: Un entero sin signo de 16 bits que especifica el número de RRs en la sección "additional".
- *Formato Sección "Question"*. La siguiente sección contiene las consultas al servidor de nombres. Contiene QDcount (generalmente 1) entradas, cada una con el formato mostrado en la figura:

Figura 4. Formato Sección "Question".



donde:

- *Length*: Un byte que indica la longitud de la siguiente etiqueta.

- *Label*: Un elemento del nombre de dominio. El nombre de dominio se almacena como una serie de etiquetas de longitud variable, cada una precedida por un campo "length".
00: Indica el fin del dominio y representa la etiqueta nula del dominio raíz.
- *Type*: 2 bytes especificando el tipo de consulta. Puede tener cualquier valor del campo "Type" del registro.
- *Class*: 2 bytes especificando la clase de consulta. Para consultas en Internet, será "IN".
- *Secciones "Answer", "Authority" y "Additional Resource"*: Estas tres secciones contienen un número variable de RRs. El número se especifica en el campo correspondiente de la cabecera.
- **Resolución de Nombres de Dominio**. La resolución de nombres de dominio es un proceso cliente / servidor. La función del cliente ("resolver") es transparente al usuario y la llama una aplicación para mapear nombres de dominio a direcciones IP o viceversa. El servidor de nombres (llamado también servidor de nombres de dominio) es una aplicación servidora que traduce los nombres de las máquinas a direcciones IP.

Las peticiones sobre nombres de dominio pueden ser de dos tipos: *recursivas* o *iterativas* (llamadas también no-recursivas). Un bit de flag en la consulta (RD) especifica si el cliente desea una consulta recursiva y un bit de flag en la respuesta (RA) indica si el servidor soporta peticiones recursivas. La diferencia entre una consulta recursiva y una iterativa aparece cuando el servidor recibe una solicitud a la que por sí mismo no

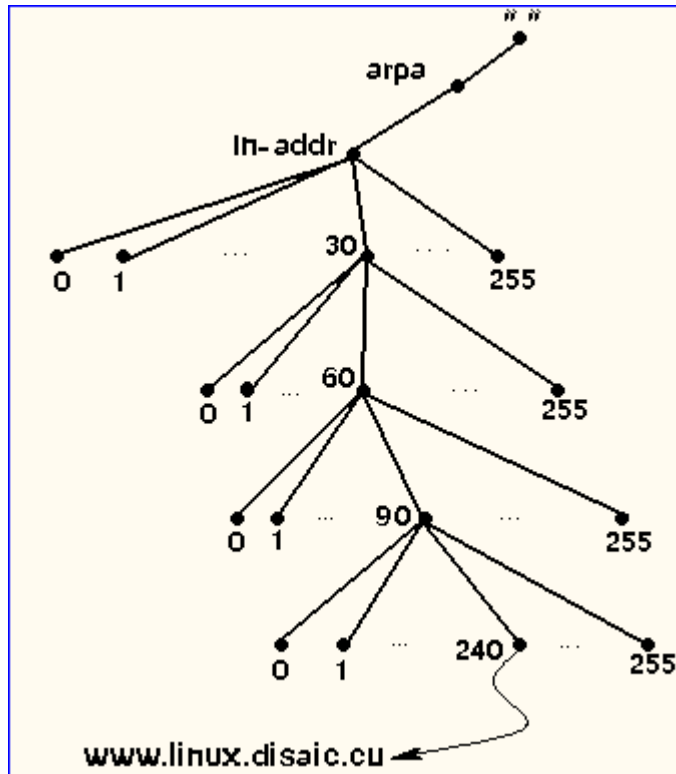
puede dar una respuesta completa. Una consulta recursiva demanda que el servidor lance a su vez una consulta para determinar la información buscada y luego devolvérsela al cliente. Una consulta iterativa implica que el servidor de nombres debería devolver la información de la que disponga además de una lista de servidores adicionales con los que el cliente puede contactar para completar su consulta.

Las respuestas de nombres de dominio pueden ser de dos tipos: *autoritativas* y *no-autoritativas*. Un bit de flag en la respuesta (AA) indica de qué tipo es la respuesta. Cuando un servidor de nombres recibe una consulta para un dominio en una zona en la que tiene autoridad, devuelve una respuesta autoritativa. Si no tiene autoridad en esa zona, su reacción depende de si el flag de recursividad (RD) está o no activo:

- Si el flag de recursividad deseada (RD) está activo y el servidor soporta recursividad, dirigirá su consulta a otro servidor de nombres. Este será un servidor con autoridad sobre el dominio de la consulta, o uno de los servidores de nombres de la raíz. Si el segundo servidor no devuelve una respuesta autoritativa, el proceso se repite. Cuando un servidor recibe una respuesta, la cacheará para mejorar el rendimiento de consultas repetidas. La entrada de la caché se almacena con un tiempo máximo especificado en la respuesta en un campo TTL ("time to live") de 32 bits. 86.400 segundos (1 día) es un valor típico.
- Si el flag de recursividad deseada (RD) no está activo o el servidor no soporta recursividad, devolverá la información que tenga en su caché o una lista de servidores capaces de dar respuestas autoritativas.

- Traducción de Direcciones a Nombres.

Figura 5. Dominio IN-ADDR.ARPA



De acuerdo a la estructura el DNS hasta el momento, la traducción de direcciones a nombres se vuelve imposible debido a que es necesario explorar todo el espacio de nombres para encontrar un número IP determinado y su nombre equivalente. Es por ello que se le adiciona una parte especial al espacio de nombres, conocida como dominio *in-addr.arpa*, en el cual se utilizan las direcciones en lugar de los nombres para identificar los dominios. Las direcciones o números IP se expresan en el formato de cuatro números entre 0 y 255 separados por puntos. En la figura anterior se representa la forma de este dominio especial donde se encuentran todas las posibles direcciones IP y sus equivalentes en

nombres. En este caso se ejemplifica con un host hipotético de nombre www.linux.disaic.cu con dirección IP 30.60.120.240.

3.1.4 Transporte

Los mensajes DNS se transmiten como datagramas (UDP) o sobre un canal (TCP).

- **UDP.** Puerto del servidor: 53 (decimal). Los mensajes transportados por UDP se restringen a 512 bytes. Los mensajes más largos se truncan y el bit TC de la cabecera se activa. Ya que las tramas UDP se pueden perder, hace falta una estrategia de retransmisión.
- **TCP.** Puerto del servidor: 53 (decimal). En este caso, el mensaje va precedido de un campo de 2 bytes que indica la longitud total de la trama.

3.1.5 Referencias

Los siguientes RFC's definen el estándar DNS y la información el mismo:

RFC 1032 - Guía del administrador de dominios

RFC 1033 - Guía operativa del administrador de dominios

RFC 1034 - Nombres de dominio - Conceptos y servicios

RFC 1035 - Nombres de dominio - Implementación y especificación

RFC 1101 - Codificación DNS de nombres de red y de otros tipos

RFC 1183 - Nuevas definiciones de RRs de DNS

RFC 1706 - Registros de recursos DNS NSAP

3.2 FUNDAMENTOS PROTOCOLO DATAGRAMAS DE USUARIO

3.2.1 Introducción

En la capa del Protocolo Internet (IP), una dirección de destino identifica una computadora anfitrión; no se hace ninguna otra distinción con respecto a qué usuario o qué aplicación recibirá el datagrama. El Protocolo de Datagramas de Usuario (UDP) distingue entre muchos destinos dentro de un anfitrión, permitiendo que varios programas de aplicación que se ejecutan en una computadora, envíen y reciban datagramas en forma independiente.

UDP asume que el IP se utiliza como protocolo subyacente, es orientado a transacciones y no garantiza la entrega ni la protección ante duplicados. Las aplicaciones que requieran una entrega fiable y ordenada de secuencias de datos deben usar el Protocolo de Control de Transmisión TCP. UDP utiliza el puerto 17.

Las principales especificaciones para UDP están definidas en el RFC 768 .

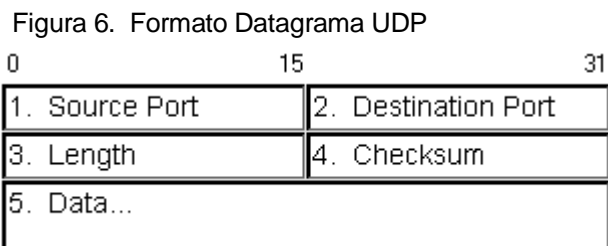
3.2.2 Definición

El módulo UDP es un fino recubrimiento a la capa IP que cumple las funciones equivalentes al nivel OSI de Transporte de datagramas. Se basa en el intercambio de datagramas permitiendo su envío a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera; el inconveniente de esta forma de actuación es que no hay confirmación de recepción ni de haber recibido los datagramas en el orden adecuado, debiendo ser la aplicación la que se encargue de controlarlo. El UDP puede

ser la alternativa al TCP en algunos casos en los que no sea necesario el gran nivel de complejidad proporcionado por el TCP. Puesto que UDP no admite numeración de los datagramas, éste protocolo se utiliza principalmente cuando el orden en que se reciben los mismos no es un factor fundamental, o también cuando se quiere enviar información de poco tamaño que cabe en un único datagrama. UDP proporciona puertos de protocolo utilizados para distinguir entre muchas aplicaciones, además de los datos, cada mensaje UDP contiene tanto el número de puerto de origen como el número de puerto destino, haciendo posible que el mensaje se entregue al receptor correcto y éste envíe una respuesta.

3.2.3 Formato de Datagrama UDP

Cada mensaje UDP se conoce como datagrama de usuario. Conceptualmente un datagrama está compuesto por dos partes: un encabezado y un área de datos, este formato se muestra en la siguiente figura:



donde:

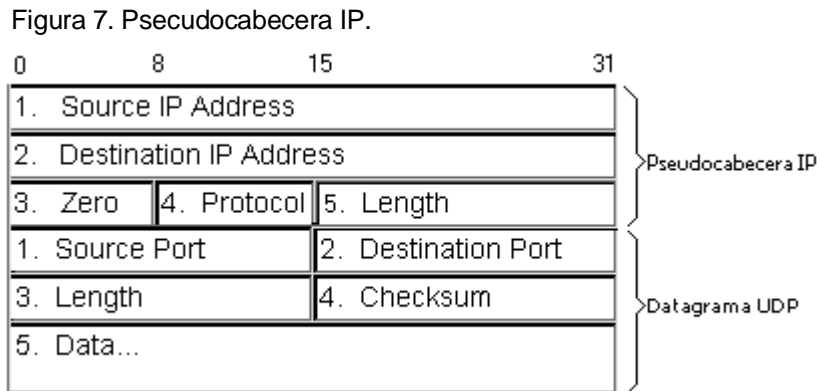
- **Source Port.** Campo puerto de origen. Este valor de 16 bits es opcional, indica el puerto del proceso emisor y puede que se asuma que este es el puerto al cual la respuesta será dirigida en ausencia de otra información. Si no se utiliza debe ser cero. Existen dos tipos de puertos: los bien conocidos

y los efímeros, los primeros pertenecen a servidores estándar, por ejemplo DNS usa el puerto 53, estos se hallan en el rango de 1 a 1023 y suelen tener números impares. Los puertos efímeros son asignados por un host en funcionamiento a cada proceso mientras éste lo necesite, están en el rango de 1024 a 5000. Los clientes no necesitan puertos bien conocidos porque inician la comunicación con los servidores mediante datagramas UDP que contienen su número de puerto, los clientes pueden usar cualquier número de puerto efímero siempre que la combinación <protocolo de transporte, dirección IP, número de puerto> sea única.

- **Destination Port.** Campo puerto destino. Este valor de 16 bits contiene el número de puerto de la aplicación a la que se enviará el datagrama.
- **Length.** Campo longitud. Valor de 16 bits que representa lo longitud en octetos del datagrama de usuario incluyendo el encabezado y los datos. Por lo tanto, el valor mínimo para el campo longitud es de ocho (longitud del encabezado).
- **Checksum.** Campo suma de verificación. Valor de 16 bits opcional que almacena la suma de comprobación de errores del datagrama. Si su valor es cero, significa que no se calculó y puede ser ignorado. Se calcula sobre los datos y encabezado del datagrama UDP y la pseudocabecera IP.
- **Data.** Campo obligatorio que contiene los datos que envían las aplicaciones con el datagrama. Ejemplo: mensaje DNS.

3.2.4 Pseudocabecera IP

El propósito de utilizar un pseudoencabezado IP es verificar que el datagrama UDP llegó a su destino, ya que éste proporciona protección frente a datagramas mal encaminados, consiste en 12 bytes de datos distribuidos como se muestra a continuación:



donde:

- **Source IP Address.** Campo dirección IP de origen. Valor de 32 bits que se utilizará cuando se envíe el mensaje UDP para identificar el ordenador que envía la consulta.
- **Destination IP Address.** Campo dirección IP de destino. Valor de 32 bits que se utilizará cuando se envíe el mensaje UDP para identificar el ordenador que recibe la consulta.
- **Zero.** Campo de 8 bits de ceros para completar el datagrama.
- **Protocol.** Campo de 8 bits que contiene el código del tipo de protocolo (17 para UDP).

- **Length.** Campo de 16 bits que indica la longitud del datagrama UDP.

3.2.5 Interfaz IP

El módulo UDP debe ser capaz de determinar las direcciones de origen y destino en un entorno de Internet así como el campo Protocolo de la cabecera del IP. Una posible interfaz UDP/IP devolvería el datagrama completo, incluyendo toda la cabecera, en respuesta a una operación de recepción. Una interfaz de este tipo permitiría también al módulo UDP pasar un datagrama de Internet completo con cabecera al módulo IP para ser enviado. IP verificaría ciertos campos por consistencia y calcularía la suma de verificación.

3.2.6 Encapsulación de UDP Y Estratificación por Capas de Protocolos

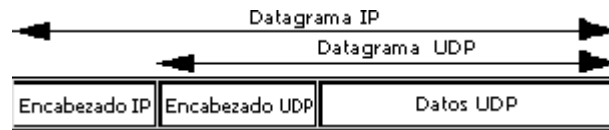
El lugar de UDP en las capas de TCP/IP es:

Figura 8. Lugar de UDP en las capas de TCP/IP



Estratificar por capas el UDP por encima del IP significa que un mensaje UDP completo incluyendo el encabezado y los datos se encapsula en un datagrama IP mientras viaja a través de una red de redes tal y como se muestra a continuación:

Figura 9. Encapsulación del datagrama UDP



La capa IP sólo es responsable de transferir datos entre un par de anfitriones dentro de una red de redes, mientras que la capa UDP solamente es responsable de diferenciar entre varias fuentes o destinos dentro de un anfitrión.

3.2.7 Aplicaciones del UDP

Los usos principales del protocolo son:

- DNS- Sistema de nombres de dominio.
- TFTP- Transferencia trivial de archivos.
- RPC- Llamada de procedimiento remoto (usado por el NFS).

3.2.8 Números de Puertos UDP Reservados y Disponibles

Los diseñadores del TCP adoptaron un enfoque que preasigna algunos números de puertos, pero que dejan muchos de ellos disponibles para los sitios locales o programas de aplicación. En la siguiente tabla se listan algunos de los puertos UDP actualmente asignados:

Tabla 2. Números de puerto UDP Reservados y Disponibles

Decimal	Palabra Clave	Descripción
0	-	Reservado
7	ECHO	Eco
9	DISCARD	Descartar
11	USERS	Usuarios Activos
13	DAYTIME	Hora del día
15	-	Quién está ahí o NETSTAT

17	QUOTE	Cita del día
19	CHARGEN	Generador de caracteres
37	TIME	Hora
42	NAMESERVER	Servidor de nombre de anfitriones
43	NICNAME	Quién es
53	DOMAIN	Servidor de nombres de Dominio
67	BOOTPS	Servidor de protocolo bootstrap
68	BOOTPC	Cliente de protocolo bootstrap
69	TFTP	Transferencia trivial de archivos
111	SUNRPC	RCP de Sun Microsystems
123	NTP	Protocolo de tiempo de red
161		Monitor de res SNMP
162		Interrupciones SNMP
512		COMSAT UNIX
513		Rwho daemon UNIX
514		Conexión de sistema

3.3 INFORMATICA EDUCATIVA

3.3.1 Definición de Software Educativo

3.3.1.1 Conceptualización. El software educativo es un programa para ordenador creado con la finalidad específica de ser utilizado como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje.

Esta definición engloba todos los programas que han sido elaborados con fines didácticos, desde los tradicionales programas basados en los modelos conductistas de los programas experimentales de Enseñanza Inteligente Asistida por Ordenador (EIAO), que, utilizando técnicas propias del campo de los Sistemas Expertos y de la Inteligencia Artificial en general, pretenden imitar la labor tutorial personalizada que realizan los profesores y presentan modelos de representación del conocimiento en consonancia con los procesos cognitivos que desarrollan los usuarios.

No obstante según esta definición, más basada en un criterio de finalidad que de funcionalidad, se excluyen del software educativo todos los programas de uso general en el mundo empresarial que también se utilizan en los centros educativos con funciones didácticas o instrumentales como por ejemplo: procesadores de textos, gestores de bases de datos, hojas de cálculo, editores gráficos. Estos programas, aunque puedan desarrollar una función didáctica, no han estado elaborados específicamente con esta finalidad.

3.3.1.2 Características Esenciales de los Programas Educativos. Los programas educativos pueden tratar diferentes temas de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada al usuario, mediante la simulación de procesos...) y ofrecer un entorno de trabajo sensible a las circunstancias del usuario y rico en posibilidades de interacción; pero todos comparten cinco características esenciales:

- Son materiales elaborados con una *finalidad didáctica*.
- Utilizan el *ordenador* como soporte en cual el usuario realiza las actividades que ellos proponen.
- Son *interactivos*, responden inmediatamente a la acción del usuario y permite un diálogo y un intercambio de información entre el ordenador y el usuario.
- *Individualizan el trabajo* del usuario, ya que se adaptan al ritmo de trabajo de cada uno, ajustando sus actividades según las actuaciones de éste.
- Son *fáciles de usar*. Los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son similares a los conocimientos de electrónica necesarios para usar un video, es decir, son mínimos, aunque cada programa tiene unas reglas de funcionamiento que es necesario conocer.

3.3.2 Estructura Básica de los Programas Educativos.

La mayoría de los programas didácticos, igual que muchos de los programas informáticos nacidos sin finalidad educativa, tienen tres módulos principales claramente definidos: el módulo que gestiona la comunicación con el usuario (sistema input/output), el módulo que contiene debidamente organizados los contenidos informativos del programa (bases de datos) y el módulo que gestiona las actuaciones del ordenador y sus respuestas a las acciones de los usuarios (motor).

3.3.2.1 El Entorno de Comunicación o Interfaz. La interfaz es el entorno a través del cual los programas establecen el diálogo con sus usuarios, y es la que posibilita la interactividad característica de estos materiales. Está integrada por dos sistemas:

- El sistema de comunicación *programa-usuario*, que facilita la transmisión de informaciones al usuario por parte del ordenador, incluye:
 - Las pantallas a través de las cuales los programas presentan información a los usuarios.
 - Los informes y las fichas que proporcionen mediante las impresoras.
 - El empleo de otros periféricos: altavoces, sintetizadores de voz, robots, módems, convertidores digitales-analógicos...
- El sistema de comunicación *usuario-programa*, que facilita la transmisión de información del usuario hacia el ordenador, incluye:

- El uso del teclado y el ratón, mediante los cuales los usuarios introducen al ordenador un conjunto de órdenes o respuestas que los programas reconocen.
- El empleo de otros periféricos: micrófonos, lectores de fichas, teclados conceptuales, pantallas táctiles, lápices ópticos, módems, lectores de tarjetas, convertidores analógico-digitales...

Con la ayuda de las técnicas de la Inteligencia Artificial y del desarrollo de las tecnologías multimedia, se investiga la elaboración de entornos de comunicación cada vez más intuitivos y capaces de proporcionar un diálogo abierto y próximo al lenguaje natural.

3.3.2.2 Las Bases de Datos. Las bases de datos contienen la información específica que cada programa presentará a los usuarios. Pueden estar constituidas por:

- **Modelos de Comportamiento.** Representan la dinámica de los sistemas. Distinguimos:
 - *Modelos físico-matemáticos*, que tienen unas leyes perfectamente determinadas por unas ecuaciones.
 - *Modelos no deterministas*, regidos por unas leyes no totalmente deterministas, que son representadas por ecuaciones con variables aleatorias, por grafos y por tablas de comportamiento.
- **Datos de Tipo Texto.** Información alfanumérica.
- **Datos Gráficos.** Las bases de datos pueden estar constituidas por dibujos, fotografías, secuencias de vídeo, etc.

- **Sonido.** Como los programas que permiten componer música, escuchar determinadas composiciones musicales y visionar sus partituras.

3.3.2.3 El Motor o Algoritmo. El algoritmo del programa, en función de las acciones de los usuarios, gestiona las secuencias en que se presenta la información de las bases de datos y las actividades que pueden realizar los usuarios. Distinguimos 4 tipos de algoritmo:

- **Lineal.** Cuando la secuencia de las actividades es única.
- **Ramificado.** Cuando están predeterminadas posibles secuencias según las respuestas de los usuarios.
- **Tipo Entorno.** Cuando no hay secuencias predeterminadas para el acceso del usuario a la información principal y a las diferentes actividades. El usuario elige *qué* ha de hacer y *cuándo* lo ha de hacer. Este entorno puede ser:
 - *Estático*, si el usuario sólo puede consultar (y en algunos casos aumentar o disminuir) la información que proporciona el entorno, pero no puede modificar su estructura.
 - *Dinámico*, si el usuario, además de consultar la información, también puede modificar el estado de los elementos que configuran el entorno.
 - *Programable*, si a partir de una serie de elementos el usuario puede construir diversos entornos.
 - *Instrumental*, si ofrece a los usuarios diversos instrumentos para realizar determinados trabajos.

- **Tipo Sistema Experto.** Cuando el programa tiene un motor de inferencias y, mediante un diálogo bastante inteligente y libre con el usuario, asesora al usuario inteligentemente en el aprendizaje. Su desarrollo está muy ligado con los avances en el campo de la Inteligencia Artificial.

3.3.3 Clasificación de los Programas Educativos

Los programas educativos a pesar de tener unos rasgos esenciales básicos y una estructura general común se presentan con unas características muy diversas: unos aparentan ser un laboratorio o una biblioteca, otros se limitan a ofrecer una función instrumental del tipo máquina de escribir o calculadora, otros se presentan como un juego o como un libro, bastantes tienen vocación de examen, unos pocos se creen expertos... y, por si no fuera bastante, la mayoría participan en mayor o menor medida de algunas de estas peculiaridades. Para poner orden a esta disparidad, se han elaborado múltiples tipologías que clasifican los programas didácticos a partir de diferentes criterios.

Uno de estos criterios se basa en la consideración del tratamiento de los errores que cometen los usuarios, distinguiendo:

- *Programas Tutoriales Directivos.* Hacen preguntas a los usuarios y controlan en todo momento su actividad. El ordenador adopta el papel de juez poseedor de la verdad y examina al usuario. Se producen errores cuando la respuesta del usuario está en desacuerdo con la que el ordenador tiene como correcta. En los programas más tradicionales el error lleva implícita la noción de fracaso.

- *Programas no Directivos.* El ordenador adopta el papel de un laboratorio o instrumento a disposición de la iniciativa de un usuario que pregunta y tiene una libertad de acción sólo limitada por las normas del programa. El ordenador no juzga las acciones del usuario, se limita a procesar los datos que éste introduce y a mostrar las consecuencias de sus acciones sobre un entorno. Objetivamente no se producen errores, sólo desacuerdos entre los efectos esperados por el usuario y los efectos reales de sus acciones sobre el entorno. No está implícita la noción de fracaso. El error es sencillamente una hipótesis de trabajo que no se ha verificado y que se debe sustituir por otra. En general, siguen un modelo pedagógico de inspiración cognitivista, potencian el aprendizaje a través de la exploración, favorecen la reflexión y el pensamiento crítico y propician la utilización del método científico.

Otra clasificación interesante de los programas atiende a la posibilidad de *modificar los contenidos* del programa y distingue entre *programas cerrados* (que no pueden modificarse) y *programas abiertos*, que proporcionan un esqueleto, una estructura, sobre la cual los usuarios pueden añadir el contenido que les interese. De esta manera se facilita su adecuación a los diversos contextos educativos y permite un mejor tratamiento de la diversidad de los usuarios.

No obstante, de todas las clasificaciones la que posiblemente proporciona categorías más claras y útiles a los usuarios es la que tiene en cuenta el grado de control del programa sobre la actividad de los usuarios y la estructura de su algoritmo, se presenta a continuación.

3.3.3.1 Programas Tutoriales. Son programas que en mayor o menor medida dirigen, tutorizan, el trabajo de los usuarios. Pretenden que, a partir de unas informaciones y mediante la realización de ciertas actividades previstas de antemano, los usuarios pongan en juego determinadas capacidades y aprendan o refuercen sus conocimientos y habilidades. Cuando se limitan a proponer ejercicios de refuerzo sin proporcionar explicaciones conceptuales previas se denominan programas tutoriales de ejercitación, como es el caso de los programas de preguntas, programas de adiestramiento psicomotor, que desarrollan la coordinación neuromotriz en actividades relacionadas con el dibujo, la escritura y otras habilidades psicomotrices.

En cualquier caso, son programas basados en los *planteamientos conductistas* de la enseñanza que comparan las respuestas de los usuarios con los patrones que tienen como correctos, guían los aprendizajes de los usuarios y facilitan la realización de prácticas rutinarias y su evaluación; en algunos casos una evaluación negativa genera una nueva serie de ejercicios de repaso. A partir de la estructura de su algoritmo, se distinguen cuatro categorías:

- **Programas Lineales.** Presentan al usuario una secuencia de información y/o ejercicios (siempre la misma o determinada aleatoriamente) con independencia de la corrección o incorrección de sus respuestas. Herederos de la enseñanza programada, transforman el ordenador en una máquina de enseñar transmisora de conocimientos y adiestradora de habilidades. No obstante, su interactividad resulta pobre y el programa se hace largo de recorrer.

- **Programas Ramificados.** Basados inicialmente también en modelos conductistas, siguen recorridos pedagógicos diferentes según el juicio que hace el ordenador sobre la corrección de las respuestas de los usuarios o según su decisión de profundizar más en ciertos temas. Ofrecen mayor interacción, más opciones, pero la organización de la materia suele estar menos compartimentada que en los programas lineales y exigen un esfuerzo más grande al usuario. Pertenecen a éste grupo los programas multinivel, que estructuran los contenidos en niveles de dificultad y previenen diversos caminos, y los programas ramificados con dientes de sierra, que establecen una diferenciación entre los conceptos y las preguntas de profundización, que son opcionales.
- **Entornos Tutoriales.** En general están inspirados en *modelos pedagógicos cognitivistas*, y proporcionan a los usuarios una serie de herramientas de búsqueda y de proceso de la información que pueden utilizar libremente para construir la respuesta a las preguntas del programa. Este es el caso de los *entornos de resolución de problemas*, "problem solving", donde los usuarios conocen parcialmente las informaciones necesarias para su resolución y han de buscar la información que falta y aplicar reglas, leyes y operaciones para encontrar la solución. En algunos casos, el programa no sólo comprueba la corrección del resultado, sino que también tiene en cuenta la idoneidad del camino que se ha seguido en la resolución.
- **Sistemas Tutoriales Expertos.** Como los Sistemas Tutores Inteligentes (Intelligent Tutoring Systems), que, elaborados con las técnicas de la Inteligencia Artificial y teniendo en cuenta las teorías cognitivas sobre el aprendizaje, tienden a reproducir un diálogo auténtico entre el programa y

el usuario, y pretenden comportarse como lo haría un tutor humano: guían a los usuarios paso a paso en su proceso de aprendizaje, analizan su estilo de aprender y sus errores y proporcionan en cada caso la explicación o ejercicio más conveniente.

3.3.3.2 Bases de Datos. Proporcionan unos datos organizados, en un entorno estático, según determinados criterios, y facilitan su exploración y consulta selectiva. Se pueden emplear en múltiples actividades como por ejemplo: seleccionar datos relevantes para resolver problemas, analizar y relacionar datos, extraer conclusiones, comprobar hipótesis... Las preguntas que acostumbran a realizar los usuarios son del tipo: ¿Qué características tiene este dato? ¿Qué datos hay con la característica X? ¿Qué datos hay con las características X e Y?.

Las bases de datos pueden tener una estructura *jerárquica* (si existen unos elementos subordinantes de los que dependen otros subordinados, como los organigramas), *relacional* (si están organizadas mediante unas fichas o registros con una misma estructura y rango) o *documental* (si utiliza descriptores y su finalidad es almacenar grandes volúmenes de información documental: revistas, periódicos, etc.). En cualquier caso, según la forma de acceder a la información se pueden distinguir dos tipos:

- **Bases de Datos Convencionales.** Tienen la información almacenada en ficheros, mapas o gráficos, que el usuario puede recorrer según su criterio para recopilar información.
- **Bases de Datos Tipo Sistema Experto.** Son bases de datos muy especializadas que recopilan toda la información existente de un tema

concreto y además asesoran al usuario cuando accede buscando determinadas respuestas.

3.3.3.3 Simuladores. Presentan un modelo o entorno dinámico (generalmente a través de gráficos o animaciones interactivas) y facilitan su exploración y modificación a los usuarios, que pueden realizar aprendizajes inductivos o deductivos mediante la observación y la manipulación de la estructura subyacente; de esta manera pueden descubrir los elementos del modelo, sus interrelaciones, y pueden tomar decisiones y adquirir experiencia directa delante de unas situaciones que frecuentemente resultarían difícilmente accesibles a la realidad (control de una central nuclear, contracción del tiempo, pilotaje de un avión...). También se pueden considerar simulaciones ciertos videojuegos que, al margen de otras consideraciones sobre los valores que incorporan (generalmente no muy positivos) facilitan el desarrollo de los reflejos, la percepción visual y la coordinación psicomotriz en general, además de estimular la capacidad de interpretación y de reacción ante un medio concreto.

En cualquier caso, posibilitan un *aprendizaje significativo por descubrimiento* y la investigación de los usuarios/experimentadores puede realizarse en tiempo real o en tiempo acelerado, según el simulador, mediante preguntas del tipo: *¿Qué pasa al modelo si modifico el valor de la variable X? ¿Y si modifico el parámetro Y?* Se pueden diferenciar dos tipos de simulador:

- **Modelos Físico-Matemáticos.** Presentan de manera numérica o gráfica una realidad que tiene unas leyes representadas por un sistema de ecuaciones deterministas. Se incluyen aquí los programas-laboratorio, algunos trazadores de funciones y los programas que mediante un convertidor analógico-digital captan datos analógicos de un fenómeno

externo al ordenador y presentan en pantalla un modelo del fenómeno estudiado o informaciones y gráficos que van asociados. Estos programas a veces son utilizados por tutores delante de la clase a manera de pizarra electrónica, como demostración o para ilustrar un concepto, facilitando así la transmisión de información a los usuarios, que después podrán repasar el tema interactuando con el programa.

- **Entornos Sociales.** Presentan una realidad regida por unas leyes no del todo deterministas. Se incluyen aquí los juegos de estrategia y de aventura, que exigen una estrategia cambiante a lo largo del tiempo.

3.3.3.4 Constructores. Son programas que tienen un entorno programable. Facilitan a los usuarios unos elementos simples con los cuales pueden construir elementos más complejos o entornos. De esta manera potencian el aprendizaje heurístico y, de acuerdo con las teorías cognitivistas, facilitan a los usuarios la construcción de sus propios aprendizajes, que surgirán a través de la reflexión que realizarán al diseñar programas y comprobar inmediatamente, cuando los ejecuten, la relevancia de sus ideas. El proceso de creación que realiza el usuario genera preguntas del tipo: ¿Qué sucede si añado o elimino el elemento X?. Se pueden distinguir dos tipos de constructores:

- **Constructores Específicos.** Ponen a disposición de los usuarios una serie de mecanismos de actuación (generalmente en forma de órdenes específicas) que les permiten llevar a cabo operaciones de un cierto grado de complejidad mediante la construcción de determinados entornos, modelos o estructuras, y de esta manera avanzan en el conocimiento de una disciplina o entorno específico.

- **Lenguajes de Programación.** Ofrecen unos "laboratorios simbólicos" en los que se pueden construir un número ilimitado de entornos. Aquí los usuarios se convierten en profesores del ordenador. Además, con los interfaces convenientes, pueden controlar pequeños robots construidos con componentes convencionales (arquitecturas, motores...), de manera que sus posibilidades educativas se ven ampliadas incluso en campos pre-tecnológicos. Así los usuarios pasan de un manejo abstracto de los conocimientos con el ordenador a una manipulación concreta y práctica en un entorno informatizado que facilita la representación y comprensión del espacio y la previsión de los movimientos.

3.3.3.5 Programas Herramienta. Son programas que proporcionan un entorno instrumental con el cual se facilita la realización de ciertos trabajos generales de tratamiento de la información: escribir, organizar, calcular, dibujar, transmitir, captar datos.... A parte de los lenguajes de autor (que también se podrían incluir en el grupo de los programas constructores), los más utilizados son programas de uso general que provienen del mundo laboral y, por tanto, quedan fuera de la definición que se ha dado de software educativo. No obstante, se han elaborado algunas versiones de estos programas "para niños" que limitan sus posibilidades a cambio de una, no siempre clara, mayor facilidad de uso. De hecho, muchas de estas versiones resultan innecesarias, ya que el uso de estos programas cada vez resulta más sencillo y cuando los usuarios necesitan utilizarlos o su uso les resulta funcional aprenden a manejarlos sin dificultad. Los programas más utilizados de este grupo son:

- **Procesadores de Texto.** Son programas que, con la ayuda de una impresora, convierten el ordenador en una fabulosa máquina de escribir. En el ámbito educativo debe hacerse una introducción gradual que puede

empezar a lo largo de la enseñanza primaria, y ha de permitir a los usuarios familiarizarse con el teclado y con el ordenador en general, y sustituir parcialmente la libreta de redacciones por un disco (donde almacenarán sus trabajos). Al escribir con los procesadores de textos los usuarios pueden concentrarse en el contenido de las redacciones y demás trabajos que tengan encomendados despreocupándose por la caligrafía. Además el corrector ortográfico que suelen incorporar les ayudará a revisar posibles faltas de ortografía antes de entregar el trabajo.

Además de este empleo instrumental, los procesadores de textos permiten realizar múltiples actividades didácticas, por ejemplo:

- Ordenar párrafos, versos, estrofas.
 - Insertar frases y completar textos.
 - Separar dos poemas...
- **Gestores de Bases de Datos.** Sirven para generar potentes sistemas de archivo ya que permiten almacenar información de manera organizada y posteriormente recuperarla y modificarla. Entre las muchas actividades con valor educativo que se pueden realizar están las siguientes:
 - Revisar una base de datos ya construida para buscar determinadas informaciones y recuperarlas.
 - Recoger información, estructurarla y construir una nueva base de datos.
 - **Hojas de Cálculo.** Son programas que convierten el ordenador en una versátil y rápida calculadora programable, facilitando la realización de actividades que requieran efectuar muchos cálculos matemáticos. Entre

las actividades didácticas que se pueden realizar con las hojas de cálculo están las siguientes:

- Aplicar hojas de cálculo ya programadas a la resolución de problemas de diversas asignaturas, evitando así la realización de pesados cálculos y ahorrando un tiempo que se puede dedicar a analizar los resultados de los problemas.
 - Programar una nueva hoja de cálculo, lo que exigirá previamente adquirir un conocimiento preciso del modelo matemático que tiene que utilizar.
- **Editores Gráficos.** Se emplean desde un punto de vista instrumental para realizar dibujos, portadas para los trabajos, murales, anuncios, etc. Además constituyen un recurso idóneo para desarrollar parte del currículum de Educación Artística: dibujo, composición artística, uso del color, etc.
 - **Programas de Comunicaciones.** Son programas que permiten que ordenadores lejanos (si disponen de módem) se comuniquen entre sí a través de las líneas telefónicas y puedan enviarse mensajes y gráficos, programas... Desde una perspectiva educativa estos sistemas abren un gran abanico de actividades posibles para los usuarios, por ejemplo:
 - Comunicarse con otros compañeros e intercambiarse informaciones.
 - Acceder a bases de datos lejanas para buscar determinadas informaciones.
 - **Programas de Experimentación Asistida.** A través de variados instrumentos y convertidores analógico-digitales, recogen datos sobre el

comportamiento de las variables que inciden en determinados fenómenos. Posteriormente con estas informaciones se podrán construir tablas y elaborar representaciones gráficas que representen relaciones significativas entre las variables estudiadas.

- **Lenguajes y Sistemas de Autor.** Son programas que facilitan la elaboración de programas tutoriales a quienes no disponen de grandes conocimientos informáticos. Utilizan unas pocas instrucciones básicas que se pueden aprender en pocas sesiones. Algunos incluso permiten controlar vídeos y dan facilidades para crear gráficos y efectos musicales, de manera que pueden generar aplicaciones multimedia.

3.3.4 Funciones del Software Educativo

Los programas didácticos, cuando se aplican a la realidad educativa, realizan las funciones básicas propias de los medios didácticos en general y además, en algunos casos, según la forma de uso que se determine, pueden proporcionar funcionalidades específicas.

Por otra parte, como ocurre con otros productos de la actual tecnología educativa, no se puede afirmar que el software educativo por sí mismo sea bueno o malo, todo dependerá del uso que de él se haga, de la manera cómo se utilice en cada situación concreta. En última instancia su funcionalidad y las ventajas e inconvenientes que pueda comportar su uso serán el resultado de las características del material, de su adecuación al contexto educativo al que se aplica y de la manera en que el profesor organice su utilización. Funciones que pueden realizar los programas:

- **Función Informativa.** La mayoría de los programas a través de sus actividades presentan unos contenidos que proporcionan una información estructurada de la realidad a los usuarios. Como todos los medios didácticos, estos materiales representan la realidad y la ordenan. Los programas *tutoriales*, los *simuladores* y, especialmente, las *bases de datos*, son los programas que realizan más marcadamente una función informativa.
- **Función Instructiva.** Todos los programas educativos orientan y regulan el aprendizaje de los usuarios ya que, explícita o implícitamente, promueven determinadas actuaciones de los mismos encaminadas a facilitar el logro de unos objetivos educativos específicos. Además condicionan el tipo de aprendizaje que se realiza pues, por ejemplo, pueden disponer un tratamiento global de la información (propio de los medios audiovisuales) o a un tratamiento secuencial (propio de los textos escritos).

Con todo, si bien el ordenador actúa en general como mediador en la construcción del conocimiento y el metac conocimiento de los usuarios, son los programas *tutoriales* los que realizan de manera más explícita esta función instructiva, ya que dirigen las actividades de los usuarios en función de sus respuestas y progresos.

- **Función Motivadora.** Generalmente los usuarios se sienten atraídos e interesados por todo el software educativo, ya que los programas suelen incluir elementos para captar la atención de los usuarios, mantener su interés y, cuando sea necesario, focalizarlo hacia los aspectos más importantes de las actividades. Por lo tanto la función motivadora es una

de las más características de este tipo de materiales didácticos, y resulta extremadamente útil para los tutores.

- **Función Evaluadora.** La interactividad propia de estos materiales, que les permite responder inmediatamente a las respuestas y acciones de los usuarios, les hace especialmente adecuados para evaluar el trabajo que se va realizando con ellos. Esta evaluación puede ser de dos tipos:
 - Implícita, cuando el usuario detecta sus errores, se evalúa, a partir de las respuestas que le da el ordenador.
 - Explícita, cuando el programa presenta informes valorando la actuación del usuario. Este tipo de evaluación sólo la realizan los programas que disponen de módulos específicos de evaluación.

- **Función Investigadora.** Los programas no directivos, especialmente las *bases de datos, simuladores y programas constructores*, ofrecen a los usuarios interesantes entornos donde investigar: buscar determinadas informaciones, cambiar los valores de las variables de un sistema, etc. Además, tanto estos programas como los *programas herramienta*, pueden proporcionar a los tutores y usuarios instrumentos de gran utilidad para el desarrollo de trabajos de investigación que se realicen básicamente al margen de los ordenadores.

- **Función Expresiva.** Dado que los ordenadores son unas máquinas capaces de procesar los símbolos mediante los cuales las personas representamos nuestros conocimientos y nos comunicamos, sus posibilidades como instrumento expresivo son muy amplias. Desde el ámbito de la informática que estamos tratando, el software educativo, los

usuarios se expresan y se comunican con el ordenador y con otros compañeros a través de las actividades de los programas y, especialmente, cuando utilizan *lenguajes de programación, procesadores de textos, editores de gráficos*, etc. Otro aspecto a considerar al respecto es que los ordenadores no suelen admitir la ambigüedad en sus "diálogos" con los usuarios, de manera que los usuarios se ven obligados a cuidar más la precisión de sus mensajes.

- **Función Metalingüística.** Mediante el uso de los sistemas operativos y los lenguajes de programación los usuarios pueden aprender los lenguajes propios de la informática.
- **Función Lúdica.** Trabajar con los ordenadores realizando actividades educativas es una labor que a menudo tiene unas connotaciones lúdicas y festivas para los usuarios. Además, algunos programas refuerzan su atractivo mediante la inclusión de determinados elementos lúdicos, con lo que potencian aún más esta función.
- **Función Innovadora.** Aunque no siempre sus planteamientos pedagógicos resulten innovadores, los programas educativos se pueden considerar materiales didácticos con esta función ya que utilizan una tecnología recientemente incorporada a los centros educativos y, en general, suelen permitir diversas formas de uso. Esta versatilidad ofrece amplias posibilidades de experimentación didáctica e innovación educativa en el aula.

3.4 INGENIERIA DEL SOFTWARE

La ingeniería del software es una disciplina relacionada con el desarrollo de productos de soporte lógico o software. Un producto de software es el conjunto completo de programas informáticos, procedimientos, documentación y datos especificados para su suministro a un cliente; el desarrollo se ocupa de todas las actividades técnicas y de gestión necesarias para crear el producto, y realizar el desarrollo eficazmente significa cumplir las necesidades del cliente ajustándose a unos límites de tiempo, coste y calidad. Típicamente, el proceso se divide en cuatro fases principales:

- El *análisis y especificación de requisitos*, donde se establece qué debe lograr el producto de software.
- El *diseño*, que determina cómo cumplirá el software esos requisitos.
- La *implementación*, que crea el producto de software que se ha diseñado (esto combina el desarrollo de nuevos componentes con la reutilización o modificación de componentes anteriores).
- La *prueba*, que garantiza que el producto de software funciona como se pretende. Los productos intermedios, como las especificaciones de requisitos y los diseños de software, también se revisan en profundidad antes de pasar a la siguiente fase de desarrollo.

En todos los casos, los principios de la ingeniería de software ayudan a garantizar que los sistemas resultantes sean fiables y funcionen del modo requerido.

3.4.1 El Proceso Unificado de Desarrollo de Software

“El proceso unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto”³.

El proceso unificado es un proceso de desarrollo de software que proporciona un conjunto de actividades con las cuales se pueden transformar los requisitos del cliente en un sistema software.

La división en etapas permite fijar metas inmediatas y controlar mejor el avance del proyecto.

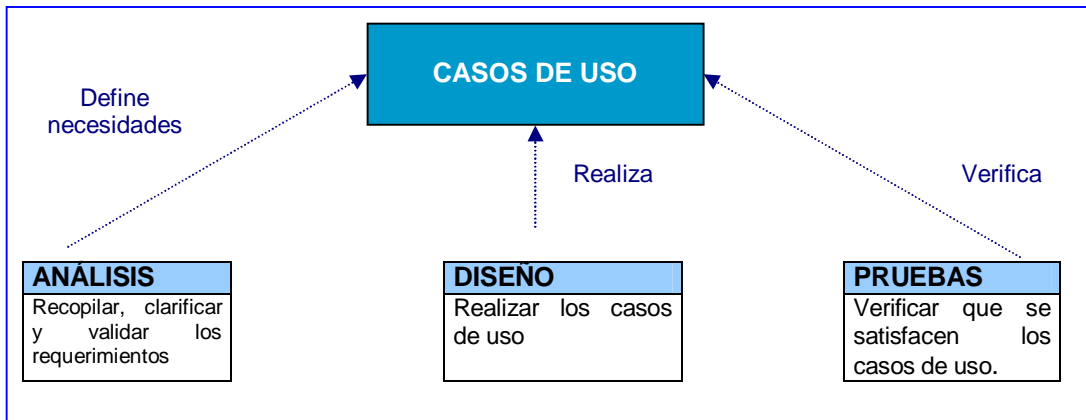
3.4.1.1 Descripción del Proceso Unificado.

El proceso unificado es flexible y extensible, entre sus ventajas se destaca que: permite varias estrategias de desarrollo, se pueden definir diferentes conjuntos de productos, actividades y encargados de las mismas. Se apoya en tres principios básicos:

- **Dirigido por Casos de Uso.** Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante.

³ JACOBSON, Ivar, BOOCH, Grady y RUMBAURGH, James. El proceso unificado de desarrollo de software. 1 ed. España, Addison Wesley, 2000. p. 4.

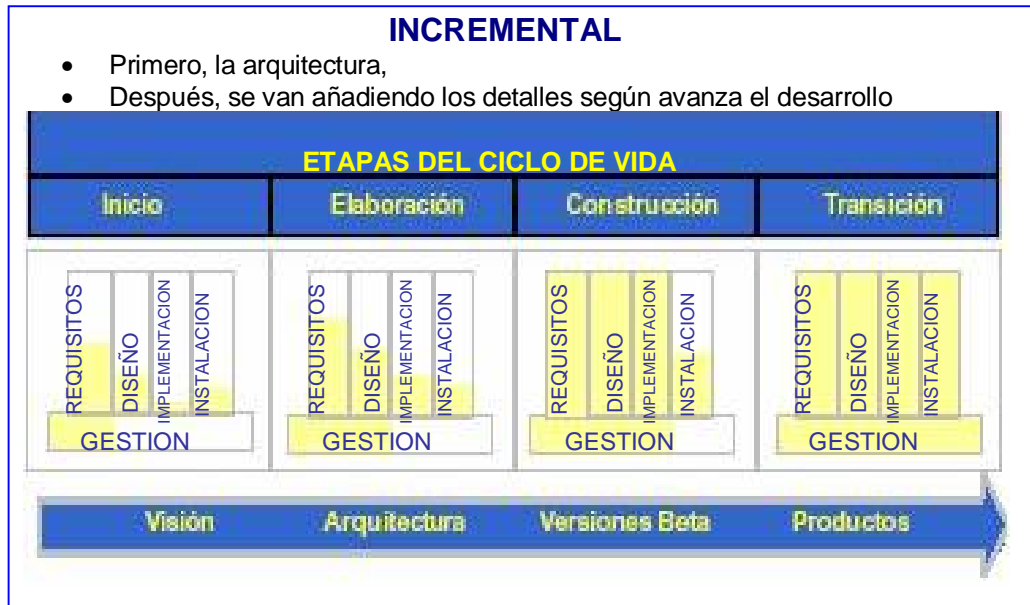
Figura 10. Principios básicos – Casos de uso.



Los casos de uso representan los requisitos funcionales, no sólo son una herramienta para especificar los requisitos de un sistema, sino que también dirigen y controlan el proceso de desarrollo en su totalidad: Desde la especificación hasta el mantenimiento; la unión de estos constituye el modelo de casos de uso que describe la funcionalidad total del sistema.

- **Centrado en la Arquitectura.** La arquitectura es la pieza clave que permite comprender, organizar y determinar la forma del sistema. Describe elementos fundamentales como subsistemas, dependencias, interfaces, colaboraciones y nodos. Es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado.
- **Proceso Iterativo e Incremental.** Una iteración es una secuencia de actividades con un plan establecido y unos criterios de evaluación, cuyo resultado es una versión ejecutable del producto.

Figura 11. Principios básicos – *Proceso iterativo e incremental*



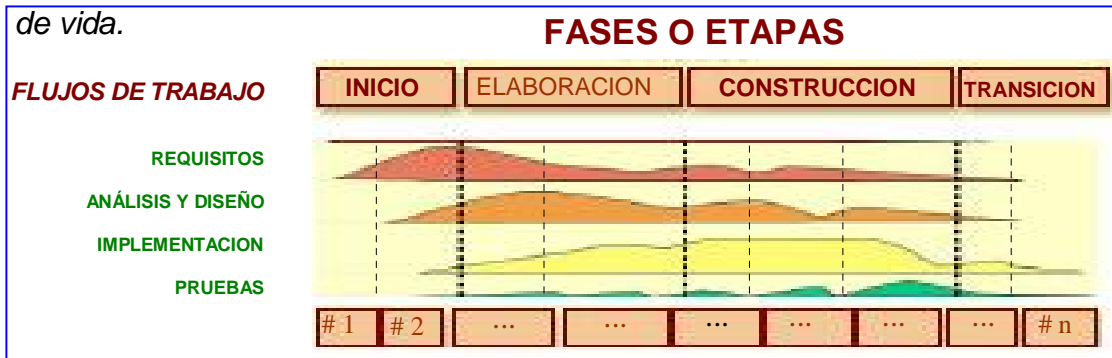
El enfoque iterativo propone un desarrollo progresivo, de tal modo que se atenúen los riesgos y se planteen las cuestiones en el instante en el que se está capacitado para resolverlas. Entre sus ventajas se encuentran: *la detección temprana de riesgos, la administración adecuada del cambio, un mayor grado de reutilización y experiencia para el grupo de desarrollo.* Se caracteriza por:

- Descomposición de un proyecto grande en miniproyectos, mediante una serie de desarrollos cortos.
- Cada miniproyecto es una iteración que debe ser controlada.
- Cada iteración trata un conjunto de casos de uso.
- Se basa en la ampliación y refinación del sistema, es decir, no solo se mejora, sino que el sistema también crece.

3.4.1.2 Elementos del Proceso Unificado. El proceso unificado contiene los siguientes elementos:

- **Fases del Ciclo de Vida.** Cada ciclo se desarrolla a lo largo del tiempo y concluye con una versión del producto. Este ciclo a su vez, se divide en cuatro fases o etapas.

Figura 12. Elementos del proceso unificado- *Flujos de trabajo en las cuatro etapas del ciclo de vida.*



Durante la *fase de inicio* se desarrolla una descripción del producto final y se presenta el análisis del negocio hasta el punto necesario para la puesta en marcha del proyecto. En esta fase, se identifican y priorizan los riesgos más importantes, se planifica en detalle la fase de elaboración, y se estima el proyecto de manera aproximada.

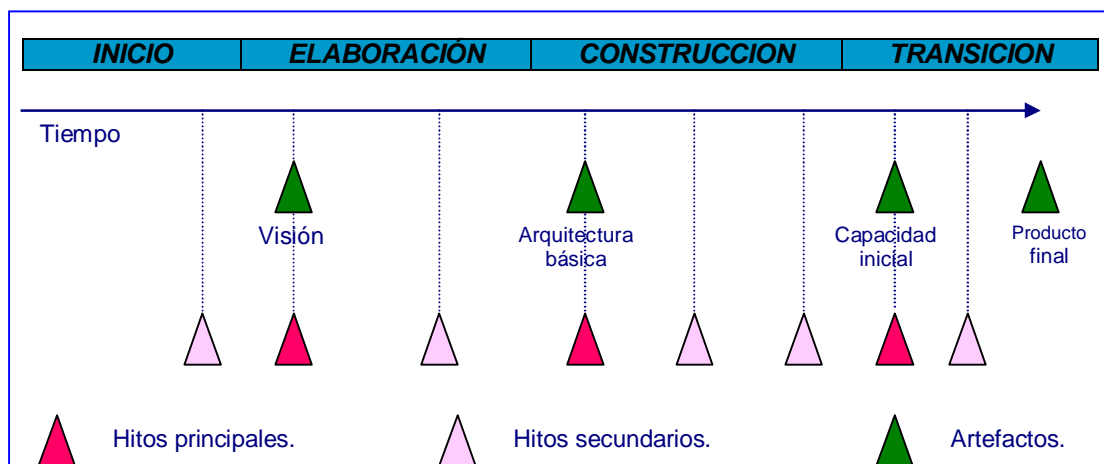
En la *fase de elaboración* se especifican en detalle la mayoría de los casos de uso y se diseña la arquitectura. Durante esta fase se realizan los casos de uso más críticos identificados y el resultado es una línea base de la arquitectura.

Durante la *fase de construcción* se crea el producto. En esta fase, la línea base de la arquitectura crece hasta convertirse en el sistema completo. La descripción evoluciona hasta convertirse en un producto preparado para ser entregado a los usuarios.

La *fase de transición* conlleva a actividades como la fabricación, formación del cliente, el proporcionar una línea de ayuda y asistencia, y la corrección de los defectos que se encuentren tras la entrega.

- **Puntos de Control o Hitos.** Los hitos son puntos de control en los cuales los desarrolladores revisan el progreso del proyecto, buscando sincronizar las expectativas y la realidad, identificar los riesgos y evaluar la situación global del proyecto. Los hitos que se realizan al final de cada fase son principales y los realizados al final de cada iteración son secundarios.

Figura 13. Elementos del proceso unificado – *Hitos principales y secundarios.*



- **Disciplinas o Flujos de Trabajo.** Organizan actividades fundamentales de gestión y desarrollo del proyecto, se pueden solapar en el tiempo son: requisitos, análisis, diseño, implementación y pruebas.
 - *Requisitos:* Las características provenientes de todas las fuentes de información se convierten en requisitos candidatos. Se deben encontrar actores y casos de uso y determinar su prioridad.

- *Análisis:* Se construye la primera versión del modelo de análisis basado en el conjunto inicial de casos de uso detallados.
 - *Diseño:* Se formulan modelos para soportar los requisitos y se descomponen los trabajos de implementación en partes más manejables, con un punto de partida y llegada para cada una de las actividades.
 - *Implementación:* Se desarrolla la arquitectura del sistema en términos de componentes, es decir, el código fuente y los ejecutables que concluirán con la integración del sistema como un todo.
 - *Prueba:* Se prueba el resultado de la implementación mediante las pruebas de cada uno de sus componentes, así como la versión final del sistema. Las construcciones en las que se detecten defectos se prueban nuevamente y si es necesario se devuelven a otros flujos de trabajo como el diseño o la implementación.
- **Artefactos.** Son el resultado de las actividades de los flujos de trabajo, cualquier producto parcial o final usado durante el proyecto. Son las entradas y salidas de las actividades. Se construyen de forma incremental.

3.4.2 Metodología de Desarrollo para la Herramienta DNS con UDP

Dado que en la elaboración de un producto software frecuentemente se detecta la conveniencia de introducir cambios a fin de cumplir a cabalidad con los objetivos propuestos, se adoptó el proceso unificado como la metodología más adecuada para el desarrollo de la herramienta porque:

- Permite establecer una mejor comunicación entre las diferentes áreas que participaron en su desarrollo a través de sus diferentes diagramas.
- Brinda la posibilidad de incluir nuevos requisitos durante la elaboración del proyecto, permitiendo considerar necesidades que no habían sido tenidas en cuenta inicialmente y que surgen a lo largo del proceso de desarrollo de la herramienta.
- Proporciona una guía para ordenar las actividades que se llevarán a cabo y que requieren continua verificación durante su desarrollo.

Cabe resaltar, que solo se abordaron las tres primeras etapas: inicio, elaboración y construcción; sin involucrar la etapa de transición, debido a que ésta implica la realización de una versión incrementada del producto.

3.4.3 El Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

UML es una especificación de notación orientada a objetos. Divide cada proyecto en un número de diagramas que representan las diferentes vistas y la arquitectura del proyecto. UML introduce nuevos diagramas que

representan una visión dinámica del sistema, la cual permite detectar problemas de la estructura en la fase de diseño al propagar errores o de las partes que necesitan ser sincronizadas, así como del estado de cada una de las instancias en cada momento. En resumen, un sistema debe estar bien diseñado, pero también debe funcionar bien.

UML también intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

UML es ahora un estándar, no existe otra especificación de diseño orientado a objetos, ya que es el resultado de las tres opciones existentes en el mercado. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otro ramo.

UML permite la modificación de todos sus miembros mediante estereotipos y restricciones. Un estereotipo nos permite indicar especificaciones del lenguaje al que se refiere el diagrama de UML. Una restricción identifica un comportamiento forzado de una clase o relación, es decir mediante la restricción estamos forzando el comportamiento que debe tener el objeto al que se le aplica.

3.4.3.1 Elementos. Son los principales constituyentes de un modelo y corresponden a las abstracciones que se realizan del sistema. Los hay de cuatro tipos:

- **Elementos Estructurales.** Son las partes estáticas de un modelo, necesarias para representar las cosas materiales o conceptuales, pueden clasificarse en siete tipos:
 - *Clases:* sirven en la descripción de objetos con atributos, operaciones, relaciones y semántica comunes.
 - *Interfaces:* Son una colección de operaciones que especifican un servicio de la clase o componente.
 - *Colaboraciones:* Son el conjunto de roles y otros elementos que generan un comportamiento holístico.
 - *Casos de Uso:* Son una descripción de las acciones propias de un sistema.
 - *Clases Activas:* Son clases cuyas instancias son objetos activos.
 - *Componentes:* Son una parte física y reemplazable que representa el empaquetamiento físico de elementos lógicos.
 - *Nodos:* Son elementos físicos que representan un recurso computacional.

- **Elementos de Comportamiento.** Son las partes dinámicas de un modelo y representan las acciones de éste. Los hay de dos tipos:
 - *Interacciones:* Son comportamientos que comprenden los mensajes relacionales entre los objetos para llevar a cabo un propósito definido.
 - *Máquinas de Estado:* Son comportamientos que especifican los estados que puede tener un objeto en respuesta a un evento.

- **Elementos de Agrupación.** Se utilizan para la organización de los modelos en UML. El principal elemento de agrupación es el paquete, que sirve para concentrar los elementos en grupos dentro de los que pueden incluirse los elementos estructurales, de comportamiento y a sí mismos.
- **Elementos de Anotación.** Son las partes explicativas de un modelo. Como su nombre lo indica permiten hacer anotaciones o comentarios de cualquier elemento del modelo, bien sea para describirlo, hacer una aclaración o una observación de éste.

3.4.3.2 Relaciones. Representan el conjunto de conexiones existentes entre los diferentes elementos de un sistema. Debe procurarse no exceder el número de relaciones con el fin de no saturar el modelo, buscando un equilibrio que lo haga fácilmente comprensible. En UML existen tres tipos de relaciones básicas.

- **Dependencias.** Son relaciones de uso que se utilizan para simbolizar que el cambio ocurrido en un elemento afecta directamente a otro.
- **Generalizaciones.** Se utilizan para relacionar las clases que se consideran generales con otras que son más específicas.
- **Asociaciones.** Permiten establecer conexiones entre los objetos de un elemento con objetos pertenecientes a otros elementos.

3.4.3.3 Diagramas. Las representaciones gráficas de los elementos y sus relaciones se pueden ver desde diferentes perspectivas que dan origen a diferentes tipos de diagramas para generar distintas proyecciones de un sistema. Existen nueve tipos de diagramas:

- **Diagrama de Clases.** Despliegan relaciones entre clases, interfaces y colaboraciones. Se utilizan para modelar sistemas orientados a objetos, siendo la base para realizar los diagramas de componentes y de despliegue. Es considerada la vista de diseño estática de un sistema.
- **Diagrama de Objetos.** Muestra las relaciones que en un momento dado se establecen entre los objetos. Se encargan de mostrar desde una perspectiva estática el diseño o los procesos de un sistema.
- **Diagrama de Casos de Uso.** Se emplea para modelar una vista estática de un sistema mostrando las relaciones existentes entre los casos de uso y los actores. Se consideran de vital importancia en el modelado del comportamiento de un sistema.
- **Diagrama de Secuencias.** Permite la ordenación temporal de los mensajes. Es más adecuado para observar la perspectiva cronológica de las interacciones.
- **Diagrama de Colaboración.** Enfatizan la interacción entre los objetos que envían y reciben mensajes. Ofrece una mejor visión espacial mostrando los enlaces de comunicación entre objetos y se obtiene a partir del correspondiente diagrama de secuencias.
- **Diagrama de Estados.** Tiene como objeto mostrar la máquina de estados, la cual es el conjunto de estados por los que atraviesa un objeto a lo largo de su vida en respuesta a eventos. Un estado es una condición

o situación en la que un objeto satisface una condición, realiza una actividad o espera un evento.

- **Diagrama de Actividades.** Como su nombre lo indica representa el flujo de actividades dentro de un sistema y es un caso especial de diagrama de estados. Se utiliza para modelar su funcionamiento mostrando el flujo de control entre actividades, para obtener la perspectiva dinámica del sistema.
- **Diagrama de Componentes.** Muestra la distribución entre un conjunto de componentes y sus dependencias. Se utiliza para modelar los aspectos físicos de los sistemas orientados a objetos. Este tipo de diagrama es un tipo especial de diagrama de clases que sólo utiliza los componentes del sistema.
- **Diagrama de Despliegue.** Al igual que los diagramas de componentes se encarga de modelar los aspectos físicos del sistema mostrando la configuración de los nodos y cada uno de los componentes que se alojan en ellos.

3.5 FUNDAMENTACION TECNICA

3.5.1 Multimedia

La multimedia hace referencia a la forma de presentar información que emplea una combinación de texto, sonido, imágenes, animación y vídeo que el usuario a través del computador puede manipular y controlar. La mayoría de las aplicaciones multimedia incluyen asociaciones predefinidas conocidas

como hipervínculos, que permiten a los usuarios moverse por la información de modo más intuitivo e interactivo.

Los productos multimedia, bien planteados, permiten que una misma información se presente de múltiples maneras, utilizando cadenas de asociaciones de ideas similares a las que emplea la mente humana.

3.5.1.1 Elementos Multimedia. Entre los elementos multimedia que apoyan el proceso de aprendizaje se encuentran:

- **Elementos Visuales.** La imagen es un elemento primordial de las aplicaciones multimedia. Cuanto mayor y más nítida sea una imagen y cuantos más colores tenga, más difícil es de presentar y manipular en la pantalla de un ordenador. Las aplicaciones multimedia también pueden incluir animación para dar movimiento a las imágenes. Las animaciones son especialmente útiles para simular situaciones de la vida real, también puede realzar elementos gráficos y de vídeo añadiendo efectos especiales, esto ofrece la posibilidad de establecer una comunicación efectiva con el usuario debido al impacto sensorial que produce las representaciones pictóricas.
- **Elementos de Sonido.** El sonido, igual que los elementos visuales, tiene que ser grabado y formateado de manera que la computadora pueda manipularlo y usarlo en presentaciones. Recientemente se han incorporado formatos de audio con una gran capacidad de compresión, lo que ha permitido incluir elementos de sonido importantes, estos elementos proporcionan apoyo a los demás elementos multimedia, reforzando el papel que desempeñan el texto y las imágenes en el

proceso de aprendizaje, además pueden ser utilizados para enriquecer los ambientes de enseñanza que motiven al usuario.

- **Elementos de Organización.** Los elementos multimedia incluidos en una aplicación necesitan un entorno que lleve al usuario a interactuar con la información y a aprender. Entre los elementos interactivos están los menús desplegables, pequeñas ventanas que aparecen en la pantalla del ordenador con una lista de instrucciones o elementos multimedia para que el usuario elija. Las barras de desplazamiento, que suelen estar situadas en un lado de la pantalla, permiten al usuario moverse a lo largo de un documento o imagen extensa.

La integración de los elementos de una presentación multimedia se ve reforzada por los hipervínculos, los cuales conectan de manera creativa los diferentes elementos de una presentación multimedia a través de texto coloreado o subrayado o de una pequeña imagen denominada icono, que el usuario señala con el cursor o puntero y activa haciendo click con el mouse.

Por otra parte, existe el hipertexto, que es el modo en que se escriben los documentos multimedia y los documentos Web, este método de presentación de la información permite hacer una lectura no secuencial de la misma. Trata de emular el modo en que el cerebro humano almacena y recupera la información, por medio de asociación de ideas, y no en el orden en el que los sucesos ocurrieron. En el hipertexto, la información se organiza en torno a una serie de palabras clave, que permiten que, al llegar a ellas, el documento pueda avanzar hacia otra parte del mismo (delante o detrás de esa posición) o presentar información que se encuentra en otro documento.

La conectividad que proporcionan los hipertextos hace que los programas multimedia no sean sólo presentaciones estáticas con imágenes y sonido, sino una experiencia interactiva infinitamente variada e informativa.

3.5.1.2 Beneficios de una Presentación Multimedia.

- **Impacto.** Al incorporar imágenes, efectos de sonido, vídeo y animación para crear presentaciones vivas y de extraordinaria calidad.
- **Flexibilidad.** El material digital puede ser fácil y rápidamente actualizado y presentado a través de innumerables medios.
- **Control por parte del Emisor.** Al seleccionar la cantidad y tipo de información y la forma como la entregará.
- **Control por parte del Receptor.** Al elegir la información que quiere recibir y el momento en que desea recibirla.
- **Costo Beneficio.** Al aprovechar todos los materiales existentes e incorporarlos a la presentación multimedia, utilizando la misma para múltiples finalidades y a través de diversos medios; ahorrando recursos en materiales impresos difíciles de actualizar y presentándola en innumerables ocasiones sin ninguna restricción.

La interactividad que ofrece multimedia significa que el usuario tiene el control y puede acceder a la información precisa que está buscando, profundizando en los tópicos que son de interés e ignorando aquellos que no conoce bien, a su propio ritmo y en el momento en que él lo decida. A

diferencia de un video o una presentación convencional (diapositivas, láminas de computadora, acetatos, etc.) la interactividad permite participar activamente, estimulando la curiosidad del usuario y permitiendo que este imponga su voluntad.

3.5.2 Bases de Datos

Son un conjunto exhaustivo de datos estructurados, organizados independientemente de su utilización y su implementación en hardware, accesibles en tiempo real y compatibles con usuarios concurrentes que poseen necesidades de información diferentes; su estructura facilita su mantenimiento y acceso de una forma estándar. La información se organiza en campos y registros. Un campo se refiere a un tipo o atributo de información, y un registro, a toda la información sobre un individuo. Los datos pueden aparecer en forma de texto, números, gráficos, sonido o vídeo. Normalmente las bases de datos presentan la posibilidad de consultar datos, bien los de un registro o los de una serie de registros que cumplan una condición. También es frecuente que se puedan ordenar los datos o realizar operaciones sencillas. Para facilitar el ingreso de los datos en la base se suelen utilizar formularios; también se pueden elaborar e imprimir informes sobre los datos almacenados.

3.5.2.1 Ventajas de las Bases de Datos.

- Independencia de datos y tratamiento, ya que un cambio de datos no implica un cambio en programas y viceversa, lo que se traduce en un menor costo de mantenimiento.

- Coherencia de resultados, permitiendo reducir la redundancia con acciones lógicamente únicas, evitando la inconsistencia de los datos.
- Mejora en la disponibilidad de datos.
- Permite el cumplimiento de ciertas normas y restricciones de seguridad.
- Posibilitan una gestión más eficiente de almacenamiento.

3.5.2.2 Diseño de Base de Datos. Existen distintos modos de organizar la información y representar las relaciones entre los datos en una base de datos. Los cuatro modelos lógicos de bases de datos utilizados por los sistemas administradores convencionales son: el jerárquico, de redes, el relacional y el orientado a objetos. Cada modelo lógico tiene ciertas ventajas de procesamiento y de negocio para hacer un seguimiento de las entidades, atributos y relaciones.

El modelo relacional es el más extendido hoy día, utilizado en el desarrollo de la base de datos necesaria para la herramienta, por su sencillez en el diseño y mantenimiento, baja complejidad de programación y alta flexibilidad.

- **Modelo de Base de Datos Relacional.** Es un tipo de base de datos o sistema de administración de bases de datos, que almacena la información en varias tablas (filas y columnas de datos) o ficheros independientes y realiza búsquedas que permiten relacionar datos que han sido almacenados en más de una tabla.

Las tablas de la base de datos relacional deben tener un campo común, es decir, un campo que almacena, en cada una de ellas, la misma

información para cada registro y que va a ser el que permita establecer la relación al realizar las consultas. El resultado de la búsqueda produce una tercera tabla que combina los datos solicitados de ambas tablas.

En otras palabras, una base de datos relacional utiliza los valores coincidentes de campos comunes de dos tablas para relacionar información de ambas; es decir, opera sobre los datos, no actúa sobre registros sino sobre operaciones completas y es posible codificar de una manera simple las consultas complejas por claves múltiples sobre varios archivos.

Por lo general, los productos de bases de datos para microcomputadoras o microordenadores son bases de datos relacionales.

3.6 SOFTWARE EN EL DESARROLLO DE LA HERRAMIENTA

3.6.1 Lenguaje de Programación JAVA

3.6.1.1 Introducción. Java fue creado en 1991 por un grupo de ingenieros de Sun Microsystems. Debido a la existencia de distintos tipos de CPUs y a los continuos cambios, era importante conseguir una herramienta independiente del tipo de CPU utilizada. Desarrollan un código “neutro” que no depende del tipo de ordenador, el cual se ejecuta sobre una “máquina hipotética o virtual” denominada Java Virtual Machine (JVM). Es la JVM quien interpreta el código neutro convirtiéndolo a código particular de la CPU utilizada. Java, como lenguaje de programación para computadores, se introdujo a finales de 1995. La clave fue la incorporación de un intérprete Java en el programa Netscape Navigator, versión 2.0, produciendo una verdadera revolución en Internet. Java 1.1 apareció a principios de 1997,

mejorando sustancialmente la primera versión del lenguaje. Java es un lenguaje ideal para aprender la informática moderna, porque incorpora los conceptos (clases, paquetes, etc.) de un modo estándar, mucho más sencillo y claro que otros lenguajes. Esto es consecuencia de haber sido diseñado más recientemente y por un único equipo. El principal objetivo del lenguaje Java es llegar a ser el “nexo universal” que conecte a los usuarios con la información, esté ésta situada en el ordenador local, en un servidor de Web, en una base de datos o en cualquier otro lugar.

La compañía Sun describe el lenguaje Java como “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico”. Además de una serie de halagos por parte de Sun hacia su propia criatura, el hecho es que todo ello describe bastante bien el lenguaje Java.

3.6.1.2 Definición. Los programas desarrollados en Java presentan diversas ventajas frente a los desarrollados en otros lenguajes como C/C++. La ejecución de programas en Java tiene muchas posibilidades:

- Ejecución como aplicación independiente.
- Ejecución como Applet⁴.
- Ejecución como Servlet⁵.

⁴ Los Applets son pequeñas aplicaciones a la que se accede en un servidor de Internet, se transmiten a través de la red, se instalan automáticamente y se ejecutan como parte de un documento de la red.

⁵ Un Servlet es una aplicación sin interfaz gráfica que se ejecuta en un servidor de Internet.

Además, Java permite fácilmente el desarrollo tanto de arquitecturas cliente-servidor como de aplicaciones distribuidas, consistentes en crear aplicaciones capaces de conectarse a otros ordenadores y ejecutar tareas en varios ordenadores simultáneamente, repartiendo por lo tanto el trabajo. Aunque también otros lenguajes de programación permiten crear aplicaciones de este tipo, Java incorpora en su propio API estas funcionalidades.

3.6.1.3 Características de JAVA. Las características principales que nos ofrece Java respecto a cualquier otro lenguaje de programación, son:

- Es *SIMPLE*. Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos.
- Es *ORIENTADO A OBJETOS*. Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje, trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: *encapsulación*, *herencia* y *polimorfismo*.
- Es *DISTRIBUIDO*. Java se ha construido con extensas capacidades de interconexión TCP/IP. Proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando.
- Es *ROBUSTO*. Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación

de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo.

- Es de *ARQUITECTURA NEUTRAL*. Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará.
- Es *SEGURO*. La seguridad en Java tiene dos facetas. En el lenguaje, características como los punteros o el *casting* implícito que hacen los compiladores de C y C++ se eliminan para prevenir el acceso ilegal a la memoria. Cuando se usa Java para crear un navegador, se combinan las características del lenguaje con protecciones de sentido común aplicadas al propio navegador.
- Es *PORTABLE*. Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo.
- Es *INTERPRETADO*. El intérprete Java (sistema run-time) puede ejecutar directamente el código objeto.
- Es *MULTITHREADED*. Java permite muchas actividades simultáneas en un programa.
- Es *DINAMICO*. Java se beneficia todo lo posible de la tecnología orientada a objetos.

3.6.1.4 Entorno de Desarrollo de JAVA. Existen distintos programas comerciales que permiten desarrollar código Java. La compañía Sun⁶, creadora de Java, distribuye gratuitamente el Java Development Kit (JDK). Se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en Java. Incorpora además la posibilidad de ejecutar parcialmente el programa, deteniendo la ejecución en el punto deseado y estudiando en cada momento el valor de cada una de las variables (es el denominado Debugger). Cualquier programador con un mínimo de experiencia sabe que una parte muy importante (muchas veces la mayor parte) del tiempo destinado a la elaboración de un programa se destina a la detección y corrección de errores. Existe también una versión reducida del JDK, denominada JRE (Java Runtime Environment) destinada únicamente a ejecutar código Java (no permite compilar).

Los IDEs (Integrated Development Environment), tal y como su nombre indica, son entornos de desarrollo integrados. En un mismo programa es posible escribir el código Java, compilarlo y ejecutarlo sin tener que cambiar de aplicación. Algunos incluyen una herramienta para realizar Debug gráficamente, frente a la versión que incorpora el JDK basada en la utilización de una consola (denominada habitualmente ventana de comandos de MS-DOS, en Windows NT/95/98) bastante difícil y pesada de utilizar. Los entornos integrados permiten desarrollar las aplicaciones de forma mucho más rápida, incorporando en muchos casos librerías con componentes ya desarrollados, los cuales se incorporan al proyecto o programa. Como inconvenientes se pueden señalar algunos fallos de compatibilidad entre

⁶ Compañía estadounidense de computadoras fundada en 1983, con sede central en Mountain View, California.

plataformas y ficheros resultantes de mayor tamaño que los basados en clases estándar.

3.6.2 Interbase 6.02

Interbase es un Sistema Administrador de Bases de Datos (DBMS), actualmente desarrollado y comercializado por Borland⁷. Se distingue de otros DBMS's por su corta trayectoria, sus pocos requisitos de administración y su arquitectura multi-generacional. Es una base de datos completa que nos permite administrar la información y mantenerla íntegra y segura, además tiene algunas características como los procedimientos almacenados que facilitan el desarrollo de las aplicaciones.

Interbase es en general un servidor de bases de datos relacional muy estable y seguro; es compatible con sistemas operativos como Linux, Microsoft Windows, y Solaris. La versión 6.02 es de código abierto.

3.6.3 Java DataBase Connectivity (JDBC)

3.6.3.1 Definición. JDBC es una especificación de procedimientos y estándares para usar fuentes de datos en aplicaciones y applets, es decir, permite su conexión a bases de datos relacionales. En otras palabras, es un API⁸ de JAVA para permitir ejecutar instrucciones SQL⁹ formada por una

⁷ Borland International, Inc., empresa estadounidense de software y lenguajes de programación para ordenadores o computadoras, con sede en Scotts Valley, California.

⁸ API (Application Programming Interface), interfaz de programación de aplicación. Documento que contiene las especificaciones para el lenguaje de programación Java.

⁹ SQL (Structured Query Language), lenguaje estructurado de consultas. Es un lenguaje de alto nivel para crear, manipular, examinar y gestionar bases de datos relacionales.

colección de interfaces y clases abstractas, que deben utilizar todos los fabricantes de drivers que quieran realizar una implementación de su driver 100% JAVA..

Debido a que JDBC está escrito completamente en JAVA también posee la ventaja de ser independiente de la plataforma, no será necesario escribir un programa para cada tipo de base de datos, una misma aplicación escrita utilizando JDBC podrá manejar diferentes DBMS's. Además podrá ejecutarse en cualquier sistema que posea una Máquina Virtual de Java, es decir, serán aplicaciones completamente independientes de la plataforma.

Con la habilidad de JAVA para integrarse con DBMS's comerciales y su naturaleza orientada al manejo de la Red, es posible crear un ambiente ideal tipo *cliente-servidor*. Muchas compañías están usando bases de datos relacionales para manejar la información en sus sitios Web. Las bases de datos relacionales son ideales para el almacenamiento de grandes cantidades de información, la cual puede ser accedida por muchos usuarios.

Hoy en día, la mayoría de los gestores de bases de datos relacionales tienen soporte para la utilización de interfaces HTML¹⁰. Para algunas aplicaciones, las páginas HTML favorecen la interfaz, pero, en aplicaciones más complejas se presentan ciertas limitaciones que no permiten generar un buen trabajo. En estos casos, es conveniente la utilización de lenguajes de programación como JAVA, que permitan elaborar aplicaciones para generar la interfaz con la base de datos. En realidad, todo lo que se pueda hacer en C++ se puede

¹⁰ HTML (HyperText Markup Language), lenguaje de marcas de hipertexto. Formato estándar de los documentos que circulan en la World Wide Web (WWW).

hacer con JAVA. Y mejor aún, al combinar JAVA con JDBC, se presentan nuevas expectativas para comunicarse con las bases de datos a través de un esquema similar al de las aplicaciones en C y C++. Se puede definir en cuatro frases lo que hace JDBC:

- Establece una conexión con una BD, que puede ser remota o no.
- Envía sentencias SQL a la BD.
- Manipula los datos.
- Procesa los resultados obtenidos de la BD.

3.6.3.2 Tipos de Drivers JDBC. Los drivers permiten la conexión con una base de datos determinada. Existen cuatro tipos de drivers JDBC, cada uno presenta una filosofía de trabajo diferente, a continuación se describen cada uno de ellos:

- **Puente JDBC-ODBC.** Este driver conecta JAVA con el ODBC¹¹ de Microsoft vía métodos nativos.
 - *Ventajas:* Dada la gran popularidad de ODBC, casi todas las bases de datos cuentan actualmente con un driver ODBC. Por esta razón, la conectividad con Java se puede hacer inmediatamente sin hacer ningún desarrollo adicional.
 - *Desventajas:* El uso de métodos nativos implica que esta conectividad no se puede usar en applets. Además la conexión se vuelve más lenta por usar ODBC.

¹¹ ODBC (Open Database Connectivity), conectividad abierta a base de datos.

- **Driver Mixto Java - API Nativo.** Este driver convierte las llamadas JDBC en llamados a métodos nativos del cliente de la base de datos, típicamente es desarrollado por el mismo fabricante de la base de datos.
 - *Ventajas:* Casi todos los motores tienen un cliente ya probado y depurado, que hace que el desarrollo de estos driver pueda ser implementado fácilmente por todos los proveedores.
 - *Desventajas:* No se puede usar en applets debido al llamado a métodos nativos.

- **Driver Gateway.** Este driver traduce los llamados JDBC en un protocolo independiente del motor, que luego es traducido a un protocolo específico de la base de datos en un servidor intermedio. Este servidor (gateway) puede conectar los clientes Java con muchos proveedores de bases de datos diferentes usando un sólo driver en el cliente.
 - *Ventajas:* Esta es la alternativa más flexible para conectarse con diferentes motores de bases de datos simultáneamente. Además, como el driver está totalmente desarrollado en Java, puede funcionar en applets y aplicaciones.
 - *Desventajas:* El uso de una máquina intermedia hace necesariamente más lenta la transferencia de datos.

- **Driver JDBC 100% Java Puro con Protocolo Nativo.** Este driver convierte todas las llamadas JDBC a un protocolo de red específico al manejador de bases de datos. Esta es la solución ideal para casi todas las situaciones.

- *Ventajas:* La eliminación de intermediarios, hace que el rendimiento al acceder a la base de datos sea óptimo. Al ser un driver sólo Java, puede funcionar tanto en applets como en aplicaciones.
- *Desventajas:* Es el tipo de driver más difícil de implementar por los proveedores de bases de datos, porque involucra desarrollo en el cliente y en el servidor. Muchos proveedores no ofrecen aún esta forma de conectividad.

La forma más apropiada de acceder bases de datos en JAVA es vía JDBC. Para usarlo debemos conseguir un driver JDBC apropiado para el motor. Los mejores son los Gateway y el driver 100% Java con Protocolo Nativo.

3.6.3.3 Interclient. Es un driver totalmente desarrollado en Java para proveer conectividad JDBC con Interbase, éste incorpora un protocolo remoto JDBC para intercambiar y almacenar datos entre un cliente y un servidor.

Interclient permite que el browser habilitado como cliente acceda a los datos de Interbase a través de la red sin que el cliente tenga librerías previamente instaladas. Es un tipo de driver Gateway.

3.6.4 Lenguaje de Marcas de Hipertexto (HTML)

HTML es un lenguaje descriptivo que tiene como objeto dar formato al texto y las imágenes que se pretenden visualizar en el navegador.

A partir de este lenguaje se pueden introducir enlaces, seleccionar el tamaño de las fuentes o intercalar imágenes, todo esto de una manera prefijada y en

ningún caso inteligente. En efecto, el HTML no permite el realizar un simple cálculo matemático o crear una página de la nada a partir de una base de datos. A decir verdad, el HTML, aunque muy útil a pequeña escala, resulta bastante limitado a la hora de concebir grandes sitios o portales.

Es esta deficiencia del HTML la que ha hecho necesario el empleo de otros lenguajes accesorios mucho más versátiles y de un aprendizaje relativamente más complicado, capaces de responder de manera inteligente a las demandas del navegador y que permiten la automatización de determinadas tareas tediosas e irremediables como pueden ser las actualizaciones, el tratamiento de pedidos de una tienda virtual.

Los documentos HTML contienen dos tipos de información: la que se muestra en la pantalla (texto, imágenes...) y los códigos (tags o etiquetas), transparentes al usuario, que indican cómo se debe mostrar esa información. Para crear un documento HTML, basta con disponer de un editor de texto. También se pueden utilizar editores específicos que insertan automáticamente las etiquetas correspondientes al formato del texto que se escriba o a las imágenes que se inserten; casi todos los procesadores de texto actuales incluyen esta posibilidad.

En un documento HTML hay etiquetas que indican los atributos del texto (negrita, centrado...). Otras indican al sistema cómo debe responder a eventos que genere el usuario. La etiqueta más importante es la que indica un vínculo (link), que puede contener como destinatario otro lugar del mismo documento o el URL de otro documento; este último puede residir en el mismo lugar de la Web que el documento actual o en cualquier otro ordenador de WWW.

A medida que se ha ido avanzando, se han estandarizado distintas versiones del lenguaje HTML. Cada una de ellas amplía el número de etiquetas, lo que permite nuevas posibilidades para los documentos; así, se le ha dotado de marcas para rellenar formularios (forms) de manera interactiva, que permiten al usuario enviar la información necesaria para realizar consultas en bases de datos, comprar o solicitar un servicio. Otras marcas permiten mejorar la presentación de los documentos, por ejemplo, añadiendo fondos, tablas de contenido o textos intermitentes.

El encargado de interpretar las etiquetas y de mostrar el documento en la pantalla es el navegador. La evolución de las distintas versiones del lenguaje HTML ha estado determinada por un desarrollo paralelo de los navegadores, que permitiesen interpretar las nuevas etiquetas. El avance de unos y otros ha contribuido al crecimiento exponencial que ha experimentado WWW

4 DESARROLLO DE LA HERRAMIENTA

4.1 ANALISIS DE REQUISITOS

Para el estudio de los requisitos del software se tuvo en cuenta factores como:

Complejidad e importancia del tema. El Sistema de Nombres de Dominio (DNS) es un tema muy interesante, dado que hace parte esencial del mundo de las comunicaciones, aunque su función es transparente al usuario, su papel es vital en el momento en que éste ingresa a internet. Por otra parte el Protocolo de Datagramas de Usuario (UDP), complementa el proceso realizado por el DNS mediante el transporte de datagramas.

Software disponible para la implantación de la herramienta. Dada la gran variedad de software que existe actualmente, se eligió la plataforma JAVA por ser un lenguaje robusto, y que ofrece diversas opciones que permiten la elaboración de una herramienta completa, de fácil entendimiento e interfaz agradable. Además, JAVA es de código libre y es ideal para trabajar en entorno web, éste combinado con DBMS's como Interbase 6.02 que también es de código libre, permiten la construcción de un software práctico y a bajo costo.

4.1.1. Falencias en el proceso enseñanza-aprendizaje

La informática hace posible una verdadera enseñanza individualizada. Permite el planteamiento de prácticas adaptadas al nivel real del usuario. Cualquier libro de texto ofrece la información y hace planteamientos de

acuerdo con un usuario ideal que casi nunca existe en la realidad. En el caso específico de las asignaturas, son muchos los docentes que se quejan de que los libros de texto adoptados tienen un nivel alto para sus alumnos, o demasiado bajo, según los casos. Y, si bien los libros constituyen un material valiosísimo e imprescindible, nunca se puede contar con el libro ideal para cada uno de los alumnos, ni siquiera para cada nivel. Ahora bien, constituiría una gran ventaja que los docentes dispusieran de medios que, sin llevarles mucho tiempo, les permitieran secuenciar la información según las exigencias del programa académico y del nivel cognitivo de los alumnos y les facilitaran la creación de prácticas adecuadas dirigidas a los alumnos que les permita comprender en forma rápida y didáctica un tema específico. Tales medios están vinculados necesariamente a la Informática. De ahí la necesidad de que los docentes en ejercicio y, más aún, los futuros maestros, estudiantes en la actualidad, reciban la formación necesaria para demandar tales medios, potenciar su desarrollo y servirse de ellos en la actividad profesional.

4.1.2 Especificación de Requisitos

Elaborar un software educativo para el aprendizaje del proceso del DNS con el protocolo UDP considerando que:

- El tema que compone la herramienta es el Sistema de Nombres de Dominio y el Protocolo de Datagramas de Usuario.
- Se debe implementar una base de datos que emule al DNS.

- Se deben especificar laboratorios que permitan simular el proceso de resolución de nombres de dominio y direcciones IP.
- Los laboratorios deben permitir la estructuración del tema a fin de mostrarlo de una manera comprensible y didáctica al usuario.
- Debe existir una guía que especifique: introducción, objetivos, procedimiento y evaluación para cada uno de los laboratorios.
- Es necesario implementar un módulo de ayuda para guiar al usuario en la utilización del software.
- Integrar elementos multimedia para apoyar las estrategias educativas y mantener motivados a los usuarios.
- Se debe ofrecer información en línea al usuario.
- Se debe anexar un módulo de documentación, mostrando los detalles de los temas abordados.

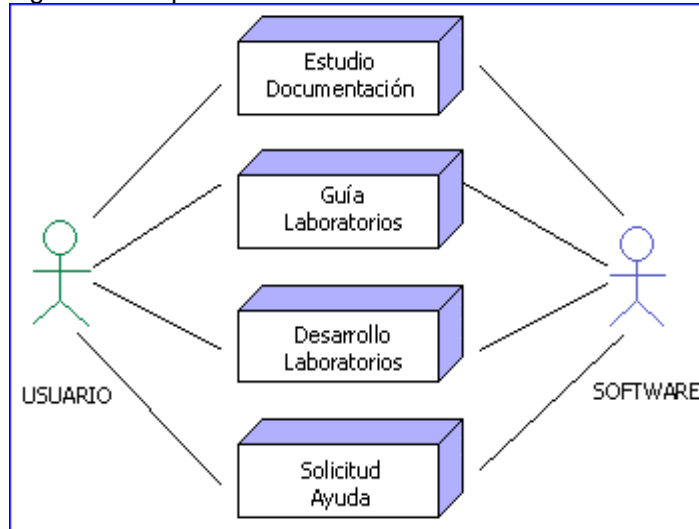
4.2 DISEÑO

Para el diseño de la herramienta, se utilizó el Lenguaje Unificado de Modelado (UML). Inicialmente se hicieron modelos desde el punto de vista del usuario y luego teniendo en cuenta la operabilidad de la herramienta.

Se presentan los diagramas de casos de uso y de actividad para visualizar y documentar el software.

4.2.1 Arquitectura del Sistema

Figura 14. Arquitectura del Sistema.



La arquitectura de la herramienta plantea la necesidad de crear cuatro operaciones fundamentales que involucran al usuario del software como el destinatario de todos los servicios que ésta ofrece. El software es el encargado de satisfacer los requerimientos del usuario, permitiendo su acceso a los diferentes módulos del sistema.

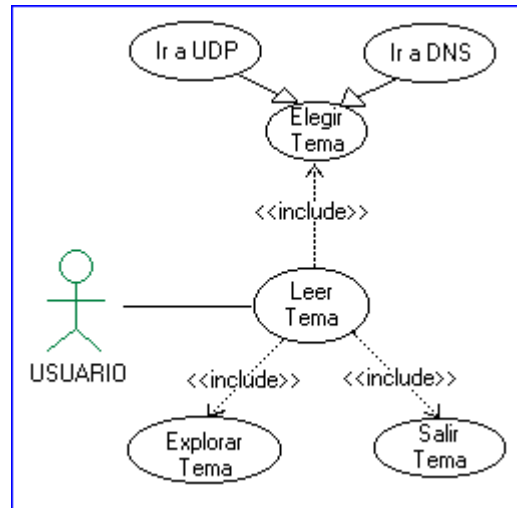
4.2.2 Módulos del Sistema

La herramienta se diseñó en base a los siguientes módulos:

- Módulo de Estudio.
- Módulo de Guías.
- Módulo de Laboratorios.
- Módulo de Ayuda.

4.2.2.1 Módulo de Estudio.

Figura 15. Diagrama Caso de Uso - Módulo de Estudio.



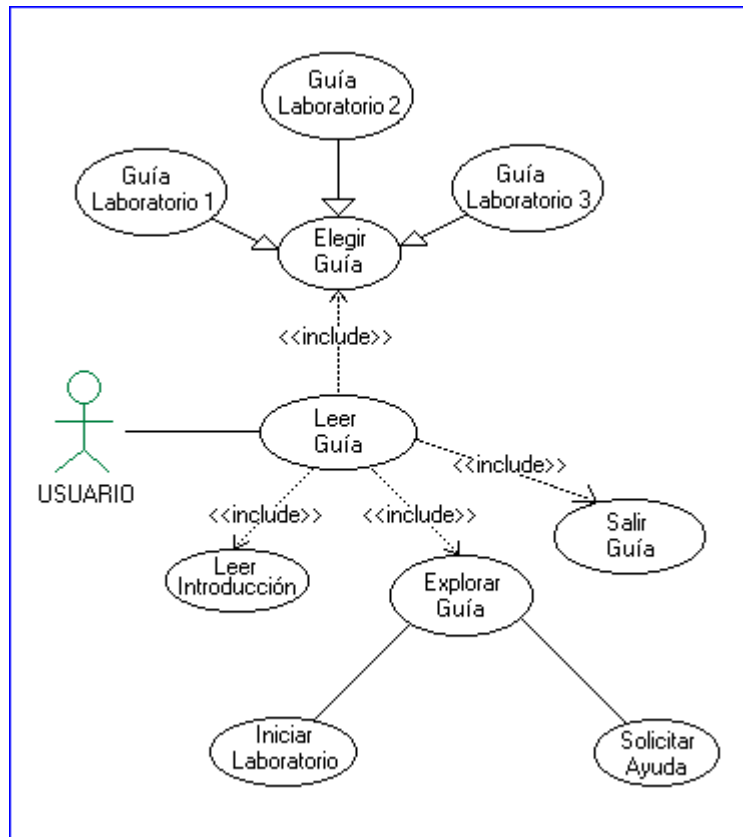
Cuando el usuario decide estudiar uno de los temas, debe elegirlo, teniendo la posibilidad de escoger entre: DNS y UDP.

4.2.2.2 Módulo de Guías.

Cuando el usuario decide leer una guía, debe elegirla teniendo la posibilidad de escoger entre: guía laboratorio 1, guía laboratorio 2 o guía laboratorio 3. Una vez que se encuentre explorando una de ellas, tiene la opción de:

- *Iniciar el laboratorio de acuerdo a guía elegida.* Se muestra la ventana inicial del laboratorio.
- *Solicitar ayuda.* Se muestra la ventana de Ayuda, desde la cual el usuario puede elegir el tema o subtema determinado que desea consultar.

Figura 16. Diagrama Caso de Uso - Módulo de Guías.

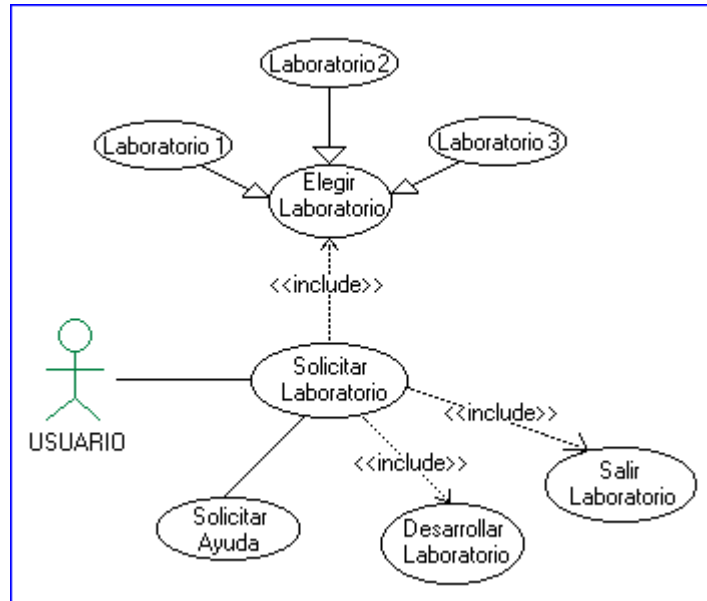


En la siguiente figura se muestra en detalle el diagrama de caso de uso del módulo de guías.

4.2.2.3 Módulo de Laboratorios.

El usuario solicita un laboratorio teniendo la opción de escoger entre los tres laboratorios que existen. Después de haber solicitado un determinado laboratorio puede iniciar el desarrollo de éste o puede solicitar ayuda, ésta le brinda mediante un ejemplo una explicación detallada de cómo realizar el laboratorio. En la figura se explica en general el diagrama de caso de uso que representa el proceso:

Figura 17. Diagrama Caso de Uso – Módulo de Laboratorios.

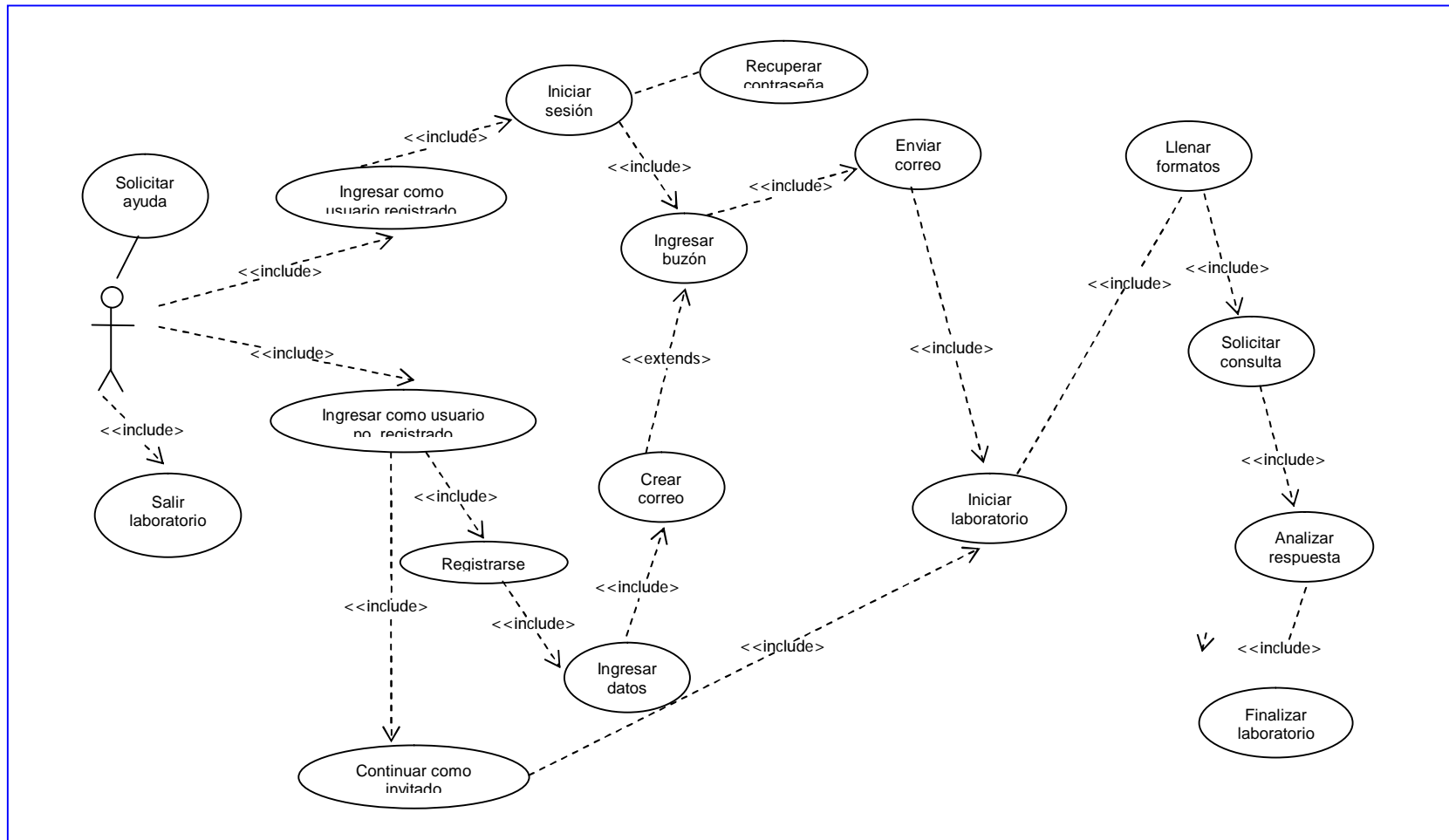


A continuación se detallarán los diagramas de caso de uso para cada uno de los laboratorios desde el punto de vista del usuario y del software.

- **Módulo Laboratorio 1 – Consulta Estándar.** En la figura 18 se muestran las opciones que tiene el usuario en el módulo laboratorio 1.
 - *Ingresar como Usuario Registrado:* Permite el ingreso al laboratorio a un usuario que ya se ha registrado previamente.
 - *Iniciar Sesión:* Permite que un usuario registrado ingrese a su buzón. Si éste olvidó su contraseña, puede recuperarla.
 - *Ingresar como Usuario no Registrado:* Permite el ingreso al laboratorio a un usuario nuevo, ofreciendo dos opciones: registrarse o continuar como invitado.
 - *Continuar como Invitado:* Permite que el usuario inicie el laboratorio sin un registro previo.

- *Registrarse*: Permite que el usuario se registre ingresando los datos respectivos.
- *Crear Correo*: Permite que el usuario cree una cuenta de correo personal para este laboratorio.
- *Ingresar a Buzón*: Muestra el buzón del usuario y le permite enviar un mensaje con el fin de iniciar el laboratorio.
- *Iniciar*: Permite que el usuario empiece a desarrollar el laboratorio siguiendo los demás procesos (llenar formato, solicitar consulta y analizar respuesta).

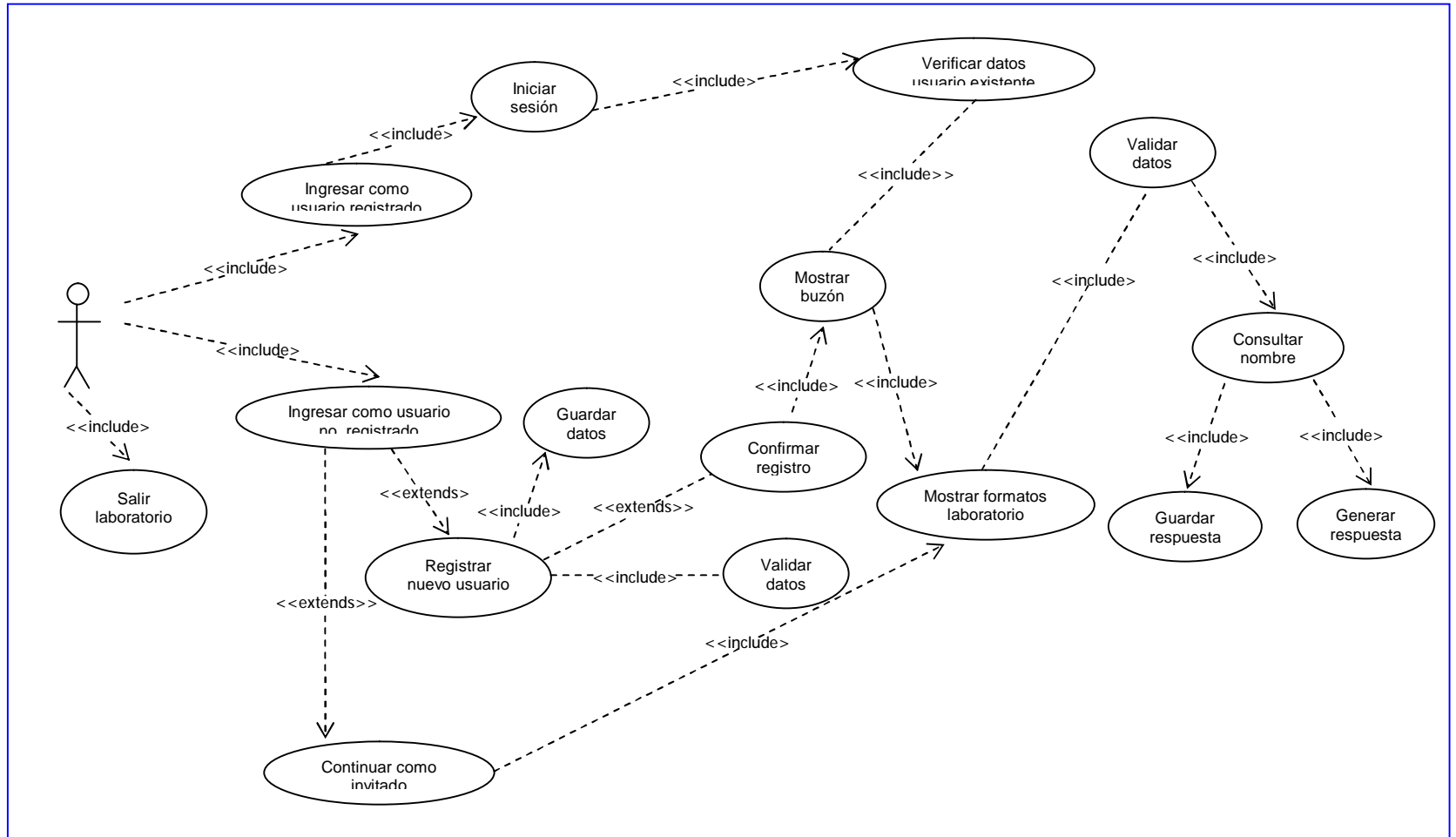
Figura 18. Diagrama de Caso de uso. Módulo Laboratorios- Laboratorio 1 – (Vista usuario).



En la figura 19 se muestra la vista software del módulo laboratorio 1.

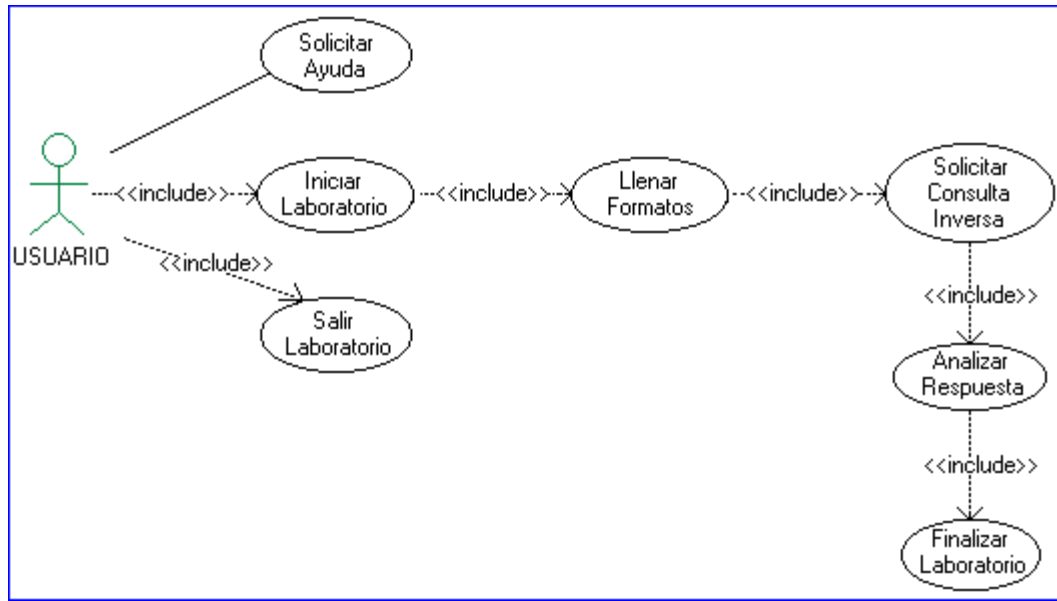
- *Iniciar Sesión:* Valida los datos que el usuario ingresa.
- *Mostrar Buzón:* Muestra la ventana del buzón de acuerdo a los datos del usuario.
- *Registrar Nuevo Usuario:* Muestra las ventanas de registro y crear correo, valida los datos y los almacena.
- *Confirmar Registro:* Muestra un mensaje con información acerca del correo creado.
- *Verificar Datos Usuario Existente:* verifica que el usuario nuevo no esté registrado.
- *Mostrar Formatos Laboratorio:* Muestra los formatos que el usuario debe llenar.
- *Consultar Nombre:* De acuerdo a la consulta solicitada, valida los datos, realiza una búsqueda y arroja los resultados.
- *Generar Respuesta:* Muestra una ventana con los datos consultados, organizados en el formato.
- *Guardar Respuesta:* Almacena los datos consultados.

Figura 19. Diagrama de Caso de uso. Módulo Laboratorios- Laboratorio 1 – (Vista software).



- **Módulo Laboratorio 2 – Consulta Inversa.**

Figura 20. Diagrama de Caso de Uso. Módulo Laboratorios - Laboratorio 2- (Vista usuario).



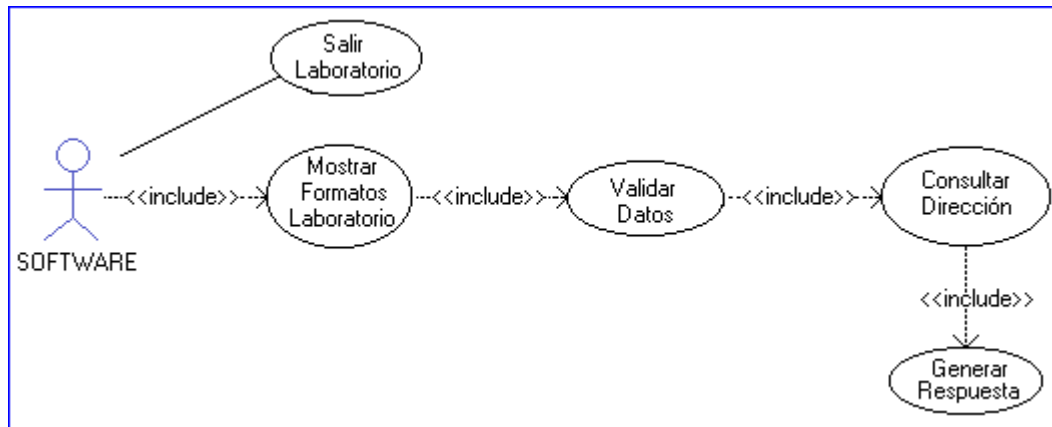
- *Iniciar Laboratorio:* Permite al usuario iniciar el laboratorio.
- *Llenar Formatos:* El usuario introduce los datos de la consulta en los diferentes formatos.
- *Solicitar Consulta:* Inicia la consulta de la dirección.
- *Analizar Respuesta:* Analiza los resultados arrojados y saca conclusiones.

En la figura 21 se muestra la vista software para el caso de uso anterior:

- *Mostrar Formatos del Laboratorio:* Muestra los formatos correspondientes al laboratorio y permite que el usuario ingrese los datos que desea consultar.
- *Validar Datos:* Verifica que los datos que el usuario ingresó estén correctos y que los formatos se hayan llenados adecuadamente.

- *Consultar Dirección:* De acuerdo al tipo de búsqueda solicitado realiza la consulta y arroja los resultados.
- *Generar Respuesta:* Muestra el formato de respuesta con los datos estructurados.

Figura 21. Diagrama de Caso de Uso. Módulo Laboratorios - Laboratorio 2- (Vista software).

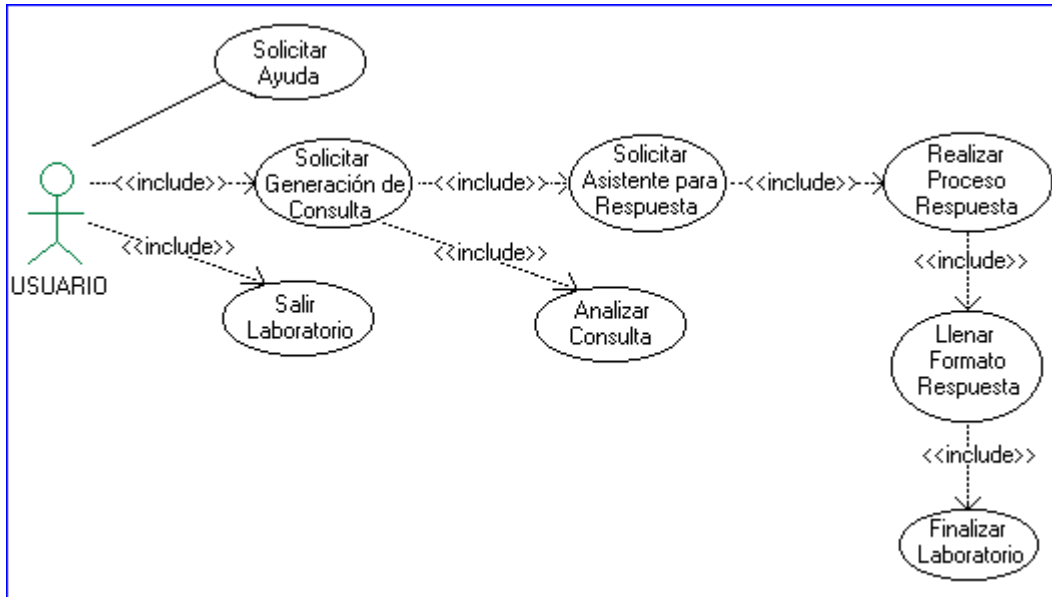


- **Módulo Laboratorio 3 – Respuesta.**

En la figura 22, se muestra el diagrama caso de uso vista usuario. En él se encuentra lo siguiente:

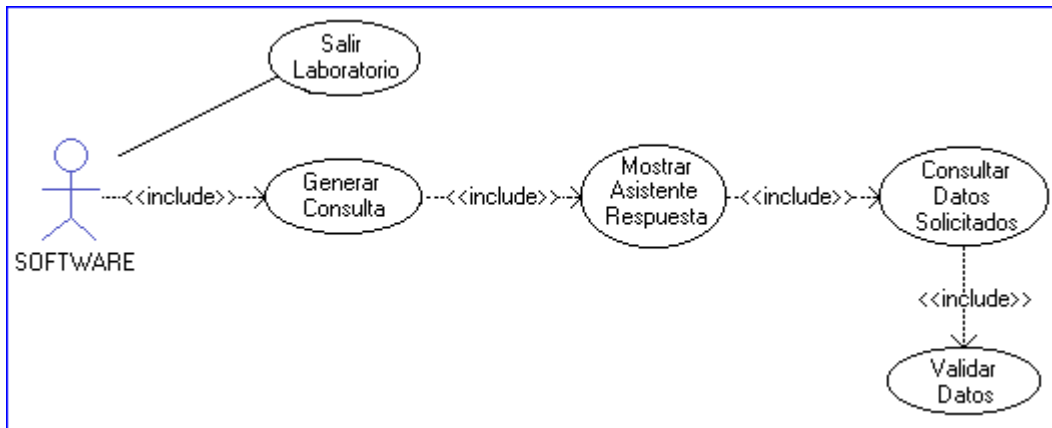
- *Solicitar Generación de Consulta:* Permite al usuario generar una consulta para iniciar el proceso de respuesta.
- *Solicitar Asistente para Respuesta:* Permite al usuario realizar el proceso de respuesta mediante un asistente.
- *Realizar Proceso de Respuesta:* A través del asistente puede realizar diferentes consultas según lo requiera.
- *Llenar Formato de Respuesta:* Permite al usuario ingresar los datos que consultó.

Figura 22. Diagrama de Caso de Uso. Módulo Laboratorios - Laboratorio 3- (Vista usuario).



En la figura 23 se muestra la vista software del caso de uso anterior.

Figura 23. Diagrama de Caso de Uso. Módulo Laboratorios - Laboratorio 3- (Vista software).

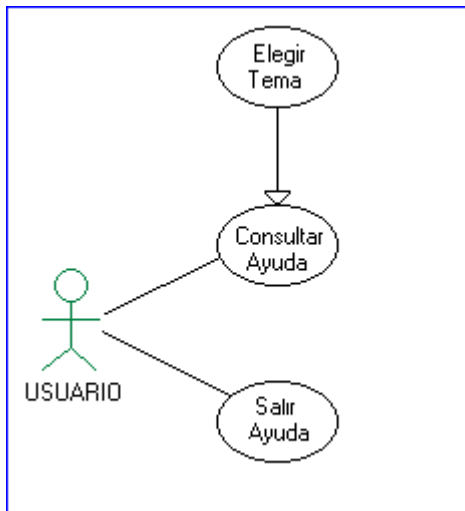


- *Generar Consulta:* Realiza una consulta a la base de datos y muestra información seleccionada aleatoriamente en base a la consulta que generó.

- *Mostrar Asistente Respuesta:* Muestra una ventana con los datos generados que permite que el usuario seleccione la información que requiera.
- *Consultar datos Solicitados:* Realiza las consultas solicitadas por el usuario a través del proceso de respuesta.
- *Validar Datos:* Verifica que el formato de respuesta se haya llenado adecuadamente y que los datos sean correctos.

4.2.2.4 Módulo de Ayuda.

Figura 24. Diagrama Caso de Uso - Módulo de Guías.



El módulo de ayuda le informa al usuario como utilizar la herramienta y lo orienta acerca de los temas.

4.3 IMPLEMENTACION

4.3.1 Descripción de la Base de Datos

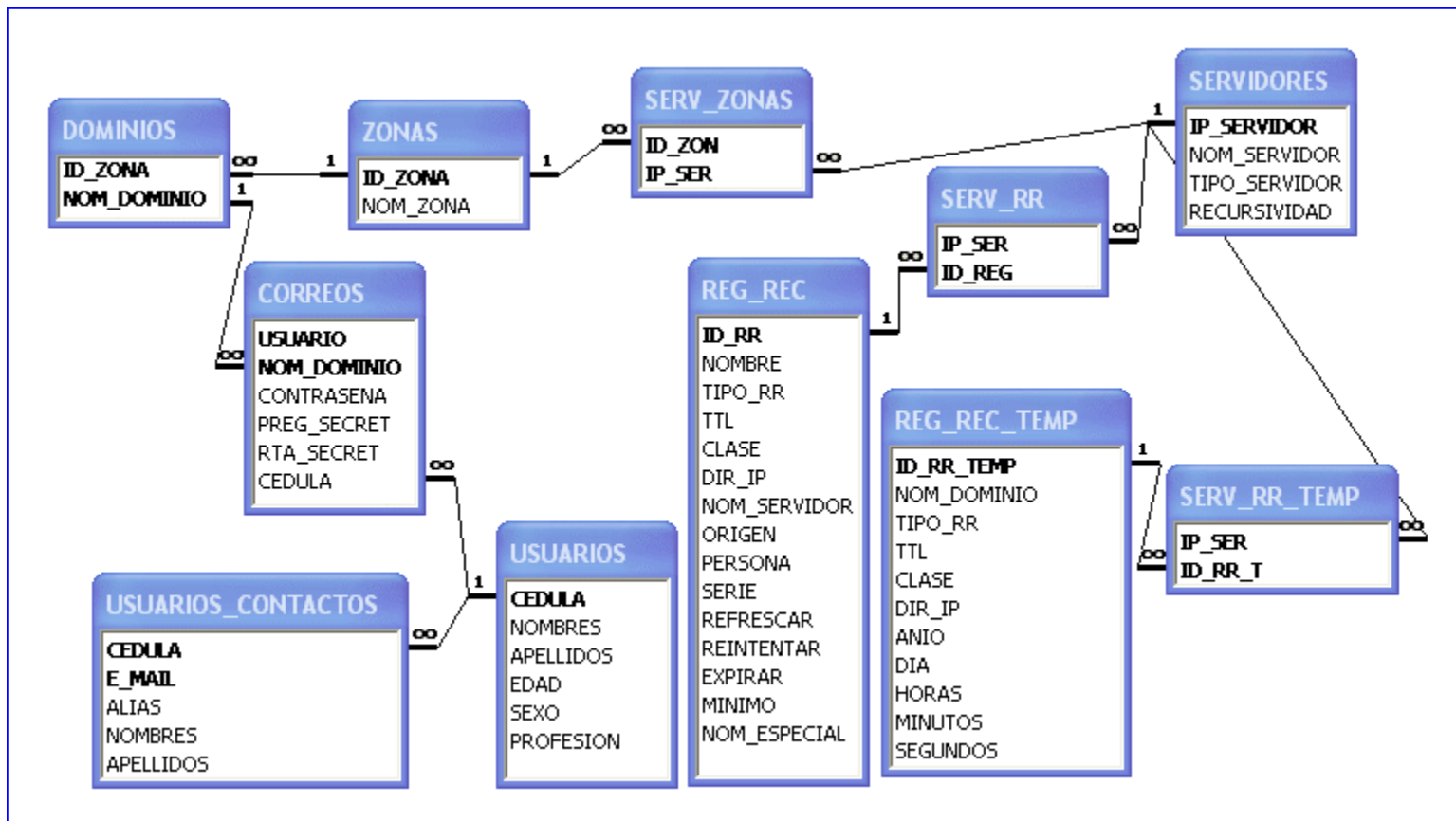
La base de datos del software, se muestra en la Figura 25 y contiene las siguientes tablas:

- **REG_REC:** Contiene los registros de recurso tipo A, NS, SOA pertenecientes a cada nombre de dominio, servidor o zona, según el caso.
 - ID_RR: Llave principal de la tabla, identifica al registro de recurso. Tipo entero.
 - NOMBRE: Nombre de dominio. Tipo texto.
 - TIPO_RR: Indica el tipo de registro de recurso. Tipo texto.
 - TTL: Indica el tiempo de vida en caché. Tipo entero.
 - CLASE: Indica la clase de registro de recurso (IN para Internet). Tipo texto.
 - DIR_IP: Dirección IP. Tipo texto.
 - NOM_SERVIDOR: Nombre de servidor. Tipo texto.
 - ORIGEN: Origen de los datos para la zona. Tipo texto.
 - PERSONA: Correo del administrador de la zona. Tipo texto.
 - SERIE: Versión de la zona. Tipo entero.
 - REFRESCAR: Tiempo en segundos que indica cuando debe refrescar los datos de la zona. Tipo entero.
 - REINTENTAR: Tiempo en segundos para reintentar un refresh fallido. Tipo entero.

- EXPIRAR: Máximo tiempo en segundos para que el servidor secundario use los datos antes de que expiren por no haber conseguido un refresh. Tipo numérico.
- MINIMO: Tiempo mínimo en segundos que puede tomar el TTL de un registro de recurso para esa zona. Tipo entero.
- NOM_ESPECIAL: Nombre especial para un registro tipo PTR, formado por la dirección IP invertida añadiendo el dominio IN-ADDR.ARPA.. Tipo alfanumérico.
- **SERVIDORES:** Contiene los datos de los servidores existentes en la base de datos.
 - IP_SERVIDOR: Llave principal de la tabla, identifica al servidor. Tipo texto.
 - NOM_SERVIDOR: Nombre del servidor. Tipo texto.
 - TIPO_SERVIDOR: Tipo de servidor. Tipo numérico.
 - RECURSIVIDAD: Indica si el servidor es o no recursivo. Tipo entero.
- **SERV_RR:** Contiene los registros de recurso por cada servidor.
 - IP_SER: Dirección IP del servidor. Tipo texto.
 - ID_REG: Identifica el registro de recurso. Tipo entero.
- **REG_REC_TEMP:** Contiene los registros de recurso tipo A pertenecientes a cada nombre de dominio que son cacheados por el servidor.

- ID_RR_TEMP: Llave principal de la tabla, identifica al registro de recurso. Tipo entero.
 - NOM_DOMINIO: Nombre de dominio. Tipo texto.
 - TIPO_RR: Indica el tipo de registro de recurso. Tipo texto.
 - TTL: Indica el tiempo de vida en caché. Tipo entero.
 - CLASE: Indica la clase de registro de recurso (IN para Internet). Tipo texto.
 - DIR_IP: Dirección IP. Tipo texto.
 - ANIO: Año en que se ingresó el registro de recurso. Tipo entero.
 - DIA: Día en que se ingresó el registro de recurso. Tipo entero.
 - HORAS: Hora en que se ingresó el registro de recurso. Tipo entero.
 - MINUTOS: Minutos en que se ingresó el registro de recurso. Tipo entero.
 - SEGUNDOS: Segundos en que se ingresó el registro de recurso. Tipo entero.
- **SERV_RR_TEMP:** Contiene los registros de recurso en caché por cada servidor.
 - IP_SER: Dirección IP del servidor. Tipo texto.
 - ID_RR_T: Identifica el registro de recurso. Tipo entero.
- **ZONAS:** Contiene información de la zona.
 - ID_ZONA: Llave principal de la tabla, identifica la zona. Tipo entero.
 - NOM_ZONA: Nombre de la zona. Tipo texto.

Figura 25. Modelo Entidad Relación.



- **SERV_ZONAS:** Contiene los datos de los servidores por zona.
 - ID_ZON: Identifica la zona. Tipo entero.
 - IP_SER: Dirección IP del servidor. Tipo texto.

- **DOMINIOS:** Contiene información de los dominios.
 - NOM_DOMINIO: Llave principal de la tabla, identifica el dominio. Tipo texto.
 - ID_ZONA: Identifica la zona a la que pertenece el dominio. Tipo entero.

- **USUARIOS:** Contiene información del usuario registrado.
 - CEDULA: Llave principal de la tabla, identifica al usuario. Tipo texto.
 - NOMBRES: Nombre(s) del usuario. Tipo texto.
 - APELLIDOS: Apellido(s) del usuario. Tipo texto.
 - EDAD: Edad del usuario. Tipo entero.
 - SEXO: Sexo del usuario. Tipo texto.
 - PROFESION: Profesión del usuario. Tipo texto.

- **CORREOS:** Contiene información del correo del usuario. La llave principal para esta tabla está conformada por los campos USUARIO y NOM_DOMINIO.
 - USUARIO: Indica el nombre de usuario para el correo. Tipo texto.
 - NOM_DOMINIO: Indica el nombre de dominio para el correo. Tipo texto.

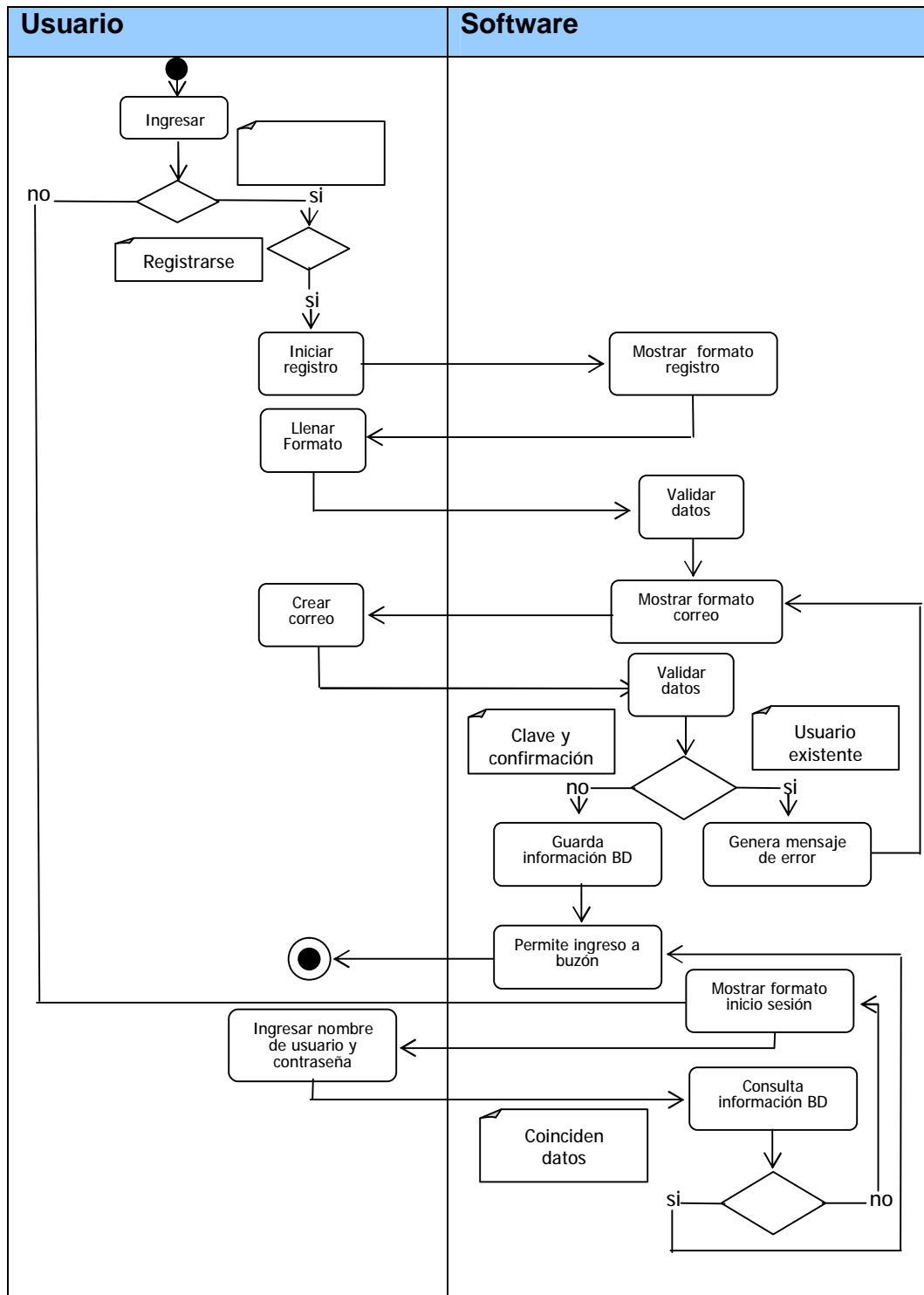
- CONTRASENA: Clave personal del usuario para ingresar al buzón. Tipo alfanumérico.
- PREG_SECRET: Pregunta que ayuda al usuario a recordar la clave en caso de olvido. Tipo texto.
- RTA_SECRET: Respuesta a la pregunta secreta. Tipo texto.
- CEDULA: Identificación del usuario. Tipo texto.
- **USUARIOS_CONTACTOS:** Contiene información de los contactos por usuario. La llave principal está conformada por los campos CEDULA y E_MAIL.
 - CEDULA: Identifica al usuario. Tipo texto.
 - E_MAIL: Dirección de correo del usuario. Tipo texto.
 - ALIAS: Alias para el contacto del usuario. Tipo texto.
 - NOMBRES: Nombres del contacto. Tipo texto.
 - APELLIDOS: Apellidos del contacto. Tipo texto.

4.3.2 Diagramas de Actividad

Un diagrama de actividades es un caso especial de un diagrama de estados en el cual casi todos los estados son estados de acción (identifican que acción se ejecuta al estar en él) y casi todas las transiciones son enviadas al terminar la acción ejecutada en el estado anterior. Estos diagramas fueron elaborados con el objeto de dar detalle los casos de uso del módulo laboratorio, el cuál está dividido en tres módulos: laboratorio 1, 2 y 3. Los siguientes diagramas representan transiciones internas, sin hacer mucho énfasis en transiciones o eventos externos:

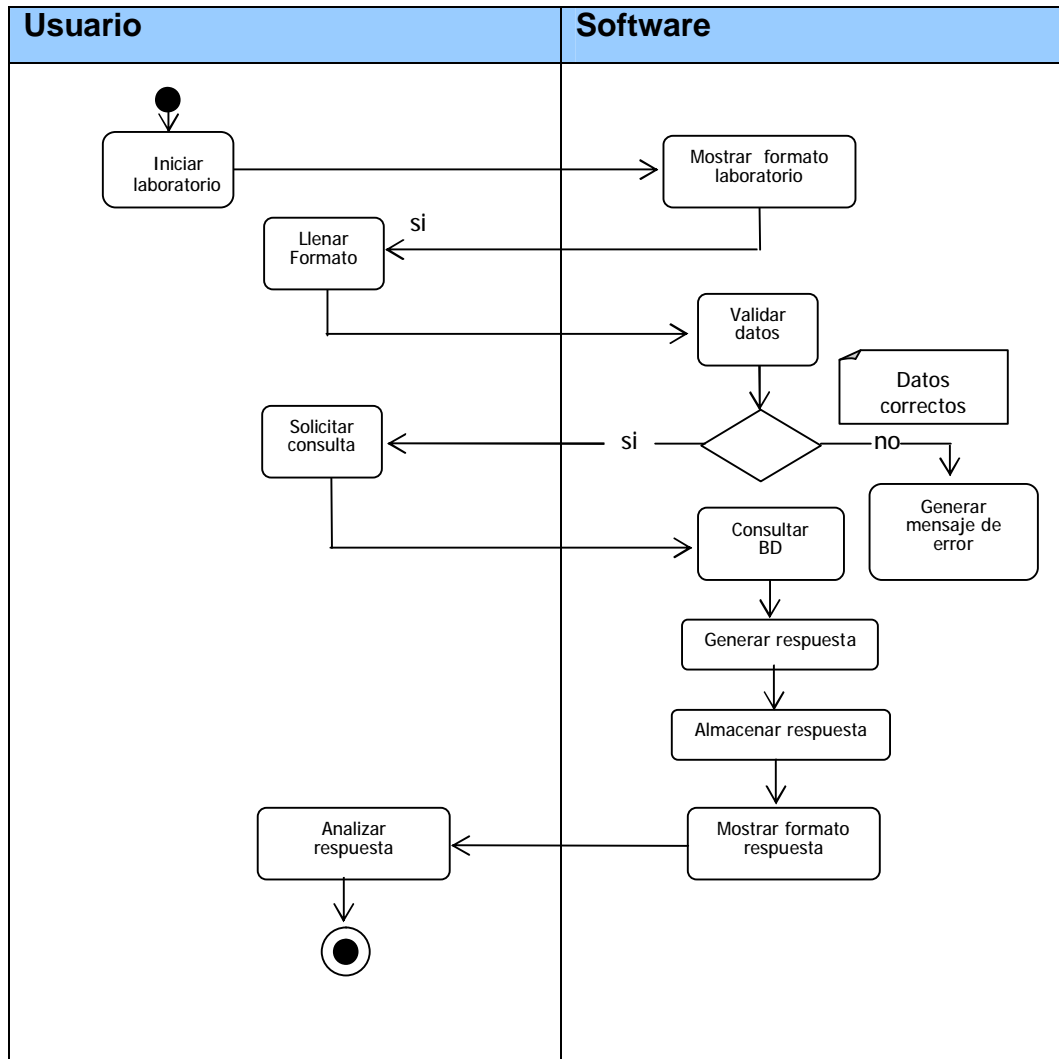
4.3.2.1 Modelo de Actividad. Módulo Laboratorios - Laboratorio 1.

Figura 26. Diagrama de Actividad. Módulo Laboratorios- Laboratorio 1 – Ingreso.



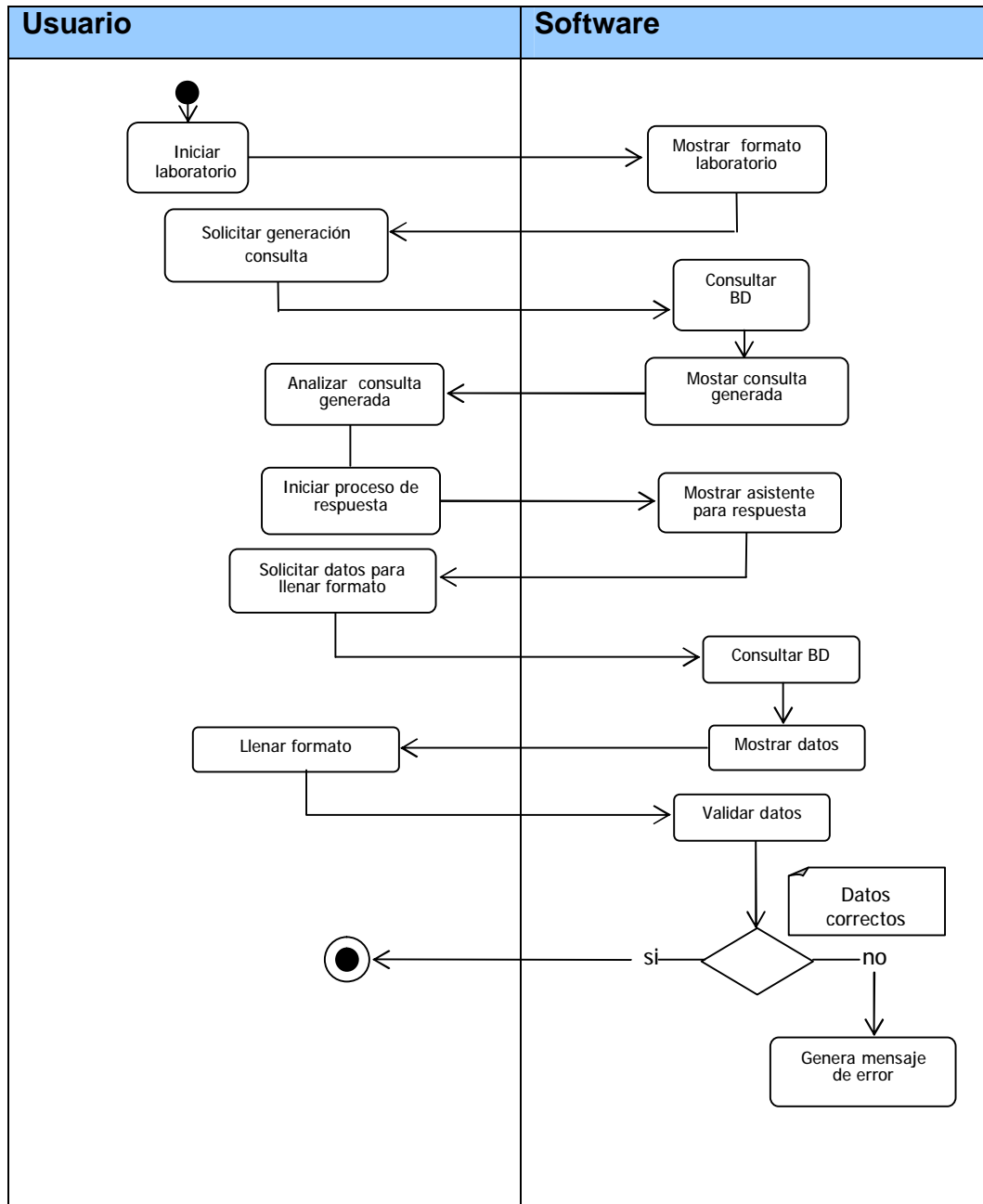
4.3.2.1 Modelo de Actividad. Módulo Laboratorios - Laboratorios 1 y 2.

Figura 27. Diagrama de Actividad. Módulo Laboratorios- Laboratorios 1 y 2 – Desarrollo.



4.3.2.1 Modelo de Actividad. Módulo Laboratorios - Laboratorio 3.

Figura 28. Diagrama de Actividad. Módulo Laboratorios- Laboratorio 3 – Desarrollo.



4.3.3 Funcionamiento de la Herramienta DNS con UDP

A continuación se presenta una descripción de las principales funcionalidades de la herramienta:

Figura 29. Pantalla inicial de la herramienta.



La figura 29 corresponde a la pantalla inicial de la herramienta en ella se muestra el menú desplegado con las diferentes opciones.

Si el usuario elige un ítem (DNS o UDP) del menú Documentación, aparecerá una ventana con el tema estructurado, lo que permite que sea explorado fácilmente. Si por el contrario, el ítem seleccionado es del menú Laboratorios, se mostrará la guía del laboratorio seleccionado, desde la cual podrá dar inicio a éste. En la figura 30 se muestra la pantalla inicial del

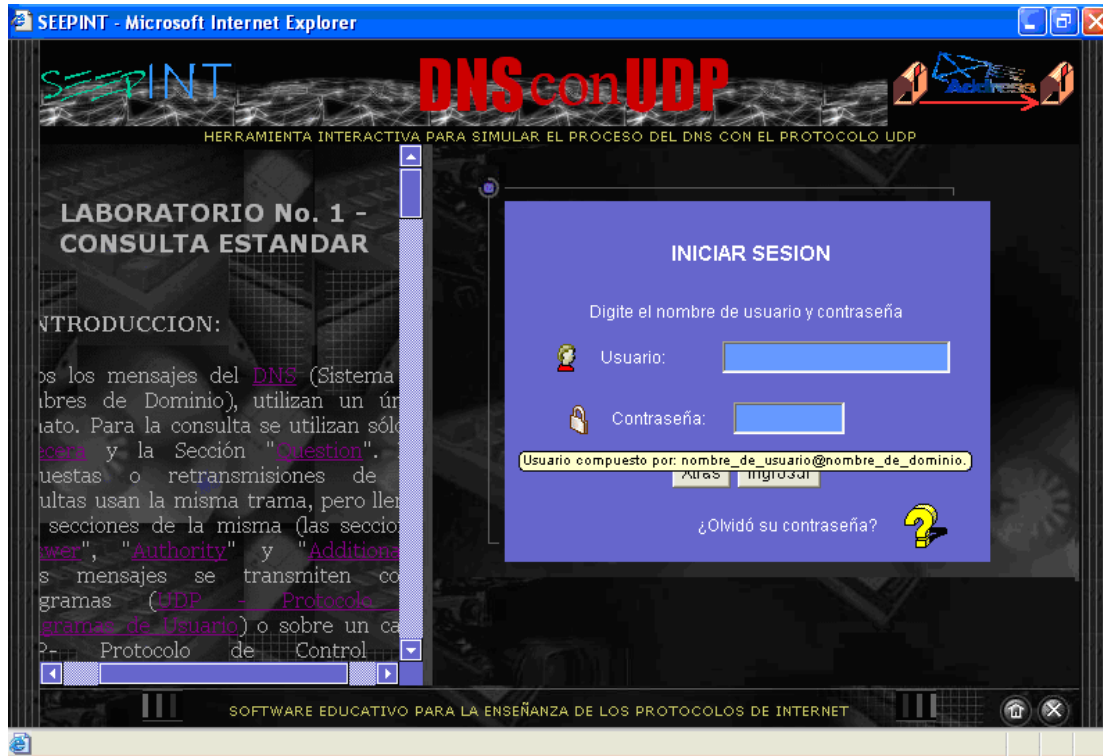
laboratorio 1. En ésta el usuario puede elegir ingresar al laboratorio como usuario registrado o invitado. Se puede observar en la parte izquierda de la figura, la guía del laboratorio, así el usuario puede consultar la guía mientras desarrolla el laboratorio de acuerdo con el procedimiento especificado en ésta, desde la guía también puede acceder al módulo de ayuda.

Figura 30. Pantalla Principal – Laboratorio 1.



Si ingresa como usuario registrado, aparecerá la ventana de la figura 31, donde el usuario podrá iniciar sesión digitando su dirección de correo y contraseña; en caso de haber olvidado su clave podrá crear una nueva desde la ventana que aparece al dar click en la imagen del signo interrogación.

Figura 31. Pantalla Iniciar Sesión – Laboratorio 1.



Si ingresa como usuario no registrado aparece la ventana mostrada en la figura 32. Desde allí el usuario tiene dos opciones: continuar como invitado o registrarse. Si decide continuar como invitado el usuario debe dar click en la imagen que aparece en la ventana, desde allí podrá iniciar el laboratorio. Si decide registrarse, debe dar click en el botón “Registrar”, aparecerá una ventana con un formato de registro en el cual el usuario ingresa los datos solicitados. Posteriormente, se mostrará una ventana que le permite crear un correo, con esto puede finalizar el registro. Una vez finalizado éste, se mostrará un mensaje de confirmación indicando el correo creado e inmediatamente el usuario podrá dar inicio al laboratorio.

Cabe resaltar, que por cualquiera de los dos caminos (registrado o invitado) que elija el usuario, el paso previo al inicio del laboratorio es enviar un correo.

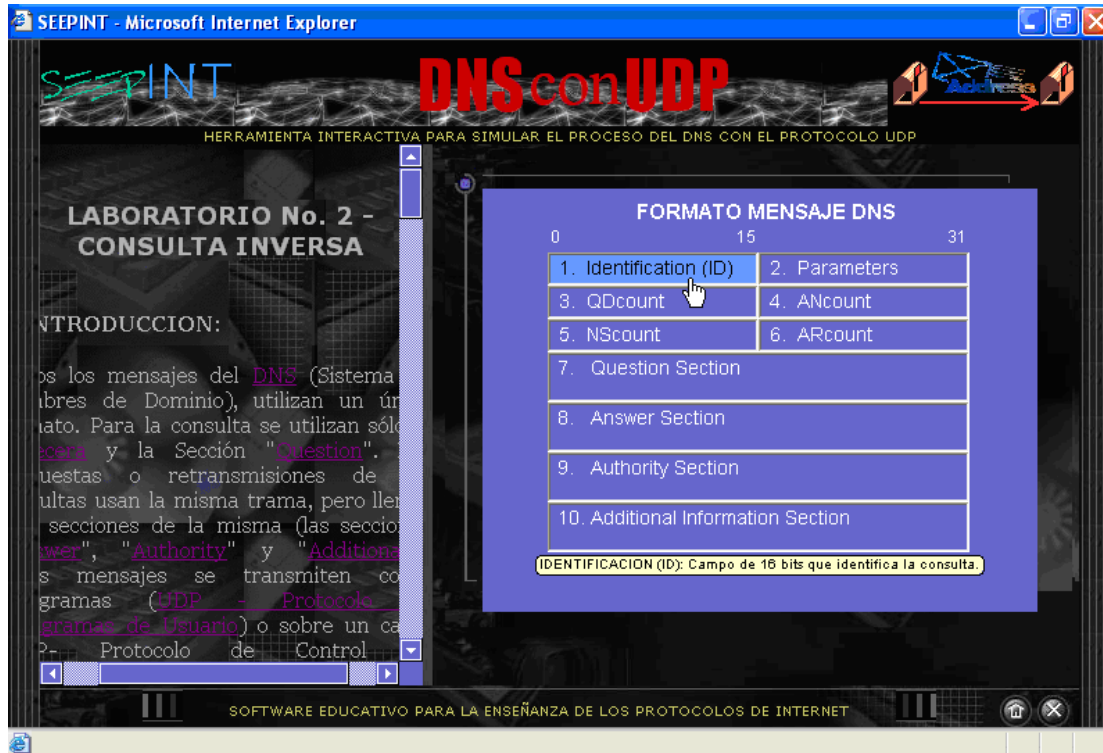
Figura 32. Pantalla Usuario No Registrado – Laboratorio 1.



En la figura 33, se muestra la ventana que da inicio al laboratorio 2. Esta corresponde al formato de mensaje DNS donde el usuario debe llenar los campos adecuadamente para realizar la consulta inversa. Una vez llenado éste formato aparecerán las ventanas de los formatos de datagrama UDP y de la pseudocabecera IP.

La herramienta valida los campos y no permite al usuario continuar con la siguiente ventana hasta haber llenado correctamente dichos campos. Para obtener información, basta con pasar el Mouse sobre el campo deseado y en la parte inferior se mostrará una breve explicación. Si el usuario desea profundizar puede consultar la ayuda, dando click en el botón de ayuda que se encuentra al final de la guía.

Figura 33. Pantalla Inicio Laboratorio 2.



En la figura 34 se muestra la ventana que da inicio al laboratorio 3. Desde allí el usuario podrá generar una consulta para iniciar el proceso de respuesta. Este laboratorio es uno de los más importantes, dado que el usuario va a realizar el papel opuesto al de los laboratorios anteriores, es decir, éste se comportará como el cliente (“Resolver”) al tratar de dar respuesta a la consulta generada por el software. Para esto, contará con el asistente para respuesta que le permitirá realizar consultas a la base de datos según lo requiera. Este proceso se torna muy interesante cuando en la consulta generada se solicita búsqueda recursiva. El software hace una consulta a la base de datos y en base a esta información elige aleatoriamente el nombre a consultar, el tipo de búsqueda y el servidor desde el cuál el usuario debe iniciar el proceso. Algunas de las ventanas del asistente de respuesta se muestran en las figuras 35 y 36.

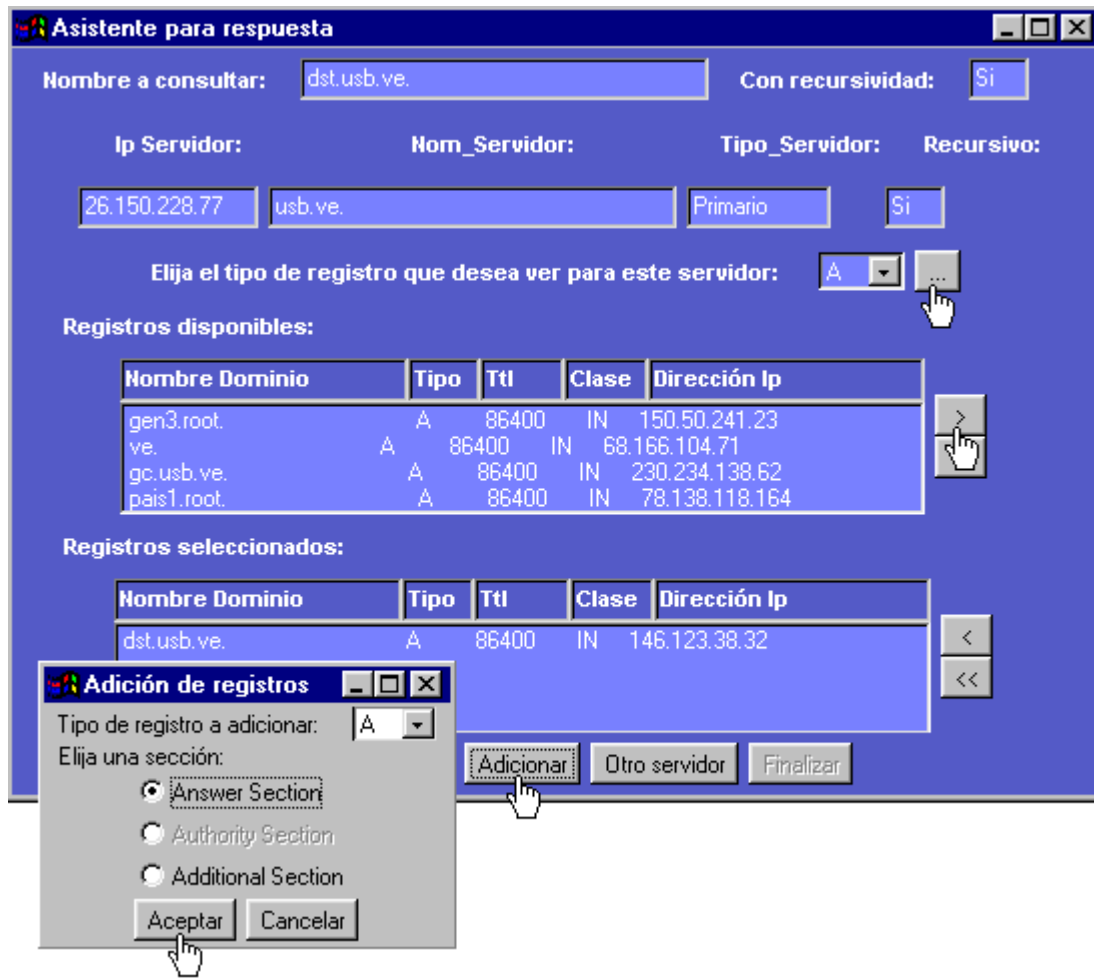
Figura 34. Pantalla Inicio Laboratorio 3.



Figura 35. Asistente para Respuesta - Laboratorio 3.

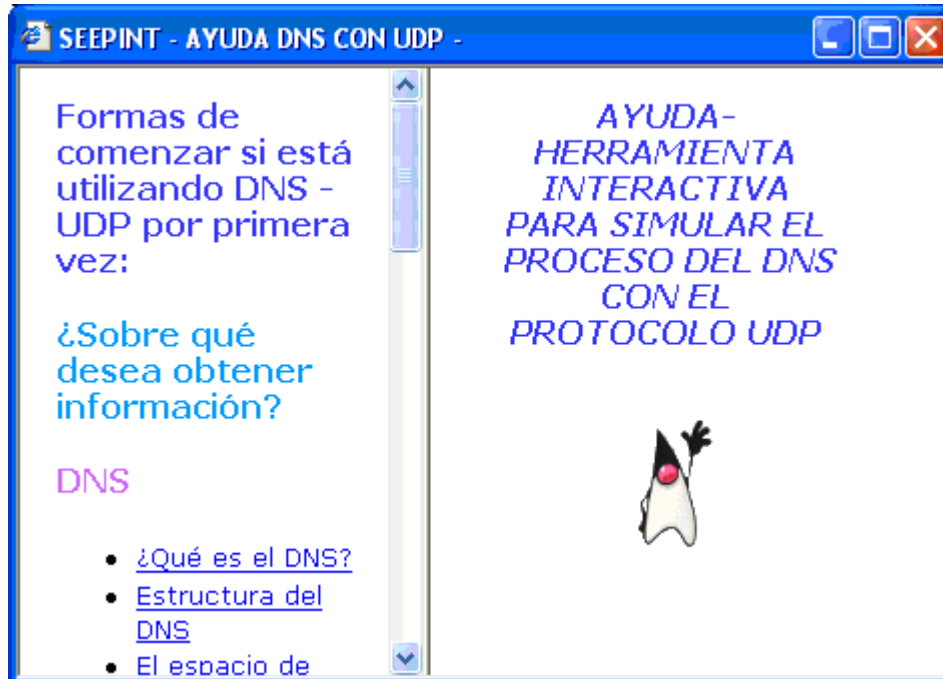


Figura 36. Asistente para Respuesta – Laboratorio 3.



Finalmente, el módulo de ayuda se muestra en la figura 37. Este funciona de una manera muy sencilla. El usuario puede consultarlo en cualquier momento y elegir el tema del cual desea obtener información. La ayuda consta de un menú principal en la parte izquierda de la ventana donde se enuncia los diferentes temas que el usuario puede consultar. No sólo se ofrece información acerca de los temas, sino también acerca del manejo de la herramienta. Para esta última opción se especifica paso a paso el desarrollo de cada uno de los laboratorios a través de un ejemplo.

Figura 37. Ayuda.



5 CONCLUSIONES

- Los objetivos del Proyecto se alcanzaron en su totalidad, la herramienta interactiva para simular el proceso del DNS con el protocolo UDP satisfizo las expectativas expuestas en el Plan de Proyecto.
- El tema de la herramienta, resultó muy interesante dada su importancia en Internet, que actualmente es la red más importante en el mundo.
- El estímulo para la realización de proyectos como éste en la ingeniería de sistemas, es muy importante ya que contribuye a mejorar el proceso enseñanza-aprendizaje, mediante herramientas que permitan la comprensión de temas complejos y que involucran procesos técnicos, que en ocasiones resultan difíciles de explicar y de asimilar con una simple lectura. Se requiere que el estudiante interactúe y se involucre de alguna manera en el proceso. Esta es precisamente una de las ventajas que ofrece la herramienta DNS con UDP.
- La utilización de software de código libre como JAVA e INTERBASE 6.02, permite construir herramientas didácticas y educativas muy completas. Las pocas limitaciones de lenguajes como éste combinada con ingenio, creatividad y un buen diseño (metodología de desarrollo de software adecuada), permiten que se implementen proyectos completos, que pueden satisfacer necesidades de diversas fuentes a muy bajo costo. Además es importante que se exploren herramientas software diferentes a las que existen en el mercado actualmente y que en ocasiones son un poco limitantes y costosas.

- El lenguaje JAVA, mediante los applets ofrece una ventaja muy importante como lo es el desarrollo Web, éste combinado con código HTML permite construir interfaces agradables al usuario, además con la implementación adecuada, herramientas como ésta se pueden acceder desde Internet, de esta forma el usuario puede acceder a ella desde cualquier parte sin limitaciones.
- Con la realización de la herramienta no sólo se afianzaron las bases de programación adquiridas en el transcurso de la carrera sino también las de bases de datos. Es una herramienta que involucra algunos conocimientos de sistemas distribuidos, redes, sistemas de información y programación en general.
- La experiencia adquirida en el desarrollo del presente proyecto fue muy enriquecedora, y nos brindó una visión más amplia acerca de lo que puede abarcar la ingeniería de sistemas como tal y de las diferentes áreas relacionadas con ella.

6 RECOMENDACIONES

- Se sugiere que se incentive a los estudiantes para el desarrollo de herramientas similares a estas que abarquen los diferentes protocolos de Internet y de esta forma crear un sitio completo con todos los niveles de TCP/IP.
- Mantener la metodología de diseño aplicada en caso de futuras implementaciones.

BIBLIOGRAFIA

COMER, Douglas E. Redes de computadoras, Internet e interredes. México, Prentice Hall, 1997.

GARCIA TOMAS, Jesús, FERRANDO, Santiago y PIATTINI, Mario. Redes para Proceso Distribuido. México, Alfaomega grupo editor, 1997.

HALSALL, Fred. Comunicación de datos, redes de computadores y sistemas abiertos. 4 ed. México, Addison Wesley iberoamericana, S.A., 1998.

JACOBSON, Ivar, BOOCH, Grady y RUMBAURGH, James. El proceso unificado de desarrollo de software. 1 ed. España, Addison Wesley, 2000.

PRESSMAN, Roger S. Ingeniería del Software: Un Enfoque Práctico, Tercera Edición. México, McGrawHill, 1993

RAYA, José Luis. Redes locales y TCP/IP. México, Alfaomega grupo editor, 1997.

TANENBAUM, Andrews. Redes de computadoras. 3 ed. México, Prentice Hall, 1997.

[www.java.sun.com] JAVA.

[www.agave.com] JAVA.

[www.programacion.com] JAVA.

[www.rfc-editor.org] RFC UDP y DNS.

[www.info-ab.uclm.es] Proceso Unificado.

[www.inf-cr.uclm.es] Proceso Unificado.

[www.borland.com] Internase 6.02.

[www.sun.com] JAVA.

ANEXO A



Las pruebas se realizaron el 22 de Junio de 2004 con estudiantes de ingeniería de sistemas de octavo y noveno nivel. A continuación se muestra un informe de dichas pruebas.

1. ERRORES DE CONTENIDO Y DE NAVEGACION

1.1 Contenido del sitio

Permite determinar posibles errores en el contenido del sitio.

CASO DE PRUEBA	contnav_01	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Página principal de la herramienta				
ENTRADA	Revisar: 1. Ortografía y gramática. 2. Si la información es presentada en forma estructurada y comprensible al usuario. 3. Si los recursos gráficos contribuyen a la fácil asimilación del tema. 4. Distribución del espacio. 5. Tamaño de fuentes. 6. Manejo de colores.				
ACCIÓN	El sistema responde a las solicitudes del usuario.				

1.2 Facilidad de exploración del sitio

Permite verificar aspectos relacionados con la exploración del sitio.

CASO DE PRUEBA	contnav_02	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Página principal de la herramienta				
ENTRADA	1. Las opciones presentadas en el menú principal permiten acceder al sitio totalmente. 2. Mientras explora el sitio le resulta fácil retornar a un punto determinado de éste. 3. Los íconos de navegación se identifican fácilmente y se presentan en forma estándar. 4. Existe coherencia entre el icono de direccionamiento y la página mostrada.				
ACCIÓN	El sistema responde a las solicitudes del usuario.				



2. INTERESES EN USABILIDAD

Permite determinar la utilidad de la herramienta, teniendo en cuenta el tema y los intereses del usuario.

CASO DE PRUEBA	inteusa_01	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Página principal de la herramienta				
ENTRADA	<ol style="list-style-type: none">1. El tema presentado es de utilidad para complementar sus conocimientos profesionales.2. La herramienta le ofrece ventajas de aprendizaje en comparación a los métodos tradicionales de estudio en lo referente al tema.3. La herramienta es adecuada para ser utilizada como medio didáctico en la ilustración del tema.4. Considera que éste tipo de herramientas podría vincularse dentro del contenido de una materia que contemple este tema como medio para lograr los objetivos trazados.				
ACCIÓN	El sistema responde a las solicitudes del usuario.				

3. RENDIMIENTO

Permite verificar el rendimiento de la herramienta.

CASO DE PRUEBA	rendi_01	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Página principal de la herramienta				
ENTRADA	<ol style="list-style-type: none">1. Los componentes del sistema ofrecen el rendimiento de acuerdo con las especificaciones de su equipo.				
ACCIÓN	El sistema responde a las solicitudes del usuario.				

4. MODULO LABORATORIOS - LABORATORIO 1.

4.1 Inicio de Sesión

Interfaz que permite a los usuarios registrados ingresar al laboratorio 1

CASO DE PRUEBA	usureg_01	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios registrados				
ENTRADA	<ol style="list-style-type: none">1. Introducir un nombre de usuario válido.2. Introducir una contraseña de usuario válida y que corresponda al nombre de usuario.				
ACCIÓN	<ul style="list-style-type: none">• El sistema verifica que el usuario se encuentre registrado.• El sistema verifica que el nombre de usuario y contraseña sean correctos.				
RESPUESTA ESPERADA	Entrada al buzón.				



DOCUMENTACIÓN DE PRUEBAS

HERRAMIENTA INTERACTIVA PARA SIMULAR EL PROCESO DEL DNS CON EL PROTOCOLO UDP

CASO DE PRUEBA	usureg_02	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios registrados				
ENTRADA	1. Introducir un nombre de usuario válido. 2. Introducir una contraseña de usuario no válida.				
ACCIÓN	<ul style="list-style-type: none">• El sistema verifica que el usuario se encuentre registrado.• El sistema verifica que el nombre de usuario y contraseña sean correctos.				
RESPUESTA ESPERADA	Mensaje de error: ! Contraseña no válida...La contraseña que ingresó no es válida.				

CASO DE PRUEBA	usureg_03	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios registrados				
ENTRADA	1. Introducir un nombre de usuario no válido. 2. Introducir una contraseña de usuario válida.				
ACCIÓN	<ul style="list-style-type: none">• El sistema verifica que el usuario se encuentre registrado.• El sistema verifica que el nombre de usuario y contraseña sean correctos.				
RESPUESTA ESPERADA	Mensaje de error: ! Usuario no válido...El nombre de usuario o el nombre de dominio no existe. Verifique que los dos nombres sean válidos.				

4.2 Recuperar Contraseña

Formato que permite cambiar contraseña en caso de olvido.

CASO DE PRUEBA	recclave_01	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios registrados/olvidó contraseña?				
ENTRADA	1. Introducir nombre de usuario válido. 2. Dar click en imagen "?". 3. Introducir respuesta válida a la pregunta secreta. 4. Ingresar nueva contraseña válida. 5. Confirmar nueva contraseña válida.				
ACCIÓN	<ul style="list-style-type: none">• El sistema verifica que el nombre de usuario sea válido.• El sistema muestra formato para cambiar contraseña.• El sistema verifica que la respuesta a la pregunta secreta sea correcta.• El sistema verifica que las dos contraseñas sean válidas y coincidan.				
RESPUESTA ESPERADA	Entrada al buzón.				



DOCUMENTACIÓN DE PRUEBAS

HERRAMIENTA INTERACTIVA PARA SIMULAR EL PROCESO DEL DNS CON EL PROTOCOLO UDP

CASO DE PRUEBA	recclave_02	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios registrados/olvidó contraseña?				
ENTRADA	1. Introducir nombre de usuario no válido. 2. Dar click en imagen "?".				
ACCIÓN	<ul style="list-style-type: none">El sistema verifica que el nombre de usuario sea válido.				
RESPUESTA ESPERADA	Mensaje de error: ! Usuario no válido...El nombre de usuario o el nombre de dominio no existe. Verifique que los dos nombres sean válidos.				

CASO DE PRUEBA	recclave_03	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios registrados/olvidó contraseña?				
ENTRADA	1. Introducir nombre de usuario válido. 2. Dar click en imagen "?". 3. Introducir respuesta no válida a la pregunta secreta.				
ACCIÓN	<ul style="list-style-type: none">El sistema verifica que el nombre de usuario sea válido.El sistema muestra formato para cambiar contraseña.El sistema verifica que la respuesta a la pregunta secreta sea correcta.				
RESPUESTA ESPERADA	Mensaje de error: ! Respuesta no válida...La respuesta no coincide.				

CASO DE PRUEBA	recclave_04	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios registrados/olvidó contraseña?				
ENTRADA	1. Introducir nombre de usuario válido. 2. Dar click en imagen "?". 3. Introducir respuesta válida a la pregunta secreta. 4. Ingresar nueva contraseña no válida. 5. Confirmar nueva contraseña no válida.				
ACCIÓN	<ul style="list-style-type: none">El sistema verifica que el nombre de usuario sea válido.El sistema muestra formato para cambiar contraseña.El sistema verifica que la respuesta a la pregunta secreta sea correcta.El sistema verifica que las dos contraseñas sean válidas y coincidan.				
RESPUESTA ESPERADA	Mensaje de error: ! Contraseña no válida...La contraseña debe tener máximo 8 caracteres los cuales pueden ser letras o dígitos.				



CASO DE PRUEBA	recclave_05	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios registrados/olvidó contraseña?				
ENTRADA	<ol style="list-style-type: none"> 1. Introducir nombre de usuario válido. 2. Dar click en imagen "?". 3. Introducir respuesta válida a la pregunta secreta. 4. Ingresar nueva contraseña válida. 5. Confirmar nueva contraseña válida pero no coincide. 				
ACCIÓN	<ul style="list-style-type: none"> • El sistema verifica que el nombre de usuario sea válido. • El sistema muestra formato para cambiar contraseña. • El sistema verifica que la respuesta a la pregunta secreta sea correcta. • El sistema verifica que las dos contraseñas sean válidas y coincidan. 				
RESPUESTA ESPERADA	Mensaje de error: ! Error en contraseña...Las contraseñas no coinciden!				

4.3 Registrarse

Interfaz que permite al usuario registrarse.

CASO DE PRUEBA	registrar_01	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/registrar				
ENTRADA	<p>Introducir los siguientes datos válidos de usuario no registrado:</p> <ol style="list-style-type: none"> 1. Introducir nombres. 2. Introducir apellidos. 3. Introducir identificación. 4. Seleccionar tipo de identificación. 				
ACCIÓN	<ul style="list-style-type: none"> • El sistema verifica que los datos sean válidos. • El sistema verifica que el usuario no se encuentre registrado. 				
RESPUESTA ESPERADA	Entrada a la ventana de crear correo.				

CASO DE PRUEBA	registrar_02	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/registrar				
ENTRADA	<p>Introducir los siguientes datos no válidos de usuario no registrado:</p> <ol style="list-style-type: none"> 1. Introducir nombres. 2. Introducir apellidos. 3. Introducir identificación. 4. Seleccionar tipo de identificación. 				
ACCIÓN	<ul style="list-style-type: none"> • El sistema verifica que los datos sean válidos. • El sistema verifica que el usuario no se encuentre registrado. 				
RESPUESTA ESPERADA	Mensaje de error: ! Error en el campo...Los campos nombre, apellidos e identificación son datos requeridos, cada uno debe tener máximo 50, 50 y 15 caracteres respectivamente.				



CASO DE PRUEBA	registrar_03	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/registrar				
ENTRADA	Introducir los siguientes datos válidos de usuario registrado: 1. Introducir nombres. 2. Introducir apellidos. 3. Introducir identificación. 4. Seleccionar tipo de identificación.				
ACCIÓN	<ul style="list-style-type: none"> El sistema verifica que los datos sean válidos. El sistema verifica que el usuario no se encuentre registrado. 				
RESPUESTA ESPERADA	Mensaje de información: ! Mensaje de Registro... Usted se encuentra actualmente registrado, si desea crear otra cuenta de correo llene los datos solicitados y dé click en 'Siguiete', de lo contrario dé click en 'Cancelar'.				

4.4 Crear correo

Interfaz que permite al usuario crear un correo después de haber llenado el formato de registro.

CASO DE PRUEBA	crearcorreo_01	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/registrar/crear correo				
ENTRADA	1. Introducir un nombre de usuario válido. 2. Seleccionar una extensión para el correo. 3. Introducir una contraseña válida. 4. Confirmar contraseña válida. 5. Seleccionar una pregunta secreta. 6. Introducir respuesta válida a la pregunta secreta.				
ACCIÓN	<ul style="list-style-type: none"> El sistema verifica que el nombre de usuario sea válido. El sistema verifica que el nombre de usuario con la extensión (dirección de correo) no exista. El sistema verifica que la contraseña sea válida. El sistema verifica que las contraseñas coincidan El sistema verifica que la respuesta a la pregunta secreta sea válida. 				
RESPUESTA ESPERADA	Mensaje de confirmación: Tu registro ha finalizado satisfactoriamente ¡Ahora puedes iniciar tu consulta!.				



DOCUMENTACIÓN DE PRUEBAS

HERRAMIENTA INTERACTIVA PARA SIMULAR EL PROCESO DEL DNS CON EL PROTOCOLO UDP

CASO DE PRUEBA	crearcorreos_02	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/registrar/crear correo				
ENTRADA	<ol style="list-style-type: none">1. Introducir un nombre de usuario no válido.2. Seleccionar una extensión para el correo.3. Introducir una contraseña válida.4. Confirmar contraseña válida.5. Seleccionar una pregunta secreta.6. Introducir respuesta válida a la pregunta secreta.				
ACCIÓN	<ul style="list-style-type: none">• El sistema verifica que el nombre de usuario sea válido.• El sistema verifica que el nombre de usuario con la extensión (dirección de correo) no exista.• El sistema verifica que la contraseña sea válida.• El sistema verifica que las contraseñas coincidan• El sistema verifica que la respuesta a la pregunta secreta sea válida.				
RESPUESTA ESPERADA	Mensaje de error: ! Nombre de Usuario no válido...El nombre de usuario debe tener máximo 10 caracteres, empezar con letra, terminar con letra o dígito y tener como caracteres internos sólo letras, dígitos o guiones.				

CASO DE PRUEBA	crearcorreos_03	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/registrar/crear correo				
ENTRADA	<ol style="list-style-type: none">1. Introducir un nombre de usuario válido que ya existe.2. Seleccionar una extensión para el correo que ya existe.3. Introducir una contraseña válida.4. Confirmar contraseña válida.5. Seleccionar una pregunta secreta.6. Introducir respuesta válida a la pregunta secreta.				
ACCIÓN	<ul style="list-style-type: none">• El sistema verifica que el nombre de usuario sea válido.• El sistema verifica que el nombre de usuario con la extensión (dirección de correo) no exista.• El sistema verifica que la contraseña sea válida.• El sistema verifica que las contraseñas coincidan• El sistema verifica que la respuesta a la pregunta secreta sea válida.				
RESPUESTA ESPERADA	Mensaje de error: ! E-mail duplicados... El correo que intenta crear ya existe. Por favor digite un nombre de usuario diferente o elija otro dominio.				



CASO DE PRUEBA	crearcorreos_04	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/registrar/crear correo				
ENTRADA	<ol style="list-style-type: none">1. Introducir un nombre de usuario válido.2. Seleccionar una extensión para el correo.3. Introducir una contraseña no válida.4. Confirmar contraseña no válida.5. Seleccionar una pregunta secreta.6. Introducir respuesta válida a la pregunta secreta.				
ACCIÓN	<ul style="list-style-type: none">• El sistema verifica que el nombre de usuario sea válido.• El sistema verifica que el nombre de usuario con la extensión (dirección de correo) no exista.• El sistema verifica que la contraseña sea válida.• El sistema verifica que las contraseñas coincidan• El sistema verifica que la respuesta a la pregunta secreta sea válida.				
RESPUESTA ESPERADA	Mensaje de error: ! Contraseña no válida... La contraseña debe tener máximo 8 caracteres, los cuales pueden ser letras o dígitos.				

CASO DE PRUEBA	crearcorreos_05	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/registrar/crear correo				
ENTRADA	<ol style="list-style-type: none">1. Introducir un nombre de usuario válido.2. Seleccionar una extensión para el correo.3. Introducir una contraseña válida.4. Confirmar contraseña válida pero no coincide.5. Seleccionar una pregunta secreta.6. Introducir respuesta válida a la pregunta secreta.				
ACCIÓN	<ul style="list-style-type: none">• El sistema verifica que el nombre de usuario sea válido.• El sistema verifica que el nombre de usuario con la extensión (dirección de correo) no exista.• El sistema verifica que la contraseña sea válida.• El sistema verifica que las contraseñas coincidan• El sistema verifica que la respuesta a la pregunta secreta sea válida.				
RESPUESTA ESPERADA	Mensaje de error: ! Error en contraseña... Las contraseñas no coinciden!				



DOCUMENTACIÓN DE PRUEBAS

HERRAMIENTA INTERACTIVA PARA SIMULAR EL PROCESO DEL DNS CON EL PROTOCOLO UDP

CASO DE PRUEBA	crearcorreo_05	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/registrar/crear correo				
ENTRADA	7. Introducir un nombre de usuario válido. 8. Seleccionar una extensión para el correo. 9. Introducir una contraseña válida. 10. Confirmar contraseña válida pero no coincide. 11. Seleccionar una pregunta secreta. 12. Introducir respuesta válida a la pregunta secreta.				
ACCIÓN	<ul style="list-style-type: none">• El sistema verifica que el nombre de usuario sea válido.• El sistema verifica que el nombre de usuario con la extensión (dirección de correo) no exista.• El sistema verifica que la contraseña sea válida.• El sistema verifica que las contraseñas coincidan• El sistema verifica que la respuesta a la pregunta secreta sea válida.				
RESPUESTA ESPERADA	Mensaje de error: ! Error en contraseña... Las contraseñas no coinciden!				

CASO DE PRUEBA	crearcorreo_06	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/registrar/crear correo				
ENTRADA	1. Introducir un nombre de usuario válido. 2. Seleccionar una extensión para el correo. 3. Introducir una contraseña válida. 4. Confirmar contraseña válida pero no coincide. 5. Seleccionar una pregunta secreta. 6. Introducir respuesta no válida a la pregunta secreta.				
ACCIÓN	<ul style="list-style-type: none">• El sistema verifica que el nombre de usuario sea válido.• El sistema verifica que el nombre de usuario con la extensión (dirección de correo) no exista.• El sistema verifica que la contraseña sea válida.• El sistema verifica que las contraseñas coincidan• El sistema verifica que la respuesta a la pregunta secreta sea válida.				
RESPUESTA ESPERADA	Mensaje de error: ! Respuesta no válida...campo requerido máximo de 15 caracteres, los cuales deben ser letras.				



4.5 Ingresar como invitado

Interfaz que permite a la persona ingresar como un usuario no registrado.

CASO DE PRUEBA	invitado_01	PREPARÓ	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/correo				
ENTRADA	1. Introducir dirección de correo de origen válida. 2. Introducir dirección de correo de destino válida.				
ACCIÓN	<ul style="list-style-type: none"> El sistema verifica que la dirección de correo de origen sea válida. El sistema verifica que la dirección de correo de destino sea válida. 				
RESPUESTA ESPERADA	Muestra formato de mensaje DNS para iniciar el laboratorio.				

CASO DE PRUEBA	invitado_02	PREPARÓ	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/correo				
ENTRADA	1. Introducir dirección de correo de origen no válida. 2. Introducir dirección de correo de destino válida.				
ACCIÓN	<ul style="list-style-type: none"> El sistema verifica que la dirección de correo de origen sea válida. El sistema verifica que la dirección de correo de destino sea válida. 				
RESPUESTA ESPERADA	Mensaje de error: ! Dirección de correo de origen no válida... El nombre de usuario no debe superar los 10 caracteres y el de dominio debe estar compuesto de una o más etiquetas, máximo de 63 caracteres separadas por puntos. Las etiquetas, y el nombre de usuario deben empezar con una letra, terminar con una letra o dígito y tener como caracteres internos sólo letras, dígitos y guiones(_,-).				

CASO DE PRUEBA	invitado_03	PREPARÓ	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1/usuarios no registrados/correo				
ENTRADA	1. Introducir dirección de correo de origen válida. 2. Introducir dirección de correo de destino no válida.				
ACCIÓN	<ul style="list-style-type: none"> El sistema verifica que la dirección de correo de origen sea válida. El sistema verifica que la dirección de correo de destino sea válida. 				
RESPUESTA ESPERADA	Mensaje de error: ! Dirección de correo de destino no válida... El nombre de usuario no debe superar los 10 caracteres y el de dominio debe estar compuesto de una o más etiquetas, máximo de 63 caracteres separadas por puntos. Las etiquetas, y el nombre de usuario deben empezar con una letra, terminar con una letra o dígito y tener como caracteres internos sólo letras, dígitos y guiones(_,-).				



5. MODULO LABORATORIOS - LABORATORIOS 1, 2 Y 3.

5.1 Desarrollo de laboratorios

Aquí se efectúa una prueba general al desarrollo de los laboratorios.

CASO DE PRUEBA	deslab_01	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1, 2 o 3/formato				
ENTRADA	1. Introducir datos válidos al formato.				
ACCION	<ul style="list-style-type: none">El sistema verifica que cada uno de los datos sean válidos.				
RESPUESTA ESPERADA	Muestra consecutivamente las ventanas que hacen parte del laboratorio.				

CASO DE PRUEBA	deslab_02	PREPARO	AG-BM	APROBADO	SI
RUTA DE USUARIO	Laboratorios/laboratorio 1, 2 o 3/formato				
ENTRADA	2. Introducir datos no válidos al formato.				
ACCION	<ul style="list-style-type: none">El sistema verifica que cada uno de los datos sean válidos.				
RESPUESTA ESPERADA	Muestra mensaje de error dependiendo de el campo donde detectó el error.				