



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES E INFORMÁTICOS**

TEMA:

**PORTAL WEB PARA LA GESTIÓN DE LA INFORMACIÓN DE LA
UNIDAD DESCONCENTRADA DE TERMINALES DE AMBATO.**

Proyecto de Trabajo de Graduación. Modalidad Proyecto de Investigación, presentada previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

SUBLÍNEA DE INVESTIGACIÓN:

Aplicaciones para web

AUTOR: Landa Manotoa Angel Patricio
TUTOR: Ing. Núñez Miranda Carlos Israel, Mg

Ambato - Ecuador
Septiembre, 2018

CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor del Trabajo de Investigación sobre el tema: PORTAL WEB PARA LA GESTIÓN DE LA INFORMACIÓN DE LA UNIDAD DESCONCENTRADA DE TERMINALES DE AMBATO, del señor Angel Patricio Landa Manotoa, estudiante de la Carrera de Ingeniería en Sistemas Computaciones e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato septiembre, 2018

EL TUTOR



Ing. Carlos Núñez, Mg.

AUTORÍA DEL TRABAJO

El presente Proyecto de Investigación titulado: PORTAL WEB PARA LA GESTIÓN DE LA INFORMACIÓN DE LA UNIDAD DESCONCENTRADA DE TERMINALES DE AMBATO, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato septiembre, 2018



Angel Patricio Landa Manotoa

CC: 1804600912

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato septiembre, 2018



Angel Patricio Landa Manotoa

CC: 1804600912

APROBACIÓN DEL TRIBUNAL DE GRADO

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Mg. Hernán Naranjo e PhD. Félix Fernández, revisó y aprobó el Informe Final del Proyecto de Investigación titulado PORTAL WEB PARA LA GESTIÓN DE LA INFORMACIÓN DE LA UNIDAD DESCONCENTRADA DE TERMINALES DE AMBATO, presentado por el señor Angel Patricio Landa Manotoa de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



Ing. Mg. Elsa Pilar Urrutia Urrutia
PRESIDENTE DEL TRIBUNAL



Ing. Mg. Hernán Naranjo
DOCENTE CALIFICADOR



PhD. Felix Fernández
DOCENTE CALIFICADOR

DEDICATORIA:

Dedico el presente trabajo a mi familia en especial a mis padres, que con su continuo apoyo y amor han impulsado la culminación de esta parte de mi vida. De igual manera a mis amigos que me han acompañado a lo largo de mi carrera universitaria.

Angel Patricio Landa Manotoa.

AGRADECIMIENTO:

Agradezco a todos los docentes de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial que me impartieron su cátedra y han inculcado en mí valores. A mi tutor el Ingeniero Carlos Núñez por apoyarme durante el desarrollo del proyecto de investigación. A toda mi familia, amigos y compañeros por su apoyo incondicional y confianza que han puesto en mí.

Angel Patricio Landa Manotoa.

ÍNDICE

CERTIFICACIÓN DEL TUTOR	ii
AUTORÍA DEL TRABAJO	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
RESUMEN.....	xv
INTRODUCCIÓN	xvii
CAPÍTULO 1	1
El problema	1
1.1. Tema de Investigación.....	1
1.2. Planteamiento del problema	1
1.3. Delimitación	2
1.3.1. De contenidos.....	2
1.3.2. Espacial	2
1.3.3. Temporal	2
1.4. Justificación.....	2
1.5. Objetivos	4
1.5.1. Objetivo General	4
1.5.2. Objetivos Específicos.....	4
CAPÍTULO 2.....	5
Marco Teórico:.....	5
2.1. Antecedentes Investigativos	5
2.2. Fundamentación teórica	6
2.2.1. Concepto de Portal Web.....	6
2.2.2. Servidor Web	7
2.2.3. Java.....	7
2.2.3.1. Características de Java	8
2.2.4. Framework	9
2.2.5. Spring	9

2.2.5.1.	Arquitectura de Spring.....	9
2.2.5.2.	Spring MVC.....	11
2.2.5.3.	Ciclo de vida de una petición HTTP en Spring Web MVC.....	12
2.2.5.4.	MVC en arquitectura de tres niveles.....	13
2.2.5.5.	Proceso de autenticación en Spring Security	14
2.2.6.	Patrón DAO (Data Access Object)	15
2.2.7.	JDBC Driver para SQL Server.....	17
2.2.8.	Froala Editor.....	17
2.2.8.1.	Froala Editor SDKs.....	18
2.2.9.	Apache Tiles.....	18
2.2.10.	MAVEN	19
2.2.10.1.	Arquetipos.....	19
2.2.10.2.	POM.xml.....	19
2.2.10.3.	Principales etiquetas del archivo POM.xml.....	19
2.2.11.	Hibernate	20
2.2.12.	Bootstrap	23
2.2.13.	JQuery	23
2.2.14.	SGBD	24
2.2.15.	Microsoft SQL Server	24
2.2.16.	IDE	25
2.2.17.	Eclipse	25
2.2.18.	Servidor de aplicaciones JBoss	25
2.2.19.	Metodologías ágiles en el ámbito web	26
2.2.19.1.	OOHDM (Metodología de desarrollo de hipermedia orientada a objetos) 26	
2.2.19.2.	Metodología WSDM (método de diseño para sitios web).....	27
2.2.19.3.	Metodología MIDAS	28
2.2.19.4.	UWE (Metodología de Ingeniería Web basada en UML)	30
CAPÍTULO 3		31
Metodología		31
3.1.	Modalidad de la investigación.....	31
3.2.	Recolección de información	31
3.3.	Procesamiento y análisis de datos	31
3.4.	Desarrollo del proyecto	31

CAPÍTULO 4	33
Desarrollo de la propuesta.....	33
4.1 Datos informativos	33
4.2. Justificación de la Metodología.....	33
4.3. MIDAS/SD (Requisitos sistema)	35
4.3.1. Especificación de Requerimientos de Software.....	35
4.3.1.1. Introducción	35
4.3.1.2. Descripción General	38
4.3.1.3. Requisitos Específicos	44
4.3.2. Requerimientos del portal web.....	71
4.3.2.1. Listado de requerimientos funcionales	71
4.3.2.2. Listado de requerimientos no funcionales	74
4.3.2.3. Diagrama de casos de uso.....	74
4.4. MIDAS/HT (Hipertexto).....	79
4.4.1. Actividad de Análisis	79
4.4.2. Actividades de Diseño e Implementación.....	84
4.4.2.1. Actividad de Diseño.....	84
4.4.2.2. Actividad de Implementación.....	88
4.5. MIDAS/DB (Base de Datos).....	95
4.5.1. Modelo entidad relación.....	98
4.5.2. Diagrama Entidad/Relación completo	99
4.5.3. Diseño físico de base de datos	100
4.5.4. Diccionario de datos.....	101
4.6. MIDAS/FC (Funcionalidad).....	105
4.6.1. Plataforma de desarrollo	105
4.6.2. Desarrollo.....	105
4.6.2.1. Estructura del proyecto	105
4.6.2.2. Modelo	107
4.6.2.3. Creación de la configuración de hibernate	107
4.6.2.4. Hibernate console	108
4.6.2.5. Generación de código	109
4.6.2.6. Capa DAO.....	111
4.6.2.7. Capa de Servicio	111

4.6.2.8.	Vista.....	112
4.6.2.9.	Validaciones.....	113
4.6.2.10.	Controlador	114
4.6.3.	Configuraciones	115
4.6.3.1.	Configuración spring framework.....	115
4.6.3.2.	Configuración spring security.....	117
4.6.3.3.	Configuración hibernate	117
4.6.3.4.	Archivo de propiedades	118
4.6.3.5.	Archivo hibernate.cfg.xml	119
4.7.	MIDAS/TST (Pruebas del sistema).....	120
4.7.1.	Plan de pruebas	120
4.7.1.1.	Ambiente de pruebas	121
4.7.1.2.	Pruebas funcionales	122
4.7.1.3.	Pruebas no funcionales	130
4.7.1.4.	Pruebas de Stress	131
CAPITULO 5	135
Conclusiones y recomendaciones	135
5.1.	CONCLUSIONES.....	135
5.2.	RECOMENDACIONES	136
Bibliografía	137
ANEXOS	140

ÍNDICE DE FIGURAS

Figura 1 Módulos de Spring Framework [11].....	10
Figura 2 Ciclo de vida de una petición http en Spring MVC [12].....	12
Figura 3 MVC en arquitectura de 3 capas [13]	13
Figura 4 Proceso de autenticación en Spring Security [14].....	15
Figura 5 Patrón de diseño DAO	16
Figura 6 Implementación patrón DAO con múltiples bases de datos [15]	16
Figura 7 Implementación patrón DAO con frameworks [15].....	17
Figura 8 Estructura de plantilla con Apache Tiles.....	18
Figura 9 Arquitectura mínima de Hibernate [21]	21
Figura 10 Esquema de fases de WSDM.....	28
Figura 11 Proceso MIDAS	29
Figura 12 Diagrama de caso de uso (módulo Transportistas).....	74
Figura 13 Diagrama de caso de uso (módulo Frecuencias).....	75
Figura 14 Diagrama de caso de uso (módulo Ingresos).....	75
Figura 15 Diagrama de caso de uso (módulo áreas y equipos).....	76
Figura 16 Diagrama de caso de uso (módulo usuarios y perfiles).....	76
Figura 17 Diagrama de caso de uso (módulo dispositivos RFID).....	77
Figura 18 Diagrama de caso de uso (módulo institución).....	77
Figura 19 Diagrama de caso de uso (módulo noticias, comunicados y recursos).....	77
Figura 20 Diagrama de caso de uso (módulo áreas de transferencia)	78
Figura 21 Diagrama de caso de uso (módulo preguntas frecuentes y links de interés).....	78
Figura 22 Diagrama de caso de uso (diagrama general).....	79
Figura 23 Modelo Conceptual de Datos	80
Figura 24 Modelo de Navegación (Administración 1)	81
Figura 25 Modelo de Navegación (Administración 2)	82
Figura 26 Modelo de Navegación parte pública.....	83
Figura 27 Wireframe página de inicio del portal web	85
Figura 28 Wireframe panel de administración	86
Figura 29 Wireframe usuario	86
Figura 30 Wireframe unidad de transporte	87
Figura 31 Wireframe institución.....	87
Figura 32 Wireframe dispositivos RFID	88
Figura 33 Plantilla de Bootstrap	89
Figura 34 Verificación del funcionamiento de la plantilla Bootstrap	89
Figura 35 Prototipo página de inicio del portal web	90
Figura 36 Prototipo panel de administración del portal web	91
Figura 37 Prototipo formulario registro de usuario	92
Figura 38 Prototipo formulario registro unidad de transporte	92

Figura 39 Prototipo formulario registro datos de la institución	93
Figura 40 Prototipo formulario registro frecuencias	93
Figura 41 Prototipo formulario registro de equipos	94
Figura 42 Prototipo listado dispositivos RFID.....	94
Figura 43 Prototipo formulario registro concepto de cobro	95
Figura 44 Base de datos SICTERT	96
Figura 45 Base de datos sictert_lite_america.....	97
Figura 46 Modelo entidad-relación	98
Figura 47 Diagrama Entidad/Relación completo	99
Figura 48 Diseño físico de base de datos	100
Figura 49 Estructura del proyecto generado con maven.....	106
Figura 50 Estructura del archivo de configuración de maven	106
Figura 51 Configuración de hibernate	108
Figura 52 Configuración hibernate console.....	109
Figura 53 Generación de código con hibernate.....	110
Figura 54 Código generado con hibernate tools.....	110
Figura 55 Implementación de la capa DAO	111
Figura 56 Implementación de la capa de servicio.....	112
Figura 57 Plantilla web según la estructura de apache tiles	113
Figura 58 Implementación clase validator	113
Figura 59 Validación de formulario usuario.....	114
Figura 60 Controlador del proyecto	115
Figura 61 Configuración Spring framework.....	116
Figura 62 Configuración Spring Security	117
Figura 63 Esquema de configuración de hibernate	118
Figura 64 Archivo hibernate.properties	118
Figura 65 Configuración archivo hibernate.cfg.xml	119
Figura 66 Configuración de JMeter	132
Figura 67 Informe agregado para la prueba de 625 usuarios.....	133
Figura 68 Resultado del procesamiento de peticiones.....	134
Figura 69 Informe agregado para la prueba de 600 usuarios.....	134
Figura 70 Ingreso a la administración del portal web.....	141
Figura 71 Administración de información institucional	142
Figura 72 Mapa interactivo de las instalaciones del terminal.....	143
Figura 73 Administración áreas de transferencia	143
Figura 74 Administración de comunicados/noticias	144
Figura 75 Administración directorio de cooperativas	145
Figura 76 Administración servicios	145
Figura 77 Administración tarifas de transporte.....	146
Figura 78 Administración recursos del portal web	147
Figura 79 Editor de texto froala – opción imágenes	148
Figura 80 Editor de texto froala – opción archivos.....	148
Figura 81 Editor de texto froala – opción videos	149
Figura 82 Administración de usuarios.....	150
Figura 83 Mapa del sitio (público).....	151
Figura 84 Mapa del sitio (administración)	154

ÍNDICE DE TABLAS

Tabla 1 Cuadro comparativo metodología ágiles web [30].....	34
Tabla 2 Requerimientos funcionales	73
Tabla 3 Requerimientos no funcionales	74
Tabla 4 Tabla institución	101
Tabla 5 Tabla servicio.....	101
Tabla 6 Tabla preguntas frecuentes y glosario de términos.....	102
Tabla 7 Tabla comunicado noticia	102
Tabla 8 Tabla área transferencia	103
Tabla 9 Tabla links de interés.....	103
Tabla 10 Tabla recurso.....	103
Tabla 11 Tabla directorio de cooperativas.....	104
Tabla 12 Tabla tarifa de transporte.....	104
Tabla 13 Parámetros configuración en archivo hibernate.properties	119
Tabla 14 Detalle del plan de pruebas.....	121
Tabla 15 Caso de prueba CPF-01	123
Tabla 16 Caso de prueba CPF-02	124
Tabla 17 Caso de prueba CPF-03	124
Tabla 18 Caso de prueba CPF-04	125
Tabla 19 Caso de prueba CPF-05	126
Tabla 20 Caso de prueba CPF-06	126
Tabla 21 Caso de prueba CPF-07	127
Tabla 22 Caso de prueba CPF-08	127
Tabla 23 Caso de prueba CPF-09	128
Tabla 24 Caso de prueba CPF-10	129
Tabla 25 Caso de prueba CPF-11	129
Tabla 26 Caso de prueba CPNF-01	130
Tabla 27 Caso de prueba CPNF-02	131

RESUMEN

La Unidad Desconcentrada de Terminales Ambato es una institución pública ecuatoriana localizada en Ambato dedicada a brindar servicio de transporte interprovincial, cuya finalidad es la de contribuir al progreso del país, generando fuentes de empleo y la satisfacción de necesidades de la ciudadanía. La institución actualmente no posee un portal web para su difusión. La falta de un portal web, ha hecho que muchas personas desconozcan de la labor empresarial que desempeña, así como la variedad de servicios que tiene para ofrecer a la colectividad en general. Además, se ve perjudicada la institución ya que no se inserta en el ambiente de comunicación y difusión tan usado en la actualidad, como lo es el internet.

El presente trabajo denominado “PORTAL WEB PARA LA GESTIÓN DE LA INFORMACIÓN DE LA UNIDAD DESCONCENTRADA DE TERMINALES DE AMBATO”, mostrará el uso de la tecnología web para ayudar a la institución a gestionar la información y promocionar sus servicios a través de internet.

El portal web fue desarrollado con las siguientes herramientas: Microsoft SQL Server 2017 en su versión Developer Edition, Entorno de Desarrollo Integrado (IDE) Spring Tool Suite 3.6.4, Java como lenguaje de programación.

El producto final obtenido ha sido un portal web que representa un punto de encuentro en el cual los visitantes se mantendrán siempre actualizados, contando con acceso a la información de manera rápida y segura.

ABSTRACT

The Unidad Desconcentrada de Terminales Ambato is an Ecuadorian public institution located in Ambato dedicated to providing interprovincial transportation service, whose purpose is to contribute to the progress of the country, generating sources of employment and meeting the needs of citizens. The institution does not currently have a web portal for its dissemination. The lack of a web portal, has made many people unaware of the business they perform, as well as the variety of services they have to offer to the community in general. In addition, the institution is harmed because it is not inserted in the communication and dissemination environment so used today, as is the Internet.

The present work called "PORTAL WEB PARA LA GESTIÓN DE LA INFORMACIÓN DE LA UNIDAD DESCONCENTRADA DE TERMINALES DE AMBATO", will show the use of web technology to help the institution to manage the information and promote its services through the internet.

The web portal was developed with the following tools: Microsoft SQL Server 2017 in its Developer Edition version, Integrated Development Environment (IDE) Spring Tool Suite 3.6.4, Java as a programming language.

The final product obtained has been a web portal that represents a meeting point in which visitors will always be updated, with access to information quickly and safely.

INTRODUCCIÓN

El presente trabajo Estructurado de manera Independiente denominado “PORTAL WEB PARA LA GESTIÓN DE LA INFORMACIÓN DE LA UNIDAD DESCONCENTRADA DE TERMINALES DE AMBATO.”, para el entendimiento del mismo se ha dividido en los siguientes capítulos:

CAPÍTULO I

“EL PROBLEMA”. - Expone la carencia de un portal web en la Unidad Desconcentrada de Terminales de Ambato para la promoción de sus servicios.

CAPÍTULO II

“MARCO TEÓRICO”. - Consta de las investigaciones previas realizadas para la implementación del portal web conjuntamente con la metodología de desarrollo.

CAPÍTULO III

“METODOLOGÍA”. - Se determina la metodología de investigación a utilizar, la modalidad de la investigación, el tipo de investigación.

CAPÍTULO IV

“DESARROLLO DE LA PROPUESTA”. - Presenta el desarrollo del portal web en conjunto con la metodología.

CAPÍTULO V

“CONCLUSIONES Y RECOMENDACIONES”. - Dentro de este capítulo se presentan las conclusiones y recomendaciones del proyecto desarrollado.

CAPÍTULO 1

El problema

1.1. Tema de Investigación

PORTAL WEB PARA LA GESTIÓN DE LA INFORMACIÓN DE LA UNIDAD DESCONCENTRADA DE TERMINALES DE AMBATO.

1.2. Planteamiento del problema

Debido al gran auge tecnológico se hace indispensable que el acceso a la información sea de manera rápida, eficaz y segura. Por este motivo millones de personas están utilizando el Internet como herramienta para tratar de solucionar este problema, y que mejor manera que con un portal web, al que se puede acceder desde cualquier parte del mundo. [1]

La Unidad Desconcentrada de Terminales (UDT) no cuenta con un Portal Web, que permita la actualización permanente de datos y la difusión de los servicios que presta como el transporte, encargos y encomiendas.

Actualmente la UDT cuenta con un Sistema de Control de Terminales (implantado localmente), del cual la institución no posee código fuente que permita el mantenimiento o mejoramiento de la aplicación. El sistema cuenta con módulos de facturación, recaudación, venta de frecuencias, entre otras; sin embargo, debido al diseño de sus interfaces hace difícil su manejo y solo puede ser operado por personal debidamente capacitado. Además, por ser un sistema que está implantado en la intranet el usuario está limitada a acudir a las instalaciones de la entidad para realizar cualquier gestión en el mismo.

Teniendo en cuenta que la Unidad Desconcentrada de Terminales trabaja las 24 horas del día y los 7 días de la semana, se hace necesario la implementación de una herramienta que facilite la gestión de la información que se maneja. Esta herramienta no debe estar limitada a funcionar solamente dentro de las instalaciones de la

institución sino más bien estar disponible en internet. Para de esta manera dar facilidades a la UDT y que la gestión de la información sea, de una manera mucho más flexible, desde cualquier lugar y a cualquier hora.

En lo que se refiere a publicidad y difusión, la falta de un medio digital propio para promocionarse, ha hecho que recurran a medios tradicionales como: radio, televisión y prensa. Dichos medios no llegan a muchas personas como si lo hace el internet el mismo que no está limitado a ninguna región geográfica específica.

Por lo mencionado, es necesario dar solución al problema establecido mediante la implementación de un Portal Web en la Unidad Desconcentrada de Terminales de Ambato. Dicho portal brindará tanto las facilidades para la gestión de información de la institución, así como un medio informativo y de difusión, proporcionando una mayor oportunidad de dar a conocer todos los servicios que ofrece a la ciudadanía.

1.3. Delimitación

1.3.1. De contenidos

Área Académica: Software

Línea de investigación: Desarrollo de Software

Sublínea de investigación: Aplicación web

1.3.2. Espacial

Este portal web no está limitado para un área geográfica específica, el objetivo es publicar el proyecto en un servidor web, por lo que la delimitación espacial del proyecto es a nivel global.

1.3.3. Temporal

La presente investigación se desarrollará durante los 6 meses posteriores a la aprobación del proyecto por parte del Consejo Directivo.

1.4. Justificación

La implementación de un Portal Web para la Unidad Desconcentrada de Terminales de Ambato se lo hace; por una parte, para ser un complemento a su Sistema Local y, por otra, para gestionar la información de carácter público, sirviendo como un medio

informativo, tratando de solventar la falta de difusión y publicación de información de la institución.

En nuestro país existen leyes que promueven el acceso a la información como es la Ley Orgánica de Transparencia y Acceso a la Información Pública – LOTAIP. En cuyo artículo 4, se establece que la información pública pertenece a los ciudadanos y ciudadanas y que su acceso es gratuito a excepción de los costos de reproducción, así como que el ejercicio de la función pública está sometido al principio de apertura y publicidad de sus actuaciones. El artículo 7 de la LOTAIP dispone que todas las instituciones, organismos y entidades, personas jurídicas de derecho público o privado que tengan participación del Estado o sean concesionarios de éste, difundirán a través de un portal de información, página web u otros medios a disposición del público implementados en la institución, la información que se describe en los literales del artículo enunciado.

Esta ley promueve que las instituciones públicas den a conocer a la ciudadanía la información de carácter público, a través de medios digitales como portales web, de ahí la justificación de la importancia de esta herramienta informática que; por un lado será un complemento a un Sistema Informático local brindando las facilidades para gestionar la información y, por el otro, será un medio informativo que pondrá a disponibilidad de los visitantes una información verídica, fácil de acceder , segura, sobre todo de manera rápida y oportuna.

El desarrollo de este proyecto es factible, ya que la administración de la Unidad Desconcentrada de Terminales de Ambato ha dado el visto bueno y está de acuerdo para que se desarrolle el portal web, proporcionando todas las facilidades e información necesaria para el desarrollo del mismo.

Se propone desarrollar un portal web para la Unidad Desconcentrada de Terminales de Ambato, mediante el cual se busca proporcionar un medio que facilite la gestión de la información relevante para la institución.

1.5. Objetivos

1.5.1. Objetivo General

Implementar un portal web para la gestión de la información de la Unidad Desconcentrada de Terminales de Ambato.

1.5.2. Objetivos Específicos

- Recolectar los requerimientos para el análisis y la implementación del portal web
- Seleccionar una metodología ágil orientada al campo de aplicaciones web que facilite el desarrollo del proyecto planteado.
- Implementar el portal web para la gestión de la información de la Unidad Desconcentrada de Terminales de Ambato.

CAPÍTULO 2

Marco Teórico:

2.1. Antecedentes Investigativos

Se debe manifestar que en la Unidad Desconcentrada de Terminales de Ambato no se han realizado temas relacionados con la implantación de un Portal Web, así como tampoco ha sido abordado por otros investigadores, constituyéndolo en un tema exclusivo.

A nivel del país existen proyectos afines realizados por la Universidad de Guayaquil en la Facultad de Ingeniería Industrial y por la Universidad Técnica del Norte en la Facultad de Ciencias Aplicadas respectivamente, los autores de los proyectos que se citan a continuación mencionan la importancia de un portal web.

Javier Gaibor, en su propuesta, desarrolló un portal web para la compra de boletos en línea para la cooperativa Transportes Ecuador en el cual manifiesta que el propósito de un portal web es beneficiar, principalmente, a la sociedad proporcionando herramientas más eficientes, seguras y asequibles en este caso para la compra de boletos, además que la entidad reducirá gastos y mejorará su imagen institucional. [1]

Hernán Castro, en su proyecto manifiesta que hoy en día debido al crecimiento tecnológico existe la necesidad de estar comunicado no solo con organizaciones locales sino con el mundo entero y por esta razón es necesario contar con un portal web, ya que además de permitir la difusión, administración de recursos y prestación de servicios también permite crear una imagen competitiva dentro del entorno web. [2]

Revisando los archivos de la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, se encontró que existen varios temas de desarrollo Web entre los que destacan.

Edwin Peñaloza, en su proyecto, manifiesta que la Web se ha constituido en una herramienta de bastante utilidad, permitiendo el acceso simultáneo de gran cantidad de usuarios a servidores de datos en una forma rápida y económica a través de documentos (páginas WEB). Gracias a las nuevas tecnologías, es posible distribuir y ejecutar aplicaciones a través de la red. Permitiendo de esta forma un ahorro sustancial en los costos de licencia de software y equipos de cómputo. [3]

Javier Díaz, en su propuesta, hace énfasis en que la razón fundamental de implementar un Sitio Web en una empresa es que el Internet a día de hoy se constituye como un medio de comunicación muy eficiente y económico para impulsar una relación comercial entre las empresas que ofertan sus productos, servicios y los mercados consumidores. Por sus facilidades, alcance y bajo costo, el Internet es el medio de comunicación de negocios por excelencia. [4]

Edwin Pilco, en su proyecto, da a conocer que debido a la globalización mundial y a la alta competitividad existente, se hace indispensable el buscar nuevas formas de llegar a la gente, innovando en ideas y propuestas en la presentación de productos y/o servicios. Una de las alternativas que las empresas e instituciones han considerado más fiable es el Internet, ya que este es visitado por la mayoría de personas no solo a nivel provincial o nacional sino mundial. [5]

2.2. Fundamentación teórica

2.2.1. Concepto de Portal Web

El portal es un sitio web que provee un único punto de interacción con aplicaciones, información, personas y procesos, personalizados a las necesidades y responsabilidades del usuario. Constituyen la próxima generación de escritorios de trabajo, permitiendo la ejecución de aplicaciones a través de Internet utilizando no sólo una PC, sino además otros dispositivos como PDA y teléfonos celulares. Un portal brinda de una manera integrada contenidos y aplicaciones, con el agregado de un espacio de trabajo unificado y colaborativo, con el objeto de proveer al usuario de toda la información relevante que necesita para poder tomar decisiones de manera acertada,

acorde a sus necesidades y responsabilidades, en cualquier lugar y a cualquier hora. [6]

2.2.2. Servidor Web

Un servidor web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP). Un servidor web básico tiene un esquema de funcionamiento muy sencillo, ejecutando de forma infinita el bucle siguiente:

1. Espera peticiones en el puerto TCP asignado (el estándar para HTTP es el 80).
2. Recibe una petición.
3. Busca el recurso en la cadena de petición.
4. Envía el recurso por la misma conexión por donde ha recibido la petición.
5. Vuelve al punto 2.

Un servidor web que siguiese el esquema anterior cumpliría los requisitos básicos de los servidores HTTP, aunque, eso sí, sólo podría servir ficheros estáticos. A partir del esquema anterior se han diseñado y construido todos los programas servidores de HTTP que existen, variando sólo el tipo de peticiones (páginas estáticas, CGI, Servlets, etc.) que pueden atender, en función de que sean o no multi-proceso, multi-hilados, etc. [6]

2.2.3. Java

Es un lenguaje de programación creado por Sun Microsystems, (empresa que posteriormente fue comprada por Oracle) para poder funcionar en distintos tipos de procesadores. Su sintaxis es muy parecida a la de C o C++, e incorpora como propias algunas características que en otros lenguajes son extensiones: gestión de hilos, ejecución remota, etc. [7]

El código Java, una vez compilado, puede llevarse sin modificación alguna sobre cualquier máquina, y ejecutarlo. Esto se debe a que el código se ejecuta sobre una máquina hipotética o virtual, la Java Virtual Machine, que se encarga de interpretar el código (ficheros compilados .class) y convertirlo a código particular de la CPU que se esté utilizando (siempre que se soporte dicha máquina virtual). [7]

2.2.3.1. Características de Java

Las características principales que nos ofrece Java son:

- ✓ **Lenguaje Simple.** Es un lenguaje sencillo de aprender. Su sintaxis es la de C++ “simplificada”. Los creadores de Java partieron de la sintaxis de C++ y trataron de eliminar de este todo lo que resultase complicado.
- ✓ **Orientado a Objetos.** Posiblemente sea el lenguaje más orientado a objetos de todos los existentes; en Java todo, a excepción de los tipos fundamentales de variables (int, char, long...) es un objeto.
- ✓ **Distribuido.** Java está muy orientado al trabajo en red, soportando protocolos como TCP/IP, UDP, HTTP y FTP. Por otro lado, el uso de estos protocolos es bastante sencillo comparándolo con otros lenguajes que los soportan.
- ✓ **Robusto.** El compilador Java detecta muchos errores que otros compiladores solo detectarían en tiempo de ejecución o incluso nunca. A esclarecer así por ejemplo " if(a=b) then ... " o " int i; h=i*2; " son dos ejemplos en los que el compilador Java no nos dejaría compilar este código; sin embargo, un compilador C compilaría el código y generaría un ejecutable que ejecutaría esta sentencia sin dar ningún tipo de error).
- ✓ **Seguro.** Sobre todo, un tipo de desarrollo: los Applet. Estos son programas diseñados para ser ejecutados en una página web. Java garantiza que ningún Applet puede escribir o leer de nuestro disco o mandar información del usuario que accede a la página a través de la red (como, por ejemplo, la dirección de correo electrónico). En general no permite realizar cualquier acción que pudiera dañar la máquina o violar la intimidad del que visita la página web.
- ✓ **Portable.** En Java no hay aspectos dependientes de la implementación, todas las implementaciones de Java siguen los mismos estándares en cuanto a tamaño y almacenamiento de los datos.
- ✓ **Arquitectura Neutral.** El código generado por el compilador Java es independiente de la arquitectura: podría ejecutarse en un entorno UNIX, Mac o Windows. El motivo de esto es que el que realmente ejecuta el código

generado por el compilador no es el procesador del ordenador directamente, sino que este se ejecuta mediante una máquina virtual.

- ✓ **Multihilo.** Soporta de modo nativo los hilos, sin necesidad del uso de librerías específicas (como es el caso de C++). Esto le permite además que cada Hilo de una aplicación java pueda correr en una CPU distinta, si la aplicación se ejecuta en una máquina que posee varias CPU.

Estas características hacen que Java sea uno de los lenguajes más empleados para programar a día de hoy. [7]

2.2.4. Framework

El término framework, hace referencia a una estructura de software compuesta por componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta. [8]

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. [9]

2.2.5. Spring

Spring es un framework de desarrollo de código libre para la plataforma java que está compuesta por una serie de módulos que se comunican y facilitan el desarrollo de aplicaciones empresariales bajo un lenguaje de programación java. Spring es liviano y ligero, ya que no es una aplicación que requiere de muchos recursos para su ejecución, además el framework completo puede ser distribuido en un archivo .jar de alrededor de 1 MB lo cual representa muy poco espacio. [10]

2.2.5.1. Arquitectura de Spring

Spring se compone de distintos módulos (véase **Figura 1**), permitiendo seleccionar solo algunos de ellos o todos, dependiendo de la naturaleza de la aplicación.

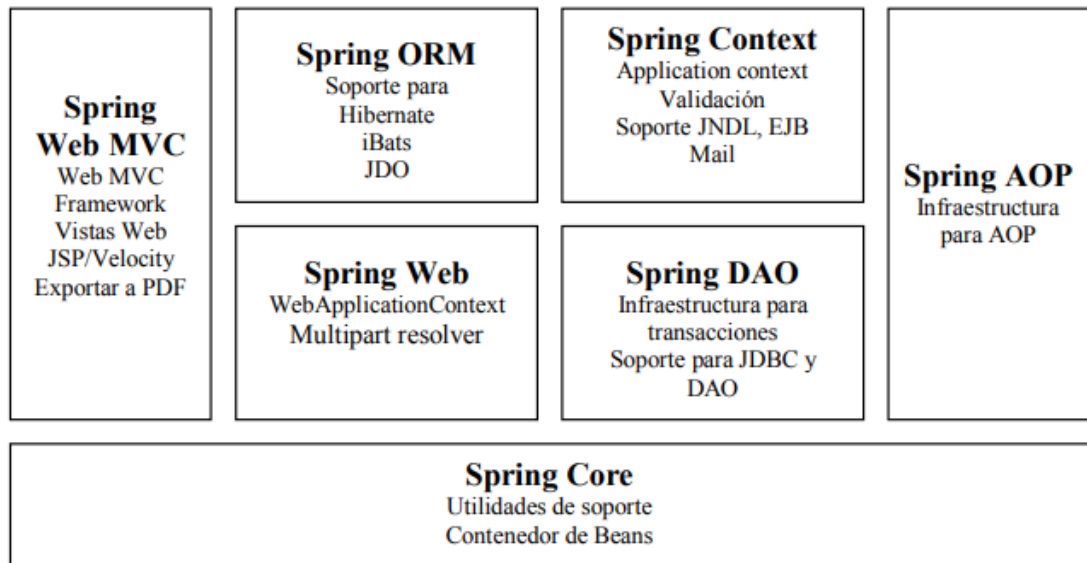


Figura 1 Módulos de Spring Framework [11]

- ✓ **Spring Core:** Este módulo provee la funcionalidad básica de la fábrica de Spring. El componente principal es BeanFactory, el cual aplica el concepto de Inversion of Control (IoC) o también conocido como Dependency Injection (DI).
- ✓ **Spring Context:** Aquí es donde se realiza la configuración del framework, generalmente en un archivo de configuración XML. Incluye la configuración de beans de la capa Web MVC, de servicios empresariales tales como DataSource, JDBC, Hibernate, Repository, Internacionalización, validación, entre varios más.
- ✓ **Spring AOP:** Permite aplicar los conceptos de Programación Orientada a Aspectos (AOP), además incluye clases de soporte para el manejo transaccional, logger, la seguridad, entre varias clases más, permitiendo desacoplar estas características de nuestra aplicación. Además, nos permite desarrollar interceptores de método y puntos de corte para desacoplar el código de las funcionalidades transversales.
- ✓ **Spring DAO Support:** Permite aplicar conceptos de la capa de datos Data Access Object (DAO) a través de POJOs (Plain Old Java Object), abstrayendo la complejidad, permitiendo crear un código JDBC, Hibernate u JPA más limpio y simple. Además, cuenta con una capa de excepciones sobre los mensajes de error provistos por cada servidor específico de base de datos y con manejo de transacciones a través de AOP.

- ✓ **Spring ORM:** Permite integrarse con tecnologías tales como JPA, Hibernate e iBatis, entre otras integradas con DAO Support.
- ✓ **Spring Web:** Permite el desarrollo y la integración con tecnologías como Struts, JSF, Tapestry, entre otros.
- ✓ **Spring MVC:** Este módulo implementa el patrón MVC para ser utilizado en la capa de presentación.

2.2.5.2. Spring MVC

Spring MVC es uno de los componentes del Framework de Spring, y como su propio nombre nos indica implementa una arquitectura Modelo - Vista - Controlador utilizado para desarrollar aplicaciones web. Spring MVC es considerado uno de los más poderosos y a la vez flexible Framework Modelo-Vista Controlador (MVC) para aplicaciones basadas en Web. Se encarga de mapear las peticiones (o request) hacia los controladores, y de los controladores cargar las vistas. Cuenta con excelentes capacidades de manejo de formularios y validaciones de datos, se integra con las tecnologías o motores de vistas más populares, tales como JSP, Thymeleaf, Velocity, FreeMarker, JasperReports, Excel y PDF [11].

La pieza central de Spring MVC es un componente llamado el "DispatcherServlet", el cual sigue el patrón de diseño front controller, este envía las peticiones a los componentes designados para manejarlas.

Las interfaces más importantes definidas por Spring MVC son:

- ✓ **Controller:** Se encuentra entre el Modelo y la Vista, maneja las peticiones que entran y redirige a las respuestas apropiadas.
- ✓ **HandlerAdapter:** Es quien realiza en realidad la invocación del manejador de la petición (el DispatcherServlet delega esa tarea a este componente), esto incluye inyectar los parámetros adecuados usando reflection.
- ✓ **HandlerInterceptor:** Intercepta las peticiones de entrada. Es comparable, pero no igual, a los filtros en el API de Servlets.
- ✓ **HandlerMapping:** Selecciona objetos que manejan las peticiones de entrada basado en cualquier atributo o condición interna o externa a esos atributos.

- ✓ **LocaleResolver:** Resuelve y opcionalmente guarda el Locale de un usuario individual.
- ✓ **MultipartResolver:** Facilita trabajar con la carga de archivos, envolviendo las peticiones de entrada.
- ✓ **View:** Responsable de regresar una respuesta al cliente. Algunas peticiones pueden ir directo a la vista sin pasar por el modelo.
- ✓ **ViewResolver:** Selecciona una vista basado en un nombre lógico para la vista.

2.2.5.3. Ciclo de vida de una petición HTTP en Spring Web MVC

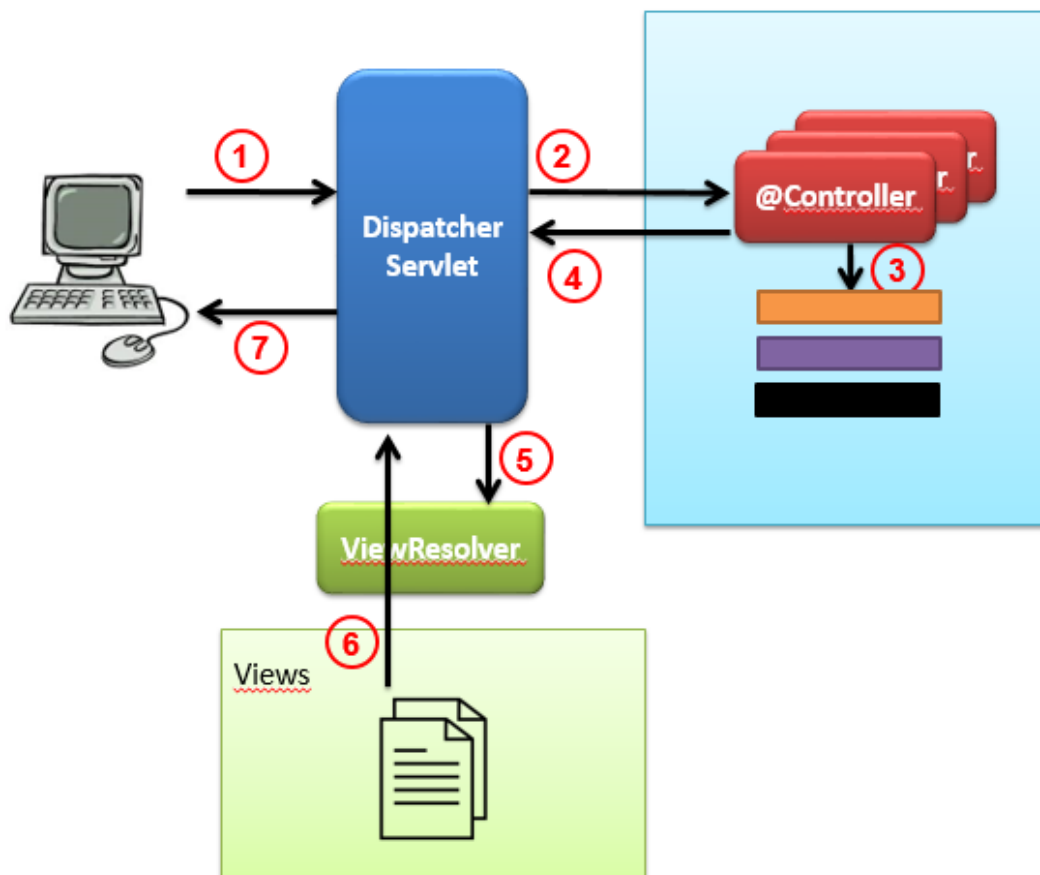


Figura 2 Ciclo de vida de una petición http en Spring MVC [12]

1. El cliente hace una petición a la aplicación web, esta petición llega al DispatcherServlet.
2. El DispatcherServlet determina qué componente debe atender la petición y la envía a este.

3. El Controller implementa la lógica específica para responder la petición, para lo que puede hacer uso de cualquier recurso que esté al alcance de cualquier aplicación Java (incluyendo conexiones con servicios web o con base de datos).
4. Una vez que el Controller termina su proceso, regresa la petición al DispatcherServlet, estableciendo los datos adecuados del modelo e indicando el nombre lógico de la vista que debe regresarse al cliente y un modelo lleno con los nombres y valores de los atributos que se usarán para generar la vista final.
5. En base al nombre lógico regresado por el Controller, el DispatcherServlet usa un ViewResolver para determinar qué recurso debe utilizar para generar la vista final que se mostrará al usuario, este recurso puede ser una JSP, una página HTML, un template de Velocity, un archivo de Excel, un PDF, etc.
6. El DispatcherServlet obtiene la vista que será regresada al cliente.
7. El DispatcherServlet finalmente regresa la vista adecuada al cliente.

2.2.5.4. MVC en arquitectura de tres niveles

En la arquitectura de tres niveles tenemos Capa Presentación, Capa Servicio y Capa DAO.

La capa de presentación interactúa con la capa de Servicio y la capa de Servicio interactúa con la capa DAO (véase **Figura 3**).

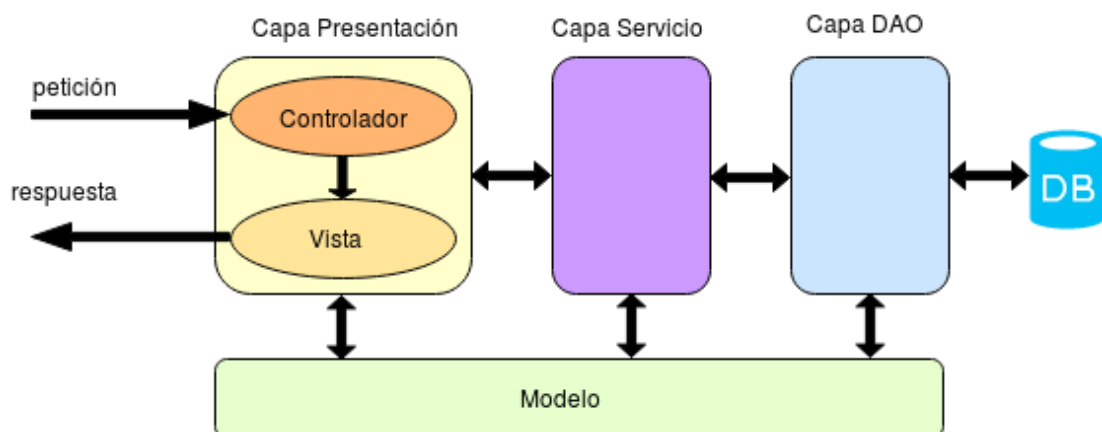


Figura 3 MVC en arquitectura de 3 capas [13]

El flujo será así:

1. La solicitud llega al controlador y el controlador decide si esta solicitud necesita alguna lógica de negocios para implementarse o no.

2. Si no, simplemente devuelve una vista.
3. En caso afirmativo, interactúa con la capa de servicio y ésta a su vez interactúa con la capa DAO.
4. Más adelante, el controlador mostrará la respuesta apropiada de la capa de servicio. [13]

Spring recomienda que el mismo modelo se pueda usar para todas las tres capas.

2.2.5.5. Proceso de autenticación en Spring Security

Este proceso se basa en un conjunto de interfaces y clases que interactúan conjuntamente en un orden establecido para terminar decidiendo si una petición tiene o no acceso. Spring realiza esto mediante una cadena de filtros en los que se irán haciendo las distintas acciones y comprobaciones necesarias hasta decidir si la autenticación es correcta o no. [14]

En este proceso intervienen 3 actores principales:

- ✓ **AuthenticationFilter:** Es el responsable de crear una instancia concreta del usuario y sus credenciales de autenticación.
- ✓ **AuthenticationManager:** Responsable de la validación del usuario y credenciales y rellenar los permisos/roles que tiene el usuario o de lanzar los distintos tipos de excepciones en caso de fallo en la autenticación. Esto lo hace apoyándose en uno o varios proveedores de autenticación.
- ✓ **AuthenticationProvider:** Es en quien se delega la correcta validación del usuario y recuperación de roles.

Este proceso a alto nivel se ilustra en la **Figura 4:**

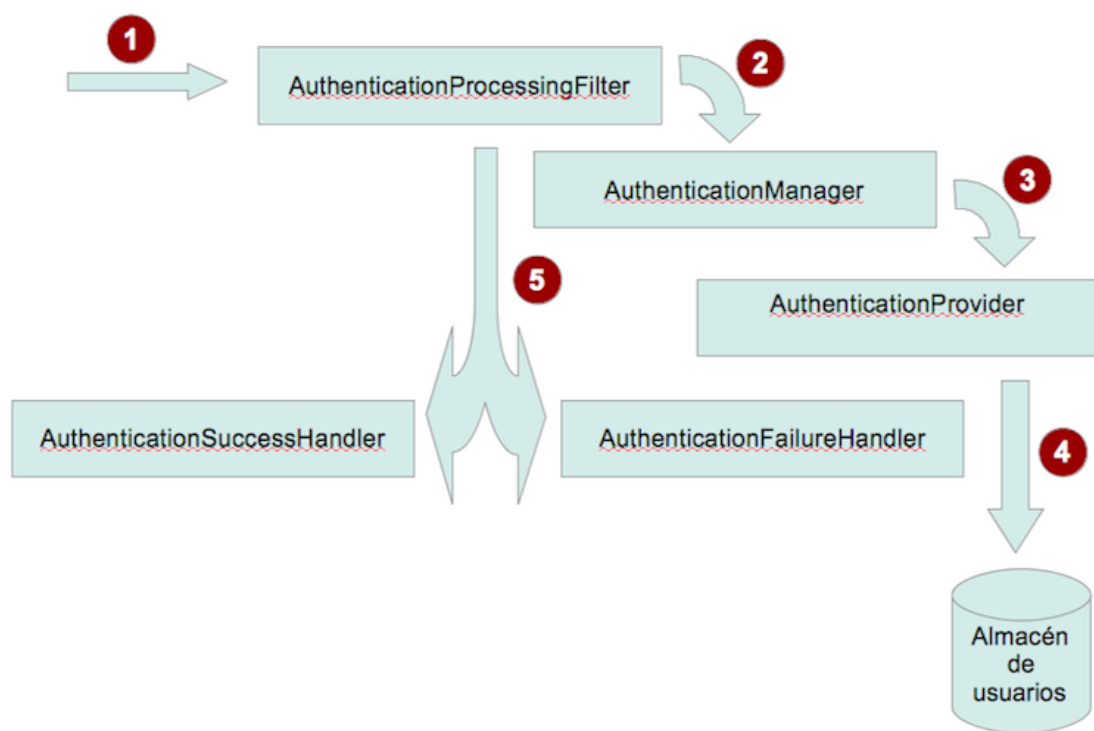


Figura 4 Proceso de autenticación en Spring Security [14]

1. El filtro de seguridad intercepta la petición de autenticación. Crea una instancia de Authentication para su validación.
2. El filtro de seguridad pasa al AuthenticationManager el objeto creado para que realice su validación. El AuthenticationManager delegará esta validación al proveedor o proveedores de autenticación.
3. El proveedor de autenticación comprobará si los datos de autenticación son correctos. En caso de haber algún tipo de error lanzará una excepción.
4. El filtro de seguridad comprueba el resultado de la autenticación solicitada al AuthenticationManager. Si se produjo una excepción redirigirá el flujo al AuthenticationFailureHandler y si hubo éxito al AuthenticationSuccessHanlder.

2.2.6. Patrón DAO (Data Access Object)

El patrón de diseño DAO está relacionado con la persistencia de una aplicación, este patrón permite que nuestra aplicación sea lo más independiente posible de una base de datos concreta, de cómo se accede a los datos o incluso de si hay o no base de datos detrás. La idea de este patrón es sencilla. En primer lugar, debemos hacernos las clases que representan nuestros datos. Por ejemplo, podemos hacer una clase Persona con los datos de la persona y los métodos set() y get() correspondientes.

Luego hacemos una interface. Esta interface tiene que tener los métodos necesarios para obtener y almacenar Personas. Esta interface no debe tener nada que la relacione con una base de datos ni cualquier otra cosa específica del medio de almacenamiento que vayamos a usar, es decir, ningún parámetro debería ser una Connection, ni un nombre de fichero, etc. [15]

```
public interface InterfaceDAO {
    public List<Persona> getPersonas();
    public Persona getPersonaPorNombre (String nombre);
    ...
    public void salvaPersona (Persona persona);
    public void modificaPersona (Persona persona);
    ...
    public void borraPersonaPorNombre (String nombre);
    ...
}
```

Figura 5 Patrón de diseño DAO

Con esto deberíamos construir nuestra aplicación, usando la clase Persona y usando la InterfaceDAO para obtener y modificar Personas.

Aparte, hacemos la implementación de la InterfaceDAO, ya contra una base de datos concreta o usando una herramienta determinada como iBATIS, Hibernate, etc. Al usar nuestra aplicación la InterfaceDAO, podremos pasarle cualquier implementación que queramos o incluso cambiar una por otra en un momento dado. [15]

Por ejemplo, podemos hacer implementaciones según usemos MySQL, Oracle, etc

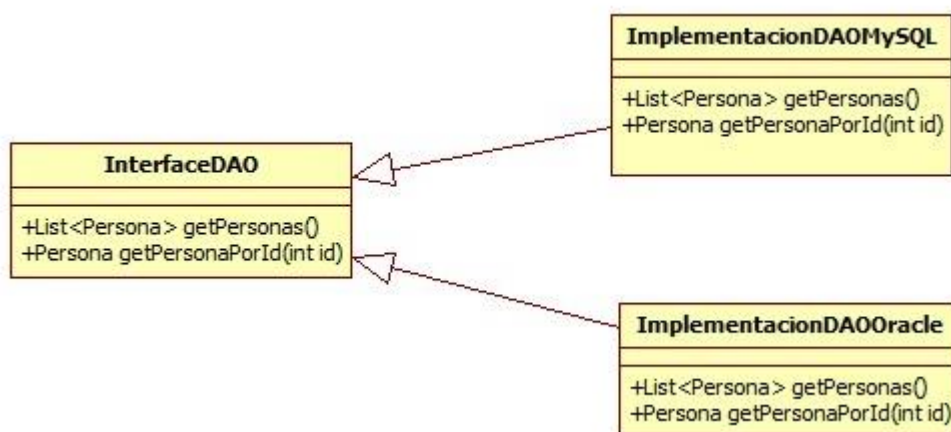


Figura 6 Implementación patrón DAO con múltiples bases de datos [15]

O bien según usemos iBATIS, JDBC, Hibernate, etc.

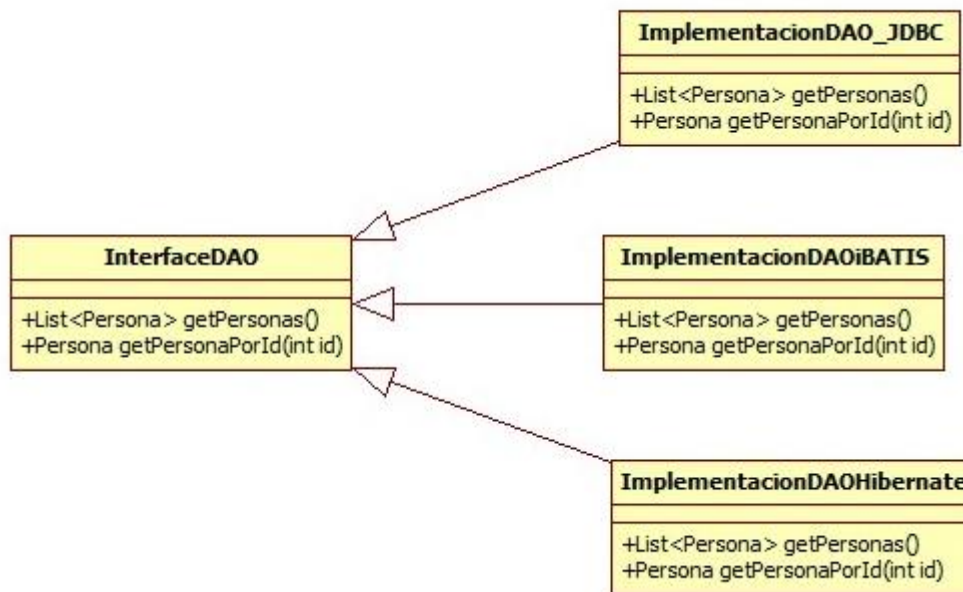


Figura 7 Implementación patrón DAO con frameworks [15]

2.2.7. JDBC Driver para SQL Server

- ✓ **Microsoft JDBC Driver for SQL Server.** El conector JDBC para SQL Server provee acceso a SQL Server 2000-2016 desde cualquier aplicación Java.
- ✓ **jTDS JDBC Driver.** Es un conector JDBC de código abierto tanto para Microsoft SQL Server (6.5, 7, 2000-2016) y también para Sybase (10, 11, 12, 15).

Se opta por utilizar jTDS JDBC Driver ya que a diferencia del primero este está disponible en el repositorio de maven por ende su integración con el proyecto es muy sencilla basta con declarar como dependencia en el archivo pom.xml y maven se encargará de buscar, descargar e incorporarlo a nuestro proyecto.

2.2.8. Froala Editor

Es un editor de texto enriquecido WYSIWYG basado en la web. Es un editor de texto WYSIWYG JavaScript / HTML independiente de la plataforma, basado en la web, hecho por Froala. Froala Editor está disponible bajo 4 licencias: licencia de sitio web único, licencia de desarrollador, licencia comercial y licencia OEM. [16]

El código central del editor de Froala está escrito en JavaScript y, por lo tanto, se puede integrar con cualquier documento HTML.

Froala WYSIWYG HTML Editor es un plugin fácil de integrar y fácil de usar que requiere un conocimiento mínimo de codificación. Requiere jQuery 1.11.0 o superior y la fuente llamada Font Awesome 4.4.0. También puede usar versiones anteriores de Font Awesome, pero algunos de los íconos del editor no aparecerán.

2.2.8.1. Froala Editor SDKs

Son bibliotecas diseñadas para simplificar la interacción del editor con el back-end y hacer cosas como almacenar imágenes y archivos disponibles para los lenguajes como PHP, .NET y Java. El SDK de Java para el editor de HTML WYSIWYG de Froala está diseñado para funcionar con HttpServlet.

2.2.9. Apache Tiles

Apache Tiles es un framework que nos permite organizar de mejor manera la presentación de nuestro sitio web creado con Spring MVC, Tiles funciona como un sistema de plantillas en donde podemos crear pequeños fragmentos de código que pueden ser insertados en cualquier parte de nuestro sitio, organizando cada fragmento de modo que represente una parte de un sitio web, cabecera, barras laterales, pie de página, contenido principal, etc., (véase **Figura 8**), de esta manera evitar la duplicación de código y se hace más fácil realizar cambios en cualquiera de las partes, no tendremos que realizar el cambio página por página. [17]

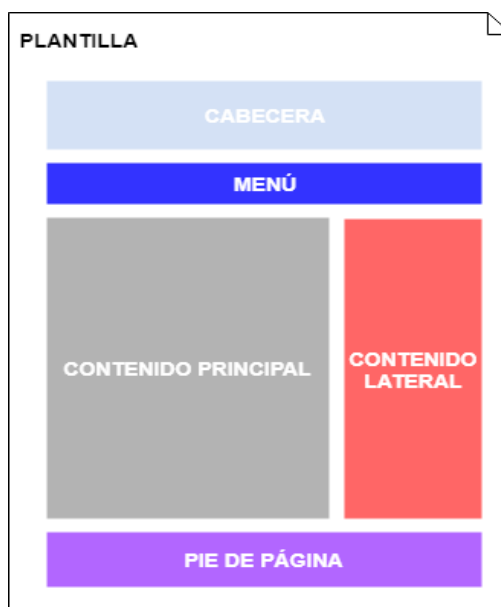


Figura 8 Estructura de plantilla con Apache Tiles

2.2.10. MAVEN

Maven es una herramienta para la gestión de proyectos de software, que se basa en el concepto de POM (Project Object Model). Es decir, con Maven vamos a poder compilar, empaquetar, generar documentación, pasar los test, preparar las builds [18].

Algunas de las principales ventajas de Maven son:

- ✓ Maven se basa en patrones y en estándares. Esto permite a los desarrolladores moverse entre proyectos y no necesitan aprender como compilar o empaquetar. Esto mejora el mantenimiento y la reusabilidad.
- ✓ Maven hace la gestión de librerías, incluso teniendo en cuenta las dependencias transitivas. Es decir, si A depende de B y B depende de C, es que A depende de C. Esto quiere decir que cuando empaquetemos A, Maven se encargará de añadir tanto B como C en el paquete.

2.2.10.1. Arquetipos

Un arquetipo para Maven es una plantilla. Es decir, gracias a un arquetipo Maven es capaz de generar una estructura de directorios y ficheros.

A la hora de empezar un proyecto, basta con decirle a Maven que tipo de proyecto queremos y nos creará la estructura base [18].

2.2.10.2. POM.xml

POM (Project Object Model) es un fichero XML que representa la unidad principal de un proyecto Maven. Contiene información acerca del proyecto, fuentes, test, dependencias, plugins, versión, etc. Antes de ser ejecutado una tarea o proyecto, lo primero que hace Maven es buscar y ejecutar las directivas que fueron definidas en el fichero pom.xml para descargar las librerías a ser usadas en el proyecto [19].

2.2.10.3. Principales etiquetas del archivo POM.xml

- ✓ **<project>** Es la etiqueta principal de un fichero pom.xml. Cuando creamos un nuevo proyecto las primeras etiquetas que son definidas son **<groupId>**, **<artifactId>** estas últimas etiquetas indican la estructura de paquetes del proyecto y el nombre del proyecto respectivamente.
- ✓ **<properties>** Donde se añaden las propiedades del proyecto

- ✓ **<packaging>** Donde se especifica el tipo de binario a generar (.jar aplicaciones java, .war- aplicaciones web).
- ✓ **<name>** y **<url>** indican el nombre como se conoce al proyecto y la URL del proyecto respectivamente.
- ✓ **<dependencies>** Para definir las librerías que vamos a necesitar. Estas serán descargadas automáticamente cuando compilemos el proyecto. Una vez descargadas Maven las pone en nuestro repositorio local así pueden ser usadas en próximas aplicaciones sin necesidad de volver a descargarlas.
- ✓ **<finalName>** Donde indicamos el nombre final que tendrá nuestro archivo ya empaquetado.

2.2.11. Hibernate

Hibernate es una capa de persistencia objeto/relacional y un generador de sentencias SQL que permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada se puede generar BBDD en cualquiera de los entornos soportados: Oracle, DB2, MySql, PostgreSQL, etc. Y lo más importante de todos, es open source, lo que supone, entre otras cosas que no se tiene que pagar ningún costo para adquirirlo. [20]

La arquitectura "mínima" hace que la aplicación proporcione sus propias conexiones JDBC y que administre sus propias transacciones. Este enfoque utiliza un subgrupo mínimo de las APIs de Hibernate:

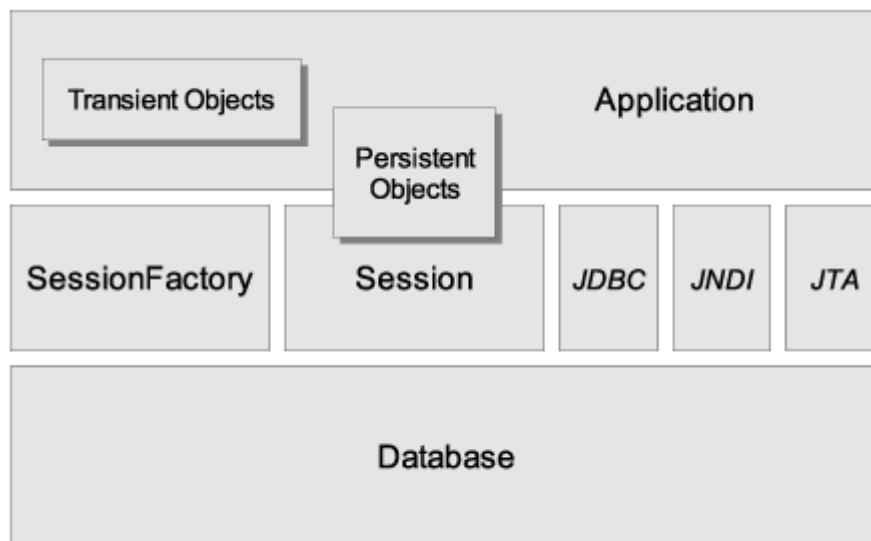


Figura 9 Arquitectura mínima de Hibernate [21]

Definiciones de los objetos en los diagramas:

- ✓ **JDBC** proporciona un nivel rudimentario de la abstracción de la funcionalidad común a las bases de datos relacionales, lo que permite casi cualquier base de datos con un driver JDBC que sea implementada en Hibernate.
- ✓ **JNDI y JTA** permiten Hibernate para ser integrado con servidores de aplicaciones J2EE.
- ✓ **SessionFactory (org.hibernate.SessionFactory)**. Un caché thread-safe (inmutable) de mapeos compilados para una sola base de datos. Una fábrica de Session, SessionFactory puede mantener un caché opcional (de segundo nivel) de datos reusables entre transacciones a nivel de proceso o de clúster. [21]
- ✓ **Session (org.hibernate.Session)**. Una Session se utiliza para obtener una conexión física con una base de datos mediante JDBC. El objeto Session es ligero y está diseñado para ejecutarse cada vez que se necesita una interacción con la base de datos. [21]
- ✓ **Transient Objects**. Hibernate crea una instancia de clase de entidad (clase Java mapeado con la capa de base de datos). Este objeto se llama objeto transitorio.

- ✓ **Objeto de Configuración Hibernate.** El objeto de configuración es el primer objeto de Hibernate que se crea en cualquier aplicación que utilice Hibernate y generalmente se crea una sola vez durante la inicialización de la aplicación. Representa un archivo de configuración o propiedades requeridas por Hibernate. El objeto de configuración proporciona dos componentes claves:
 1. **Conexión de base de datos:** Esto se maneja a través de uno o más archivos de configuración soportadas por Hibernate. Estos archivos son `hibernate.properties` y `hibernate.cfg.xml`.
 2. **Configuración de Mapeo Clase:** Este componente crea la conexión entre las clases de Java y tablas de la base de datos.

Hibernate Tools

Es un conjunto de herramientas para Eclipse y para Apache Ant, que están integradas dentro de JBoss Tools, diseñadas para facilitar el uso de Hibernate. Estas son las herramientas que se incluyen para Eclipse:

- ✓ **Editor de Mapeo.** Un editor para los archivos de mapeo XML de Hibernate que proporciona una ayuda en la escritura del mismo teniendo la opción de autocompletar y del resaltado de sintaxis. El editor soporta incluso autocompletar semántico para nombres de clases, nombres de propiedad, nombres de tabla y nombres de columna. [22]
- ✓ **Consola.** La perspectiva de la consola Hibernate le permite configurar conexiones de bases de datos, proporciona visualización de clases y sus relaciones y le permite ejecutar consultas HQL interactivamente contra su base de datos y examinar los resultados de la consulta. [22]
- ✓ **Ingeniería inversa.** La característica más poderosa de Hibernate Tools es una herramienta de ingeniería inversa de base de datos que puede generar clases de modelo de dominio y archivos de mapeo Hibernate, beans de entidad EJB3 anotados, documentación HTML o incluso una aplicación JBoss Seam completa en segundos. [22]

- ✓ **Wizards.** Se proporcionan varios asistentes, incluidos los asistentes para generar rápidamente archivos de configuración de Hibernate (cfg.xml) y las configuraciones de la consola de Hibernate. [22]

2.2.12. Bootstrap

Se trata de un framework que ofrece la posibilidad de crear un sitio web totalmente responsive mediante el uso de librerías CSS. En estas librerías, nos podemos encontrar un gran número de elementos ya desarrollados y listos para ser utilizados como pueden ser botones, menús, cuadros e incluso un amplio listado de tipografías. En un principio fue desarrollado por Twitter, aunque posteriormente fue liberado bajo licencia MIT. Hoy en día continúa su desarrollo en un repositorio de GitHub. [23]

Desde que vio la luz por el año de 2011, Bootstrap se ha caracterizado por tratarse de una excelente herramienta para crear interfaces de usuarios limpias y totalmente adaptables a cualquier tipo de dispositivo y pantalla, independientemente de su tamaño. [23]

2.2.13. JQuery

Es una biblioteca de JavaScript rápida y concisa que simplifica el documento HTML, manejo de eventos, animación y las interacciones AJAX para el desarrollo web. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. [24]

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

Características

- ✓ Selección de elementos.
- ✓ Eventos.
- ✓ Manipulación de la hoja de estilos CSS.
- ✓ Efectos y animaciones.
- ✓ Animaciones personalizadas.
- ✓ AJAX.

- ✓ Soporta extensiones.
- ✓ Utilidades varias como obtener información del navegador, operar con objetos y vectores.

2.2.14. SGBD

Es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. [25]

Un Sistema de Gestión de Bases de Datos (SGBD) proporciona al usuario de la Base de Datos las herramientas necesarias para realizar, al menos, las siguientes tareas:

- ✓ Definir las estructuras de los datos.
- ✓ Manipular los datos. Es decir, insertar nuevos datos, así como modificar, borrar y consultar los datos existentes.
- ✓ Mantener la integridad de la información.
- ✓ Proporcionar control de la privacidad y seguridad de los datos en la Base de Datos, permitiendo sólo el acceso a los mismos a los usuarios autorizados.

Algunos de los productos más difundidos son:

- ✓ ORACLE de Oracle Corporation.
- ✓ MySQL de Oracle Corporation.
- ✓ DB2 de I.B.M. Corporation
- ✓ SYBASE de Sybase Inc.
- ✓ Informix de Informix Software Inc.
- ✓ SQL Server de Microsoft Corporation.

2.2.15. Microsoft SQL Server

Microsoft SQL Server es un sistema de gestión de base de datos relacional (RDBMS) producido por Microsoft. Su principal lenguaje de consulta es Transact-SQL, una aplicación de las normas ANSI / ISO estándar Structured Query Language (SQL) utilizado por Microsoft. [26]

Características de Microsoft SQL Server:

- ✓ Soporte de transacciones.
- ✓ Escalabilidad, estabilidad y seguridad.
- ✓ Soporta procedimientos almacenados.
- ✓ Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- ✓ Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- ✓ Además, permite administrar información de otros servidores de datos.

2.2.16. IDE

Es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

2.2.17. Eclipse

Eclipse es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de plug-ins. Fue concebida desde sus orígenes para convertirse en una plataforma de integración de herramientas de desarrollo. No tiene en mente un lenguaje específico, sino que es un IDE genérico, aunque goza de mucha popularidad entre la comunidad de desarrolladores del lenguaje Java usando el plug-in JDT que viene incluido en la distribución estándar del IDE. Proporciona herramientas para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones. [27]

2.2.18. Servidor de aplicaciones JBoss

Es un servidor de aplicaciones Java EE de código abierto implementado en Java. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible la máquina virtual de Java. JBoss implementa todo el estándar JEE.

Ofrece una plataforma de alto rendimiento. Combinando una arquitectura orientada a servicios SOA, con una licencia GNU de código abierto, JBoss puede ser descargado, utilizado, incrustado y distribuido sin restricciones por la licencia. [28]

2.2.19. Metodologías ágiles en el ámbito web

Todo proyecto, para completarse de manera exitosa, debe seguir una metodología probada y basada en la experiencia que guíe el proceso de desarrollo hasta alcanzar los objetivos propuestos.

2.2.19.1. OOHDM (Metodología de desarrollo de hipermedia orientada a objetos)

Metodología para diseño de aplicaciones hipermedia y para la Web, es una extensión de HDM con orientación a objetos, que se está convirtiendo en una de las metodologías más utilizadas. Ha sido usada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia y, sobre todo, numerosos sitios web. [29]

OOHDM propone el desarrollo de aplicaciones hipermedia mediante un proceso de 4 etapas:

- ✓ Diseño conceptual
- ✓ Diseño navegacional
- ✓ Diseño de interfaces abstractas
- ✓ Implementación

En la primera etapa se construye un esquema conceptual representado por los objetos de dominio o clases y las relaciones entre dichos objetos.

En la segunda etapa, el diseñador define clases navegacionales tales como nodos, enlaces y estructuras de acceso (índices y visitas guiadas) inducidas del esquema conceptual. Los enlaces derivan de las relaciones y los nodos representan ventanas lógicas (views) sobre las clases conceptuales.

La tercera etapa está dedicada a la especificación de la interfaz abstracta. Así, se define la forma en la cual deben aparecer los contextos navegacionales.

La cuarta etapa, dedicada a la puesta en práctica, es donde se hacen corresponder los objetos de interfaz con los objetos de implementación.

2.2.19.2. Metodología WSDM (método de diseño para sitios web)

Metodología para diseño de sitios web (WSDM) es una propuesta en la que el sistema se define en base a los grupos de usuarios. [30]

Fases de WSDM

El modelo de diseño de sitios web se divide en cuatro fases: modelo de usuario, diseño conceptual, diseño de la implementación e implementación. A su vez, el modelo de usuario se divide en dos subfases: clasificación y descripción. Por su parte el diseño conceptual se divide en otras dos subfases: modelado de objetos y diseño navegacional (véase **Figura 10**).

Fase 1- Modelado de usuarios

En esta primera fase se intenta detectar los perfiles de usuarios que se van a presentar en la aplicación. Para ello, se deben realizar dos tareas:

- ✓ Clasificación de usuarios
- ✓ Descripción de los grupos de usuarios

Fase 2- Diseño Conceptual

Se construye un modelo orientado a objetos que represente el dominio de la aplicación usando las técnicas propias de la orientación a objetos.

Esta fase en WSDM se divide en las siguientes dos etapas:

- ✓ Modelado de objetos
- ✓ Diseño navegacional

Fase 3- Diseño de la implementación

Una vez definido el modelo es necesario diseñar la interfaz y el entorno de usuario del sistema.

Fase 4- Implementación

En esta fase se pretende conseguir la aplicación ejecutable final en base a los resultados de las fases anteriores.

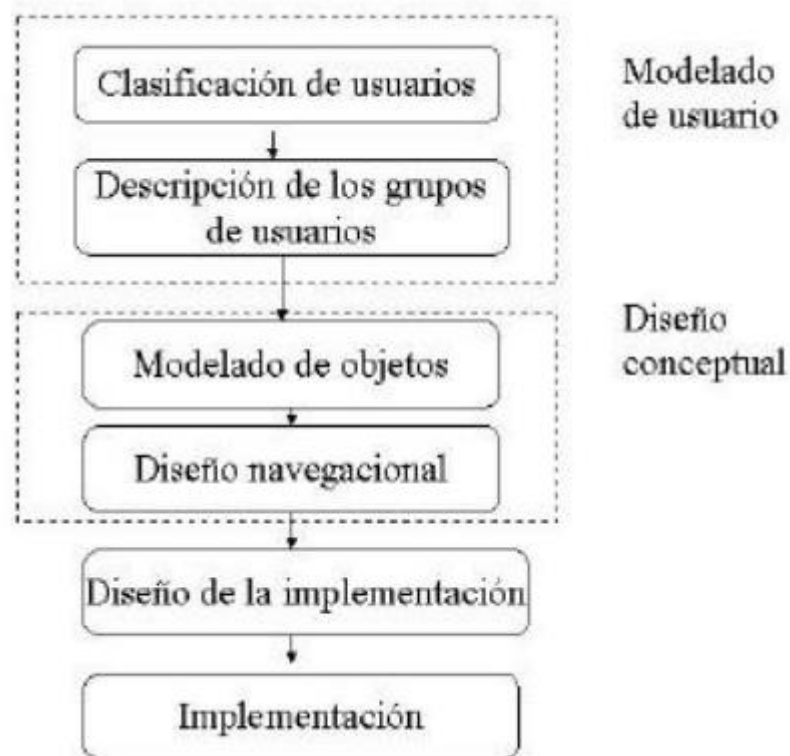


Figura 10 Esquema de fases de WSDM

2.2.19.3. Metodología MIDAS

MIDAS es una metodología genérica que se basa en la utilización de modelos para el desarrollo de Sistemas de Información Web (SIW). MIDAS propone un proceso interactivo e incremental basado en prototipado, y utiliza prácticas extraídas de metodologías ágiles, como XP (eXtreme Programming). Por lo tanto, una característica diferenciadora de MIDAS es que es una metodología ligera, que se ha definido para satisfacer tanto las necesidades de los clientes como de los desarrolladores [31].

MIDAS se ha definido con el fin de satisfacer los siguientes objetivos:

- ✓ Proporcionar a los desarrolladores una metodología basada en modelos que guíe su trabajo siguiendo su forma habitual de trabajar.
- ✓ Soportar un desarrollo de software rápido, con el fin de asegurar a los clientes una primera versión del software en el menor tiempo posible.
- ✓ Reducir la cantidad de documentación generada durante el desarrollo del SIW.

MIDAS proporcionará al cliente los productos en un tiempo corto, permitiendo introducir en cada iteración nuevos requisitos que no se identificaron en iteraciones anteriores del desarrollo. Otra ventaja será que las pruebas se harán para cada iteración con lo que se reducirá el riesgo de que se produzcan fallos [31].

MIDAS propone distintas iteraciones (véase **Figura 11**) y al final de cada una de ellas se obtiene una nueva versión del producto.

- ✓ En una primera iteración, MIDAS/SD, que constituye el núcleo del proceso, se definen los requisitos y la arquitectura del sistema.
- ✓ En la segunda iteración denominada MIDAS/HT, se desarrolla un primer prototipo del SIW, construyendo el hipertexto con páginas estáticas en HTML para proporcionar al cliente una primera versión del producto en un corto periodo de tiempo.
- ✓ En la tercera iteración, denominada MIDAS/DB, se implementará una nueva versión del hipertexto con páginas dinámicas en HTML o XML, recibiendo como entrada el prototipo definido en la iteración previa.
- ✓ En una iteración adicional, denominada MIDAS/FC, se desarrollan los servicios y la lógica del SIW.
- ✓ En otra iteración, MIDAS/TST, se probará el sistema.

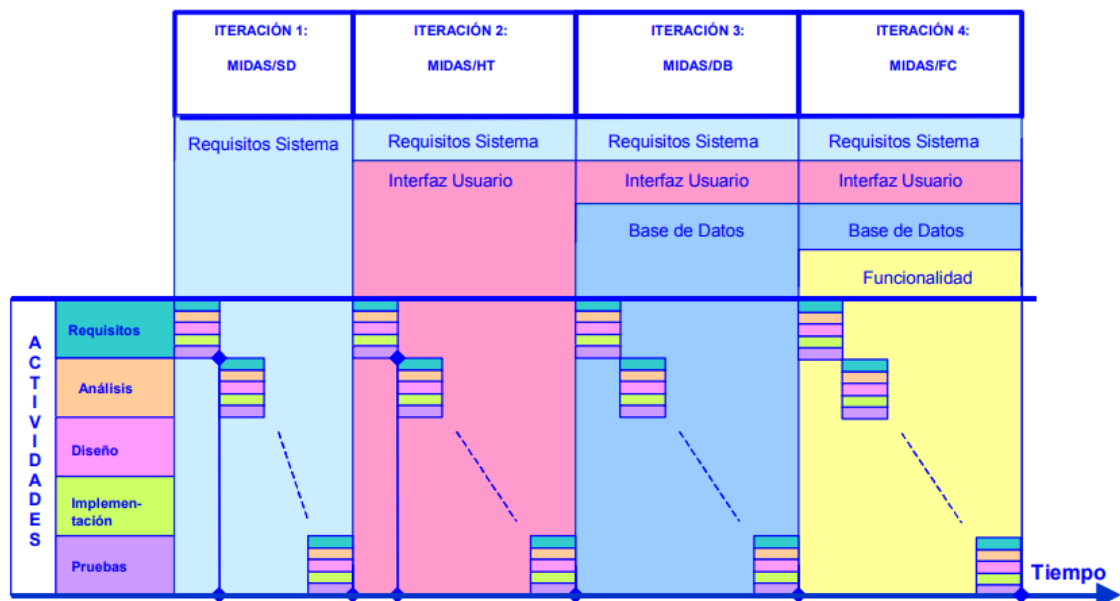


Figura 11 Proceso MIDAS

2.2.19.4. UWE (Metodología de Ingeniería Web basada en UML)

Es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario. [32]

Principales aspectos

Los principales de aspectos en los que se fundamenta UWE son los siguientes:

- ✓ Uso de una notación estándar, para todos los modelos (UML: Lenguaje de modelado unificado).
- ✓ Definición de métodos: Definición de los pasos para la construcción de los diferentes modelos.
- ✓ Especificación de Restricciones: Se recomienda el uso de restricciones escritas (OCL: Lenguaje de restricciones de objetos) para aumentar la exactitud de los modelos.

Fases del Desarrollo Web

- ✓ **Análisis de Requisitos:** Fija los requisitos funcionales de la aplicación Web para reflejarlos en un modelo de casos de uso.
- ✓ **Diseño Conceptual:** Materializado en un modelo de dominio, considerando los requisitos reflejados en los casos de uso.
- ✓ **Diseño Navegacional:** Lo podemos subdividir en:
 1. Modelo del Espacio de Navegacional.
 2. Modelo de la Estructura de navegación: Muestra la forma de navegar ante el espacio de navegación.
- ✓ **Diseño de Presentación:** Representa las vistas del interfaz del usuario mediante modelos estándares de interacción UML.

CAPÍTULO 3

Metodología

3.1. Modalidad de la investigación

El presente trabajo tiene las siguientes modalidades de investigación:

Modalidad Bibliográfica o Documentada Se considera esta modalidad ya que se recurre a diferentes fuentes obtenidas de libros, artículos científicos, tesis desarrolladas en Universidades para profundizar enfoques con respecto al tema de la investigación.

Modalidad aplicada Por la utilización de los conocimientos adquiridos a lo largo de la carrera universitaria.

3.2. Recolección de información

Se obtuvo la información relevante para el portal web a partir de una entrevista con el Administrador y el personal encargado de Recursos Informáticos de la Unidad Desconcentrada de Terminales.

3.3. Procesamiento y análisis de datos

Para el procesamiento y análisis de la información se realizaron las siguientes actividades:

1. Recolección de datos mediante entrevista.
2. Revisión y análisis de la información recogida (véase anexo 3).

3.4. Desarrollo del proyecto

Para la composición del presente proyecto se registrarán a los siguientes procesos:

Las actividades para el cumplimiento del objetivo “Recolectar los requerimientos para el análisis y la implementación del portal web”

- ✓ Visitar sitios web afines con el proyecto para determinar aspectos relacionados a funcionalidad y diseño.

- ✓ Entrevistar al Administrador y al personal encargado de Recursos Informáticos de la UDT para la recolección de información requerida en el portal web.

Las actividades para el cumplimiento del objetivo “Seleccionar una metodología ágil orientada al campo de aplicaciones web que facilite el desarrollo del proyecto planteado”

- ✓ Investigar las metodologías más utilizadas en el ámbito del desarrollo web
- ✓ Escoger una metodología que vaya más acorde al proyecto

Las actividades para el cumplimiento del objetivo “Implementar el portal web para la gestión de la información de la Unidad Desconcentrada de Terminales de Ambato”

- ✓ Diseño de la base de datos para el portal web.
- ✓ Diseño del portal web.
- ✓ Desarrollo del portal web.
- ✓ Pruebas finales
- ✓ Documentación del manual técnico y el manual de usuario

CAPÍTULO 4

Desarrollo de la propuesta

4.1 Datos informativos

Título

PORTAL WEB PARA LA GESTIÓN DE LA INFORMACIÓN DE LA UNIDAD DESCONCENTRADA DE TERMINALES DE AMBATO.

Instituciones

Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

Unidad Desconcentrada de Terminales.

Beneficiarios

Administración de la Unidad Desconcentrada de Terminales y usuarios en general.

Ubicación de la empresa

Provincia: Tungurahua

Cantón: Ambato

Dirección: Av. Colombia 332 y Paraguay

Teléfono: (03) 2 522367

Equipo responsable

Tutor: Ing. Carlos Núñez.

Investigador: Sr. Patricio Landa.

4.2. Justificación de la Metodología

Al momento de desarrollar aplicaciones web nos encontramos con una gran variedad de metodologías ágiles de entre las cuales se ha escogido las siguientes OOHDM, WSDM, UWE, MIDAS para realizar una comparativa y conocer más sobre cada una de ellas (véase **Tabla 1**).

Nombre de la metodología	Ventajas	Desventajas	Tipo de proyectos
OOHDM (Metodología de desarrollo de hipertexto orientada a objetos)	Hace una separación clara entre las capas (conceptual, navegacional y visual). Realiza un estudio profundo de los aspectos de interfaz	Carece en cuanto a funcionamiento del sistema. No ofrece ningún mecanismo para trabajar con múltiples actores.	Diseñado para aplicaciones multimedia
WSDM (método de diseño para sitios web)	Es adaptable a nuevos requisitos. Mayor separación de capas.	Al utilizar más tiempo en la obtención de requerimientos utiliza más tiempo para su desarrollo.	Diseñado para proyectos con aplicaciones web muy robustas.
UWE (Método de Ingeniería Web basada en UML)	Uso exclusivo de estándares reconocidos como UML. Establece un formalismo más rígido.	Uso de restricciones escritas.	Diseñado para proyectos especializados en la especificación de aplicaciones adaptativas.
MIDAS (Metodología Interactiva de Desarrollo de Aplicaciones de Software)	Ligera y basado en prototipado. Permite introducir en cada iteración nuevos requisitos que no se identificaron en iteraciones anteriores. Soporta un desarrollo de software rápido, con el fin de asegurar a los clientes una primera versión del software en el menor tiempo posible.		Sistema de Información Web

Tabla 1 Cuadro comparativo metodología ágiles web [30]

Se ha optado por recurrir a la metodología MIDAS, en función del cuadro comparativo anterior, para la implementación de este proyecto. Esta metodología, toma buenas prácticas de otras metodologías ágiles como la XP, la cual da mucha prioridad a las necesidades del cliente, permitiendo incorporarlo como si fuera un miembro más del equipo. Al ser ligera e iterativa permite obtener productos tangibles para el cliente en

cada iteración, ya sea un documento o un prototipo del software y gracias a los prototipos el cliente tiene la posibilidad de validar el producto continuamente.

Además, permite incorporar o modificar necesidades no detectadas o no planteadas en las primeras etapas de definición del producto. Esto garantiza sucesivas entregas al cliente hasta completar el producto final.

Permite disminuir la cantidad de documentación generada durante el desarrollo del proyecto. Admite ejecutar pruebas en cada iteración, con ello se reducirán los riesgos de que se produzcan fallos.

4.3. MIDAS/SD (Requisitos sistema)

En la primera fase de la metodología midas (MIDAS/SD), se establecieron los requerimientos del sistema, el alcance que tendría; también se procedió a entender los procesos realizados en la institución, por medio de la entrevista al administrador y encargado de recursos informáticos.

Con las primeras entrevistas, conversaciones, se ha elaborado un ERS para determinar los requisitos.

4.3.1. Especificación de Requerimientos de Software

4.3.1.1. Introducción

Propósito

La presente Especificación de Requerimientos de Software (ERS) del Portal Web de la Unidad Desconcentrada de Terminales se ha elaborado para ser utilizado como un manual durante el desarrollo y posterior implementación. También se ha descrito cada uno de los requerimientos, que se logró obtener de la investigación realizada, las características del portal web, lo que puede y no puede realizar.

La ERS podrá ser utilizada como descripción, para obtener información sobre la administración, funcionamiento y mantenimiento. Además, en caso de que se requiera realizar mejoras o modificaciones, esta ERS servirá como guía para que cualquier otro desarrollador conozca el sistema.

Esta ERS está dirigida a los desarrolladores y a todo el personal que tiene acceso al sistema, como son los administradores.

Alcance

Identificación del sistema

El Portal Web de la Unidad Desconcentrada de Terminales es un sistema que se implementará para realizar de manera sencilla, rápida y segura la gestión de información del Terminal Terrestre y se considera como un complemento a su sistema local.

Objetivo del sistema

Al igual que el sistema local del Terminal Terrestre el Portal Web permite automatizar los procesos con la diferencia que se podrá gestionar desde cualquier lugar puesto que será visible desde Internet remediando así las limitaciones que trae consigo un sistema local.

Módulos a considerar

Módulo de seguridad de acceso al sistema

Mediante el uso de usuarios y contraseñas se controlará el ingreso al sistema en base a perfiles. Se deberá determinar a qué módulos pueden acceder un usuario.

Módulo de Comunicados/Noticias

Se podrán registrar los comunicados y noticias que la institución desea dar a conocer a la ciudadanía (incluyendo imágenes o videos en caso de disponer de ellos).

Módulo de Recursos

Se administrará los recursos entendiéndose esto como los archivos de tipo imagen, video, pdf que se manejen en el portal web.

Módulo de Servicios

Se administrará los diferentes servicios que la institución ofrece a la ciudadanía (incluyendo imágenes en caso de disponer de ellos).

Módulo áreas de transferencia

Se administrará la información (de contacto y ubicación incluyendo imágenes en caso de disponer de ellos) correspondiente a las áreas de transferencia entendiéndose estas como pequeños terminales cuya finalidad es proporcionar el servicio de transporte únicamente dentro de la provincia de Tungurahua.

Módulo de tarifas de transporte

Se administrará las tarifas de transporte los cuales deben ser en base a lo establecido por la Agencia Nacional de Tránsito.

Módulo preguntas frecuentes/glosario de términos

Se podrán registrar aquellas preguntas que con más frecuencia la ciudadanía hace llegar a la administración del terminal y también definir aquellos términos ambiguos que son utilizados en el portal web.

Módulo institución

Se administrará la información correspondiente a la planificación estratégica del terminal terrestre, así como también la de contacto.

Módulo transportistas

Se administrará la información correspondiente a las operadoras de transporte, las unidades de transporte además de sus propietarios y conductores.

Módulo frecuencias

Se administrará la información correspondiente a las frecuencias de viaje autorizadas por la Agencia Nacional de Tránsito, destinos y andenes.

Módulo áreas y equipos

Se gestionará la información referente a los equipos electrónicos que se usan en las instalaciones del terminal como computadores, impresoras, cámaras de vigilancia, etc. así como también las áreas donde se encuentran ubicados dichos equipos.

Módulo ingresos

Se administrará la información correspondiente a los conceptos de cobro, los mismos son las razones por las cuales se realizan las recaudaciones como por ejemplo: arrojar basura en los andenes de abordaje, recoger pasajeros en zonas no autorizadas, etc.

Módulo de usuarios y perfiles

Se gestionará los perfiles y usuarios que tendrán acceso a la administración del portal web tomando en cuenta aspectos de seguridad como la encriptación de contraseñas.

Módulo dispositivos RFID

Mediante el uso de dispositivos RFID se realiza la identificación de las diferentes unidades de transporte que entran y salen del terminal, estas lecturas quedan registradas en el sistema local que el terminal posee, pero para el contexto de este portal web únicamente se manejará estos dispositivos con el propósito de asignarlos o cambiarlos de unidad de transporte.

Definiciones, siglas y abreviaciones

Del negocio

- a. **Próximas salidas:** Unidades de transporte que a partir de una hora específica en adelante están listas para salir desde el terminal terrestre hacia su respectivo destino.
- b. **Cooperativa:** Entidad que presta los servicios de transporte, cargas y encomiendas.
- c. **Directorio cooperativas:** Información de todas las cooperativas de transporte que laboran en el terminal terrestre.
- d. **Tarifa transporte:** Valor de pasaje para trasladarse desde un origen hacia un destino.
- e. **Comunicado:** Es una declaración, informe, nota o un parte que comunica una información para su conocimiento público.
- f. **Área transferencia:** Terminal terrestre cuyo objetivo es brindar servicio de transporte únicamente dentro de una provincia determinada.
- g. **Interprovincial:** El servicio de transporte público interprovincial es aquel que opera, bajo cualquier tipo, dentro de los límites del territorio nacional.
- h. **Intraprovincial:** El servicio de transporte público intraprovincial conocido también como intercantonal es aquel que opera dentro de los límites provinciales.
- i. **Interpacantonal:** El servicio de transporte público intracantonal conocido también como interparroquial, es aquel que opera dentro de los límites cantonales.
- j. **Frecuencia:** Horario asignado por la Agencia Nacional de Tránsito (ANT) para la salida de unidades de transporte hacia diferentes destinos.

Descripción

El ERS muestra información sobre los requisitos del sistema, de una manera general, sin describir de manera profunda el sistema, lo que permitió obtener un gran entendimiento del sistema con gran facilidad. Siguiendo el documento se detallan los requerimientos específicos del sistema, de manera profunda para permitir un diseño del sistema que cumplan los requerimientos del usuario, y después poder realizar pruebas que corroboren que el sistema efectuó los requisitos planteados en este documento.

4.3.1.2. Descripción General

Perspectivas del Producto

El Portal Web es un sistema informático que será desarrollado para la Unidad Desconcentrada de Terminales Ambato. El sistema está orientado a la administración de información que la entidad maneja y considera importante.

Funciones del Producto

Las funciones que deberá contener el sistema serán los siguientes agrupados por cada módulo del sistema:

a) Funciones del Módulo usuarios y perfiles

Sub Módulo Usuarios

- Consultar todos los usuarios existentes
- Consultar un usuario
- Ingresar un nuevo usuario
- Modificar un usuario existente
- Eliminar un usuario
- Validar el ingreso de un usuario al sistema

Sub Módulo Perfiles

- Consultar todos los perfiles existentes
- Consultar un perfil
- Ingresar un nuevo perfil
- Modificar un perfil existente
- Eliminar un perfil

b) Funciones del Módulo de Comunicados/Noticias

- Consultar todos los comunicados y noticias
- Consultar un comunicado o noticia
- Ingresar un nuevo comunicado o noticia
- Modificar un comunicado o noticia existente
- Eliminar un comunicado o noticia
- Validar el contenido imágenes (gif, jpeg, jpg, png, svg, blob), archivos (txt, pdf, doc), videos (mp4, webm, ogg)

c) Funciones del Módulo de Recursos

- Consultar todos los recursos
- Consultar un recurso
- Ingresar un nuevo recurso
- Modificar un recurso existente
- Eliminar un recurso
- Validar el tipo de recurso (imagen, audio, video, archivos tipo word y pdf)

d) Funciones del Módulo de Servicios

- Consultar todos los servicios
- Consultar un servicio
- Ingresar un nuevo servicio
- Modificar un servicio existente
- Eliminar un servicio
- Validar imagen (gif, jpeg, jpg, png, svg, blob)

e) Funciones del Módulo de Áreas de Transferencia

- Consultar todas las áreas de transferencia
- Consultar un área de transferencia
- Ingresar una nueva área de transferencia
- Modificar un área de transferencia existente
- Eliminar un área de transferencia
- Validar imagen (gif, jpeg, jpg, png, svg, blob)
- Validar nombre de las áreas de transferencia

f) Funciones del Módulo de Tarifas de Transporte

- Consultar todas las tarifas de transporte
- Consultar una tarifa de transporte
- Ingresar una nueva tarifa de transporte
- Modificar una tarifa de transporte existente
- Eliminar una tarifa de transporte

g) Funciones del Módulo de Preguntas frecuentes/Glosario de términos

- Consultar todas las preguntas frecuentes o términos de glosario
- Consultar una pregunta frecuente o término de glosario
- Ingresar una nueva pregunta frecuente o término de glosario
- Modificar una pregunta frecuente o término de glosario existente
- Eliminar una pregunta frecuente o término de glosario

h) Funciones del Módulo de Institución

- Consultar información institucional, de contacto y ubicación, planificación estratégica
- Modificar información institucional, de contacto y ubicación, planificación estratégica
- Validar logo (gif, jpeg, jpg, png, svg, blob)

i) Funciones del Módulo de Transportistas

Sub Módulo Propietarios/Socios

- Consultar todos los propietarios/socios de las unidades de transporte
- Consultar un propietario/socio de unidad de transporte
- Ingresar un nuevo propietario/socio de unidad de transporte
- Modificar un propietario/socio de unidad de transporte existente
- Eliminar un propietario/socio de unidad de transporte

Sub Módulo Operadoras de Transporte

- Consultar todas las operadoras de transporte
- Consultar una operadora de transporte
- Ingresar una nueva operadora de transporte
- Modificar una operadora de transporte existente
- Eliminar una operadora de transporte

Sub Módulo Unidades de Transporte

- Consultar todas las unidades de transporte
- Consultar una unidad de transporte
- Ingresar una nueva unidad de transporte
- Modificar una unidad de transporte existente
- Eliminar una unidad de transporte

Sub Módulo Conductores

- Consultar todos los conductores
- Consultar un conductor
- Ingresar un nuevo conductor
- Modificar un conductor existente
- Eliminar un conductor

j) Funciones del Módulo de Frecuencias

Sub Módulo Frecuencias de Viaje

- Consultar todas las frecuencias de viaje
- Consultar una frecuencia de viaje
- Ingresar una nueva frecuencia de viaje (autorizada por la Agencia Nacional de Tránsito)
- Modificar una frecuencia de viaje existente
- Eliminar una frecuencia de viaje

Sub Módulo Tipos de Frecuencia

- Consultar todos los tipos de frecuencias
- Consultar un tipo de frecuencia
- Ingresar un nuevo tipo de frecuencia
- Modificar un tipo de frecuencia existente
- Eliminar un tipo de frecuencia

Sub Módulo Tipos de Ruta

- Consultar todos los tipos de rutas
- Consultar un tipo de ruta
- Ingresar un nuevo tipo de ruta
- Modificar un tipo de ruta existente
- Eliminar un tipo de ruta

Sub Módulo Destinos

- Consultar todos los destinos
- Consultar un destino
- Ingresar un nuevo destino
- Modificar un destino existente
- Eliminar un destino

Sub Módulo Regiones

- Consultar todas las regiones
- Consultar una región
- Ingresar una nueva región
- Modificar una región existente
- Eliminar una región

Sub Módulo Andenes

- Consultar todos los andenes
- Consultar un andén
- Ingresar un nuevo andén
- Modificar un andén existente
- Eliminar un andén

k) Funciones del Módulo de Áreas y equipos

Sub Módulo Áreas

- Consultar todas las áreas del terminal terrestre
- Consultar un área
- Ingresar una nueva área
- Modificar un área existente
- Eliminar un área

Sub Módulo Equipos

- Consultar todos los equipos electrónicos
- Consultar un equipo
- Ingresar un nuevo equipo
- Modificar un equipo existente
- Eliminar un equipo

Sub Módulo Tipos de Equipos

- Consultar todos los tipos de equipos
- Consultar un tipo de equipo
- Ingresar un nuevo tipo de equipo
- Modificar un tipo de equipo existente
- Eliminar un tipo de equipo

l) Funciones del Módulo de Ingresos

Sub Módulo Conceptos de Cobro

- Consultar todos los conceptos de cobro

- Consultar un concepto de cobro
- Ingresar un nuevo concepto de cobro
- Modificar un concepto de cobro existente
- Eliminar un concepto de cobro

Sub Módulo Grupos de Cobro

- Consultar todos los grupos o categorías de cobro
- Consultar un grupo de cobro
- Ingresar un nuevo grupo de cobro
- Modificar un grupo de cobro existente
- Eliminar un grupo de cobro

m) Funciones del Módulo dispositivos RFID

- Consultar todos los dispositivos RFID
- Consultar un dispositivo RFID
- Ingresar un nuevo dispositivo RFID
- Modificar un dispositivo RFID existente
- Eliminar un dispositivo RFID

Características de usuarios

Para el Portal Web se tienen dos grupos de usuarios que interactuarán y administrarán: Visitante, Administrador del Sistema.

Perfiles de Usuario:

Cada usuario tendrá un perfil específico para que su interacción con el sistema sea correcta y no conlleve a fallos.

Administrador. Se encarga de la administración de toda la información que el portal web permite gestionar.

Visitante. Cualquier usuario que comúnmente navega en Internet esta persona puede ver la información que un administrador ha registrado y se limita únicamente a consumir el contenido del portal web.

Restricciones

Consideraciones de seguridad

Cada usuario deberá autenticarse mediante nombre de usuario (alias) y contraseña. Todas las claves de seguridad deben estar seguras y en su defecto encriptadas en la base de datos para dar una buena seguridad al sistema y su información.

Interfaces de Sistema

El sistema es dependiente, y será desarrollado utilizando una arquitectura tipo MVC. Será diseñado siguiendo una metodología ágil para el desarrollo de SIW, Midas.

El Portal Web será desarrollado sobre la plataforma Java 1.7, la base de datos a utilizarse es SQL Server 2017 Developer Edition, la GUI será desarrollada para ser utilizada en un navegador web.

Interfaces de Usuario

La GUI será diseñada con el framework Bootstrap mediante el cual se obtendrá un diseño web que se adapte a los diferentes tamaños de pantalla existentes en el mercado.

Los usuarios accederán al sistema mediante páginas JSP.

Se implementarán diferentes tipos de elementos que permitan manipular la información:

- Ingreso de información mediante formularios Web.
- Métodos de validación de información en los formularios, así como también de los controles del aplicativo.
- Botones para ejecutar los diferentes procesos de mantenimiento del Portal Web, como guardar, editar, eliminar.
- Salidas mediante tablas, que mostrarán la información requerida por el usuario.

4.3.1.3. Requisitos Específicos

En esta sección de la ERS se muestran todos los requisitos para el Portal Web, con un nivel de detalle suficiente que permitirá diseñar el sistema de tal forma que cumpla con todos los requerimientos de la entidad interesada, utilizando MIDAS como metodología de desarrollo.

Interfaces Externas

El Portal Web dependerá en parte de la información que proporcione el sistema local del terminal terrestre por lo que interactuará con dicho sistema.

Requisitos funcionales

A continuación, se detallan las funciones del Portal Web que van a ser implementadas para que cumplan con todos los requisitos especificados por la Administración de la Unidad Desconcentrada de Terminales.

i) Funciones del Módulo de Perfiles y Usuarios

1) Consultar todos los usuarios existentes

- Introducción: Despliega todos los usuarios que ha registrado el sistema, en la Vista Web de la pantalla Usuarios.
- Nombre de la Función: getUsers()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Usuario, CI_Ruc, Nombre, Usuario, Clave, Estado e Id_Perfil de todos los registros de la tabla Usuario. Los ordena por Id_Usuario.
- Salida: Devuelve los registros obtenidos de la tabla Usuario.

2) Consultar un usuario

- Introducción: Despliega todos los datos de un usuario, dado su id, en el Formulario Web de la pantalla Usuarios.
- Nombre de la Función: getUserById()
- Entradas: Código del usuario.
- Proceso: Trae de la base de datos, todos los datos de un usuario, dado su código.
- Salida: Devuelve el registro obtenido.

3) Ingresar un nuevo usuario

- Introducción: Añade un nuevo usuario a la base de datos, con su respectivo perfil de acceso al sistema.
- Nombre de la Función: saveUsuario()
- Entradas: Todos los datos del usuario a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Encripta la contraseña para mayor seguridad. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

4) Modificar un usuario existente

- Introducción: Modifica uno o más datos de un usuario existente.
- Nombre de la Función: saveUsuario()
- Entradas: Todos los datos del usuario a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Encripta la contraseña para mayor seguridad. Modifica los datos del usuario.
- Salida: Éxito o fracaso en la modificación.

5) Eliminar un usuario

- Introducción: Elimina físicamente de la base de datos, el registro de un usuario, dado su código.
- Nombre de la Función: deleteUsuario()

- Entradas: Código del usuario.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del usuario que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

6) Consultar todos los perfiles existentes

- Introducción: Despliega todos los perfiles que ha registrado el sistema, en la Vista Web de la pantalla Perfiles.
- Nombre de la Función: getPerfiles()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Perfil, Nombre y Descripción de todos los registros de la tabla Perfil. Los ordena por Id_Perfil.
- Salida: Devuelve los registros obtenidos de la tabla Perfil.

7) Consultar un perfil

- Introducción: Despliega todos los datos de un perfil, dado su id, en el Formulario Web de la pantalla Perfiles.
- Nombre de la Función: getPerfilById()
- Entradas: Código de perfil.
- Proceso: Trae de la base de datos, todos los datos de un perfil, dado su código.
- Salida: Devuelve el registro obtenido.

8) Ingresar un nuevo perfil

- Introducción: Añade un nuevo perfil a la base de datos.
- Nombre de la Función: savePerfil()
- Entradas: Todos los datos del perfil a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

9) Modificar un perfil existente

- Introducción: Modifica uno o más datos de un perfil existente.
- Nombre de la Función: savePerfil()
- Entradas: Todos los datos del perfil a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del perfil.
- Salida: Éxito o fracaso en la modificación.

10) Eliminar un perfil

- Introducción: Elimina físicamente de la base de datos, el registro de un perfil, dado su código.
- Nombre de la Función: deletePerfil()

- Entradas: Código del perfil.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del perfil que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

11) Validar el ingreso de un usuario al sistema

- Introducción: Valida que el Nombre de usuario (alias) y la contraseña ingresada, sean de un usuario registrado del sistema, permitiéndole el ingreso al mismo.
- Nombre de la Función: loadUserByUsername()
- Entradas: Alias y contraseña del usuario.
- Proceso: Verifica el alias y la contraseña provistos, en la base de datos, si existen y son correctos, es decir pertenecen a un usuario del sistema, busca el código del usuario para posteriormente obtener su perfil de acceso al sistema.
- Salida: Si son correctos los datos, devuelve el código del usuario, si no lo son devuelve fracaso en la ejecución de la función.

ii) Funciones del Módulo de Comunicados/Noticias

12) Consultar todos los comunicados y noticias existentes

- Introducción: Despliega todos los comunicados y noticias que ha registrado el sistema, en la Vista Web de la pantalla Comunicado/Noticia.
- Nombre de la Función: getComunicadoNoticia()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos ID, TITULO, CONTENIDO, TIPO y FECHA de todos los registros de la tabla COMUNICADO_NOTICIA. Los ordena por ID.
- Salida: Devuelve los registros obtenidos de la tabla COMUNICADO_NOTICIA.

13) Consultar un comunicado o noticia

- Introducción: Despliega todos los datos de un comunicado o noticia, dado su id, en el Formulario Web de la pantalla Comunicado/Noticia.
- Nombre de la Función: getComunicadoNoticiaById()
- Entradas: Código de comunicado/noticia.
- Proceso: Trae de la base de datos, todos los datos de un comunicado/noticia, dado su código.
- Salida: Devuelve el registro obtenido.

14) Ingresar un nuevo comunicado o noticia

- Introducción: Añade un nuevo comunicado/noticia a la base de datos.
- Nombre de la Función: saveComunicadoNoticia()
- Entradas: Todos los datos del comunicado/noticia a ser ingresados.

- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

15) Modificar un comunicado o noticia existente

- Introducción: Modifica uno o más datos de un comunicado/noticia existente.
- Nombre de la Función: saveComunicadoNoticia ()
- Entradas: Todos los datos del comunicado/noticia a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del comunicado/noticia.
- Salida: Éxito o fracaso en la modificación.

16) Eliminar un comunicado o noticia

- Introducción: Elimina físicamente de la base de datos, el registro de un comunicado/noticia, dado su código.
- Nombre de la Función: deleteComunicadoNoticia()
- Entradas: Código del comunicado/noticia.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del comunicado/noticia que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

17) Validar imágenes

- Introducción: Valida que las imágenes que se suban al servidor sean de los formatos (gif, jpeg, jpg, png, svg, blob).
- Nombre de la Función: ImageUpload()
- Entradas: Imagen.
- Proceso: Valida que la imagen pertenezca a los formatos establecidos y permitidos.
- Salida: Si el archivo de imagen es correcto, devuelve la url donde ha sido ubicado la imagen, si no lo es devuelve fracaso en la ejecución de la función.

18) Validar archivos

- Introducción: Valida que los archivos que se suban al servidor sean de los formatos (txt, pdf, doc).
- Nombre de la Función: FileUpload()
- Entradas: Archivo.
- Proceso: Valida que el archivo pertenezca a los formatos establecidos y permitidos.
- Salida: Si el archivo es correcto, devuelve la url donde ha sido ubicado el archivo, si no lo es devuelve fracaso en la ejecución de la función.

19) Validar videos

- Introducción: Valida que los videos que se suban al servidor sean de los formatos (mp4, webm, ogg).
- Nombre de la Función: VideoUpload()
- Entradas: Video.
- Proceso: Valida que el video pertenezca a los formatos establecidos y permitidos.
- Salida: Si el video es correcto, devuelve la url donde ha sido ubicado el video, si no lo es devuelve fracaso en la ejecución de la función.

iii) Funciones del Módulo de Recursos

20) Consultar todos los recursos

- Introducción: Despliega todos los recursos que ha registrado el sistema, en la Vista Web de la pantalla Recurso.
- Nombre de la Función: getRecursos()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos ID, NOMBRE, URL, DESCRIPCIÓN y TIPO de todos los registros de la tabla RECURSO. Los ordena por ID.
- Salida: Devuelve los registros obtenidos de la tabla RECURSO.

21) Consultar un recurso

- Introducción: Despliega todos los datos de un recurso, dado su id, en el Formulario Web de la pantalla Recurso.
- Nombre de la Función: getRecursoById()
- Entradas: Código de recurso.
- Proceso: Trae de la base de datos, todos los datos de un recurso, dado su código.

22) Ingresar un nuevo recurso

- Introducción: Añade un nuevo recurso a la base de datos.
- Nombre de la Función: saveRecurso()
- Entradas: Todos los datos del recurso a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

23) Modificar un recurso existente

- Introducción: Modifica uno o más datos de un recurso existente.
- Nombre de la Función: saveRecurso ()
- Entradas: Todos los datos del recurso a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del recurso.
- Salida: Éxito o fracaso en la modificación.

24) Eliminar un recurso

- Introducción: Elimina físicamente de la base de datos, el registro de un recurso, dado su código.
- Nombre de la Función: deleteRecurso()
- Entradas: Código del recurso.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del recurso que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

iv) Funciones del Módulo de Servicios

25) Consultar todos los servicios

- Introducción: Despliega todos los servicios que ha registrado el sistema, en la Vista Web de la pantalla Servicio.
- Nombre de la Función: getServicios()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos ID_SERVICIO, NOMBRE, DESCRIPCIÓN e IMAGEN de todos los registros de la tabla SERVICIO. Los ordena por ID_SERVICIO.
- Salida: Devuelve los registros obtenidos de la tabla SERVICIO.

26) Consultar un servicio

- Introducción: Despliega todos los datos de un servicio, dado su id, en el Formulario Web de la pantalla Servicio.
- Nombre de la Función: getServicioById()
- Entradas: Código de servicio.
- Proceso: Trae de la base de datos, todos los datos de un servicio, dado su código.

27) Ingresar un nuevo servicio

- Introducción: Añade un nuevo servicio a la base de datos.
- Nombre de la Función: saveServicio()
- Entradas: Todos los datos del servicio a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

28) Modificar un servicio existente

- Introducción: Modifica uno o más datos de un servicio existente.
- Nombre de la Función: saveServicio()
- Entradas: Todos los datos del servicio a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del servicio.
- Salida: Éxito o fracaso en la modificación.

29) Eliminar un servicio

- Introducción: Elimina físicamente de la base de datos, el registro de un servicio, dado su código.
- Nombre de la Función: deleteServicio()
- Entradas: Código del servicio.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del servicio que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

v) Funciones del Módulo de Áreas de Transferencia

30) Consultar todas las áreas de transferencia

- Introducción: Despliega todas las áreas de transferencia que ha registrado el sistema, en la Vista Web de la pantalla Área de Transferencia.
- Nombre de la Función: getAreasTransferencia()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos ID_AREA, NOMBRE, DIRECCION, TELEFONO, IMG, TIPO_TRANSP, HORARIO, LAT y LON de todos los registros de la tabla AREA_TRANSFERENCIA. Los ordena por ID_AREA.
- Salida: Devuelve los registros obtenidos de la tabla AREA_TRANSFERENCIA.

31) Consultar un área de transferencia

- Introducción: Despliega todos los datos de un área de transferencia, dado su id, en el Formulario Web de la pantalla Área de Transferencia.
- Nombre de la Función: getAreaTransferenciaById()
- Entradas: Código de área de transferencia.
- Proceso: Trae de la base de datos, todos los datos de un área de transferencia, dado su código.

32) Ingresar una nueva área de transferencia

- Introducción: Añade una nueva área de transferencia a la base de datos.
- Nombre de la Función: saveAreaTransferencia()
- Entradas: Todos los datos del área de transferencia a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

33) Modificar un área de transferencia existente

- Introducción: Modifica uno o más datos de un área de transferencia existente.
- Nombre de la Función: saveAreaTransferencia()
- Entradas: Todos los datos del área de transferencia a ser modificados.

- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del área de transferencia.
- Salida: Éxito o fracaso en la modificación.

34) Eliminar un área de transferencia

- Introducción: Elimina físicamente de la base de datos, el registro de un área de transferencia, dado su código.
- Nombre de la Función: deleteAreaTransferencia()
- Entradas: Código del área de transferencia.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del área de transferencia que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

vi) Funciones del Módulo de Tarifas de Transporte

35) Consultar todas las tarifas de transporte

- Introducción: Despliega todas las tarifas de transporte que ha registrado el sistema, en la Vista Web de la pantalla Tarifa de Transporte.
- Nombre de la Función: getTarifasTransporte()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos ID, ID_DESTINO, VALOR e IMG_DESTINO de todos los registros de la tabla TARIFAS_DESTINO. Los ordena por ID.
- Salida: Devuelve los registros obtenidos de la tabla TARIFAS_DESTINO.

36) Consultar una tarifa de transporte

- Introducción: Despliega todos los datos de una tarifa de transporte, dado su id, en el Formulario Web de la pantalla Tarifa Transporte.
- Nombre de la Función: getTarifaTransporteById()
- Entradas: Código de tarifa de transporte.
- Proceso: Trae de la base de datos, todos los datos de una tarifa de transporte, dado su código.

37) Ingresar una nueva tarifa de transporte

- Introducción: Añade una nueva tarifa de transporte a la base de datos.
- Nombre de la Función: saveTarifaTransporte()
- Entradas: Todos los datos de la tarifa de transporte a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

38) Modificar una tarifa de transporte existente

- Introducción: Modifica uno o más datos de una tarifa de transporte existente.
- Nombre de la Función: saveTarifaTransporte()

- Entradas: Todos los datos de la tarifa de transporte a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos de la tarifa de transporte.
- Salida: Éxito o fracaso en la modificación.

39) Eliminar una tarifa de transporte

- Introducción: Elimina físicamente de la base de datos, el registro de una tarifa de transporte, dado su código.
- Nombre de la Función: deleteTarifaTransporte()
- Entradas: Código de la tarifa de transporte.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro de la tarifa de transporte que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

vii) Funciones del Módulo de Preguntas frecuentes/Glosario de términos

40) Consultar todas las preguntas frecuentes o términos de glosario

- Introducción: Despliega todas las preguntas frecuentes y términos de glosario que ha registrado el sistema, en la Vista Web de la pantalla Pregunta/Glosario.
- Nombre de la Función: getPreguntaGlosario()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos ID, NOMBRE, DESCRIPCION y TIPO de todos los registros de la tabla PREGUNTA_GLOSARIO. Los ordena por ID.
- Salida: Devuelve los registros obtenidos de la tabla PREGUNTA_GLOSARIO.

41) Consultar una pregunta frecuente o término de glosario

- Introducción: Despliega todos los datos de una pregunta frecuente/termino glosario, dado su id, en el Formulario Web de la pantalla Pregunta/Glosario.
- Nombre de la Función: getPreguntaGlosarioById()
- Entradas: Código de pregunta frecuente/termino glosario.
- Proceso: Trae de la base de datos, todos los datos de una pregunta frecuente/termino glosario, dado su código.

42) Ingresar una nueva pregunta frecuente o término de glosario

- Introducción: Añade una nueva pregunta frecuente/termino glosario a la base de datos.
- Nombre de la Función: savePreguntaGlosario()
- Entradas: Todos los datos de la pregunta frecuente/termino glosario a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

43) Modificar una pregunta frecuente o término de glosario existente

- Introducción: Modifica uno o más datos de una pregunta frecuente/termino glosario existente.
- Nombre de la Función: savePreguntaGlosario()
- Entradas: Todos los datos de la pregunta frecuente/termino glosario a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos de la pregunta frecuente/termino glosario.
- Salida: Éxito o fracaso en la modificación.

44) Eliminar una pregunta frecuente o término de glosario

- Introducción: Elimina físicamente de la base de datos, el registro de pregunta frecuente/termino glosario, dado su código.
- Nombre de la Función: deletepreguntaGlosario()
- Entradas: Código de la pregunta frecuente/termino glosario.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro de pregunta frecuente/termino glosario que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

viii) Funciones del Módulo de Institución

45) Consultar información institucional, de contacto y ubicación, planificación estratégica

- Introducción: Despliega la información institucional que ha registrado el sistema, en la Vista Web de la pantalla Institución.
- Nombre de la Función: getInfoInstitucion()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos ID, NOMBRE, ABREVIATURA, SLOGAN, DIRECCION, TELEFONO, CORREO_ADMIN, HORARIO_ATENCION, LOGO, MISION, VISION, VALORES, ACERCA_DE, LAT y LON del primer registro de la tabla INSTITUCION.
- Salida: Devuelve los registros obtenidos de la tabla INSTITUCION.

46) Modificar información institucional, de contacto y ubicación, planificación estratégica

- Introducción: Modifica uno o más datos de carácter institucional.
- Nombre de la Función: updateInstitucion()
- Entradas: Todos los datos acerca de la institución.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Guarda los datos de la institución.
- Salida: Éxito o fracaso en la modificación.

ix) Funciones del Módulo Propietarios/Socios

47) Consultar todos los propietarios/socios de las unidades de transporte

- Introducción: Despliega todos los propietarios/socios que ha registrado el sistema, en la Vista Web de la pantalla Propietario.
- Nombre de la Función: getPropietarioSocio()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Propietario_Socio, CI_Ruc, Nombre, Razon_Social, Telefono y Direccion de todos los registros de la tabla PROPIETARIO_SOCIO. Los ordena por Id_Propietario_Socio.
- Salida: Devuelve los registros obtenidos de la tabla PROPIETARIO_SOCIO.

48) Consultar un propietario/socio de unidad de transporte

- Introducción: Despliega todos los datos de propietario/socio, dado su id, en el Formulario Web de la pantalla Propietario.
- Nombre de la Función: getPropietarioSocioById()
- Entradas: Código de propietario/socio.
- Proceso: Trae de la base de datos, todos los datos de un propietario/socio, dado su código.

49) Ingresar un nuevo propietario/socio de unidad de transporte

- Introducción: Añade un nuevo propietario/socio a la base de datos.
- Nombre de la Función: savePropietarioSocio()
- Entradas: Todos los datos de propietario/socio a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

50) Modificar un propietario/socio de unidad de transporte existente

- Introducción: Modifica uno o más datos de un propietario/socio existente.
- Nombre de la Función: savePropietarioSocio()
- Entradas: Todos los datos de propietario/socio a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos de propietario/socio.
- Salida: Éxito o fracaso en la modificación.

51) Eliminar un propietario/socio de unidad de transporte

- Introducción: Elimina físicamente de la base de datos, el registro de un propietario/socio, dado su código.
- Nombre de la Función: deletePropietarioSocio()
- Entradas: Código de propietario/socio.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro de propietario/socio que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

x) **Funciones del Módulo Operadoras de Transporte**

52) Consultar todas las operadoras de transporte

- Introducción: Despliega todas las operadoras de transporte que ha registrado el sistema, en la Vista Web de la pantalla Operadora Transporte.
- Nombre de la Función: getCooperativas()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Cooperativa, Ruc, Nombre, Razon_Social, Telefono, Id_Propietario_Socio y Direccion de todos los registros de la tabla COOPERATIVA. Los ordena por Id_Cooperativa.
- Salida: Devuelve los registros obtenidos de la tabla COOPERATIVA.

53) Consultar una operadora de transporte

- Introducción: Despliega todos los datos de una operadora de transporte, dado su id, en el Formulario Web de la pantalla Operadora Transporte.
- Nombre de la Función: getCooperativaById()
- Entradas: Código de operadora de transporte.
- Proceso: Trae de la base de datos, todos los datos de una operadora de transporte, dado su código.

54) Ingresar una nueva operadora de transporte

- Introducción: Añade una nueva operadora de transporte a la base de datos.
- Nombre de la Función: saveCooperativa()
- Entradas: Todos los datos de operadora de transporte a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

55) Modificar una operadora de transporte existente

- Introducción: Modifica uno o más datos de una operadora de transporte existente.
- Nombre de la Función: saveCooperativa()
- Entradas: Todos los datos de operadora de transporte a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos de operadora de transporte.
- Salida: Éxito o fracaso en la modificación.

56) Eliminar una operadora de transporte

- Introducción: Elimina físicamente de la base de datos, el registro de una operadora de transporte, dado su código.
- Nombre de la Función: deleteCooperativa()
- Entradas: Código de operadora de transporte.

- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro de operadora de transporte que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xi) Funciones del Módulo Unidades de Transporte

57) Consultar todas las unidades de transporte

- Introducción: Despliega todas las unidades de transporte que ha registrado el sistema, en la Vista Web de la pantalla Unidad Transporte.
- Nombre de la Función: getUnidadTransporte()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Unidad_Transporte, Numero_Disco, Placa, Id_Propietario_Socio, Id_Cooperativa e Id_Area de todos los registros de la tabla UNIDAD_TRANSPORTE. Los ordena por Id_Unidad_Transporte.
- Salida: Devuelve los registros obtenidos de la tabla UNIDAD_TRANSPORTE.

58) Consultar una unidad de transporte

- Introducción: Despliega todos los datos de una unidad de transporte, dado su id, en el Formulario Web de la pantalla Unidad Transporte.
- Nombre de la Función: getUnidadTransporteById()
- Entradas: Código de unidad de transporte.
- Proceso: Trae de la base de datos, todos los datos de una unidad de transporte, dado su código.

59) Ingresar una nueva unidad de transporte

- Introducción: Añade una nueva unidad de transporte a la base de datos.
- Nombre de la Función: saveUnidadTransporte()
- Entradas: Todos los datos de unidad de transporte a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

60) Modificar una unidad de transporte existente

- Introducción: Modifica uno o más datos de una unidad de transporte existente.
- Nombre de la Función: saveUnidadTransporte()
- Entradas: Todos los datos de unidad de transporte a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos de unidad de transporte.
- Salida: Éxito o fracaso en la modificación.

61) Eliminar una unidad de transporte

- Introducción: Elimina físicamente de la base de datos, el registro de una unidad de transporte, dado su código.
- Nombre de la Función: deleteUnidadTransporte()
- Entradas: Código de unidad de transporte.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro de unidad de transporte que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xii) Funciones del Módulo Conductores

62) Consultar todos los conductores

- Introducción: Despliega todos los conductores que ha registrado el sistema, en la Vista Web de la pantalla Conductor.
- Nombre de la Función: getConductores()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Chofer, Nombre, Cedula, Direccion, Telefono y Licencia de todos los registros de la tabla CONDUCTOR. Los ordena por Id_Chofer.
- Salida: Devuelve los registros obtenidos de la tabla CONDUCTOR.

63) Consultar un conductor

- Introducción: Despliega todos los datos de un conductor, dado su id, en el Formulario Web de la pantalla Conductor.
- Nombre de la Función: getConductorById()
- Entradas: Código de conductor.
- Proceso: Trae de la base de datos, todos los datos de un conductor, dado su código.

64) Ingresar un nuevo conductor

- Introducción: Añade un nuevo conductor a la base de datos.
- Nombre de la Función: saveConductor()
- Entradas: Todos los datos del conductor a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

65) Modificar un conductor existente

- Introducción: Modifica uno o más datos de un conductor existente.
- Nombre de la Función: saveConductor()
- Entradas: Todos los datos del conductor a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del conductor.
- Salida: Éxito o fracaso en la modificación.

66) Eliminar un conductor

- Introducción: Elimina físicamente de la base de datos, el registro de un conductor, dado su código.
- Nombre de la Función: deleteConductor()
- Entradas: Código de conductor.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del conductor que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xiii) Funciones del Módulo Frecuencias de Viaje

67) Consultar todas las frecuencias de viaje

- Introducción: Despliega todas las frecuencias de viaje que ha registrado el sistema, en la Vista Web de la pantalla Frecuencia Viaje.
- Nombre de la Función: getFrecuencias()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Frecuencia, Hora_Salida, Tipo_Ruta, Operadora, Origen, Destino, Tiempo_Anden, Dias y Tipo frecuencia de todos los registros de la tabla FRECUENCIA. Los ordena por Id_Frecuencia.
- Salida: Devuelve los registros obtenidos de la tabla FRECUENCIA.

68) Consultar una frecuencia de viaje

- Introducción: Despliega todos los datos de una frecuencia de viaje, dado su id, en el Formulario Web de la pantalla Frecuencia Viaje.
- Nombre de la Función: getFrecuenciaById()
- Entradas: Código de la frecuencia de viaje.
- Proceso: Trae de la base de datos, todos los datos de una frecuencia de viaje, dado su código.

69) Ingresar una nueva frecuencia de viaje (autorizada por la Agencia Nacional de Tránsito)

- Introducción: Añade una nueva frecuencia de viaje a la base de datos.
- Nombre de la Función: saveFrecuencia()
- Entradas: Todos los datos de la frecuencia de viaje a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

70) Modificar una frecuencia de viaje existente

- Introducción: Modifica uno o más datos de una frecuencia de viaje existente.
- Nombre de la Función: saveFrecuencia()
- Entradas: Todos los datos de la frecuencia de viaje a ser modificados.

- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos de la frecuencia de viaje.
- Salida: Éxito o fracaso en la modificación.

71) Eliminar una frecuencia de viaje

- Introducción: Elimina físicamente de la base de datos, el registro de una frecuencia de viaje, dado su código.
- Nombre de la Función: deleteFrecuencia()
- Entradas: Código de frecuencia de viaje.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro de frecuencia de viaje que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xiv) Funciones del Módulo Tipos de Frecuencia

72) Consultar todos los tipos de frecuencias

- Introducción: Despliega todos los tipos de frecuencias que ha registrado el sistema, en la Vista Web de la pantalla Tipo Frecuencia.
- Nombre de la Función: getTipoFrecuencias()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Tipo_Frecuencia, Nombre y Prefijo de todos los registros de la tabla TIPO_FRECUENCIA. Los ordena por Id_Tipo_Frecuencia.
- Salida: Devuelve los registros obtenidos de la tabla TIPO_FRECUENCIA.

73) Consultar un tipo de frecuencia

- Introducción: Despliega todos los datos de un tipo de frecuencia, dado su id, en el Formulario Web de la pantalla Tipo Frecuencia.
- Nombre de la Función: getTipoFrecuenciaById()
- Entradas: Código del tipo de frecuencia.
- Proceso: Trae de la base de datos, todos los datos de un tipo de frecuencia, dado su código.

74) Ingresar un nuevo tipo de frecuencia

- Introducción: Añade un nuevo tipo de frecuencia a la base de datos.
- Nombre de la Función: saveTipoFrecuencia()
- Entradas: Todos los datos del tipo de frecuencia a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

75) Modificar un tipo de frecuencia existente

- Introducción: Modifica uno o más datos de un tipo de frecuencia existente.
- Nombre de la Función: saveTipoFrecuencia()

- Entradas: Todos los datos del tipo de frecuencia a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del tipo de frecuencia.
- Salida: Éxito o fracaso en la modificación.

76) Eliminar un tipo de frecuencia

- Introducción: Elimina físicamente de la base de datos, el registro de un tipo de frecuencia, dado su código.
- Nombre de la Función: deleteTipoFrecuencia()
- Entradas: Código del tipo de frecuencia.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del tipo de frecuencia que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xv) Funciones del Módulo Tipos de Ruta

77) Consultar todos los tipos de rutas

- Introducción: Despliega todos los tipos de rutas que ha registrado el sistema, en la Vista Web de la pantalla Tipo Ruta.
- Nombre de la Función: getTiposRuta()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Tipo_Ruta, Nombre y Tiempo_Parqueo de todos los registros de la tabla TIPO_RUTA. Los ordena por Id_Tipo_Ruta.
- Salida: Devuelve los registros obtenidos de la tabla TIPO_RUTA.

78) Consultar un tipo de ruta

- Introducción: Despliega todos los datos de un tipo de ruta, dado su id, en el Formulario Web de la pantalla Tipo Ruta.
- Nombre de la Función: getTipoRutaById()
- Entradas: Código del tipo de ruta.
- Proceso: Trae de la base de datos, todos los datos de un tipo de ruta, dado su código.

79) Ingresar un nuevo tipo de ruta

- Introducción: Añade un nuevo tipo de ruta a la base de datos.
- Nombre de la Función: saveTipoRuta()
- Entradas: Todos los datos del tipo de ruta a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

80) Modificar un tipo de ruta existente

- Introducción: Modifica uno o más datos de un tipo de ruta existente.
- Nombre de la Función: saveTipoRuta()

- Entradas: Todos los datos del tipo de ruta a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del tipo de ruta.
- Salida: Éxito o fracaso en la modificación.

81) Eliminar un tipo de ruta

- Introducción: Elimina físicamente de la base de datos, el registro de un tipo de ruta, dado su código.
- Nombre de la Función: deleteTipoRuta()
- Entradas: Código del tipo de ruta.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del tipo de ruta que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xvi) Funciones del Módulo Destinos

82) Consultar todos los destinos

- Introducción: Despliega todos los destinos que ha registrado el sistema, en la Vista Web de la pantalla Destino.
- Nombre de la Función: getCiudades()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Ciudad, Nombre, Abreviatura, Region y Visible de todos los registros de la tabla CIUDAD. Los ordena por Id_Ciudad.
- Salida: Devuelve los registros obtenidos de la tabla CIUDAD.

83) Consultar un destino

- Introducción: Despliega todos los datos de un destino, dado su id, en el Formulario Web de la pantalla Destino.
- Nombre de la Función: getCiudadById()
- Entradas: Código del destino.
- Proceso: Trae de la base de datos, todos los datos de un destino, dado su código.

84) Ingresar un nuevo destino

- Introducción: Añade un nuevo destino a la base de datos.
- Nombre de la Función: saveCiudad()
- Entradas: Todos los datos del destino a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

85) Modificar un destino existente

- Introducción: Modifica uno o más datos de un destino existente.
- Nombre de la Función: saveCiudad()
- Entradas: Todos los datos del destino a ser modificados.

- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del destino.
- Salida: Éxito o fracaso en la modificación.

86) Eliminar un destino

- Introducción: Elimina físicamente de la base de datos, el registro de un destino, dado su código.
- Nombre de la Función: deleteCiudad()
- Entradas: Código de destino.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del destino que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xvii) Funciones del Módulo Regiones

87) Consultar todas las regiones

- Introducción: Despliega todas las regiones que ha registrado el sistema, en la Vista Web de la pantalla Región.
- Nombre de la Función: getRegiones()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Region y Nombre de todos los registros de la tabla REGION. Los ordena por Id_Region.
- Salida: Devuelve los registros obtenidos de la tabla REGION.

88) Consultar una región

- Introducción: Despliega todos los datos de una región, dado su id, en el Formulario Web de la pantalla Región.
- Nombre de la Función: getRegionById()
- Entradas: Código de la región.
- Proceso: Trae de la base de datos, todos los datos de una región, dado su código.

89) Ingresar una nueva región

- Introducción: Añade una nueva región a la base de datos.
- Nombre de la Función: saveRegion()
- Entradas: Todos los datos de la región a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

90) Modificar una región existente

- Introducción: Modifica uno o más datos de una región existente.
- Nombre de la Función: saveRegion()
- Entradas: Todos los datos de la región a ser modificados.

- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos de la región.
- Salida: Éxito o fracaso en la modificación.

91) Eliminar una región

- Introducción: Elimina físicamente de la base de datos, el registro de una región, dado su código.
- Nombre de la Función: deleteRegion()
- Entradas: Código de región.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro de la región que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xviii) Funciones del Módulo Andenes

92) Consultar todos los andenes

- Introducción: Despliega todos los andenes que ha registrado el sistema, en la Vista Web de la pantalla Andén.
- Nombre de la Función: getAndenes()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Anden, Numero, Destino y Area de todos los registros de la tabla ANDEN. Los ordena por Id_Anden.
- Salida: Devuelve los registros obtenidos de la tabla ANDEN.

93) Consultar un andén

- Introducción: Despliega todos los datos de un andén, dado su id, en el Formulario Web de la pantalla Andén.
- Nombre de la Función: getAndenById()
- Entradas: Código de andén.
- Proceso: Trae de la base de datos, todos los datos de un andén, dado su código.

94) Ingresar un nuevo andén

- Introducción: Añade un nuevo andén a la base de datos.
- Nombre de la Función: saveAnden()
- Entradas: Todos los datos del andén a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

95) Modificar un andén existente

- Introducción: Modifica uno o más datos de un andén existente.
- Nombre de la Función: saveAnden()
- Entradas: Todos los datos del andén a ser modificados.

- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del andén.
- Salida: Éxito o fracaso en la modificación.

96) Eliminar un andén

- Introducción: Elimina físicamente de la base de datos, el registro de un andén, dado su código.
- Nombre de la Función: deleteAnden()
- Entradas: Código de andén.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del andén que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xix) Funciones del Módulo Áreas

97) Consultar todas las áreas del terminal terrestre

- Introducción: Despliega todas las áreas que ha registrado el sistema, en la Vista Web de la pantalla Área.
- Nombre de la Función: getAreas()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Area y Nombre de todos los registros de la tabla AREA. Los ordena por Id_Area.
- Salida: Devuelve los registros obtenidos de la tabla AREA.

98) Consultar un área

- Introducción: Despliega todos los datos de un área, dado su id, en el Formulario Web de la pantalla Área.
- Nombre de la Función: getAreaById()
- Entradas: Código de área.
- Proceso: Trae de la base de datos, todos los datos de un área, dado su código.

99) Ingresar una nueva área

- Introducción: Añade una nueva área a la base de datos.
- Nombre de la Función: saveArea()
- Entradas: Todos los datos del área a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

100) Modificar un área existente

- Introducción: Modifica uno o más datos de un área existente.
- Nombre de la Función: saveArea()
- Entradas: Todos los datos del área a ser modificados.

- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del área.
- Salida: Éxito o fracaso en la modificación.

101) Eliminar un área

- Introducción: Elimina físicamente de la base de datos, el registro de un área, dado su código.
- Nombre de la Función: deleteArea()
- Entradas: Código de área.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del área que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xx) Funciones del Módulo Equipos

102) Consultar todos los equipos electrónicos

- Introducción: Despliega todos los equipos que ha registrado el sistema, en la Vista Web de la pantalla Equipo.
- Nombre de la Función: getEquipos()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Equipo, Nombre, Ubicacion, Area, IP y Tipo Equipo de todos los registros de la tabla EQUIPO. Los ordena por Id_Equipo.
- Salida: Devuelve los registros obtenidos de la tabla EQUIPO.

103) Consultar un equipo

- Introducción: Despliega todos los datos de un equipo, dado su id, en el Formulario Web de la pantalla Equipo.
- Nombre de la Función: getEquipoById()
- Entradas: Código de equipo.
- Proceso: Trae de la base de datos, todos los datos de un equipo, dado su código.

104) Ingresar un nuevo equipo

- Introducción: Añade un nuevo equipo a la base de datos.
- Nombre de la Función: saveEquipo()
- Entradas: Todos los datos del equipo a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

105) Modificar un equipo existente

- Introducción: Modifica uno o más datos de un equipo existente.
- Nombre de la Función: saveEquipo()
- Entradas: Todos los datos del equipo a ser modificados.

- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del equipo.
- Salida: Éxito o fracaso en la modificación.

106) Eliminar un equipo

- Introducción: Elimina físicamente de la base de datos, el registro de un equipo, dado su código.
- Nombre de la Función: deleteEquipo()
- Entradas: Código de equipo.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del equipo que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xxi) Funciones del Módulo Tipos de Equipos

107) Consultar todos los tipos de equipos

- Introducción: Despliega todos los tipos de equipos que ha registrado el sistema, en la Vista Web de la pantalla Tipo Equipo.
- Nombre de la Función: getTipoEquipos()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Tipo_Equipo y Nombre de todos los registros de la tabla TIPO_EQUIPO. Los ordena por Id_Tipo_Equipo.
- Salida: Devuelve los registros obtenidos de la tabla TIPO_EQUIPO.

108) Consultar un tipo de equipo

- Introducción: Despliega todos los datos de un tipo de equipo, dado su id, en el Formulario Web de la pantalla Tipo Equipo.
- Nombre de la Función: getTipoEquipoById()
- Entradas: Código de tipo equipo.
- Proceso: Trae de la base de datos, todos los datos de un tipo de equipo, dado su código.

109) Ingresar un nuevo tipo de equipo

- Introducción: Añade un nuevo tipo de equipo a la base de datos.
- Nombre de la Función: saveTipoEquipo()
- Entradas: Todos los datos del tipo de equipo a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

110) Modificar un tipo de equipo existente

- Introducción: Modifica uno o más datos de un tipo de equipo existente.
- Nombre de la Función: saveTipoEquipo()
- Entradas: Todos los datos del tipo de equipo a ser modificados.

- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del tipo de equipo.
- Salida: Éxito o fracaso en la modificación.

111) Eliminar un tipo de equipo

- Introducción: Elimina físicamente de la base de datos, el registro de un tipo de equipo, dado su código.
- Nombre de la Función: deleteTipoEquipo()
- Entradas: Código de tipo equipo.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del tipo de equipo que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xxii) Funciones del Módulo Conceptos de Cobro

112) Consultar todos los conceptos de cobro

- Introducción: Despliega todos los conceptos de cobro que ha registrado el sistema, en la Vista Web de la pantalla Concepto Cobro.
- Nombre de la Función: getConceptoCuentas()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Concepto_Cuenta, Valor, Grupo, Iva y Estado de todos los registros de la tabla CONCEPTO_CUENTA. Los ordena por Id_Concepto_Cuenta.
- Salida: Devuelve los registros obtenidos de la tabla CONCEPTO_CUENTA.

113) Consultar un concepto de cobro

- Introducción: Despliega todos los datos de un concepto de cobro, dado su id, en el Formulario Web de la pantalla Concepto Cobro.
- Nombre de la Función: getConceptoCuentaById()
- Entradas: Código de concepto.
- Proceso: Trae de la base de datos, todos los datos de un concepto de cobro, dado su código.

114) Ingresar un nuevo concepto de cobro

- Introducción: Añade un nuevo concepto de cobro a la base de datos.
- Nombre de la Función: saveConcepto()
- Entradas: Todos los datos del concepto a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

115) Modificar un concepto de cobro existente

- Introducción: Modifica uno o más datos de un concepto de cobro existente.
- Nombre de la Función: saveConcepto()

- Entradas: Todos los datos del concepto a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del concepto.
- Salida: Éxito o fracaso en la modificación.

116) Eliminar un concepto de cobro

- Introducción: Elimina físicamente de la base de datos, el registro de un concepto de cobro, dado su código.
- Nombre de la Función: deleteConcepto()
- Entradas: Código de concepto.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro de concepto que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xxiii) Funciones del Módulo Grupos de Cobro

117) Consultar todos los grupos o categorías de cobro

- Introducción: Despliega todos los grupos de cobro que ha registrado el sistema, en la Vista Web de la pantalla Grupo Cobro.
- Nombre de la Función: getGrupos()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Grupo, Nombre y Nombre_Partida de todos los registros de la tabla GRUPOS. Los ordena por Id_Grupo.
- Salida: Devuelve los registros obtenidos de la tabla GRUPOS.

118) Consultar un grupo de cobro

- Introducción: Despliega todos los datos de un grupo de cobro, dado su id, en el Formulario Web de la pantalla Grupo Cobro.
- Nombre de la Función: getGruposById()
- Entradas: Código de grupo de cobro.
- Proceso: Trae de la base de datos, todos los datos de un grupo de cobro, dado su código.

119) Ingresar un nuevo grupo de cobro

- Introducción: Añade un nuevo grupo de cobro a la base de datos.
- Nombre de la Función: saveGrupo()
- Entradas: Todos los datos del grupo de cobro a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

120) Modificar un grupo de cobro existente

- Introducción: Modifica uno o más datos de un grupo de cobro existente.
- Nombre de la Función: saveGrupo()

- Entradas: Todos los datos del grupo de cobro a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del grupo de cobro.
- Salida: Éxito o fracaso en la modificación.

121) Eliminar un grupo de cobro

- Introducción: Elimina físicamente de la base de datos, el registro de un grupo de cobro, dado su código.
- Nombre de la Función: deleteGrupo()
- Entradas: Código de grupo de cobro.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro de grupo de cobro que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

xxiv) Funciones del Módulo dispositivos RFID

122) Consultar todos los dispositivos RFID

- Introducción: Despliega todos los dispositivos rfid que ha registrado el sistema, en la Vista Web de la pantalla Dispositivo RFID.
- Nombre de la Función: getDispositivosRfid ()
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos Id_Dispositivo_RFID, Tag, Estado, Descripcion, Observacion y Unidad_transporte de todos los registros de la tabla DISPOSITIVO_RFID. Los ordena por Id_Dispositivo_RFID.
- Salida: Devuelve los registros obtenidos de la tabla DISPOSITIVO_RFID.

123) Consultar un dispositivo RFID

- Introducción: Despliega todos los datos de un dispositivo rfid, dado su id, en el Formulario Web de la pantalla Dispositivo RFID.
- Nombre de la Función: getDispositivoRfidById()
- Entradas: Código de dispositivo rfid.
- Proceso: Trae de la base de datos, todos los datos de un dispositivo rfid, dado su código.

124) Ingresar un nuevo dispositivo RFID

- Introducción: Añade un nuevo dispositivo rfid a la base de datos.
- Nombre de la Función: saveDispositivoRfid()
- Entradas: Todos los datos del dispositivo rfid a ser ingresados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

125) Modificar un dispositivo RFID existente

- Introducción: Modifica uno o más datos de un dispositivo rfid existente.

- Nombre de la Función: saveDispositivoRfid()
- Entradas: Todos los datos del dispositivo rfid a ser modificados.
- Proceso: Valida que los datos no estén vacíos también valida que la longitud de cada dato sea la correcta. Modifica los datos del dispositivo rfid.
- Salida: Éxito o fracaso en la modificación.

126) Eliminar un dispositivo RFID

- Introducción: Elimina físicamente de la base de datos, el registro de un dispositivo rfid, dado su código.
- Nombre de la Función: deleteDispositivoRfid()
- Entradas: Código de dispositivo rfid.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro de dispositivo rfid que coincida con el código proporcionado.
- Salida: Éxito o fracaso en la eliminación.

4.3.2. Requerimientos del portal web

4.3.2.1. Listado de requerimientos funcionales

A partir del ERS se resumen las funciones que permitirá realizar el portal web de la Unidad Desconcentrada de Terminales:

Ref. #	Función	Rol
RF1	El portal web debe permitir la gestión de los propietarios de las unidades de transporte (agregar, modificar, eliminar, buscar, listar).	Administrador
RF2	El portal web debe permitir la gestión de las operadoras de transporte autorizadas por la ANT (agregar, modificar, eliminar, buscar, listar).	Administrador
RF3	El portal web debe permitir la gestión de las unidades de transporte autorizadas por la ANT (agregar, modificar, eliminar, buscar, listar) y asignarles un propietario, una operadora y un área.	Administrador
RF4	El portal web debe permitir la gestión de los conductores de las unidades de transporte (agregar, modificar, eliminar, buscar, listar).	Administrador
RF5	El portal web debe permitir la gestión de los tipos de ruta de viaje (agregar, modificar, eliminar, buscar, listar)	Administrador
RF6	El portal web debe permitir la gestión de los tipos de frecuencias (agregar, modificar, eliminar, buscar, listar).	Administrador
RF7	El portal web debe permitir la gestión de los andenes de salida del terminal terrestre	Administrador

	(agregar, modificar, eliminar, buscar, listar) y asignarles el destino de salida y área correspondiente.	
RF8	El portal web debe permitir la gestión de las regiones del país (agregar, modificar, eliminar, buscar, listar).	Administrador
RF9	El portal web debe permitir la gestión de las ciudades las mismas que son destinos de viaje para la frecuencia tipo interprovincial (agregar, modificar, eliminar, buscar, listar) y asignarles su respectiva región.	Administrador
RF10	El portal web debe permitir la gestión de los grupos de conceptos de recaudación, que son las categorías de los conceptos por los que se realizarán los cobros en las cajas (agregar, modificar, eliminar, buscar, listar).	Administrador
RF11	El portal web debe permitir la gestión de los conceptos de recaudación (agregar, modificar, eliminar, buscar, listar) y asignarles un grupo de cobro.	Administrador
RF12	El portal web debe permitir la gestión de las áreas del terminal terrestre (agregar, modificar, eliminar, buscar, listar).	Administrador
RF13	El portal web debe permitir la gestión de los tipos de equipos (agregar, modificar, eliminar, buscar, listar).	Administrador
RF14	El portal web debe permitir la gestión de los equipos electrónicos que se encuentran en las instalaciones de la institución (agregar, modificar, eliminar, buscar, listar) y asignarles un área y un tipo de equipo.	Administrador
RF15	El portal web debe permitir la gestión de los perfiles de usuario (agregar, modificar, eliminar, buscar, listar).	Administrador
RF16	El portal web debe permitir la gestión de las opciones de perfil (agregar, modificar, eliminar, buscar, listar) y asignarles un perfil.	Administrador
RF17	El portal web debe permitir la gestión de los usuarios con acceso al sistema (agregar, modificar, eliminar, buscar, listar) y asignarle un perfil.	Administrador
RF18	El portal web debe permitir la gestión de los dispositivos RFID para la identificación de las Unidades de Transporte que ingresan al Terminal Terrestre (agregar, modificar, eliminar, buscar, listar) y asígnale una Unidad de Transporte.	Administrador
RF19	El portal web debe permitir la gestión de las frecuencias de viaje autorizadas por la ANT	Administrador

	(agregar, modificar, eliminar, buscar, listar) y asignarle un tipo de ruta, una cooperativa, origen, destino y tipo de frecuencia.	
RF21	El portal web debe permitir la gestión de las áreas de transferencia (agregar, modificar, eliminar, listar).	Administrador
RF24	El portal web debe permitir la publicación de información institucional como planificación estratégica, información de contacto, servicios, preguntas frecuentes, atención al público.	Administrador
RF25	El portal web debe permitir la gestión de noticias (agregar, modificar, eliminar, buscar, listar, publicar).	Administrador
RF26	El portal web debe permitir la gestión de comunicados (agregar, modificar, eliminar, buscar, listar, publicar).	Administrador
RF28	El portal web debe permitir la gestión de recursos como imágenes, videos, archivos (agregar, modificar, eliminar, buscar, listar).	Administrador
RF29	El portal web debe proveer un medio de comunicación con la UDT mediante un formulario de contacto esta información será enviada al correo del administrador del portal web para su posterior contestación.	Visitante
RF30	El portal web debe permitir la visualización de la información de todas las cooperativas que forman parte de la UDT.	Visitante
RF31	El portal web debe permitir la consulta de horarios, tarifas, teléfono de cooperativa, previa selección de destino y cooperativa de transporte.	Visitante
RF32	El portal web debe permitir la visualización de la información correspondiente a las próximas salidas de las unidades de transporte a partir de la hora actual.	Visitante
RF33	El portal web debe permitir la visualización en un mapa la ubicación exacta de las instalaciones de la UDT.	Visitante
RF34	El portal web debe permitir la visualización de la información correspondiente a las áreas de transferencia esto es: información de contacto, destinos y operadoras.	Visitante

Tabla 2 *Requerimientos funcionales*

4.3.2.2. Listado de requerimientos no funcionales

Ref. #	Función	Rol
RNF1	El portal web debe de poseer un diseño responsive.	
RNF2	El portal web debe de proporcionar mensajes de error que sean informativos y orientados al usuario final.	

Tabla 3 Requerimientos no funcionales

Fue primordial representar las tareas que realizarían los actores que intervienen en el portal, en base a la información obtenida, mediante diagramas de casos de uso como se muestra a continuación.

4.3.2.3. Diagrama de casos de uso

Administración módulo transportistas

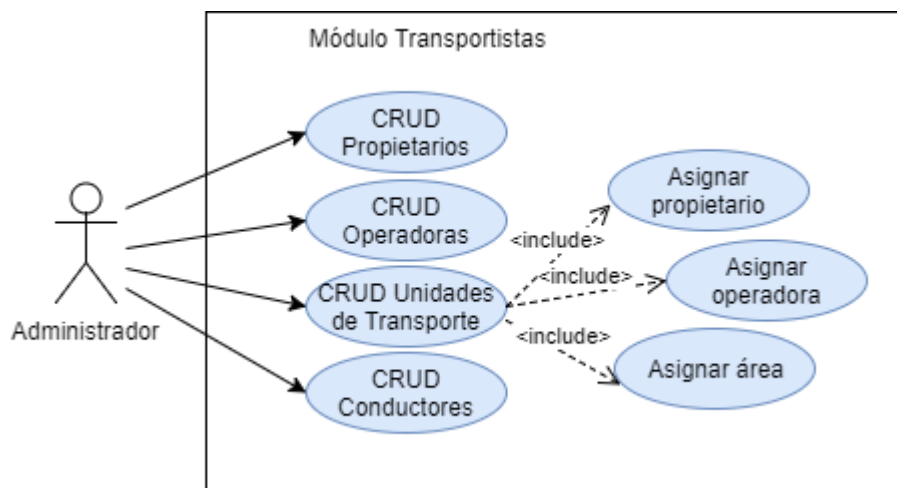


Figura 12 Diagrama de caso de uso (módulo Transportistas)

Administración módulo frecuencias

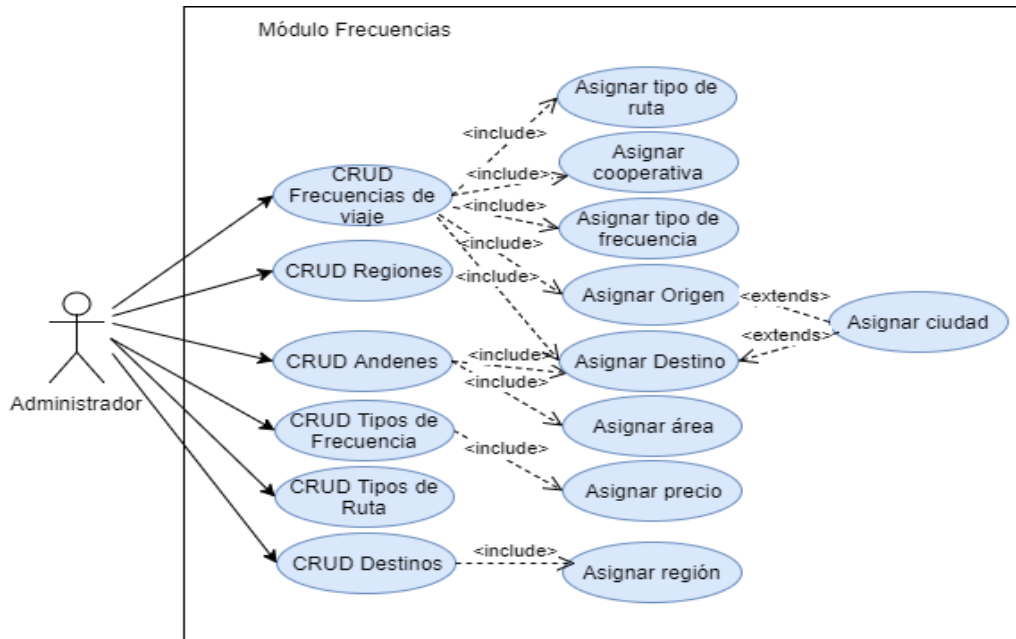


Figura 13 Diagrama de caso de uso (módulo Frecuencias)

Administración módulo ingresos

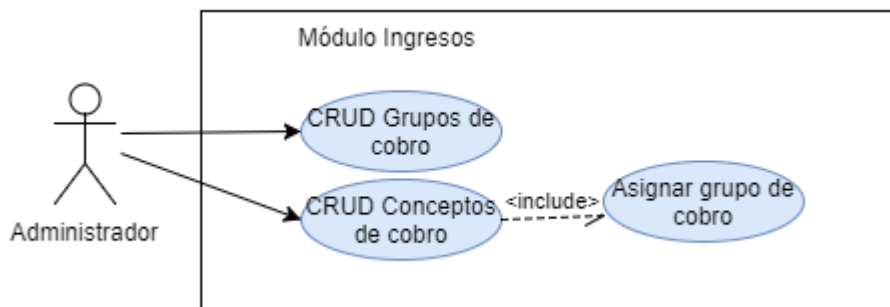


Figura 14 Diagrama de caso de uso (módulo Ingresos)

Administración módulo áreas y equipos

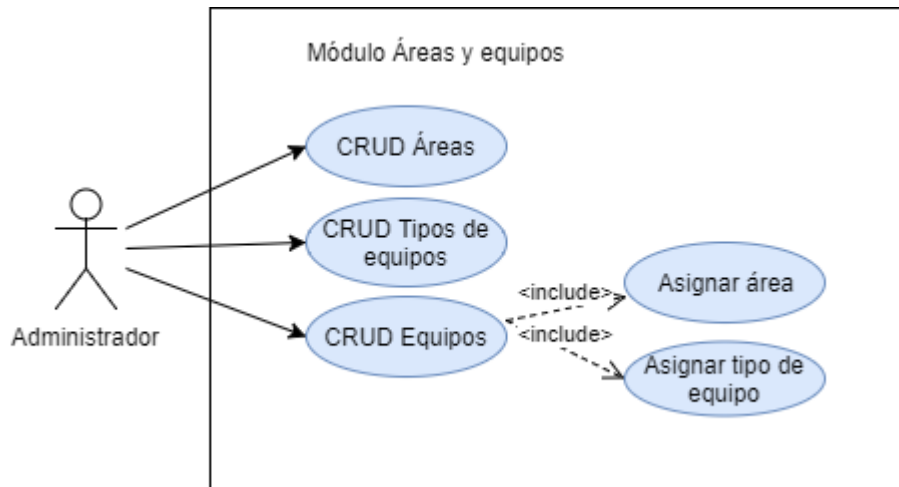


Figura 15 Diagrama de caso de uso (módulo áreas y equipos)

Administración módulo usuarios y perfiles

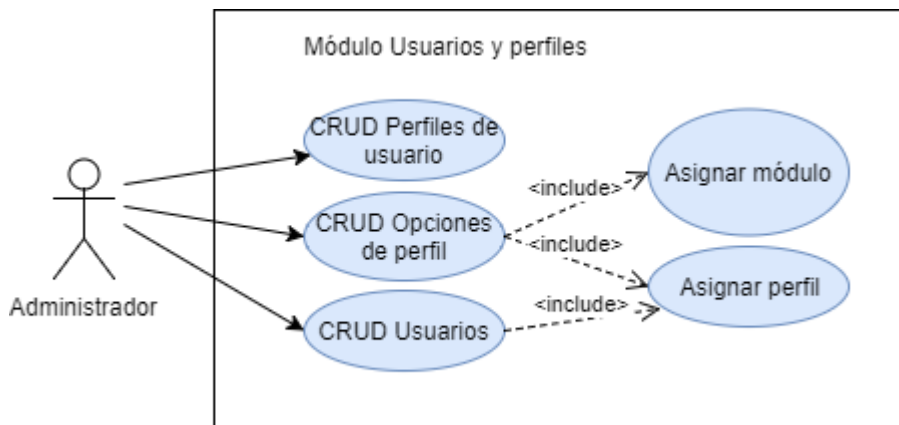


Figura 16 Diagrama de caso de uso (módulo usuarios y perfiles)

Administración módulo dispositivos RFID

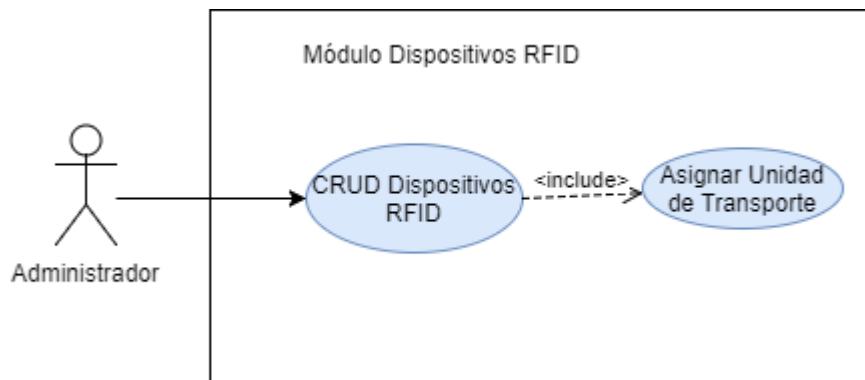


Figura 17 Diagrama de caso de uso (módulo dispositivos RFID)

Administración módulo institución

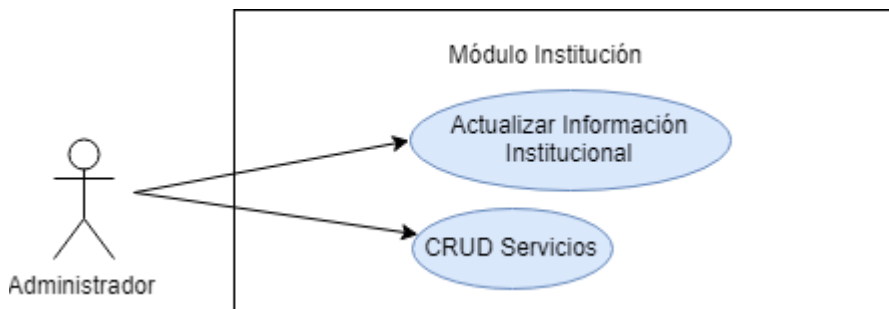


Figura 18 Diagrama de caso de uso (módulo institución)

Administración módulo noticias, comunicados y recursos

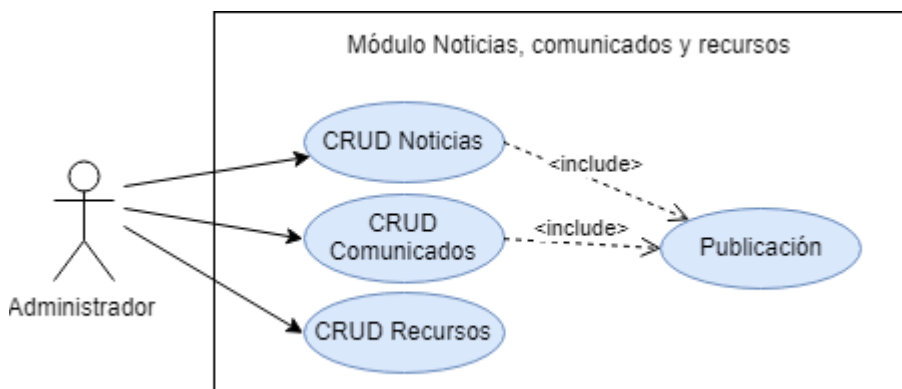


Figura 19 Diagrama de caso de uso (módulo noticias, comunicados y recursos)

Administración módulo áreas de transferencia

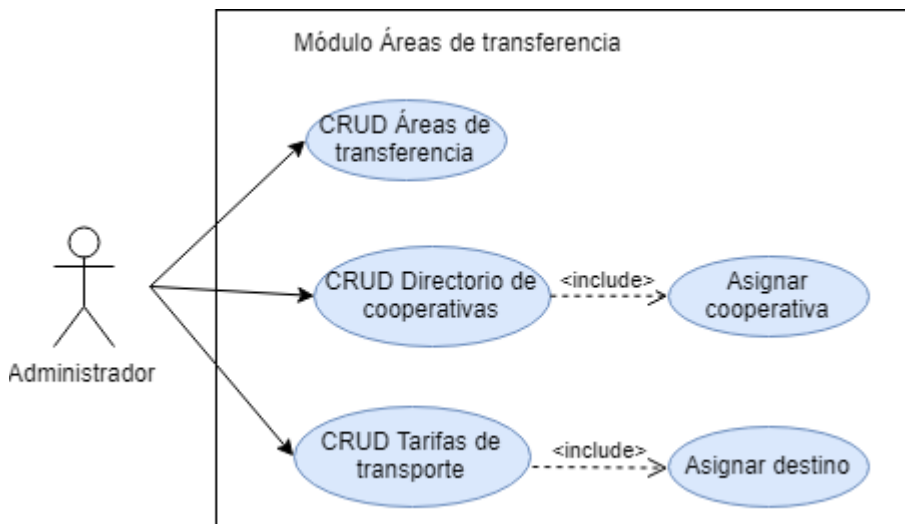


Figura 20 Diagrama de caso de uso (módulo áreas de transferencia)

Administración módulo preguntas frecuentes y links de interés

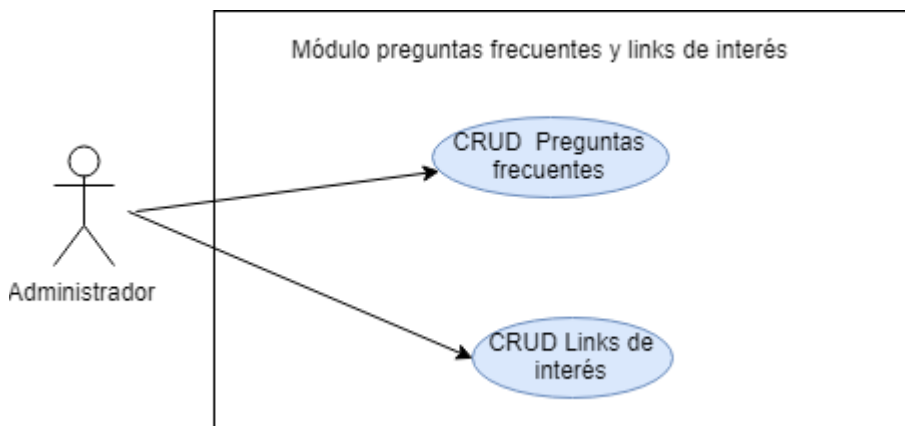


Figura 21 Diagrama de caso de uso (módulo preguntas frecuentes y links de interés)

Como se puede observar en los diagramas anteriores los usuarios administradores del portal pueden realizar varias tareas, y los visitantes podrán visualizar el resultado de la tarea realizada por los administradores.

Cabe entonces mencionar que en este portal web existirán 2 tipos de usuarios: administradores y visitantes.

Diagrama General del Usuario



Figura 22 Diagrama de caso de uso (diagrama general)

En este diagrama el usuario o visitante podrá consultar los servicios que están a su alcance, mas no podrá realizar cualquier tarea administrativa.

4.4. MIDAS/HT (Hipertexto)

En la segunda etapa del método midas (MIDAS/HT) denominada desarrollo de hipertexto (HT); se elaboraron las páginas web estáticas en HTML. Este paso es preciso para que el sistema logre accesos ágiles para la navegación en la aplicación, mejor ilustración sobre el manejo.

4.4.1. Actividad de Análisis

En la actividad de análisis se deben obtener los modelos conceptuales del sistema.

En primer lugar, se realiza el diseño conceptual de datos (véase **Figura 23**) para lo que se propone utilizar el diagrama de clases de UML. Partiendo de este diseño conceptual de datos se realiza el diseño conceptual del hipertexto. El hipertexto representa la forma en que la información es agrupada y enlazada para navegar a través de ella (véase **Figura 24**, **Figura 25**, **Figura 26**).

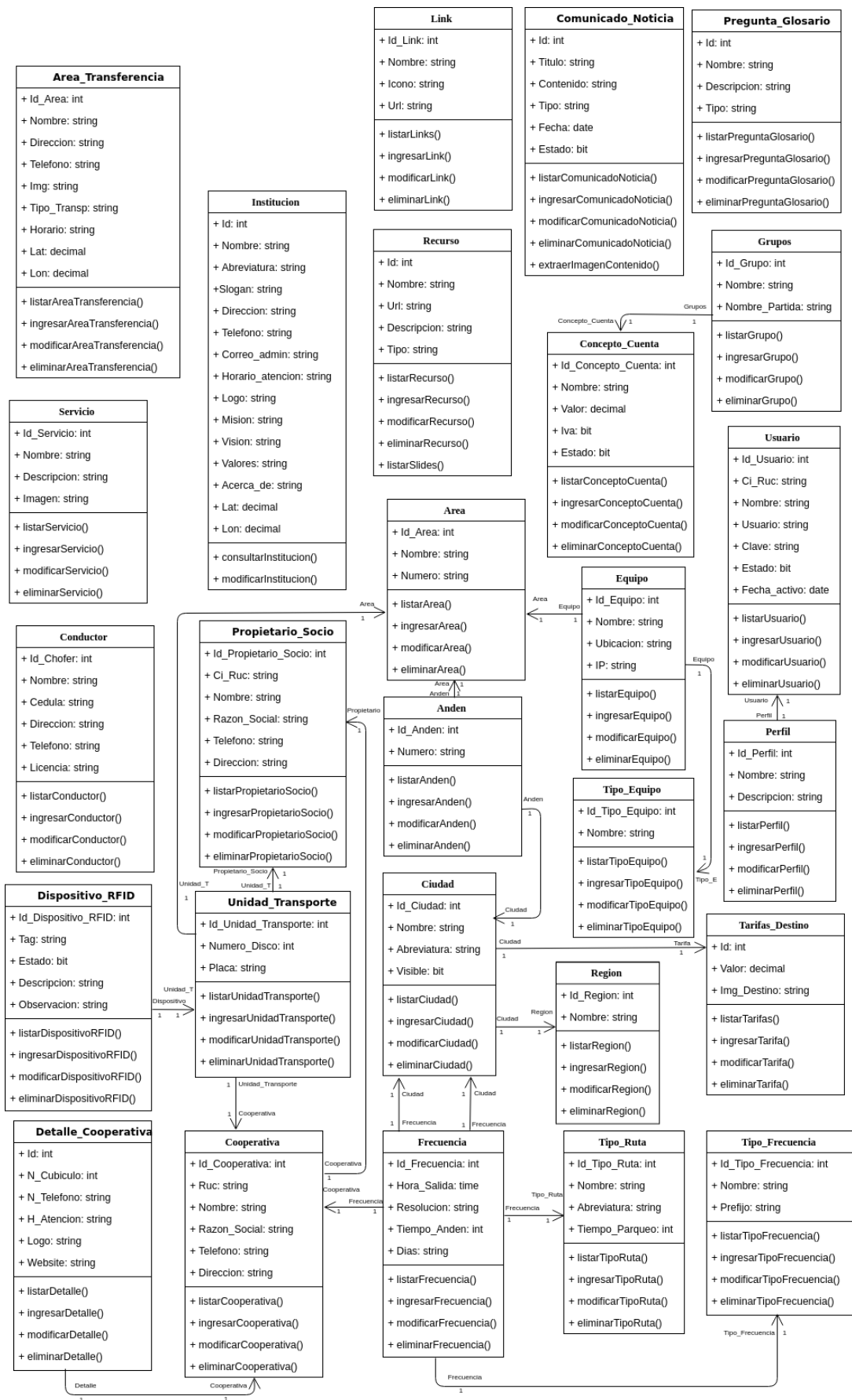


Figura 23 Modelo Conceptual de Datos

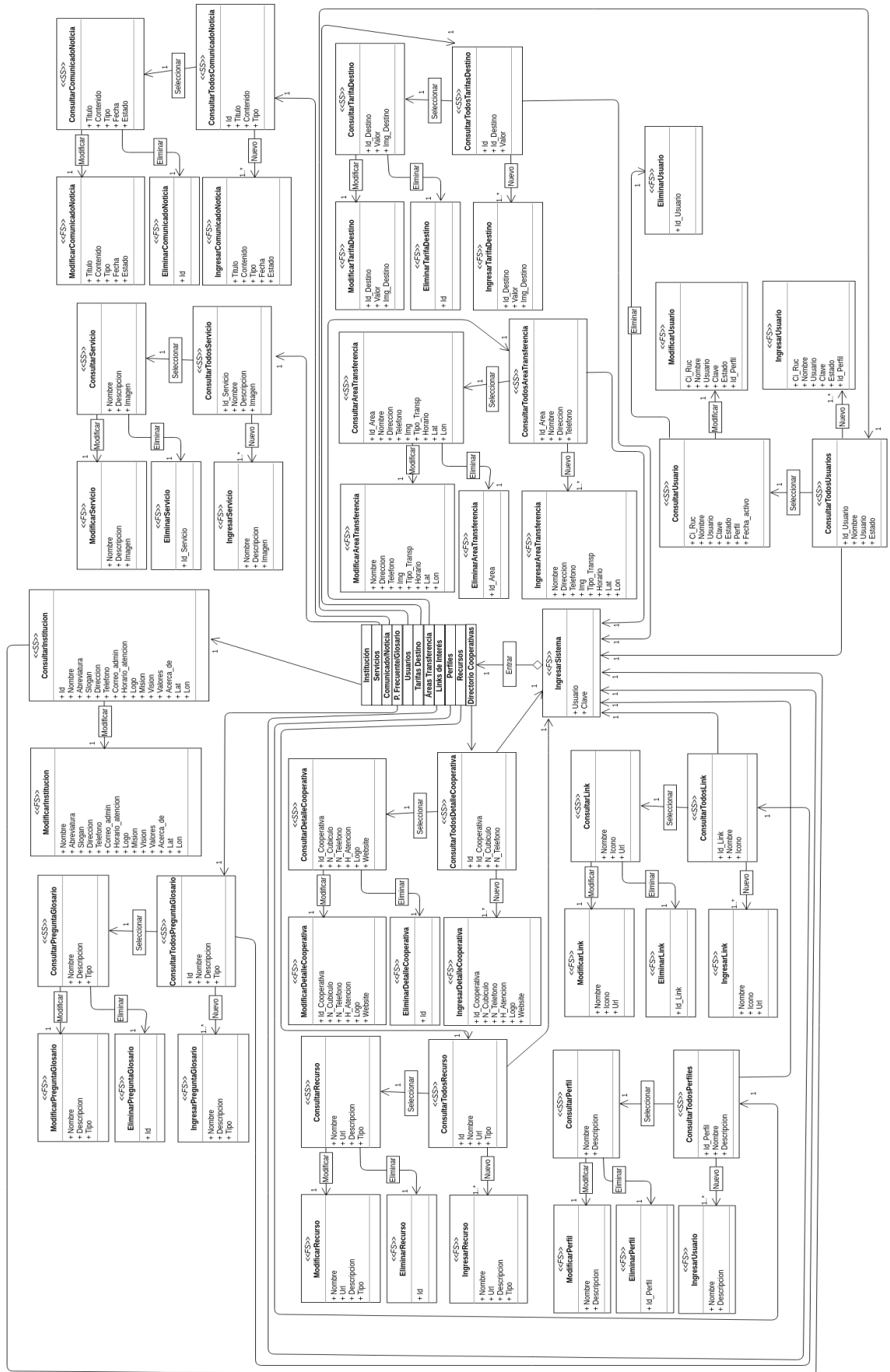


Figura 24 Modelo de Navegación (Administración 1)

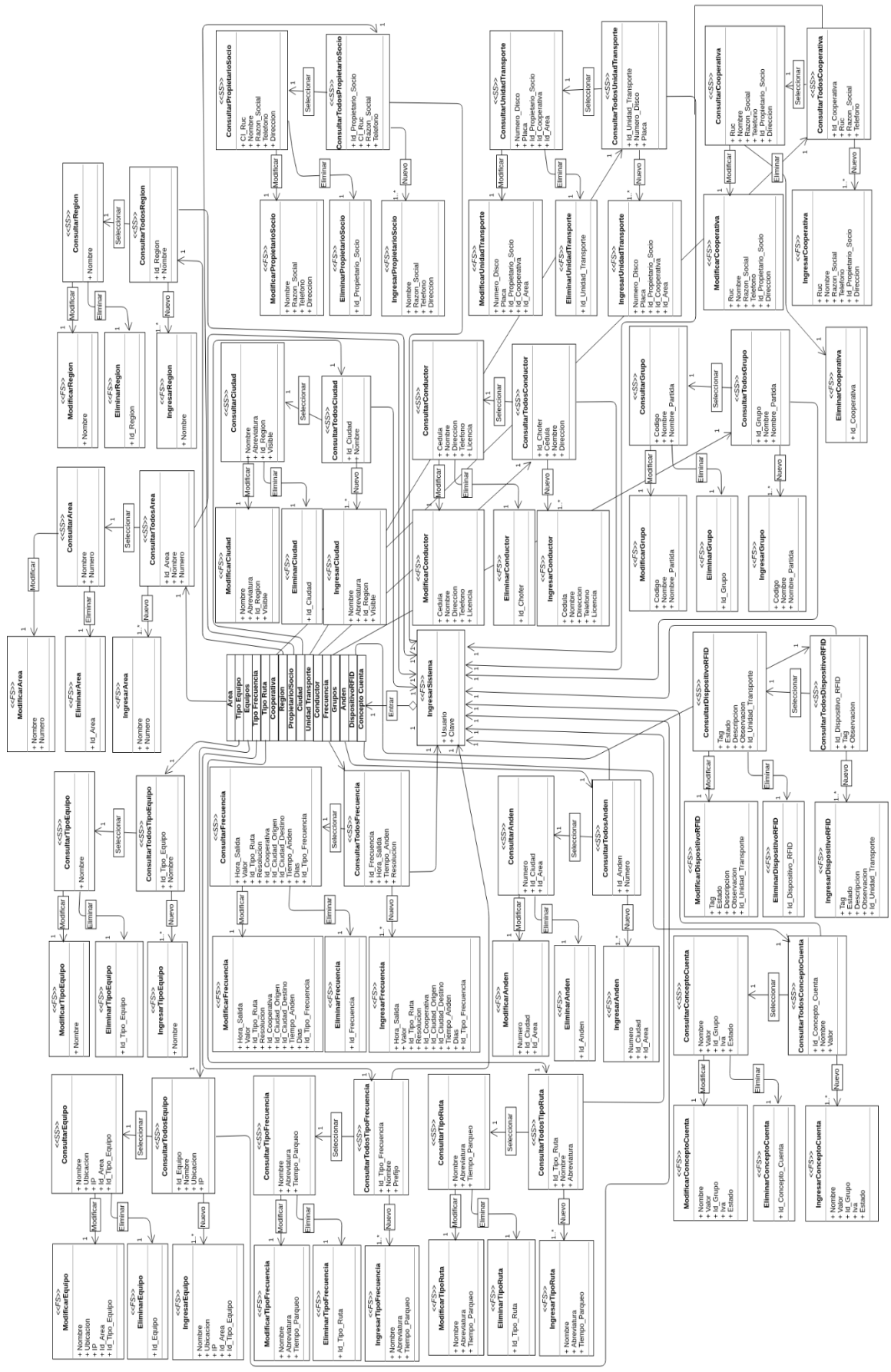


Figura 25 Modelo de Navegación (Administración 2)

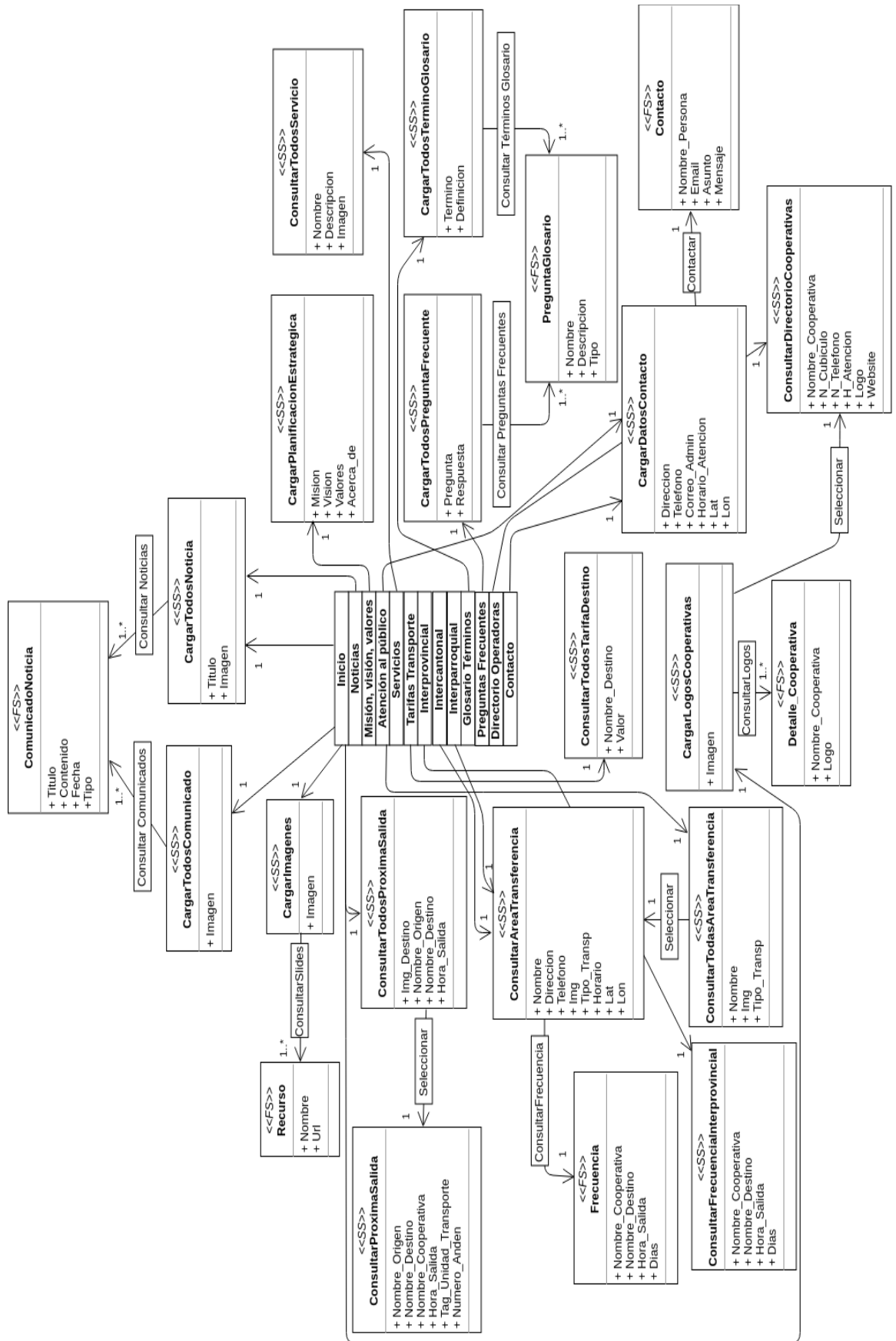


Figura 26 Modelo de Navegación parte pública

4.4.2. Actividades de Diseño e Implementación

Una vez realizado el diseño conceptual y obtenidos los modelos conceptuales de datos e hipertexto, se pasa a las actividades de diseño e implementación del hipertexto. Para ello se propone el diseño de wireframes utilizando una herramienta como draw.io y la implementación de prototipos utilizando alguna herramienta de diseño gráfico (DreamWeaver, Front Page, XMLSpy, etc.). De este modo, a la vez que se realiza el diseño de la interfaz de usuario (IU), se genera la primera versión de las páginas Web, estáticas y en HTML o XML. Una vez realizadas las pruebas correspondientes, estas páginas podrán estar disponibles en la Web en un tiempo prudencial, sirviendo además de prototipo que permita validar con el usuario los requisitos iniciales de la aplicación web. En función de esta validación con el usuario podrán modificarse, en la siguiente etapa, tanto el modelo conceptual de datos, como el del hipertexto.

4.4.2.1. Actividad de Diseño

Creación de Wireframes

Un wireframe es la primera representación de un diseño, tienen como objetivo mostrar el tipo de contenido que conforma el diseño y mostrar dónde se ubica ese contenido. Estos son una representación fiel del diseño final pero aún no tiene detalles de ciertos elementos.

Wireframe de la página de inicio

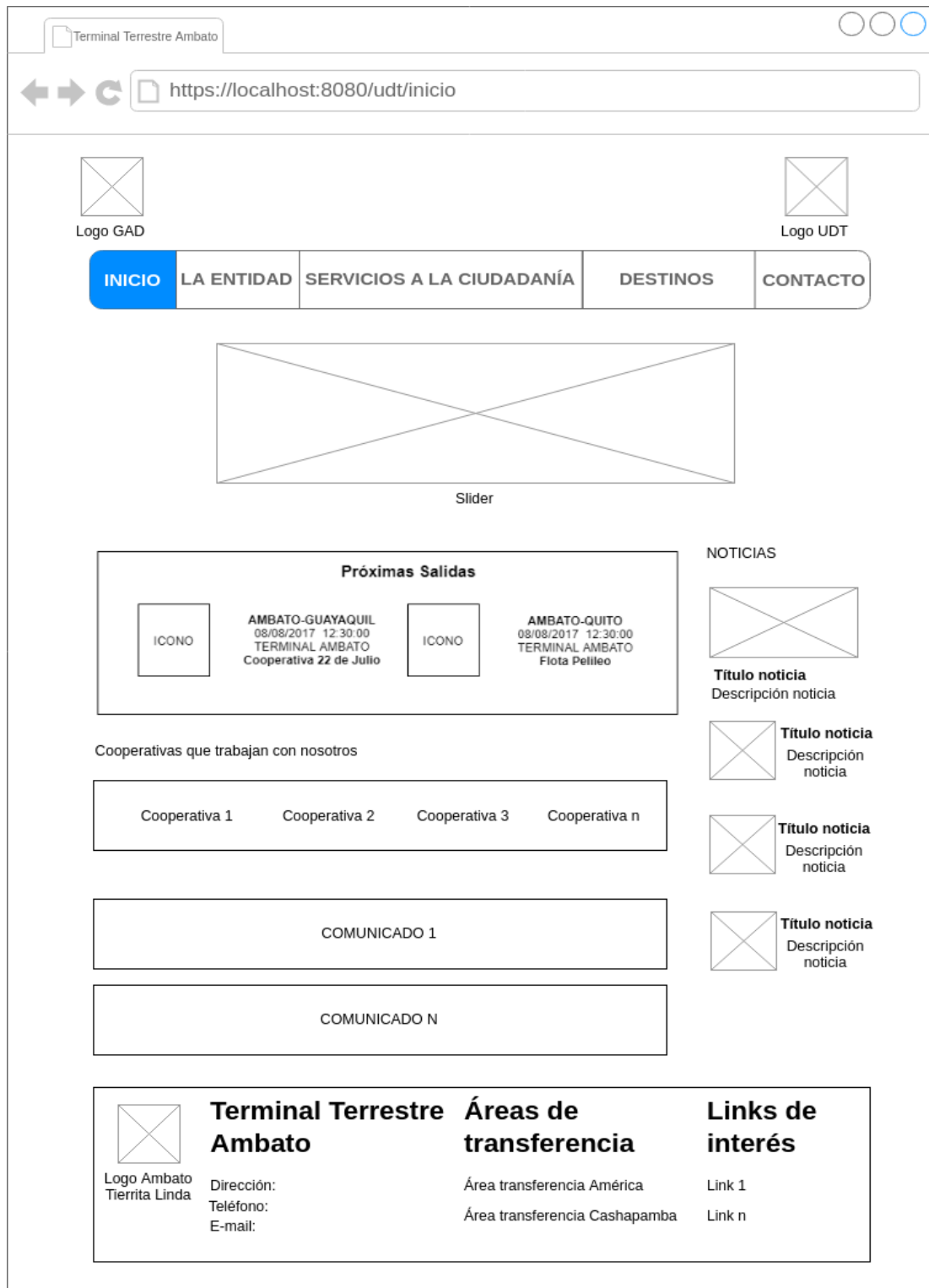


Figura 27 Wireframe página de inicio del portal web

Wireframe del panel de administración



Figura 28 Wireframe panel de administración

Wireframe usuario

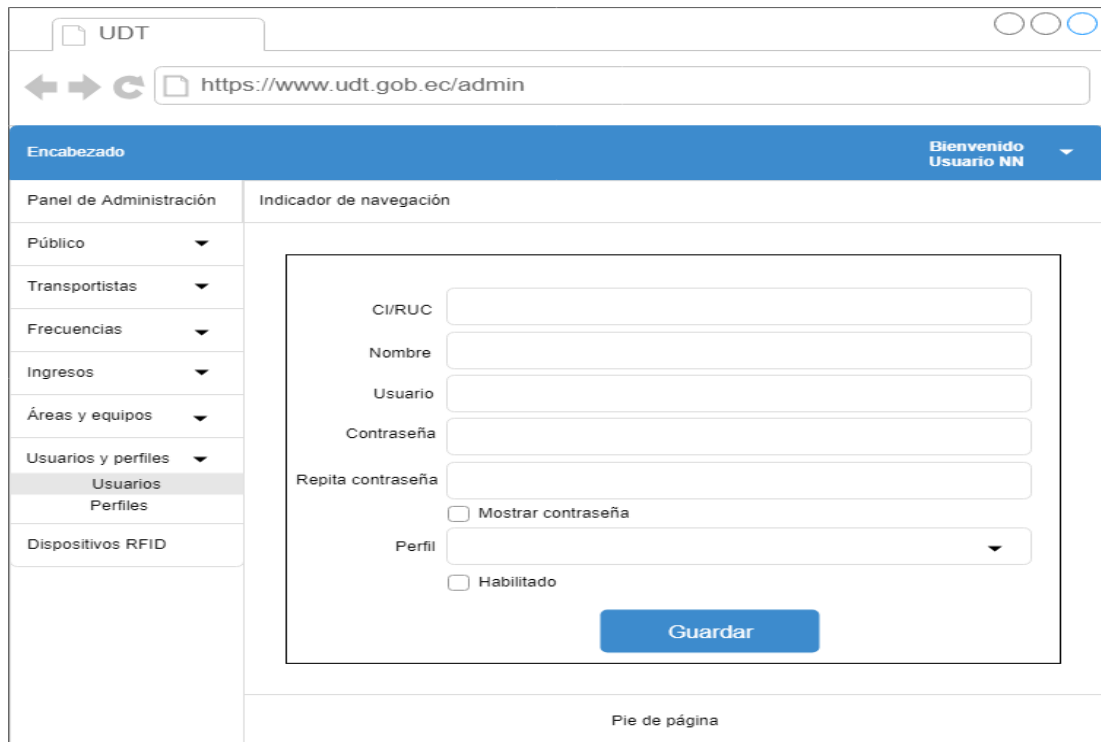


Figura 29 Wireframe usuario

Wireframe unidad de transporte

UDT

https://www.udt.gob.ec/admin

Encabezado Bienvenido Usuario NN

Panel de Administración

Indicador de navegación

Público

Transportistas

Propietarios

Operadoras

Unidades transporte

Conductores

Frecuencias

Ingresos

Áreas y equipos

Usuarios y perfiles

Dispositivos RFID

Número disco

Placa

Propietario

Cooperativa

Área

Guardar

Pie de página

Figura 30 Wireframe unidad de transporte

Wireframe institución

UDT

https://www.udt.gob.ec/admin

Encabezado Bienvenido Usuario NN

Panel de Administración

Indicador de navegación

Público

Institución

Servicios

Noticias y comunicados

Preguntas frecuentes

Directorio de cooperativas

Áreas de transferencia

Tarifas de transporte

Recursos

Links de interés

Transportistas

Frecuencias

Ingresos

Áreas y equipos

Usuarios y perfiles

Dispositivos RFID

Básico De contacto y ubicación Estratégico

Teléfono

Correo

Dirección

Ubicación

Guardar

Pie de página

Figura 31 Wireframe institución

Wireframe dispositivo RFID

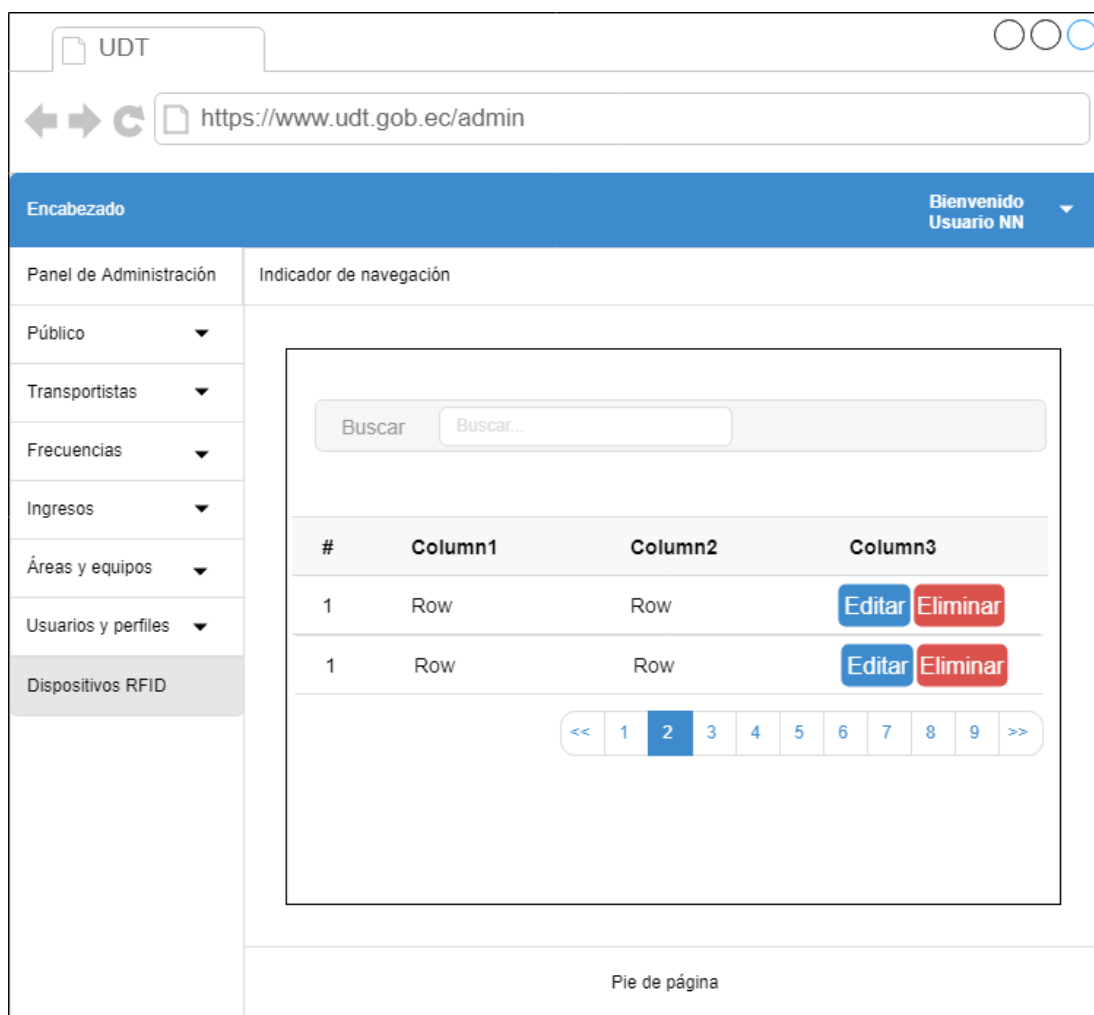


Figura 32 Wireframe dispositivos RFID

4.4.2.2. Actividad de Implementación

Implementación de una plantilla de Bootstrap

Luego de elaborar el wireframe se procedió a implementar una plantilla Bootstrap, la cual se toma como base para desarrollar los prototipos. Una plantilla de Bootstrap trae consigo componentes ya desarrollados y listos para utilizarse. Estos componentes son totalmente personalizables y permite diseñar interfaces simples, limpias e intuitivas. Para implementar una plantilla se procedió a realizar lo siguiente:

- ✓ Descargar una plantilla de Bootstrap, esta plantilla nos proporciona archivos html, hojas de estilo (css) y javascript los cuales podemos modificar a gusto para que se adapte a nuestros requerimientos.

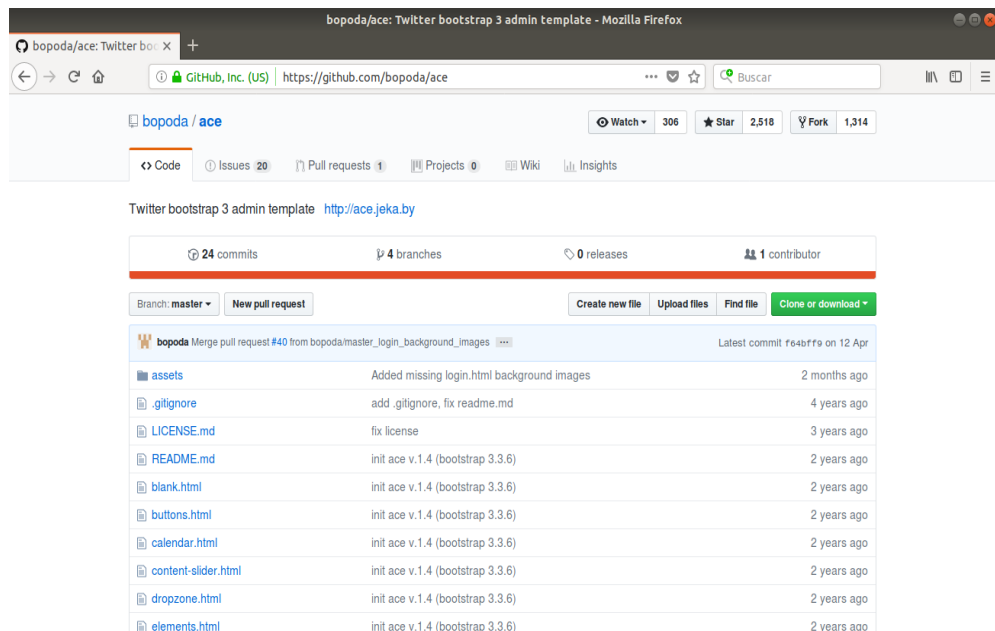


Figura 33 Plantilla de Bootstrap

- ✓ Probar el funcionamiento de la plantilla

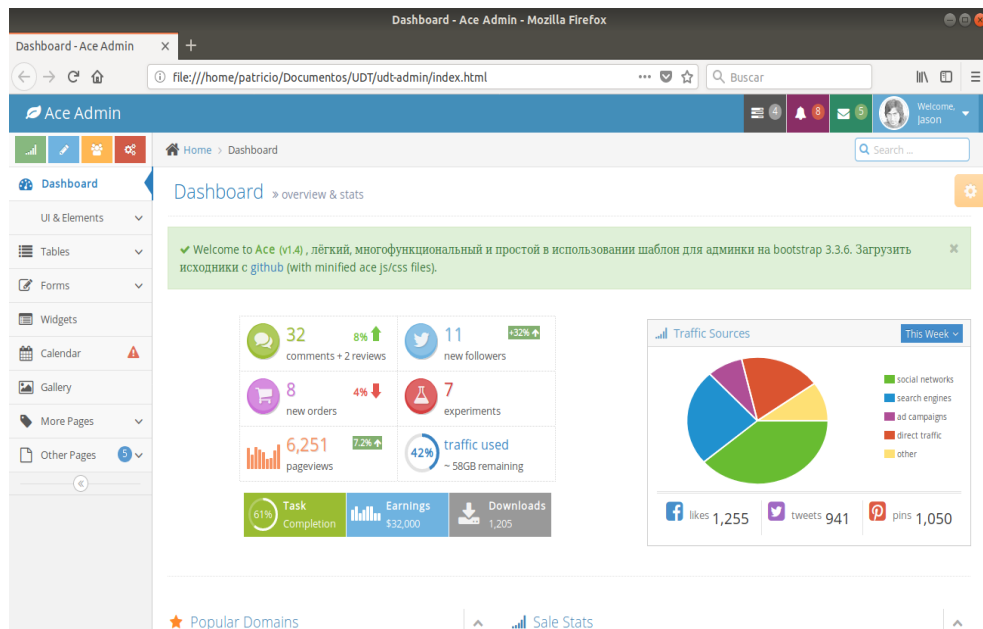


Figura 34 Verificación del funcionamiento de la plantilla Bootstrap

Creación de prototipos del portal web

Los prototipos son una representación de la versión final del diseño web que, además, simula la interactividad. Aunque es un prototipo, el diseño debe ser muy cercano a la versión final.

En base a los wireframes y la plantilla de Bootstrap se procedió a desarrollar los prototipos del portal web.

Prototipo página de inicio del portal web

La página de inicio para los visitantes del portal consta de lo siguiente:

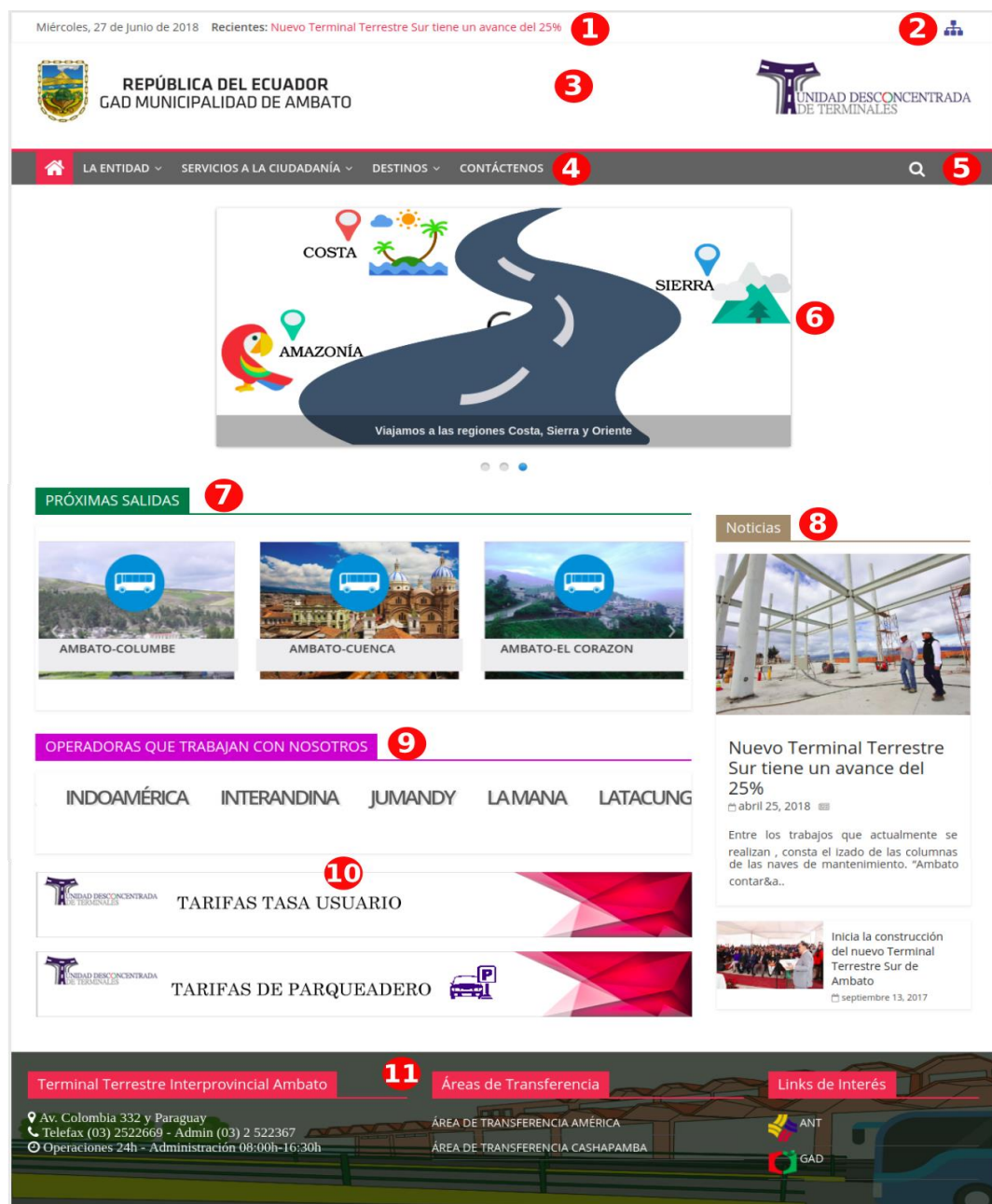


Figura 35 Prototipo página de inicio del portal web

1. Fecha actual del servidor web y acceso directo a las noticias más recientes.
2. Mapa del sitio que muestra la estructura completa de navegación del portal web.
3. Sección logos GAD y Terminal Terrestre.
4. Menú principal que contiene las diferentes secciones del portal web.
5. Caja de búsqueda para encontrar información por palabras clave.
6. Sección de slider y/o publicidad.
7. Parte que muestra los destinos, operadoras que salen del terminal a partir de la hora actual.
8. Esta sección contiene las noticias más recientes.
9. Sección que muestra el nombre o logo de las diferentes operadoras que trabajan en el Terminal Terrestre Ambato.
10. Parte del portal que muestra los comunicados.
11. Pie de página donde se encuentra información adicional.

Prototipo panel de administración

Para la parte de administración del portal se ha creado pantallas fáciles en su utilización, así como prácticas para la manipulación de la información.

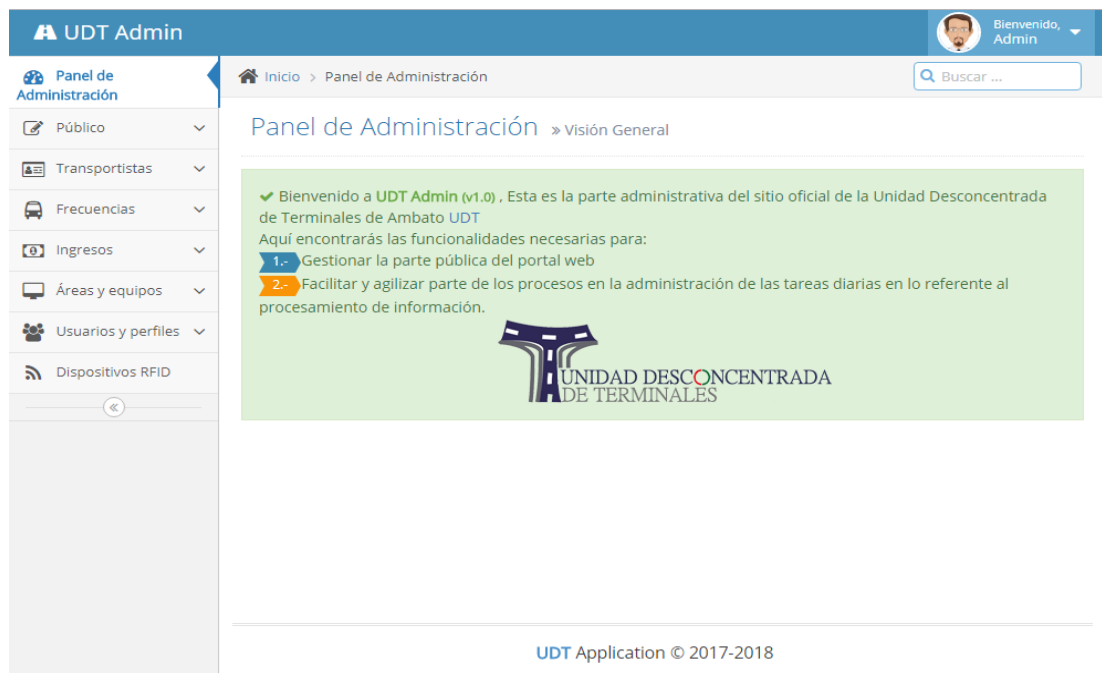
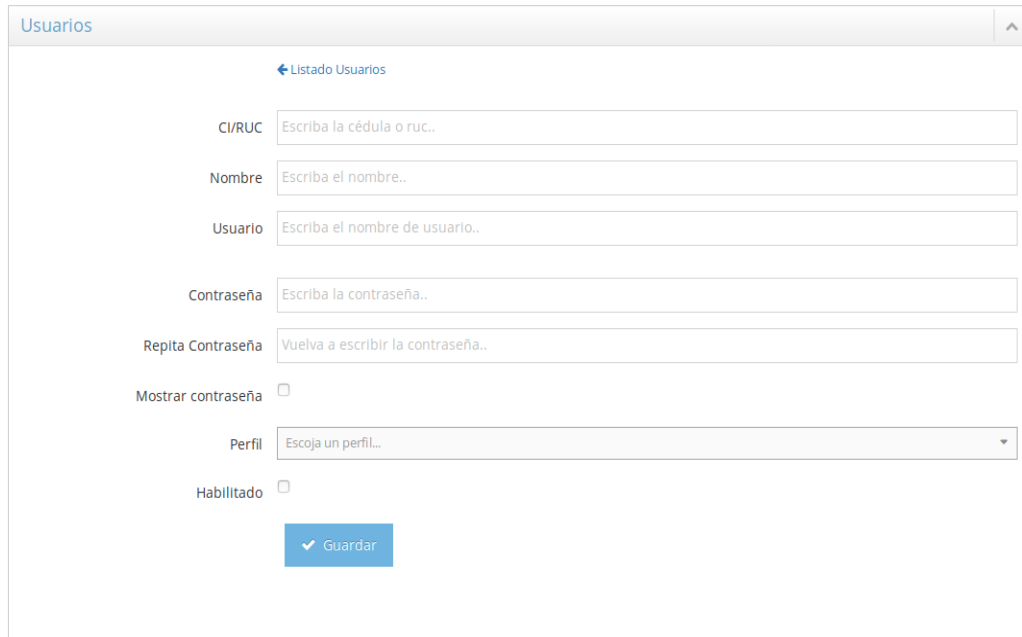


Figura 36 Prototipo panel de administración del portal web

Prototipo usuario



Usuarios

[← Listado Usuarios](#)

CI/RUC

Nombre

Usuario

Contraseña

Repita Contraseña

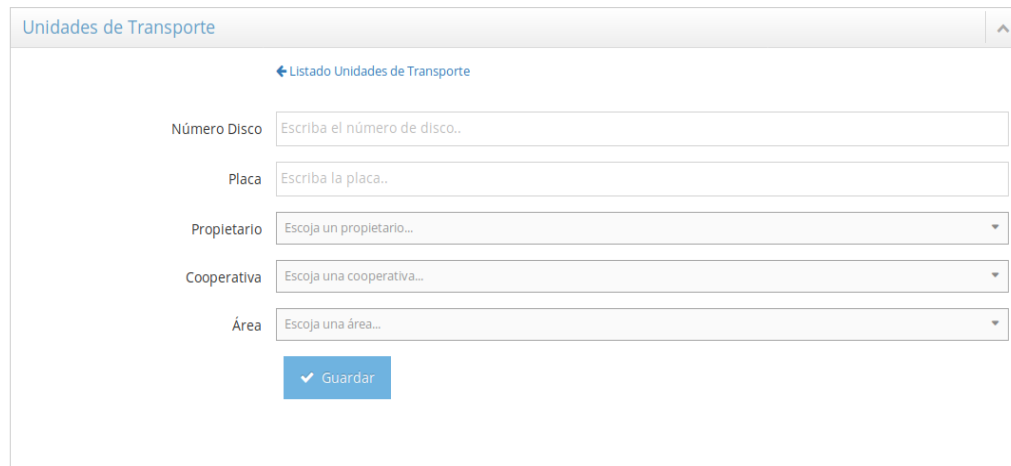
Mostrar contraseña

Perfil

Habilitado

Figura 37 Prototipo formulario registro de usuario

Prototipo Unidad de Transporte



Unidades de Transporte

[← Listado Unidades de Transporte](#)

Número Disco

Placa

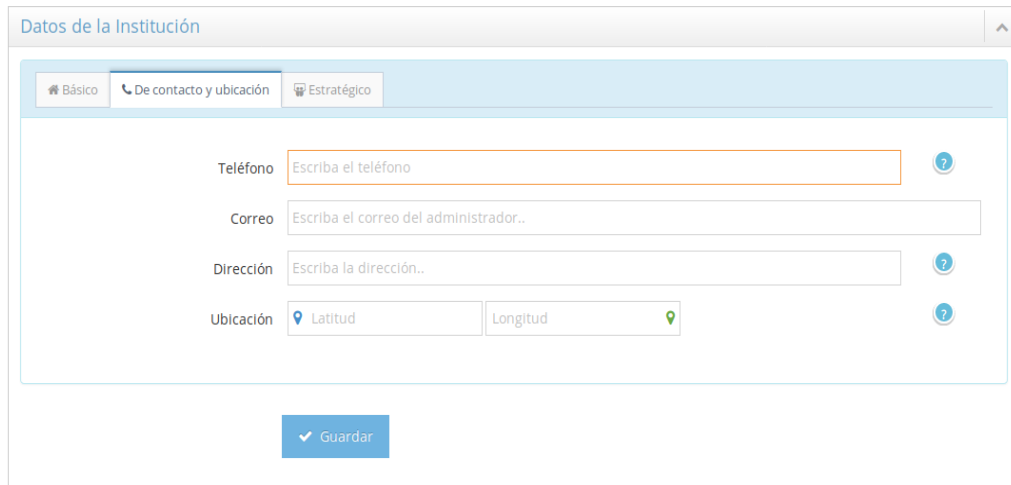
Propietario

Cooperativa

Área

Figura 38 Prototipo formulario registro unidad de transporte

Prototipo Institución



Datos de la Institución

Básico De contacto y ubicación Estratégico

Teléfono ?

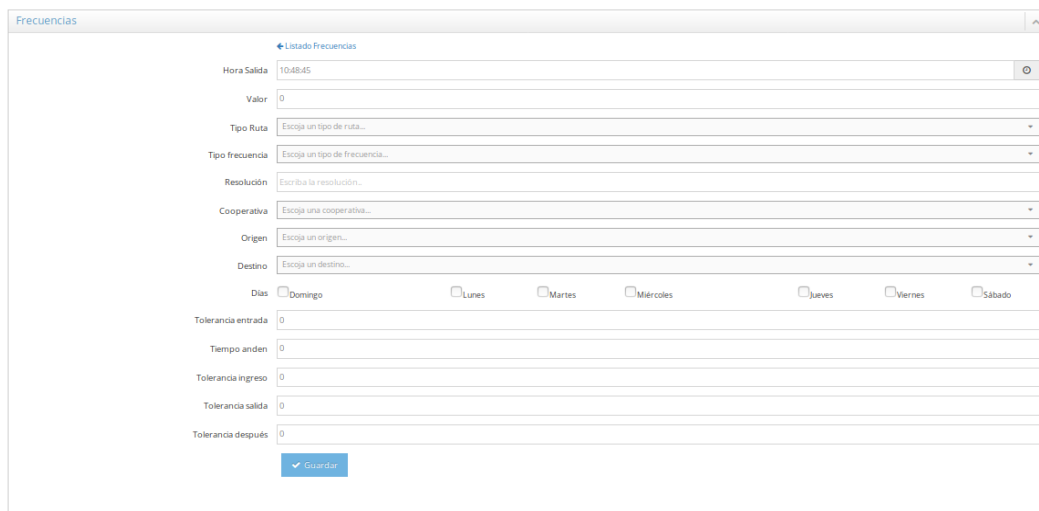
Correo

Dirección ?

Ubicación ?

Figura 39 Prototipo formulario registro datos de la institución

Prototipo Frecuencias



Frecuencias

← Listado Frecuencias

Hora Salida ○

Valor

Tipo Ruta

Tipo frecuencia

Resolución

Cooperativa

Origen

Destino

Días Domingo Lunes Martes Miércoles Jueves Viernes Sábado

Tolerancia entrada

Tiempo anden

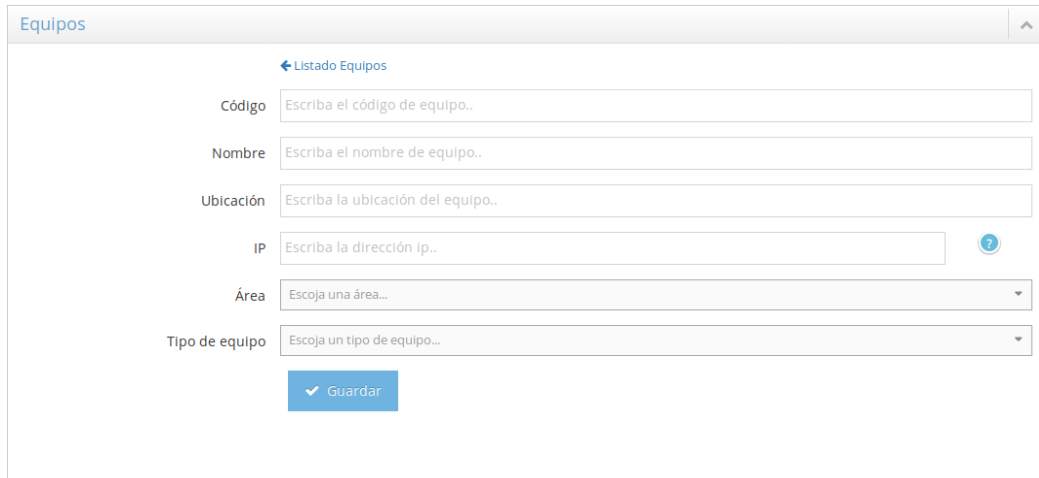
Tolerancia ingreso

Tolerancia salida

Tolerancia después

Figura 40 Prototipo formulario registro frecuencias

Prototipo Equipos




Equipos

[← Listado Equipos](#)

Código

Nombre

Ubicación

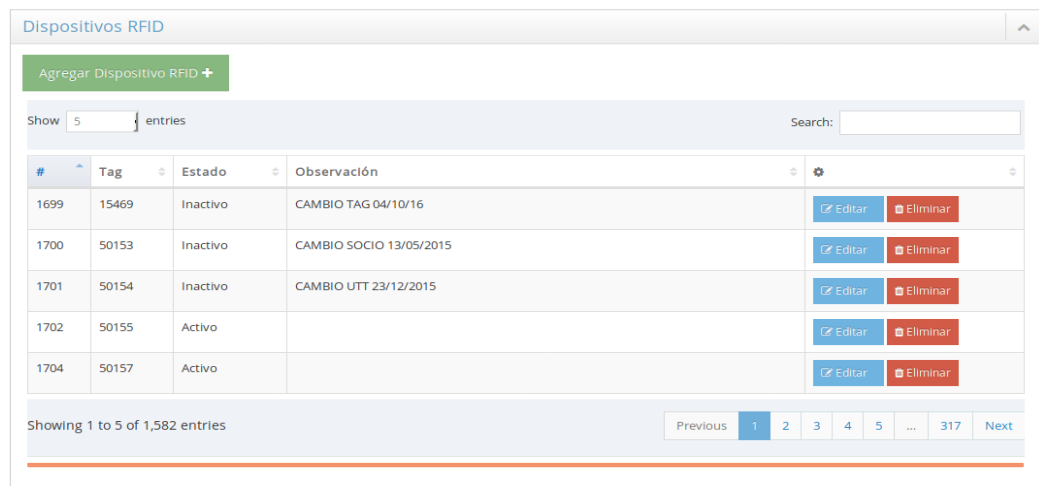
IP 

Área

Tipo de equipo

Figura 41 Prototipo formulario registro de equipos

Prototipo Dispositivos RFID



Dispositivos RFID

Show entries Search:

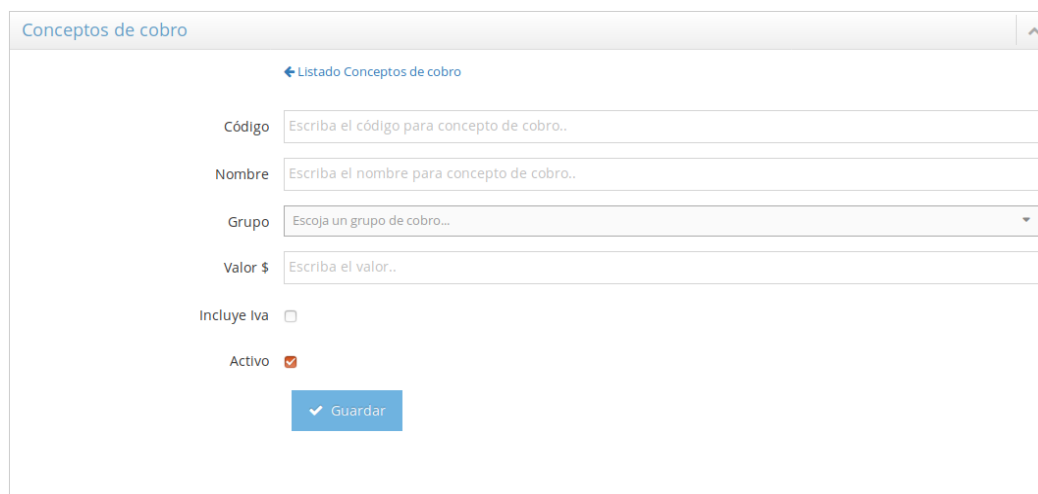
#	Tag	Estado	Observación	
1699	15469	Inactivo	CAMBIO TAG 04/10/16	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
1700	50153	Inactivo	CAMBIO SOCIO 13/05/2015	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
1701	50154	Inactivo	CAMBIO UTT 23/12/2015	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
1702	50155	Activo		<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
1704	50157	Activo		<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Showing 1 to 5 of 1,582 entries

Previous ... Next

Figura 42 Prototipo listado dispositivos RFID

Prototipo Conceptos de cobro



Conceptos de cobro

← Listado Conceptos de cobro

Código

Nombre

Grupo

Valor \$

Incluye Iva

Activo

Figura 43 Prototipo formulario registro concepto de cobro

4.5. MIDAS/DB (Base de Datos)

En esta fase teniendo en cuenta que la Unidad Desconcentrada de Terminales ha proporcionado 3 bases de datos diferentes las mismas que se detallan a continuación:

- ✓ **SICTERT.** Base de datos que almacena la información de las operaciones que a diario se llevan a cabo dentro de la institución (véase **Figura 44**).
- ✓ **sictert_lite_america.** Base de datos que almacena la información del área de transferencia América (véase **Figura 45**).
- ✓ **sictert_lite_cashapamba.** Base de datos que almacena la información del área de transferencia Cashapamba (véase **Figura 45**).

Con estas consideraciones solo se implementó las entidades necesarias y faltantes del portal web, dichas entidades formarán parte de la primera base de datos mencionada. Para el desarrollo de las entidades se utilizaron tres fases: Análisis, diseño e implementación.

La fase de análisis empezó con la captura de requisitos de la fase MIDAS/SD, se revisaron los requisitos iniciales conjuntamente con el modelo conceptual de datos, para elaborar el modelo entidad-relación (véase **Figura 46**).

BASE DE DATOS UNIDAD DESCONCENTRADA DE TERMINALES
(Proporcionado por la institución)

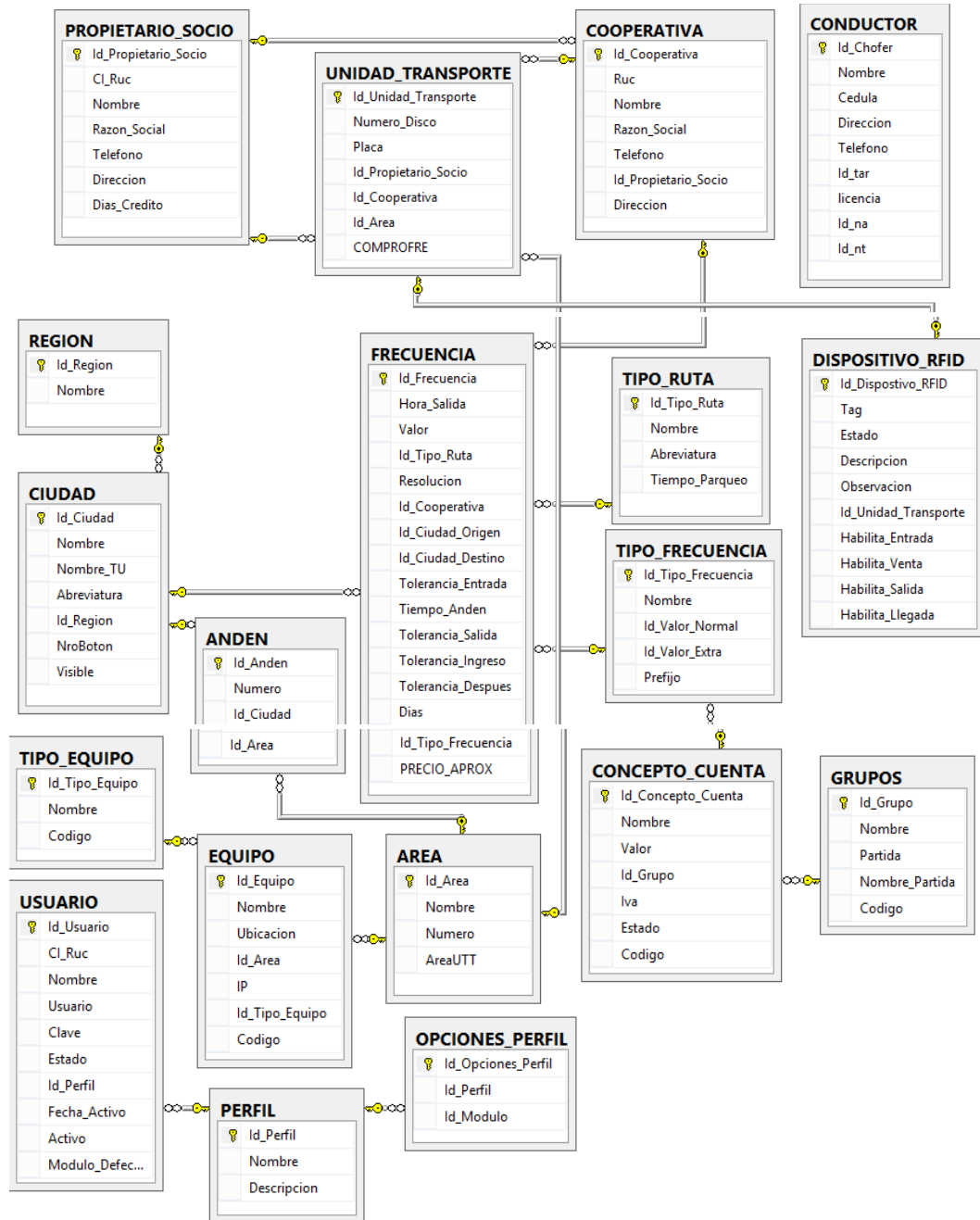


Figura 44 Base de datos SICTERT

BASE DE DATOS ÁREA DE TRANSFERENCIA ÁMERICA/CASHAPAMBA
(proporcionado por la institución)

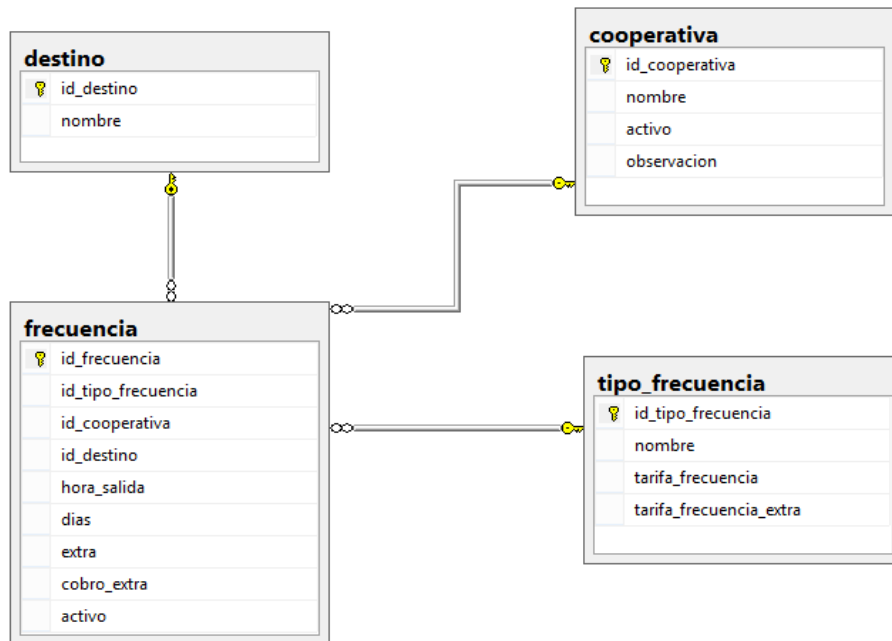


Figura 45 Base de datos sictert_lite_america

4.5.1. Modelo entidad relación

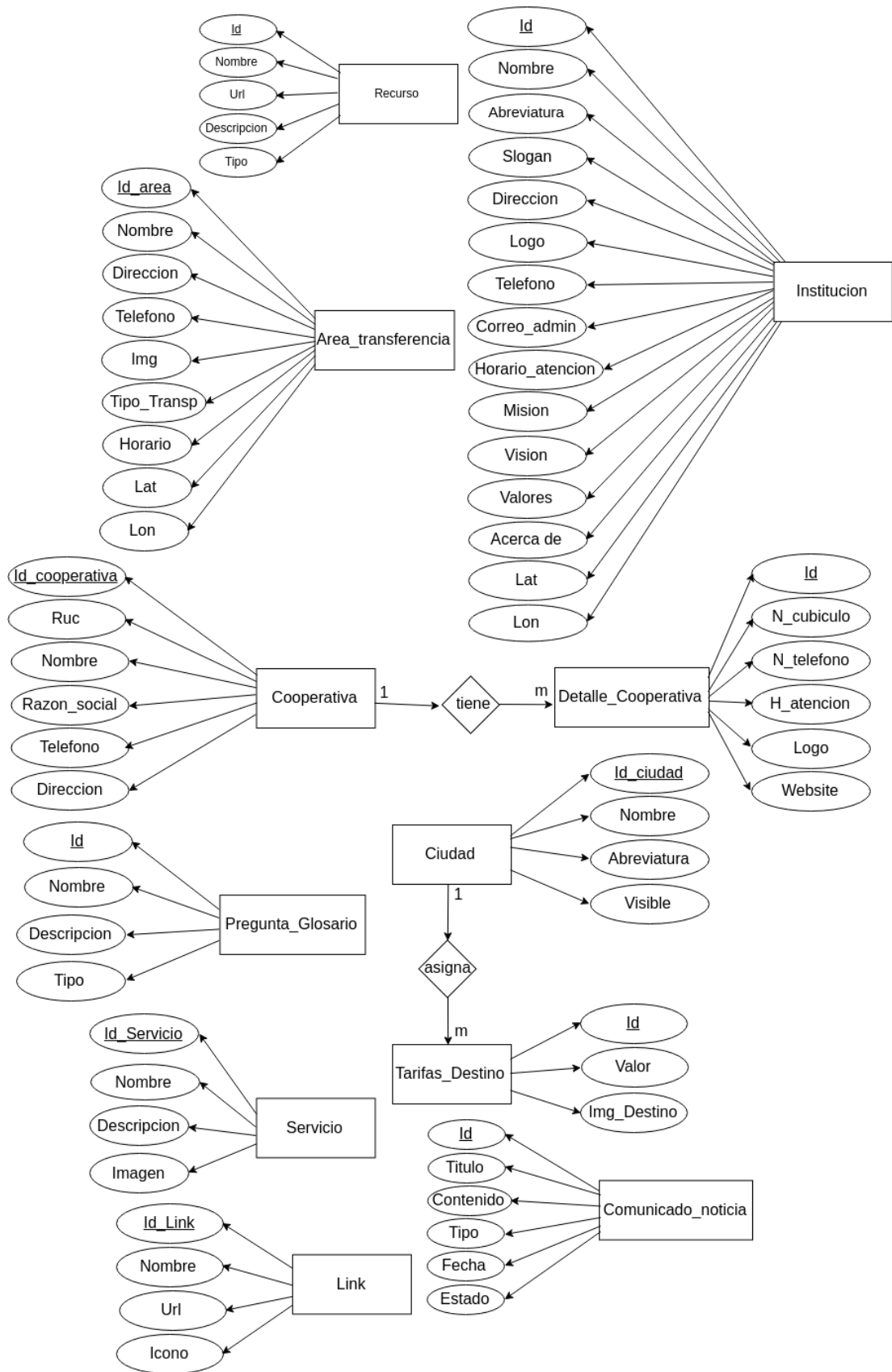


Figura 46 Modelo entidad-relación

Con la ayuda de una herramienta CASE como Power Designer se diseñó un diagrama entidad-relación completo en base al modelo anterior (véase **Figura 47**).

4.5.2. Diagrama Entidad/Relación completo

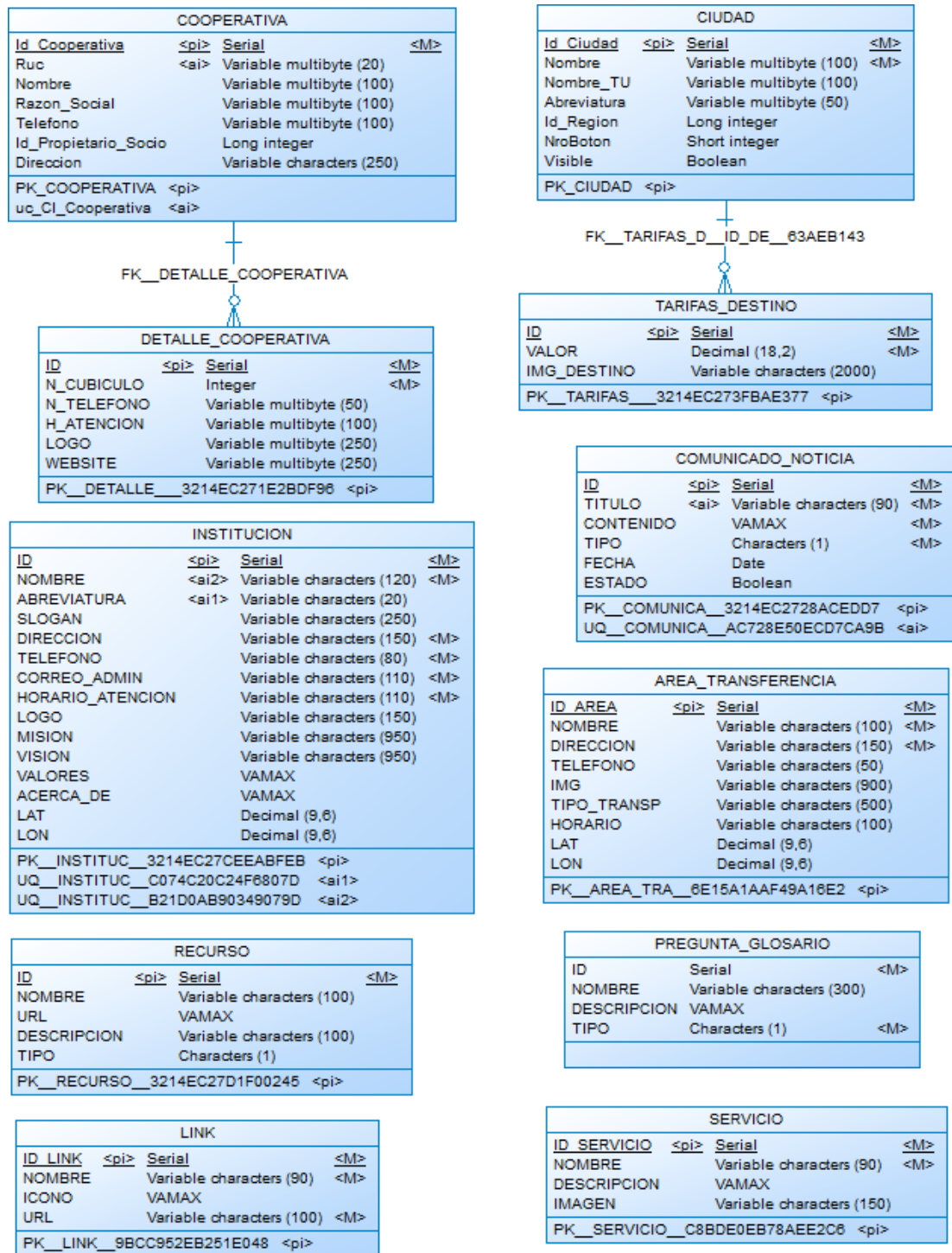


Figura 47 Diagrama Entidad/Relación completo

4.5.3. Diseño físico de base de datos

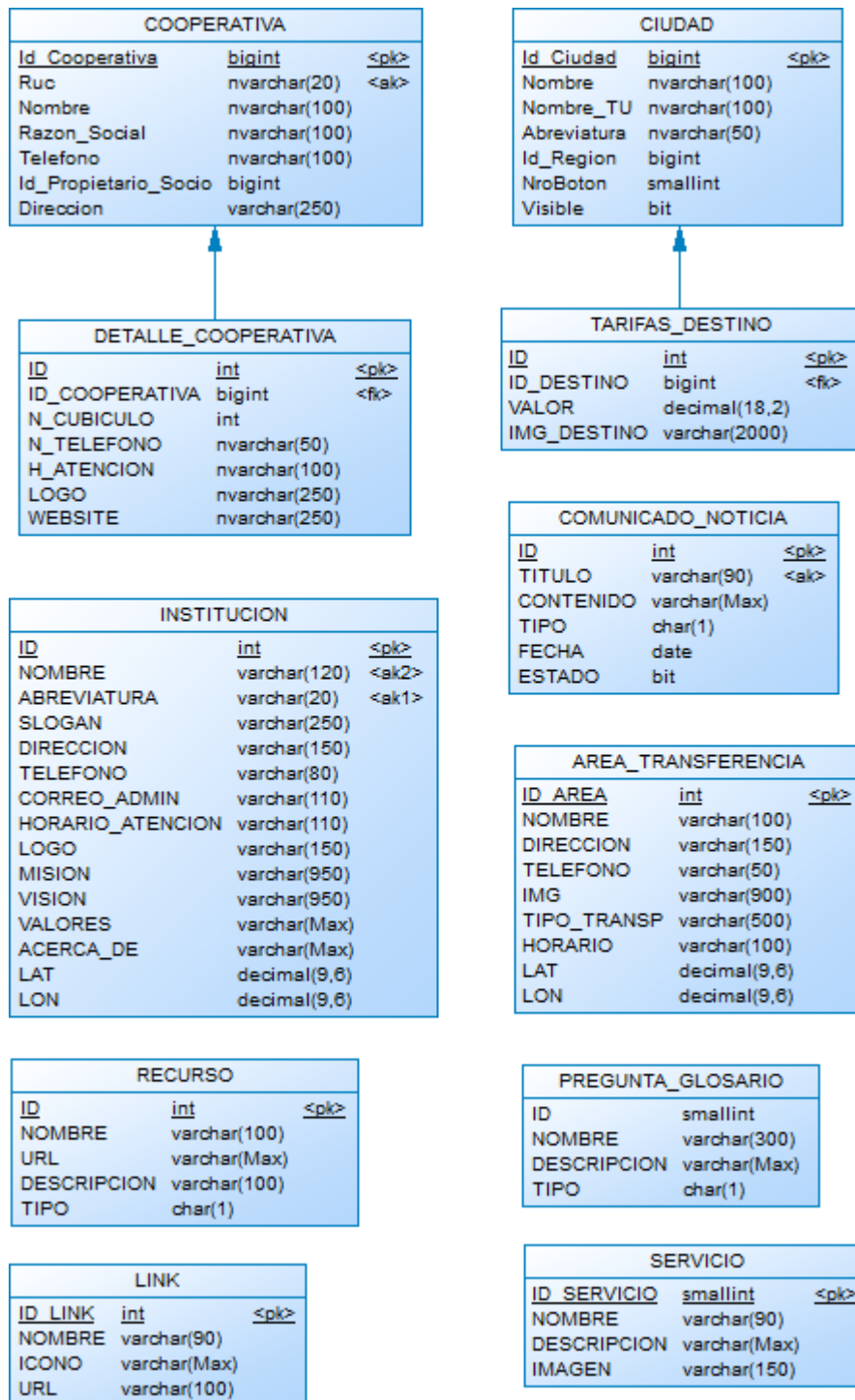


Figura 48 Diseño físico de base de datos

En la fase de implementación de base de datos se transformó la estructura obtenida del diagrama entidad/relación completa a una estructura física para el SGBD específico en este caso SQL Server (véase anexo 4).

4.5.4. Diccionario de datos

	TABLA	
NOMBRE:	Institución	
DESCRIPCIÓN:	Almacena la información institucional de la Unidad Desconcentrada de Terminales	
CAMPO:		
Id	int	Almacena el identificador de registro.
Nombre	varchar(120)	Almacena el nombre de la institución.
Abreviatura	varchar(20)	Almacena la abreviatura del nombre de la institución.
Slogan	varchar(250)	Almacena el slogan institucional.
Direccion	varchar(150)	Almacena la dirección de las instalaciones de la institución.
Telefono	varchar(80)	Almacena el o los teléfonos de contacto de la institución.
Correo_admin	varchar(110)	Almacena la dirección de correo electrónico del administrador del portal web.
Horario_atencion	varchar(110)	Almacena los horarios de atención de la institución.
Logo	varchar(150)	Almacena la ruta hacia el logo que identifica a la institución.
Mision	varchar(950)	Almacena la misión institucional.
Vision	varchar(950)	Almacena la visión institucional.
Valores	varchar(max)	Almacena los valores institucionales.
Acerca_de	varchar(max)	Almacena una descripción de lo que es o significa el portal web de la institución.
Lat	decimal(9,6)	Almacena la latitud para ubicar las instalaciones de la institución en Google maps.
Lon	decimal(9,6)	Almacena la longitud para ubicar las instalaciones de la institución en Google maps

Tabla 4 Tabla institución

	TABLA	
NOMBRE:	Servicio	
DESCRIPCIÓN:	Almacena los servicios que se ofrecen en la Unidad Desconcentrada de Terminales	
CAMPO:		
Id_Servicio	smallint	Almacena el identificador de servicio.
Nombre	varchar(90)	Almacena el nombre del servicio.
Descripcion	varchar(max)	Almacena la descripción del servicio.
Imagen	varchar(150)	Almacena la ruta hacia una imagen representativa del servicio.

Tabla 5 Tabla servicio

	TABLA	
NOMBRE:	Pregunta_Glosario	
DESCRIPCIÓN:	Almacena las preguntas frecuentes y glosario de términos.	
CAMPO:		
Id	smallint	Clave primaria, almacena el identificador de pregunta o termino.
Nombre	varchar(300)	Almacena tanto la pregunta como el término según sea el tipo.
Descripcion	varchar(max)	Almacena tanto la respuesta o definición de término según sea el tipo.
Tipo	char(1)	Almacena el tipo en el dominio P para pregunta frecuente y G para glosario de términos.

Tabla 6 Tabla preguntas frecuentes y glosario de términos

	TABLA	
NOMBRE:	Comunicado_noticia	
DESCRIPCIÓN:	Almacena tanto los comunicados como las noticias de la Unidad Desconcentrada de Terminales.	
CAMPO:		
Id	int	Clave primaria, almacena el identificador de comunicado o noticia.
Titulo	varchar(90)	Almacena el título del comunicado o noticia.
Contenido	varchar(max)	Almacena el contenido del comunicado o noticia.
Tipo	char(1)	Almacena el tipo en el dominio C para comunicado y N para noticia.
Fecha	date	Almacena la fecha de publicación.
Estado	bit	Almacena el estado del comunicado o noticia en el dominio 1 para publicado y 0 para no publicado.

Tabla 7 Tabla comunicado noticia

	TABLA	
NOMBRE:	Area_transferencia	
DESCRIPCIÓN:	Almacena la información de las áreas de transferencia.	
CAMPO:		
Id_area	tinyint	Clave primaria, almacena el identificador del área de transferencia.
Nombre	varchar(100)	Almacena el nombre del área de transferencia.
Direccion	varchar(150)	Almacena la dirección del área de transferencia.
Telefono	varchar(50)	Almacena el o los teléfonos del área de transferencia.

Img	varchar(900)	Almacena la ruta hacia la imagen del área de transferencia
Tipo_Transp	varchar(500)	Almacena una descripción que especifica el tipo de transporte que ofrece un área de transferencia
Horario	varchar(100)	Almacena el horario de atención de un área de transferencia
Lat	decimal(9,6)	Almacena la latitud para ubicar un área de transferencia en Google maps.
Lon	decimal(9,6)	Almacena la longitud para ubicar un área de transferencia en Google maps.

Tabla 8 Tabla área transferencia

TABLA		
NOMBRE:	Link	
DESCRIPCIÓN:	Almacena los links de interés.	
CAMPO:		
Id_link	tinyint	Clave primaria, almacena el identificador del link de interés.
Nombre	varchar(90)	Almacena el nombre del link de interés.
Icono	varchar(max)	Almacena la ruta hacia la imagen del link de interés.
Url	varchar(100)	Almacena la dirección url del link de interés.

Tabla 9 Tabla links de interés

TABLA		
NOMBRE:	Recurso	
DESCRIPCIÓN:	Almacena los recursos que forman parte del portal web.	
CAMPO:		
Id	int	Clave primaria, almacena el identificador del recurso.
Nombre	varchar(100)	Almacena el nombre del recurso.
Url	varchar(max)	Almacena la ruta donde se ubica el recurso.
Descripcion	varchar(100)	Almacena la descripción del recurso.
Tipo	char(1)	Almacena el tipo de recurso en el dominio I para imagen, S para slide, A para archivo y M para multimedia (audio y video).

Tabla 10 Tabla recurso

TABLA		
NOMBRE:	Detalle_Cooperativa	
DESCRIPCIÓN:	Almacena la información detallada de las operadoras de transporte que laboran en el terminal.	

CAMPO:		
Id	int	Clave primaria, almacena el identificador del registro.
Id_Cooperativa	bigint	Clave foránea, almacena el identificador de la operadora de transporte.
N_cubiculo	int	Almacena el número de cubículo o local que ocupa una operadora dentro de las instalaciones del terminal.
N_telefono	varchar(50)	Almacena el número de teléfono de una operadora dentro de las instalaciones del terminal.
H_atencion	varchar(100)	Almacena los horarios en que una operadora dentro de las instalaciones del terminal presta sus servicios.
Logo	varchar(250)	Almacena la ruta hacia el logo de una operadora de transporte.
Website	varchar(250)	Almacena la url hacia el sitio web o correo electrónico de una operadora de transporte.

Tabla 11 Tabla directorio de cooperativas

TABLA		
NOMBRE:	Tarifas_Destino	
DESCRIPCIÓN:	Almacena la información de las tarifas establecidas por la ANT para destinos interprovinciales.	
CAMPO:		
Id	int	Clave primaria, almacena el identificador de la tarifa.
Id_Destino	bigint	Clave foránea, almacena el identificador de destino.
Valor	decimal(18,2)	Almacena el costo de la tarifa.
Img_Destino	varchar(2000)	Almacena la ruta hacia la imagen de la ciudad destino.

Tabla 12 Tabla tarifa de transporte

4.6. MIDAS/FC (Funcionalidad)

4.6.1. Plataforma de desarrollo

Hardware

El portal web fue desarrollado en un computador con los requerimientos básicos, tales como:

- ✓ Procesador: Intel® Core™ i5-7200U CPU @ 2.50GHz × 4
- ✓ Memoria instalada (RAM): 8 GB
- ✓ Tamaño de disco: 1024 GB
- ✓ Sistema operativo: Ubuntu 17.10

Software

El portal web fue desarrollado haciendo uso del siguiente software:

- ✓ Máquina Virtual Java: JDK 1.7.0_80
- ✓ JBoss EAP 6.1
- ✓ SQL Server 2017 Developer Edition
- ✓ Spring Tool Suit (STS) 3.6.4
- ✓ DBeaver Community Edition 4.3.3.1
- ✓ Exploradores (Firefox Quantum 59.0.1, Google Chrome Versión 65.0.3325.162)
- ✓ JBoss Tools.

4.6.2. Desarrollo

En esta fase se tiene el desarrollo del portal web, dado que en esta se desarrolló la lógica de la aplicación, se hizo uso de la arquitectura MVC en 3 niveles, la codificación se realizó en lenguaje Java.

4.6.2.1. Estructura del proyecto

Estructura del proyecto utilizando el arquetipo web-app de maven

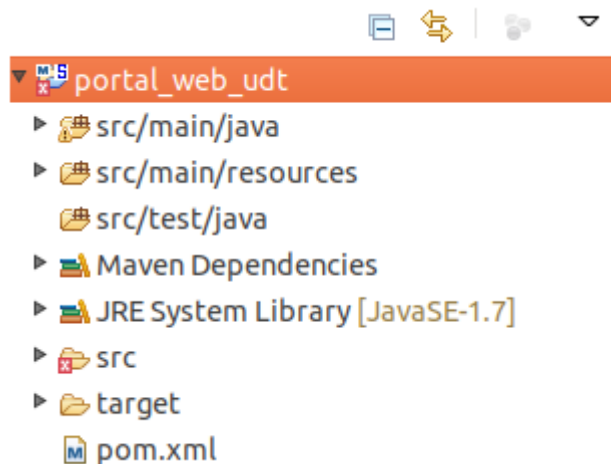


Figura 49 Estructura del proyecto generado con maven

Estructura del archivo pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>uta.gad.udt</groupId>
  <artifactId>portal_web_udt</artifactId>
  <packaging>war</packaging>
  <version>0.0.1</version>
  <name>portal_web_udt</name>
  <url>http://udt.gob.ec</url>
  <properties>[]
  <dependencies>[]
  <build>
    <plugins>
      <plugin>
        <artifactId>maven-eclipse-plugin</artifactId>
        <version>2.9</version>
        <configuration>[]
      </plugin>
      <plugin>[]
    </plugins>
    <finalName>udt</finalName>
  </build>
</project>

```

Figura 50 Estructura del archivo de configuración de maven

Principales librerías utilizadas desde el repositorio de maven

- ✓ org.springframework 4.2.0.RELEASE
- ✓ org.hibernate 4.3.11.Final
- ✓ net.sourceforge.jtds 1.3.1
- ✓ org.apache.tiles 3.0.5
- ✓ javax.mail 1.5.5
- ✓ commons-fileupload 1.3

4.6.2.2. Modelo

Se utiliza simples clases POJO (únicamente con sus constructores, getters y setters) para representar las entidades de la base de datos. Los modelos se han generado haciendo uso de Hibernate Tools.

4.6.2.3. Creación de la configuración de hibernate

Hibernate Tools, es un plugin que se agrega a Spring Tool Suite, el IDE que se emplea para el presente proyecto, y nos proporciona un asistente para crear el fichero de configuración de Hibernate (normalmente hibernate.cfg.xml). En este fichero es donde describiremos cómo se debe conectar Hibernate a la base de datos, cuáles son los ficheros xml que describen los mapeos entre las clases y las tablas de la base de datos. Para que la aplicación pueda localizar el archivo de configuración (si usamos Maven) el directorio donde guardaremos será src/main/resources.

Indicamos el dialecto que debe usar Hibernate. El dialecto, básicamente, es el idioma que ha de hablar el ORM con nuestra base de datos. También indicamos la clase del driver de acceso a la base de datos, la URL de conexión, el esquema, catálogo por defecto, el usuario y la contraseña, en definitiva, toda la información para que se pueda conectar correctamente a nuestra base de datos (véase **Figura 51**).

Hibernate Configuration File (cfg.xml)
This wizard creates a new configuration file to use with Hibernate.

Container: /proyecto_prueba/src/main/resources
File name: hibernate.cfg.xml
Session factory name:

Get values from Connection

Database dialect: SQL Server
Driver class: net.sourceforge.jtds.jdbc.Driver
Connection URL: jdbc:jtds:sqlserver://127.0.0.1:1433/SICTERT;instance=patricio-Inspir
Default Schema: dbo
Default Catalog: SICTERT
Username: sa
Password: jSsA1991

Create a console configuration

? < Back Next > Cancel Finish

Figura 51 Configuración de hibernate

4.6.2.4. Hibernate console

La consola es el eje central de Hibernate Tools, ya que cualquier otra operación que queramos hacer (generar código, lanzar sentencias HQL) dependerán de la configuración de la consola. Esta nueva configuración se ha definido al momento de crear el archivo hibernate.cfg.xml el mismo que se edita.

En la pestaña “Classpath” indicamos dónde se encuentra el driver de la base de datos (véase **Figura 52**). Esto es muy importante, ya que de lo contrario Hibernate Tools serán incapaces de conectarse con la base de datos.

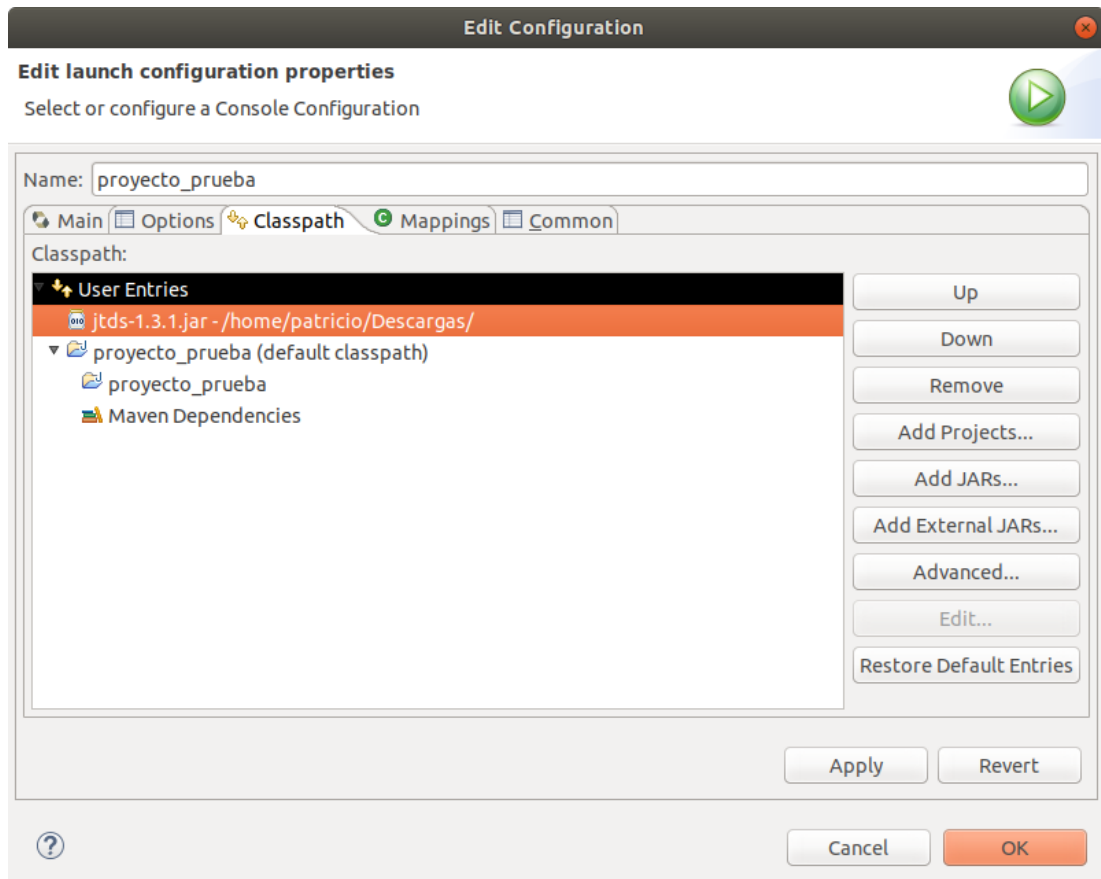


Figura 52 Configuración hibernate console

4.6.2.5. Generación de código

A partir de las tablas creadas en nuestra base de datos vamos a generar código. Para ello pulsamos sobre el menú Run>Hibernate Code Generation.

Al pulsar la opción, nos aparece una ventana, donde en el lado izquierda aparece una zona en blanco. Sobre ella hay una barra de herramientas, pulsamos sobre el primer icono “New launch configuration”. Y ahora rellenamos los datos de la zona de la derecha (véase **Figura 53**) y teniendo en cuenta lo siguiente:

- ✓ Se define un nombre al nuevo “launch configuration”.
- ✓ Indicamos el nombre de la configuración de consola que se debe utilizar.
- ✓ Marcamos la casilla “Reverse engineer from JDBC Connection”. Esto es para generar las clases a partir de la información que tenemos en la base de datos.
- ✓ Y en general marcamos las opciones que necesitemos acorde al proyecto.

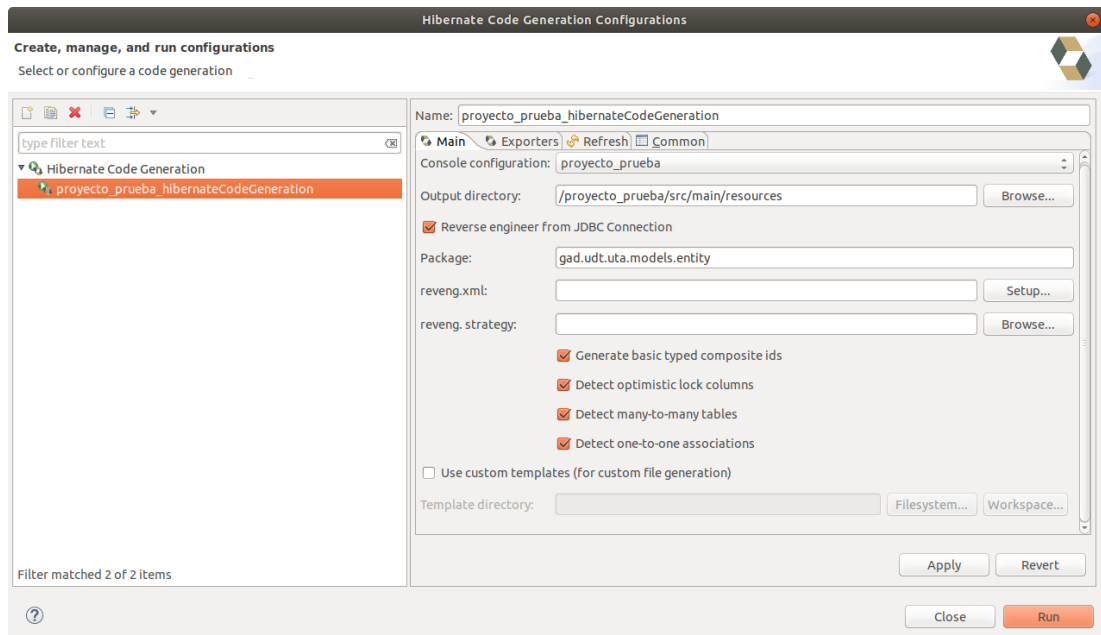


Figura 53 Generación de código con hibernate

En la pestaña de “Exporters” marcaremos las opciones “Domain code (java)” para generar los POJOs e “Hibernate XML Mappings (.hbm.xml)” para generar los ficheros de configuración xml donde se describe el mapeo entre las clases y las tablas de la base de datos; una vez configurado lo necesario pulsamos Run y el código se habrá generado (véase **Figura 54**) y nuestra capa de Modelo queda implementada y lista para interactuar con las demás capas.

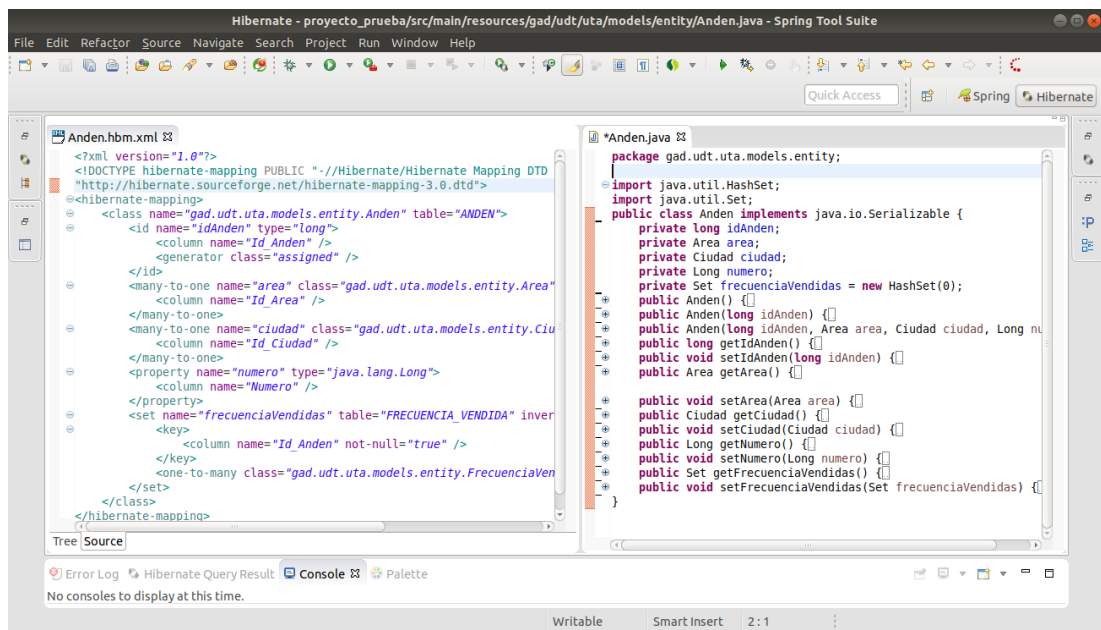


Figura 54 Código generado con hibernate tools

4.6.2.6. Capa DAO

La capa DAO contiene todos los métodos CRUD (create, read, update, delete) que son las operaciones más comunes que se realiza a la información contenida en una base de datos, la implementación se la realiza de la siguiente manera.

Se crea el acceso a los datos a través de una interface, aquí se declara todos los métodos para acceder a los datos.

Luego se crea una clase que implemente dicha interfaz, es decir darle funcionalidad a cada método de la interface, utilizamos `@Repository` que es una de las anotaciones estándar de spring para indicar que la clase es un DAO.

```
package uta.gad.udt.models.dao;
import java.util.List;
@Repository
public class AdministracionDAOImpl implements AdministracionDAO{
    @SuppressWarnings("unchecked")
    @Override
    public List<Servicio> getServicios() {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Query callStoredProcedure = session.getNamedQuery("listarServicios");
        List<Servicio> results= (List<Servicio>)callStoredProcedure.list();
        session.close();
        return results;
    }
    @Override
    public Servicio getServicio(short idServicio)
    {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Servicio servicio = null;
        Transaction tx = session.getTransaction();
        tx.begin();
        servicio = (Servicio)session.get(Servicio.class,idServicio);
        tx.commit();
        session.close();
        return servicio;
    }
    @Override
    public int saveServicio(Servicio servicio) {
        return CRUDGeneric(servicio,"s");
    }
    @Override
    public int deleteServicio(Servicio servicio)
    {
        return CRUDGeneric(servicio,"d");
    }
    public int CRUDGeneric(Object object,String crudOperacion)[]
}
```

```
package uta.gad.udt.models.dao;
import java.util.List;
import uta.gad.udt.models.entity.Servicio;
public interface AdministracionDAO {
    public List<Servicio> getServicios();
    public Servicio getServicio(short idServicio);
    public int saveServicio(Servicio servicio);
    public int deleteServicio(Servicio servicio);
}
```

Figura 55 Implementación de la capa DAO

4.6.2.7. Capa de Servicio

La capa de servicio encapsula toda la funcionalidad de la capa DAO, su implementación es muy similar a la capa DAO se basa en una interfaz y otra clase que implemente dicha interfaz, utilizamos `@Service` que es una de las anotaciones estándar de spring, para indicar que la clase es un bean de la capa de servicio. Además hacemos uso de la anotación `@Autowired` para mediante inyección resolver las dependencias del bean de la capa DAO.


```

package uta.gad.udt.models.service;

import java.util.List;

import uta.gad.udt.models.entity.Servicio;

public interface AdministracionService {

    public List<Servicio> getServicios();
    public int deleteServicio(Servicio servicio);
    public Servicio getServicio(short idServicio);
    public int saveServicio(Servicio servicio);
}

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import uta.gad.udt.models.dao.AdministracionDAO;
import uta.gad.udt.models.entity.Servicio;

@Service("administracionService")
public class AdministracionServiceImpl implements AdministracionService {

    @Autowired
    private AdministracionDAO administracionDAO;

    @Override
    public List<Servicio> getServicios() {
        return administracionDAO.getServicios();
    }

    @Override
    public int deleteServicio(Servicio servicio) {
        return administracionDAO.deleteServicio(servicio);
    }

    @Override
    public Servicio getServicio(short idServicio) {
        return administracionDAO.getServicio(idServicio);
    }

    @Override
    public int saveServicio(Servicio servicio) {
        return administracionDAO.saveServicio(servicio);
    }
}

```

Figura 56 Implementación de la capa de servicio

4.6.2.8. Vista

Aquí se define la parte visual del portal web, en la vista establecemos todas las interfaces gráficas de usuario, mediante las cuales representamos todo el modelo permitiendo la interacción entre el portal y el cliente.

Para definir las vistas partimos de una plantilla de bootstrap ésta plantilla tiene cinco atributos (véase **Figura 57**): título (de tipo cadena), cabecera, menú, cuerpo y pie de página, se creará cuatro páginas JSP que tendrán lugar en los atributos de cabecera, menú, cuerpo y pie de página dicha plantilla pasa a ser como una página maestra a partir de la cual crearemos páginas de contenido.

```

*layout.jsp cabecera.jsp menu.jsp institucion.jsp pie.jsp
<%@ taglib uri="http://tiles.apache.org/tags-tiles" prefix="tiles"%>
<!DOCTYPE html>
<html lang="en">
<head>
<body class="no-skin" >
<tiles:insertAttribute name="cabecera" />
<tiles:insertAttribute name="menu" />
<tiles:insertAttribute name="cuerpo" />
<tiles:insertAttribute name="pie" />
</body>
</html>

```

Figura 57 Plantilla web según la estructura de apache tiles

4.6.2.9. Validaciones

Para validar un modelo es necesario crear una clase que implemente la interface Validator, debemos sobre-escribir los dos métodos que se detallan a continuación:

- ✓ supports: indica la clase a la que el validador le da soporte.
- ✓ validate: realiza la validación, si hay errores registra los correspondientes mensajes.

```

package uta.gad.udt.validator;
import org.springframework.stereotype.Component;
@Component
public class UsuarioFormValidator implements Validator{
    @Override
    public boolean supports(Class<?> clazz) {
        return Usuario.class.equals(clazz);
    }
    @Override
    public void validate(Object target, Errors errors) {
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "ciRuc", "errors.empty.usuario.ciruc");
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "nombre", "errors.empty.usuario.nombre");
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "usuario", "errors.empty.usuario.usuario");
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "clave", "errors.empty.usuario.clave");
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "repiteClave", "errors.empty.usuario.clave");
        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "idPerfil", "errors.empty.usuario.idPerfil");
        Usuario usuario = (Usuario) target;
        if(usuario.getCiRuc().trim().length()>20){
            errors.rejectValue("ciRuc", "errors.length.tiporuta.nombre");
        }
        if(usuario.getNombre().trim().length()>50){
            errors.rejectValue("nombre", "errors.length.tiporuta.nombre");
        }
        if(usuario.getUsuario().trim().length()>20){
            errors.rejectValue("usuario", "errors.length.tiporuta.nombre");
        }
        if(!usuario.getClave().trim().equals(usuario.getRepiteClave().trim())){
            errors.rejectValue("repiteClave", "errors.empty.usuario.repiteclave");
        }
    }
}

```

Figura 58 Implementación clase validator

La validación se realiza con la clase ValidationUtils e implementando nuestra propia lógica. El método rejectIfEmptyOrWhitespace() lanza un error si el campo indicado

está vacío o contiene solo espacios en blanco, los parámetros son los siguientes: primero la variable en donde se almacenan los errores, luego el nombre del campo que se valida, sigue la clave usada para localizar el mensaje de error en un archivo de propiedades.

Usando el método `rejectValue()` generamos un error si un campo no cumple con una determinada condición, por ejemplo, si la contraseña y el repita contraseña no coinciden generamos el error, los parámetros son similares a lo explicado previamente. El controlador valida el formulario cuando presionamos el botón enviar del formulario utilizando HTTP POST, si detecta un error regresa a la página para mostrar los mensajes de error y procesar nuevamente el formulario (véase **Figura 59**).

Usuarios

← Listado Usuarios

CI/RUC Escriba la cédula o ruc...
CI o RUC es requerido!

Nombre Escriba el nombre...
Nombre de usuario es requerido!

Usuario Escriba el nombre de usuario...
Usuario es requerido!

Contraseña Escriba la contraseña...
Contraseña es requerido!

Repita Contraseña Vuelva a escribir la contraseña...
Contraseña es requerido!

Mostrar contraseña

Perfil Escoja un perfil...
Ha ingresado un tipo de dato inválido
Seleccione un perfil al que pertenece!

Habilitado

Figura 59 Validación de formulario usuario

4.6.2.10. Controlador

El controlador es una clase que carga el modelo de datos e invoca a la vista es decir es el encargado de establecer la conexión entre la vista y el modelo. El código se muestra a continuación y se ubica en el paquete controllers:

```

package uta.gad.udt.controllers;
import java.math.BigDecimal;[]
@Controller
@RequestMapping("admin/")
@SessionAttributes("user")
public class AdminSictertController {
    @Autowired
    private AdministracionService administracionService;
    @Autowired
    UsuarioFormValidator usuarioFormValidator;
    @RequestMapping(value="form-usuario" ,method = RequestMethod.GET)
    public ModelAndView formUsuario(@RequestParam(value = "id", required = false, defaultValue = "0") long id, Model model) {
        ModelAndView modelo = new ModelAndView("form-usuario");
        Usuario usuario = null;
        if (id > 0)
            usuario = administracionService.getUsuarioById(id);
        else
            usuario = new Usuario();
        modelo.addObject("perfiles", administracionService.getPerfiles());
        modelo.addObject("usuario", usuario);
        return modelo;
    }
    @RequestMapping(value="form-usuario" ,method = RequestMethod.POST)
    public String submitUsuario(@ModelAttribute("usuario") Usuario usuario, BindingResult result, Model model) {
        usuario.setPerfil(new Perfil(usuario.getIdPerfil()));
        modelo.addAttribute("perfiles", administracionService.getPerfiles());
        usuarioFormValidator.validate(usuario, result);
        if (result.hasErrors())
            model.addAttribute("ok", "2|La información ingresada tiene errores revisala");
        else
        {
            if(usuario.getIdUsuario()>0) //update
            {
                if(usuario.isUpdatePassword())
                    usuario.setClave(bcryptEncoder.encode(usuario.getClave()));
            }
            else //insert
                usuario.setClave(bcryptEncoder.encode(usuario.getClave()));

            if(administracionService.saveUsuario(usuario)>0)
            {
                model.addAttribute("ok", "1|Información ingresada correctamente ");
                usuario.setUpdatePassword(false);
            }
            else model.addAttribute("ok", "2|Información NO ingresada");
        }
        return "form-usuario";
    }
}

```

Figura 60 Controlador del proyecto

4.6.3. Configuraciones

4.6.3.1. Configuración spring framework

Para configurar la aplicación es necesario contar con un archivo servlet-context.xml ubicado en el directorio /WEB-INF (véase **Figura 61**), las partes más relevantes se detallan a continuación:

1. Para que Spring automáticamente busque los beans es necesario la declaración <context:component-scan>. Además, hace falta indicarle la base del paquete donde queremos que busque. Con eso buscaría en nuestra ruta todos los archivos java que en su interior tengan las siguientes anotaciones (además de otras):
 - ✓ @Component: Es de propósito general, indica que la clase es un componente Spring.
 - ✓ @Controller: Indica que la clase define un controlador Spring.
 - ✓ @Repository: La clase es un repositorio de datos.
 - ✓ @Service: La clase define un servicio.

2. Para que Spring interactúe con Apache Tiles, es necesario declarar 2 beans: viewResolver y tilesConfigurer.
3. ViewResolver será la encargada de resolver la vista, es decir, obtener el archivo físico que se usará para generar las vistas a partir del nombre lógico devuelto por el controlador.
4. En Spring, puede usar ResourceBundleMessageSource para resolver mensajes de texto de un archivo de propiedades.
5. JavaMailSenderImpl para el envío de correo electrónico.

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="
    http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd
    http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd">
  <!-- Escanea o busca en el package base de la aplicación clases beans anotados con @Components, @Controller, @Service, @Repo -->
  <context:component-scan base-package="uta.gad.udt.controllers" />
  <context:component-scan base-package="uta.gad.udt.froala" />
  <context:component-scan base-package="uta.gad.udt.validator" />
  <!-- Habilita la anotación de Spring MVC @Controller -->
  <mvc:annotation-driven />
  <!-- Controla las peticiones HTTP GET /recursos/** para los recursos estáticos (ej: imagenes), estos se almacenan en el dir -->
  <mvc:resources mapping="/recursos/**" location="/recursos/"
    cache-period="10000" />
  <!-- La url /inicio inmediatamente esta mapeada hacia la vista inicio.jsp -->
  <mvc:view-controller path="/inicio" view-name="inicio" />
  <!-- Para resolver vistas Tiles -->
  <bean id="viewResolver"
    class="org.springframework.web.servlet.view.UrlBasedViewResolver">
    <property name="order" value="1" />
    <property name="viewClass" value="org.springframework.web.servlet.view.tiles3.TilesView"/>
  </bean>
  <!-- Configuración del layout Tiles -->
  <bean id="tilesConfigurer"
    class="org.springframework.web.servlet.view.tiles3.TilesConfigurer">
    <property name="definitions">
      <list>
        <value>/WEB-INF/tiles.xml</value>
      </list>
    </property>
  </bean>
  <!-- Resuelve la ubicación de las vistas .jsp de @Controllers en la ruta /WEB-INF/vistas -->
  <bean
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="order" value="2" />
    <property name="prefix" value="/WEB-INF/vistas/" />
    <property name="suffix" value=".jsp" />
  </bean>
  <bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver"/>
  <bean id="messageSource"
    class="org.springframework.context.support.ResourceBundleMessageSource">
    <property name="basename"
    value="uta/gad/udt/validator/errors"/>
  </bean>
  <context:property-placeholder location="WEB-INF/spring/appServlet/config.properties"/>
  <!-- Spring Email Sender Bean Configuration -->
  <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">
  </bean>
  <!--
  <bean id="conversionService" class="org.springframework.format.support.FormattingConversionServiceFactoryBean">
  </bean>
  </beans>

```

Figura 61 Configuración Spring framework

4.6.3.2. Configuración spring security

La seguridad en spring está dada por el archivo spring-security.xml (véase **Figura 62**), las partes más relevantes se detallan a continuación:

1. En esta parte especificamos: los perfiles (mismos de la base de datos), la url que indique la página de inicio de sesión, los nombres de parámetros de usuario y contraseña, la url que muestre una página de acceso no autorizado.
2. authenticationSuccessHandler permite redireccionar a la página de inicio después que el usuario haya iniciado sesión correctamente.
3. BCryptPasswordEncoder permite obtener contraseñas encriptada en lugar de un texto simple.
4. customUserDetailsService es el proveedor de autenticación para el proyecto.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/security"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:beans="http://www.springframework.org/schema/beans"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.2.xsd
  http://www.springframework.org/schema/security http://www.springframework.org/schema/security/spring-security-4.0.xsd">
  <http auto-config="true">
    <intercept-url pattern="/" access="permitAll" />
    <intercept-url pattern="/admin/**" access="hasRole('WEB_ADMIN')" />
    <form-login login-page="/login"
      username-parameter="ssoId"
      authentication-success-handler-ref="authenticationSuccessHandler"
      password-parameter="password" />
    <access-denied-handler error-page="/no_authorized" />
    <csrf disabled="true"/>
    <headers>
    <frame-options disabled="true"></frame-options>
    </headers>
  </http>
  <beans:bean id="authenticationSuccessHandler"
    class="org.springframework.security.web.authentication.SavedRequestAwareAuthenticationSuccessHandler">
    <beans:property name="defaultTargetUrl" value="/admin/inicio"/>
  </beans:bean>
  <beans:bean id="bcryptEncoder"
    class="org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder" />
  <authentication-manager>
    <authentication-provider user-service-ref="customUserDetailsService">
      <password-encoder ref="bcryptEncoder" />
    </authentication-provider>
  </authentication-manager>
</beans:beans>
```

Figura 62 Configuración Spring Security

4.6.3.3. Configuración hibernate

Para la configuración de hibernate se tienen dos archivos (véase **Figura 63**) el primero es hibernate.cfg.xml que es el archivo de configuración principal de hibernate y el segundo hibernate.properties que almacena diferentes propiedades o parámetros de configuración que la aplicación pueda leer y editar.

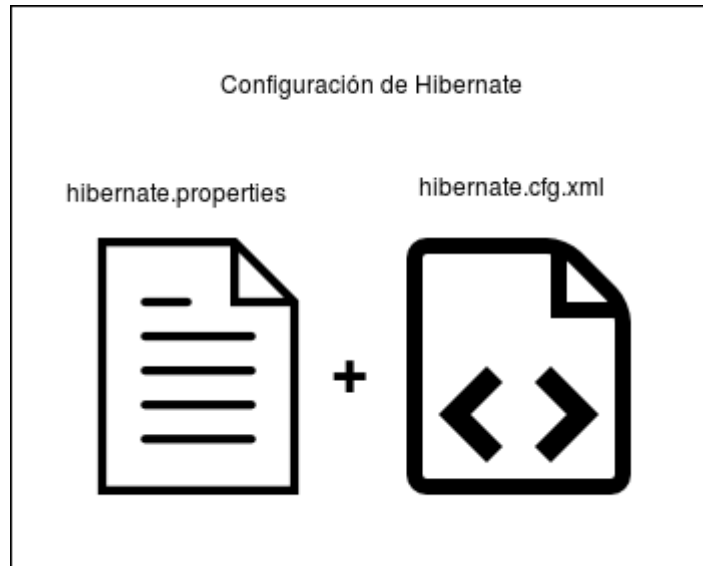


Figura 63 Esquema de configuración de hibernate

4.6.3.4. Archivo de propiedades

En java es habitual guardar algunos parámetros de configuración de nuestro programa en fichero de propiedades. Un fichero de propiedades no es más que un fichero de texto, habitualmente con la extensión .properties, que en cada línea tiene una pareja clave=valor (véase **Figura 64**).

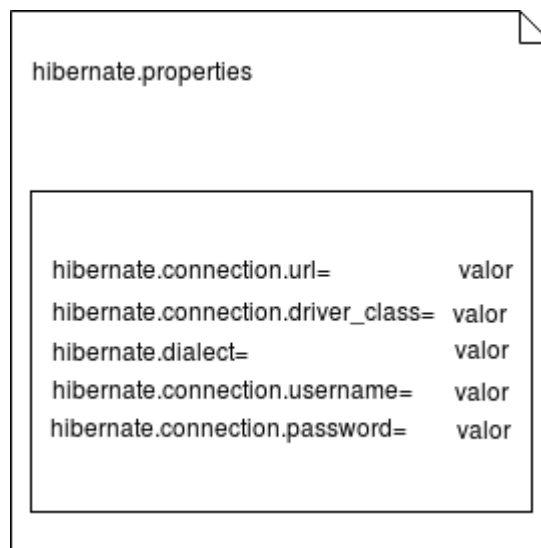


Figura 64 Archivo hibernate.properties

De la **Figura 64** se detalla que representa cada línea:

PARÁMETROS CONFIGURACIÓN HIBERNATE.PROPERTIES		
Nombre parámetro	Descripción	Ejemplo

hibernate.connection.url	Especifica la url donde se encuentra el servidor de base de datos para este proyecto SQL Server 2008 jdbc:jtds:sqlserver://<host>[:<port>][/<database>]	jdbc:jtds:sqlserver://localhost:5000/myDB
hibernate.connection.driver_class	Indica el nombre de la Clase Controlador predeterminado para este proyecto jTDS JDBC Driver	net.sourceforge.jtds.jdbc.Driver
hibernate.dialect	Representa un dialecto de SQL implementado por un RDBMS particular.	hibernate.dialect.org.hibernate.dialect.SQLServerDialect
hibernate.connection.username	Indica el nombre de usuario utilizado para la conexión con la base de datos	sa
hibernate.connection.password	Indica la contraseña utilizado para la conexión con la base de datos	sa

Tabla 13 Parámetros configuración en archivo hibernate.properties

4.6.3.5. Archivo hibernate.cfg.xml

El archivo de configuración de hibernate debe de llamarse por defecto hibernate.cfg.xml aquí se encuentra la información de mapeo que define cómo las clases Java se refieren a las tablas de la base. Además contiene un conjunto de valores de configuración relacionados con la base de datos (véase **Figura 65**).

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd" [
<hibernate-configuration>
  <session-factory name="SessionFactoryPersistencia">
    <property name="hibernate.connection.driver_class">${driver_class}</property>
    <property name="hibernate.connection.password">${password}</property>
    <property name="hibernate.connection.url">${hibernate.connection.url}</property>
    <property name="hibernate.connection.username">${username}</property>
    <property name="hibernate.default_catalog">SICTERT</property>
    <property name="hibernate.default_schema">dbo</property>
    <property name="hibernate.dialect">${dialect}</property>
    <mapping resource="entity/sictert/Frecuencia.hbm.xml"/>
    <mapping resource="entity/sictert/Grupos.hbm.xml"/>
    <mapping resource="entity/sictert/Modulo.hbm.xml"/>
    <mapping resource="entity/sictert/OpcionesPerfil.hbm.xml"/>
    <mapping resource="entity/sictert/PropietarioSocio.hbm.xml"/>
    <mapping resource="entity/sictert/Region.hbm.xml"/>
    <mapping resource="entity/sictert/TipoEquipo.hbm.xml"/>
    <mapping resource="entity/sictert/TipoFrecuencia.hbm.xml"/>
    <mapping resource="entity/sictert/TipoRuta.hbm.xml"/>
    <mapping resource="entity/sictert/UnidadTransporte.hbm.xml"/>
  </session-factory>
</hibernate-configuration>

```

Figura 65 Configuración archivo hibernate.cfg.xml

De la **Figura 65** se resaltan los siguientes puntos:

- ✓ Referencia a los parámetros de configuración definidos en el archivo hibernate.properties
- ✓ Los archivos que contienen el mapping entre las tablas de la base de datos y las clases java.

4.7. MIDAS/TST (Pruebas del sistema)

En esta etapa se detalla las pruebas a realizar para la evaluación final del portal web, se identifican los tipos de pruebas realizados con su respectiva explicación, se detalla los resultados de dichas pruebas.

4.7.1. Plan de pruebas

Este documento sirve para detallar que elementos son los que se van a probar, como realizar las pruebas y confirmar que los métodos utilizados cumplan con las soluciones a cada uno de los requerimientos solicitados por el cliente. Se incluirán los tipos de pruebas a realizar, que herramientas se ocuparon durante el desarrollo de las pruebas y su ambiente.

TIPO DE PRUEBA	DESCRIPCIÓN DE PRUEBA	HERRAMIENTA UTILIZADA
Pruebas funcionales	<p>Las pruebas funcionales son un tipo de prueba de caja negra que basa sus casos de prueba en las especificaciones del componente de software bajo prueba. Las pruebas funcionales generalmente describen lo que hace el sistema.</p> <p>Proceso a ejecutar</p> <ul style="list-style-type: none"> ✓ Elaborar los casos de prueba ✓ Ejecutar los casos de prueba ✓ Registrar los errores hallados según los casos de prueba. 	<p>Para la realización de estas pruebas no se requirió ninguna herramienta específica dado que la ejecución manual de este tipo de prueba sobre cada uno de los casos de uso fue suficiente.</p>
Pruebas no funcionales	<p>Las pruebas no funcionales se concentran en los aspectos del software que no se encuentran relacionados a una función</p>	<p>Para la realización de estas pruebas se hizo uso de las herramientas propias del navegador</p>

	<p>específica o usuario, se encarga de proveer una buena experiencia de usuario, algunos aspectos medidos podrían ser el rendimiento, compatibilidad, seguridad, fiabilidad, usabilidad, conformidad.</p> <p>Proceso a ejecutar</p> <ul style="list-style-type: none"> ✓ Elaborar los casos de prueba ✓ Ejecutar los casos de prueba ✓ Registrar los errores hallados según los casos de prueba. 	<p>Google Chrome (Device Mode) para la emulación de tamaños de pantallas en dispositivos móviles.</p>
Prueba de Stress	<p>Una prueba de stress permite forzar al sistema al máximo punto, para poder medir sus capacidades y las condiciones en las cuales trabaja, realizando una cantidad definida de peticiones y procesos.</p> <p>Proceso a ejecutar</p> <ul style="list-style-type: none"> ✓ Configurar el plan de pruebas. ✓ Ejecutar el plan de pruebas. ✓ Interpretar los resultados obtenidos. 	<p>Para la realización de estas pruebas se requirió el software JMeter, que es una aplicación de código abierto diseñada para probar el comportamiento funcional y medir el desempeño de una aplicación web.</p>

Tabla 14 Detalle del plan de pruebas

4.7.1.1. Ambiente de pruebas

Para el momento de la realización de las pruebas funcionales y no funcionales se optó por el uso de los siguientes navegadores como ambiente de pruebas:

Firefox

- ✓ Arquitectura: 64 bits
- ✓ Versión: Quantum 59.0.1

Google Chrome

- ✓ Arquitectura: 64 bits
- ✓ Versión: 65.0.3325.162

4.7.1.2. Pruebas funcionales

Al finalizar con el desarrollo de alguna de las funcionalidades del portal web se procede a realizar las pruebas de caja negra, para cada una de las tareas de los usuarios y del administrador, comprobando que se cumpla con los requerimientos funcionales solicitados por el cliente. Para el registro de los casos de prueba se ha utilizado tablas, sus campos son los siguientes:

Identificador: Identificador del caso de prueba para referencias. El formato definido es: CPF-01

Número de caso de prueba: Determina el orden del caso de prueba.

Usuario: Actor ejecutor del caso de prueba.

Referencia al caso de uso: Nombre del caso de uso que está sometido a prueba.

Nombre: Nombre del caso de prueba para la fácil comprensión del propósito del caso de prueba.

Inicialización: Datos o acciones, que se ejecutan previo a que los casos de prueba se hayan inicializado.

Salida esperada: Resultado esperado tras el cumplimiento del caso de prueba.

Propósito: Describe al caso de prueba y su finalidad.

Procedimiento de prueba: Secuencia de pasos para completar la prueba.

Salida obtenida: Descripción del comportamiento del software después de la ejecución de los casos de pruebas.

Caso de prueba “Creación, listado, actualización y eliminación de propietarios de unidades de transporte.”

CPF-01	
Nº caso de prueba	1
Usuario	Administrador
Referencia al caso de uso	Administración de propietarios de unidades de transporte.
Nombre	Creación, listado, actualización y eliminación de propietarios de unidades de transporte.
Inicialización	Iniciar sesión en la parte administrativa del portal web.
Salida esperada	Cambios (inserción, actualización, eliminación) de propietarios de unidades de transporte almacenados en el portal web.

Propósito	Realizar cambios directamente en la base de datos ubicado en el servidor a partir de páginas web.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. El administrador ingresa a la url de administración del portal web e inicia sesión con los datos correctos. 2. El administrador debe seleccionar la opción propietarios del menú. 3. El administrador visualizará un listado de todos los propietarios de unidades de transporte registrados. 4. El administrador puede optar por crear un nuevo propietario de unidad de transporte, actualizarla o eliminarla.
Salida obtenida	Inserción, actualización o eliminación de un registro en la base de datos.

Tabla 15 Caso de prueba CPF-01

Caso de prueba “Creación, listado, actualización y eliminación de operadoras de transporte.”

CPF-02	
N° caso de prueba	2
Usuario	Administrador
Referencia al caso de uso	Administración de operadoras de transporte.
Nombre	Creación, listado, actualización y eliminación de operadoras de transporte.
Inicialización	Iniciar sesión en la parte administrativa del portal web.
Salida esperada	Cambios (inserción, actualización, eliminación) de operadoras de transporte almacenados en el portal web.
Propósito	Realizar cambios directamente en la base de datos ubicado en el servidor a partir de páginas web.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. El administrador ingresa a la url de administración del portal web e inicia sesión con los datos correctos. 2. El administrador debe seleccionar la opción operadoras del menú. 3. El administrador visualizará un listado de todas las operadoras de transporte registrados. 4. El administrador puede optar por crear una nueva operadora de transporte, actualizarla o eliminarla.

Salida obtenida	Inserción, actualización o eliminación de un registro en la base de datos.
------------------------	--

Tabla 16 Caso de prueba CPF-02

Caso de prueba “Creación, listado, actualización y eliminación de áreas del terminal terrestre.”

CPF-03	
N° caso de prueba	3
Usuario	Administrador
Referencia al caso de uso	Administración de áreas del terminal terrestre.
Nombre	Creación, listado, actualización y eliminación de áreas del terminal terrestre.
Inicialización	Iniciar sesión en la parte administrativa del portal web.
Salida esperada	Cambios (inserción, actualización, eliminación) de áreas del terminal almacenados en el portal web.
Propósito	Realizar cambios directamente en la base de datos ubicado en el servidor a partir de páginas web.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. El administrador ingresa a la url de administración del portal web e inicia sesión con los datos correctos. 2. El administrador debe seleccionar la opción áreas del menú. 3. El administrador visualizará un listado de todas las áreas del terminal terrestre registrados. 4. El administrador puede optar por crear una nueva área, actualizarla o eliminarla.
Salida obtenida	Inserción, actualización o eliminación de un registro en la base de datos.

Tabla 17 Caso de prueba CPF-03

Caso de prueba “Creación, listado, actualización y eliminación de unidades de transporte.”

CPF-04	
N° caso de prueba	4
Usuario	Administrador
Referencia al caso de uso	Administración de unidades de transporte.
Nombre	Creación, listado, actualización y eliminación de unidades de transporte.

Inicialización	Iniciar sesión en la parte administrativa del portal web.
Salida esperada	Cambios (inserción, actualización, eliminación) de unidades de transporte almacenados en el portal web.
Propósito	Realizar cambios directamente en la base de datos ubicado en el servidor a partir de páginas web.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. El administrador ingresa a la url de administración del portal web e inicia sesión con los datos correctos. 2. El administrador debe seleccionar la opción unidades de transporte del menú. 3. El administrador visualizará un listado de todas las unidades de transporte registrados. 4. El administrador puede optar por crear una nueva unidad de transporte, actualizarla o eliminarla.
Salida obtenida	Inserción, actualización o eliminación de un registro en la base de datos.

Tabla 18 Caso de prueba CPF-04

Caso de prueba “Creación, listado, actualización y eliminación de conductores.”

CPF-05	
Nº caso de prueba	5
Usuario	Administrador
Referencia al caso de uso	Administración de conductores.
Nombre	Creación, listado, actualización y eliminación de conductores.
Inicialización	Iniciar sesión en la parte administrativa del portal web.
Salida esperada	Cambios (inserción, actualización, eliminación) de conductores almacenados en el portal web.
Propósito	Realizar cambios directamente en la base de datos ubicado en el servidor a partir de páginas web.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. El administrador ingresa a la url de administración del portal web e inicia sesión con los datos correctos. 2. El administrador debe seleccionar la opción conductores del menú. 3. El administrador visualizará un listado de todos los conductores registrados.

	4. El administrador puede optar por crear un nuevo conductor, actualizarlo o eliminarlo.
Salida obtenida	Inserción, actualización o eliminación de un registro en la base de datos.

Tabla 19 Caso de prueba CPF-05

Caso de prueba “Creación, listado, actualización y eliminación de destinos.”

CPF-06	
N° caso de prueba	6
Usuario	Administrador
Referencia al caso de uso	Administración de destinos.
Nombre	Creación, listado, actualización y eliminación de destinos.
Inicialización	Iniciar sesión en la parte administrativa del portal web.
Salida esperada	Cambios (inserción, actualización, eliminación) de destinos almacenados en el portal web.
Propósito	Realizar cambios directamente en la base de datos ubicado en el servidor a partir de páginas web.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. El administrador ingresa a la url de administración del portal web e inicia sesión con los datos correctos. 2. El administrador debe seleccionar la opción destinos del menú. 3. El administrador visualizará un listado de todos los destinos registrados. 4. El administrador puede optar por crear un nuevo destino, actualizarlo o eliminarlo.
Salida obtenida	Inserción, actualización o eliminación de un registro en la base de datos.

Tabla 20 Caso de prueba CPF-06

Caso de prueba “Creación, listado, actualización y eliminación de andenes.”

CPF-07	
N° caso de prueba	7
Usuario	Administrador
Referencia al caso de uso	Administración de andenes.
Nombre	Creación, listado, actualización y eliminación de andenes.
Inicialización	Iniciar sesión en la parte administrativa del portal web.

Salida esperada	Cambios (inserción, actualización, eliminación) de andenes en el portal web.
Propósito	Realizar cambios directamente en la base de datos ubicado en el servidor a partir de páginas web.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. El administrador ingresa a la url de administración del portal web e inicia sesión con los datos correctos. 2. El administrador debe seleccionar la opción anden del menú. 3. El administrador visualizará un listado de todos los andenes registrados. 4. El administrador puede optar por crear un nuevo anden, actualizarlo o eliminarlo.
Salida obtenida	Inserción, actualización o eliminación de un registro en la base de datos.

Tabla 21 Caso de prueba CPF-07

Caso de prueba “Creación, listado, actualización y eliminación de frecuencias de viaje.”

CPF-08	
Nº caso de prueba	8
Usuario	Administrador
Referencia al caso de uso	Administración de frecuencias de viaje.
Nombre	Creación, listado, actualización y eliminación de frecuencias de viaje.
Inicialización	Iniciar sesión en la parte administrativa del portal web.
Salida esperada	Cambios (inserción, actualización, eliminación) de frecuencias de viaje almacenados en el portal web.
Propósito	Realizar cambios directamente en la base de datos ubicado en el servidor a partir de páginas web.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. El administrador ingresa a la url de administración del portal web e inicia sesión con los datos correctos. 2. El administrador debe seleccionar la opción frecuencias del menú. 3. El administrador visualizará un listado de todas las frecuencias de viaje registradas. 4. El administrador puede optar por crear una nueva frecuencia de viaje, actualizarla o eliminarla.
Salida obtenida	Inserción, actualización o eliminación de un registro en la base de datos.

Tabla 22 Caso de prueba CPF-08

Caso de prueba “Creación, listado, actualización y eliminación de servicios.”

CPF-09	
N° caso de prueba	9
Usuario	Administrador
Referencia al caso de uso	Administración de servicios del terminal terrestre.
Nombre	Creación, listado, actualización y eliminación de servicios.
Inicialización	Iniciar sesión en la parte administrativa del portal web.
Salida esperada	Cambios (inserción, actualización, eliminación) de servicios en el portal web.
Propósito	Realizar cambios directamente en la base de datos ubicado en el servidor a partir de páginas web.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. El administrador ingresa a la url de administración del portal web e inicia sesión con los datos correctos. 2. El administrador debe seleccionar la opción servicios del menú. 3. El administrador visualizará un listado de todos los servicios registrados. 4. El administrador puede optar por crear un nuevo servicio, actualizarlo o eliminarlo.
Salida obtenida	Inserción, actualización o eliminación de un registro en la base de datos.

Tabla 23 *Caso de prueba CPF-09*

Caso de prueba “Creación, listado, actualización y eliminación de comunicados/noticias del terminal terrestre.”

CPF-10	
N° caso de prueba	10
Usuario	Administrador
Referencia al caso de uso	Administración de comunicados/noticias del terminal terrestre.
Nombre	Creación, listado, actualización y eliminación de comunicados/noticias.
Inicialización	Iniciar sesión en la parte administrativa del portal web.
Salida esperada	Cambios (inserción, actualización, eliminación) de comunicados/noticias en el portal web.
Propósito	Realizar cambios directamente en la base de datos ubicado en el servidor a partir de páginas web.

Procedimiento de prueba	<ol style="list-style-type: none"> 1. El administrador ingresa a la url de administración del portal web e inicia sesión con los datos correctos. 2. El administrador debe seleccionar la opción comunicados/noticias del menú. 3. El administrador visualizará un listado de todos los comunicados y noticias registrados. 4. El administrador puede optar por crear un nuevo comunicado/noticia, actualizarlo o eliminarlo.
Salida obtenida	Inserción, actualización o eliminación de un registro en la base de datos.

Tabla 24 Caso de prueba CPF-10

Caso de prueba “Creación, listado, actualización y eliminación de recursos.”

CPF-11	
N° caso de prueba	11
Usuario	Administrador
Referencia al caso de uso	Administración de recursos.
Nombre	Creación, listado, actualización y eliminación de recursos.
Inicialización	Iniciar sesión en la parte administrativa del portal web.
Salida esperada	Cambios (inserción, actualización, eliminación) de recursos en el portal web.
Propósito	Realizar cambios directamente en la base de datos ubicado en el servidor a partir de páginas web.
Procedimiento de prueba	<ol style="list-style-type: none"> 1. El administrador ingresa a la url de administración del portal web e inicia sesión con los datos correctos. 2. El administrador debe seleccionar la opción recursos del menú. 3. El administrador visualizará un listado de todos los recursos registrados. 4. El administrador puede optar por crear un nuevo recurso, actualizarlo o eliminarlo.
Salida obtenida	Inserción, actualización o eliminación de un registro en la base de datos, creación o eliminación de archivos de la carpeta uploads en el servidor.

Tabla 25 Caso de prueba CPF-11

4.7.1.3. Pruebas no funcionales

Los casos de prueba no funcionales se los ha ubicado en tablas, cada campo se detalla a continuación:

Código: Identificador del caso de prueba para referencias. El formato definido es: CPNF-01

Número de caso de prueba: Determina el orden del caso de prueba.

Nombre: Nombre del caso de prueba.

Propósito: Texto que describe al caso de prueba y su finalidad.

Procedimiento de prueba: Secuencia de pasos para completar la prueba.

Salida obtenida: Descripción del comportamiento del software después de la ejecución de los casos de pruebas.

A continuación, se mencionan los casos de prueba no funcionales definidos:

CPNF-01	
N° caso de prueba	1
Nombre	Seguridad
Propósito	Definir la privacidad de la información mostrada en el portal web.
Procedimiento de prueba	<ol style="list-style-type: none">1. Se intentó ingresar a los módulos de mantenimiento del portal web solo conociendo la url.2. Si no ha iniciado sesión aún, redirecciona a la página login, caso contrario muestra el resultado de la url solicitada.
Salida obtenida	La seguridad del portal web está establecida mediante autenticación de usuario y contraseña, de esta manera se restringe al acceso a personas no autorizadas.

Tabla 26 Caso de prueba CPNF-01

CPNF-02	
N° caso de prueba	2
Nombre	Diseño responsive
Propósito	Definir un diseño que se adapte a diferentes tamaños de pantalla.
Procedimiento de prueba	<ol style="list-style-type: none">1. Se ingresa a la url del portal web haciendo uso del navegador Google Chrome.

	2. Se hace uso de la opción Device Mode en Google Chrome para emular diferentes tamaños comúnmente encontrados en dispositivos móviles.
Salida obtenida	El portal web es 100% responsive, debiéndose al uso de un framework que se encarga de ello.

Tabla 27 Caso de prueba CPNF-02

4.7.1.4. Pruebas de Stress

Estas pruebas fueron realizadas utilizando el hardware descrito en la fase MIDAS/FC del presente capítulo, que fue el mismo con el que se desarrolló dicho software, el cual es de vital importancia, ya que el rendimiento del sistema está estrictamente ligado con el hardware de la computadora que realizará la función de servidor, ya que éste determina el desempeño del mismo.

Para la realización de las pruebas de stress se llevaron a cabo algunas tareas y se inspeccionó que el sistema funcionara y respondiera correctamente a todas las tareas, para estas pruebas se midieron dos tipos de casos:

Caso crítico: en el cual se le lleva al sistema a una situación extrema, donde se prueba realmente sus capacidades y donde se comienza a ver que puede presentar algunos errores en las peticiones de los diferentes usuarios y al llevar a cabo ciertas tareas.

Caso óptimo: en donde el sistema funciona perfectamente sin presentar ningún contratiempo ni problema; donde las peticiones de los diferentes usuarios son procesados correctamente.

Durante las pruebas se midieron 3 parámetros, el número máximo de usuarios que soporta el sistema, el tiempo promedio que tarda en procesar una petición, y el margen de error que existía para cada uno de los casos que se eligieron.

El plan de pruebas se realizó utilizando el árbol de configuración (véase **Figura 66**), para que se llevaran a cabo 7 peticiones por usuario, las cuales estaban dadas en el siguiente orden:

- Desplegar la página de inicio
- Desplegar el directorio de cooperativas
- Desplegar las frecuencias interprovinciales

- Desplegar la página de contacto
- Desplegar las tarifas de transporte
- Inicio de sesión en el sistema, que consta de ingresar usuario y contraseña y desplegar la página que sirve como panel de administración.
- Desplegar la página para la administración de servicios

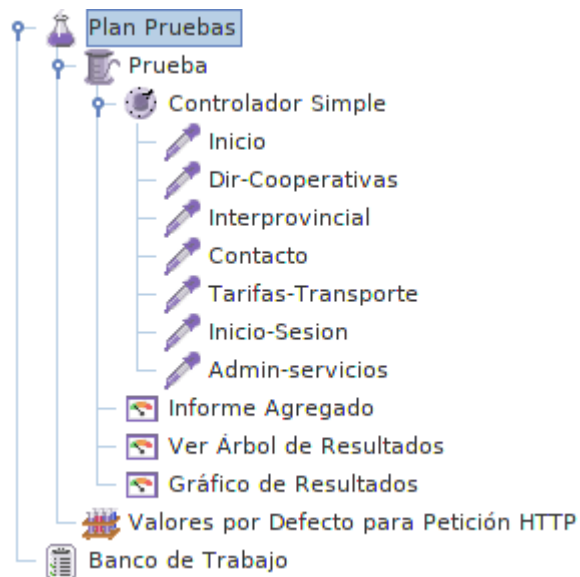


Figura 66 Configuración de JMeter

Los usuarios se conectan al mismo tiempo, cada uno con una sesión diferente y llevan a cabo estas 7 actividades, para lo cual se registraron los tiempos de respuesta y se tomaron algunos datos estadísticos que proporciona JMeter en el Informe Agregado.

Para encontrar el número correcto de usuarios después de varias pruebas incrementales, es decir se comenzó probando para un número de usuarios reducidos, y se fue aumentando para medir el desempeño del sistema. Hasta que se encontró que el caso óptimo fue con 600 usuarios, y el caso crítico con 625. A pesar de que la diferencia que existe entre estos dos números es muy pequeña, para 625 usuarios se comienza a reportar errores mínimos pero que sí afecta a algunos de los usuarios que se encuentran utilizando el sistema.

El informe agregado contiene varias columnas y renglones, los renglones representan cada una de las 7 peticiones o actividades que se realizaron y las columnas representan ciertas medidas dentro de la prueba:

Etiqueta: es la actividad que se desempeña, el request o la petición.

Muestras: es la cantidad de veces que se realizó la actividad (una vez por cada usuario).

Media: el promedio o media aritmética del tiempo en milisegundos.

Mediana: del tiempo en milisegundos.

Min: tiempo mínimo de todos los request de ese tipo.

Max: tiempo máximo de todos los request de ese tipo.

Porcentaje de error: en el cual se muestra el porcentaje de los request fallidos.

Rendimiento: está medido en request/segundo.

KB/Sec: medida de velocidad de kilobytes/sec.

En la **Figura 67** la media total fue de 3652 ms, esto quiere decir que el sistema en promedio se tardó en responder 3.6 segundos, lo cual es un tiempo bastante bueno tomando en cuenta que son 625 usuarios conectados al mismo tiempo.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimi...	Kb/sec
Inicio	625	12420	11145	30069	32949	37885	7	38035	0,16%	15,3/sec	833,9
Dir-Cooperativas	625	7652	5623	20027	32389	37598	186	38499	0,16%	15,1/sec	1588,8
Interprovincial	625	1774	288	870	12378	29730	4	38057	0,16%	15,2/sec	598,0
Contacto	625	560	312	643	866	6998	6	29289	0,00%	15,1/sec	553,0
Tarifas-Transporte	625	565	241	579	686	9004	3	37713	0,00%	15,1/sec	538,9
Inicio-Sesion	625	2577	1701	6135	7056	7737	152	39131	0,00%	14,8/sec	134,6
Admin-servicios	625	14	6	35	58	148	1	259	0,00%	14,9/sec	129,7
Total	4375	3652	407	11913	23083	35145	1	39131	0,07%	97,5/sec	4028,1

Figura 67 Informe agregado para la prueba de 625 usuarios

En la figura anterior también se puede apreciar que en las actividades Inicio, Dir-Cooperativas e Interprovincial existe un porcentaje de error del 0.16% para cada uno, estos representan 3 usuarios de los 625 con los que se realizó la prueba. Estos 3 usuarios obtuvieron una página de error al intentarse conectar al sistema (véase **Figura 68**).

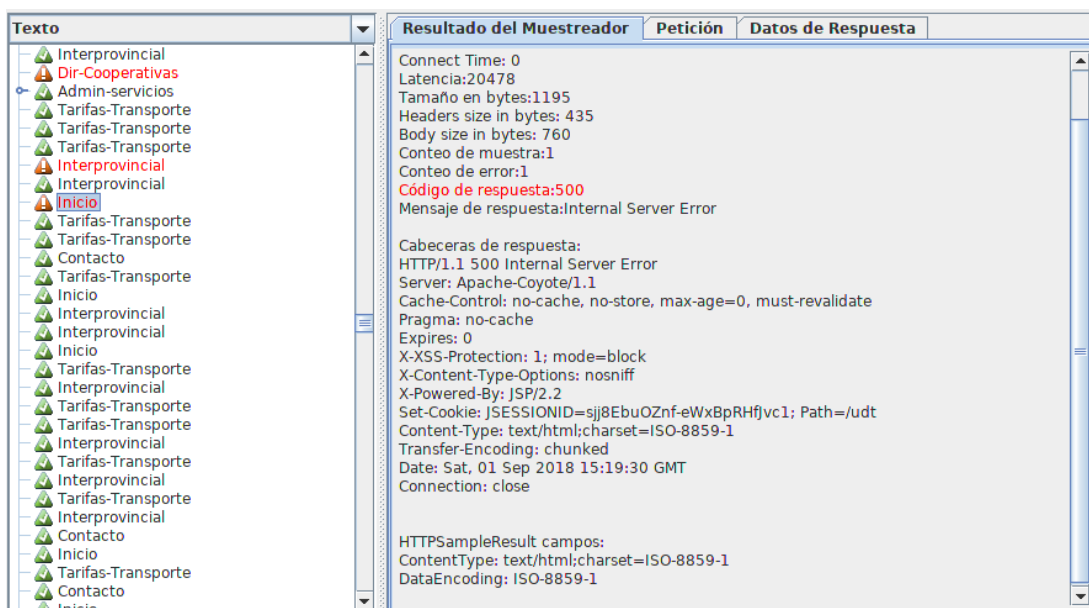


Figura 68 Resultado del procesamiento de peticiones

Para el caso óptimo se utilizaron 600 usuarios y como se muestra en la **Figura 69** no hay porcentaje de error y es un mejor rendimiento del sistema. Además, se puede apreciar que el porcentaje de error en todas las peticiones es de 0.00%, esto indica que no fue desplegada ninguna página de error, ya que todas las peticiones fueron respondidas de manera adecuada y correcta. Si se toma la media de la **Figura 67**, que es de 3652 ms, y se compara con la de la **Figura 69**, que es de 3255, se puede apreciar que existe una mínima diferencia de 0.4 segundos, lo cual es un poco menos de tiempo. Por lo que se considera como un tiempo de respuesta muy pequeño, lo que clasifica al sistema como rápido.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimi...	Kb/sec
Inicio	600	11362	10266	27298	28323	30024	5	31140	0,00%	17,6/sec	961,0
Dir-Cooperativas	600	6878	5834	15795	26279	33234	256	34011	0,00%	16,1/sec	1696,3
Interprovincial	600	1484	337	849	10531	25924	5	27542	0,00%	16,2/sec	640,1
Contacto	600	722	325	626	780	15703	6	21948	0,00%	16,1/sec	591,1
Tarifas-Transporte	600	523	300	615	736	8628	2	28882	0,00%	16,1/sec	574,4
Inicio-Sesion	600	1808	1832	2878	3083	3731	111	13677	0,00%	15,4/sec	140,0
Admin-servicios	600	10	5	25	40	93	1	220	0,00%	16,0/sec	139,1
Total	4200	3255	442	10580	20359	28588	1	34011	0,00%	105,5/sec	4360,4

Figura 69 Informe agregado para la prueba de 600 usuarios

Como se puede apreciar en los informes agregados los tiempos de respuesta son pequeños, ese tiempo también ya incluye el acceso a la base de datos por lo que se considera que el sistema se tarda muy poco tiempo en realizar el acceso a la base de datos para recuperar la información necesaria.

CAPITULO 5

Conclusiones y recomendaciones

5.1. CONCLUSIONES

- ✓ La metodología MIDAS al ser iterativo proporciona pequeñas versiones del proyecto en cada iteración, ya sea un documento o prototipo de un producto software. Además, brinda prioridad a las necesidades del cliente puesto que, esta metodología toma principios fundamentales de metodologías ágiles como eXtreme Programming XP.
- ✓ La utilización de un estándar de especificación de requerimientos de software, permite que la información analizada tenga un formato que facilite la explicación de las necesidades y requisitos de la solución tecnológica.
- ✓ Hibernate, como framework de persistencia de datos, ha facilitado la realización de las operaciones CRUD, disminuyendo gran cantidad de código y aprovechando la programación orientada a objetos, además de proporcionar cierta independencia del motor de base de datos.
- ✓ El uso de Eclipse, como IDE de programación, conjuntamente con la herramienta Maven que dispone de plantillas y gestión de librerías que se requieran incluir al proyecto, ha simplificado la creación de un proyecto tipo aplicación web. Proporcionando la estructura básica, con las carpetas que se necesitan, para organizar el proyecto de una manera adecuada.
- ✓ Mediante el portal web se genera un punto de encuentro en el cual los visitantes podrán informarse acerca de las actividades de la institución a través de las noticias, comunicados.

5.2. RECOMENDACIONES

- ✓ Para que exista una mayor compatibilidad, entre una aplicación desarrollada con el framework hibernate y las versiones de java, se recomienda usar el mapping haciendo uso de archivos xml, puesto que si usamos únicamente clases java para definir el mapping estamos limitando a que la aplicación deba usarse con la versión java 1.7 o superior.
- ✓ Es recomendable utilizar navegadores como Firefox desde la versión 2.5 en adelante para la administración del contenido del portal y cualquier navegador para los usuarios que visiten el portal, teniendo en cuenta el soporte JavaScript ya que el portal tiene partes que está desarrollado con este lenguaje de programación.
- ✓ Es recomendable utilizar un diseño responsivo al desarrollar un proyecto orientado a la web. Ya que con esto se ofrece a los visitantes una mejor experiencia de usuario, dado que en la actualidad muchas personas hacen uso de sus dispositivos móviles para navegar en internet.
- ✓ A pesar de que en el desarrollo de este proyecto se ha empleado Spring Framework no se ha utilizado todo el potencial del mismo. Una opción recomendada es usar Spring Boot, una utilidad del framework que permite simplificar más aún el desarrollo del proyecto ya que con esto las configuraciones de Spring que se hacen manualmente, aquí lo hace de manera automática con archivos de configuración más legibles y ordenados.
- ✓ Se recomienda realizar respaldos periódicos de la base de datos para preservar la información de la Institución y sobre todo mantener actualizada la información que contiene y esta publicada en el Portal Web.

Bibliografía

- [1] M. J. Paz Gaibor, «Terminal Terrestre Virtual de Guayaquil,» 2015. [En línea]. Available: <http://repositorio.ug.edu.ec/bitstream/redug/13062/1/MILTON%20JAVIER%20PAZ%20GAIBOR.pdf>.
- [2] H. Castro, «Portal Web del Ilustre Municipio del Cantón de San Pedro de Huaca,» 2011. [En línea]. Available: <http://repositorio.utn.edu.ec/handle/123456789/609>.
- [3] E. P. Peñaloza Castro, «Portal Web para la difusión de Servicios que proporcionan a las personas con capacidades especiales que forman parte de las Organizaciones que trabajan con el Parlamento Gente del Gobierno Provincial de Tungurahua,» 2010. [En línea]. Available: <http://repositorio.uta.edu.ec/bitstream/123456789/147/1/t561s.pdf>.
- [4] J. A. Díaz Pilco, «Portal Web Dinámico para la empresa Calzamatrix de la ciudad de Ambato,» 2011. [En línea]. Available: <http://repositorio.uta.edu.ec/bitstream/123456789/123/1/t584s.pdf>.
- [5] E. G. Pilco Chiluiza , «Portal Web Dinámico para la difusión y publicación de información de la sociedad de Ingenieros del Ecuador Zona Centro,» 2010. [En línea]. Available: <http://repositorio.uta.edu.ec/bitstream/123456789/156/1/t555s.pdf>.
- [6] C. Mateu, «Desarrollo de Aplicaciones Web,» 2004. [En línea]. Available: <http://libros.metabiblioteca.org/bitstream/001/591/1/004%20Desarrollo%20de%20aplicaciones%20web.pdf>.
- [7] B. Ivanov Bonev, «Introducción al lenguaje Java,» 2012. [En línea]. Available: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>.
- [8] Á. M. Valbuena Aponte, «Guía comparativa de frameworks para los lenguajes Html 5, Css Y Javascript para el desarrollo de aplicaciones web,» 2014. [En línea]. Available: <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/4577/00676V865.pdf;sequence=1>.
- [9] J. Gutiérrez, «¿Qué es un framework web?,» [En línea]. Available: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
- [10] «Spring Framework,» [En línea]. Available: <https://spring.io/>. [Último acceso: 12 04 2018].
- [11] A. Guzmán Fontecilla, «Spring MVC – Básico parte I,» Chile, 2017.
- [12] «Tutoriales de Programación Java,» [En línea]. Available: <http://www.javatutoriales.com/2015/12/spring-mvc-parte-1-configuracion.html>. [Último acceso: 22 05 2018].
- [13] R. Kotha, «How MVC belongs to only Presentation Layer,» [En línea]. Available: <https://java2practice.com/2012/09/16/how-mvc-belongs-to-only-presentation-layer/>. [Último acceso: 29 05 2018].

- [14] R. Borja Lázaro, «Token con caducidad en Spring Security,» [En línea]. Available: <https://www.adictosaltrabajo.com/tutoriales/spring-security-timeout-token/>. [Último acceso: 29 05 2018].
- [15] «Chuwiki,» [En línea]. Available: http://chuwiki.chuidiang.org/index.php?title=Patr%C3%B3n_DAO. [Último acceso: 18 06 2018].
- [16] Wikipedia, «Froala Editor,» [En línea]. Available: https://en.wikipedia.org/wiki/Froala_Editor. [Último acceso: 2018 05 2018].
- [17] C. Marin, «Integración Apache Tiles con Spring MVC,» [En línea]. Available: <http://acodigo.blogspot.com/2017/05/integracion-apache-tiles-con-spring-mvc.html>. [Último acceso: 24 05 2018].
- [18] A. Pérez García, «Maven, nunca antes resultó tan fácil compilar, empaquetar, ...» 9 16 2006. [En línea]. Available: <https://www.adictosaltrabajo.com/tutoriales/maven/>.
- [19] Y. Duarte Perez, «Tutorial General Java Como crear Aplicaciones Enterprise,» 2011.
- [20] O. A. Cando Cando, «Análisis Hibernate como tecnología de persistencia de objetos sobre bases de datos relacionales en aplicaciones empresariales: Caso Práctico control de bienes del Gobierno Municipal de Carlos Julio Arosemena Tola,» 2013. [En línea]. Available: <http://dspace.esepoch.edu.ec/bitstream/123456789/2427/1/18T00501.pdf>.
- [21] Red Hat, «Hibernate Community Documentation,» [En línea]. Available: <https://docs.jboss.org/hibernate/orm/3.5/reference/es-ES/html/architecture.html>. [Último acceso: 25 05 2018].
- [22] Red Hat, «Hibernate ORM,» [En línea]. Available: <http://hibernate.org/orm/documentation/5.3/>. [Último acceso: 25 05 2018].
- [23] «Bootstrap, un framework para diseñar portales web,» acensTechnologies, 2016. [En línea]. Available: <https://www.acens.com/wp-content/images/2016/10/bootstrap-framework-acens-wp.pdf>.
- [24] C. A. Villarreal Fuentes, «¿Qué es jQuery?,» 2013. [En línea]. Available: http://www.northware.mx/wp-content/uploads/2013/11/Noviembre_Que_es_jQuery__Northware.pdf.
- [25] R. Gutiérrez Sosa, «¿Qué es SGBD?,» [En línea]. Available: <https://rommgusa.files.wordpress.com/2011/02/que-es-sgbd-tarea1.pdf>.
- [26] J. Santamaría y J. Hernández, «Sql Server vs MySQL,» [En línea]. Available: <https://iessanvicente.com/colaboraciones/sqlserver.pdf>.
- [27] «Eclipse IDE,» Asociación Española de Programadores Informáticos, [En línea]. Available: <https://www.asociacionaepi.es/eclipse-ide/>.
- [28] N. Chaudhary, S. Gilda, D. Mison, M. Stanley-Jones, L. Costi, V. Goyal, S. Mumford, K. Verma, R. Dickenson, E. Logue, D. Ryan y T. Wells, «Plataforma de aplicaciones empresariales JBoss 6.4 Guía de inicio,» 2015. [En línea]. Available: https://access.redhat.com/documentation/es-ES/JBoss_Enterprise_Application_Platform/6.4/pdf/Getting_Started_Guide/JBoss_Enterprise_Application_Platform-6.4-Getting_Started_Guide-es-ES.pdf.

- [29] M. J. Lamarca Lapuente, «Modelo OOHDM,» 2013. [En línea]. Available: <http://www.hipertexto.info/documentos/ooohdm.htm>.
- [30] M. J. Escalona Cuaresma, «Metodologías para el desarrollo de sistemas de información global: análisis comparativo y propuesta,» 2001. [En línea]. Available: <http://www.lsi.us.es/docs/informes/EstadoActual.pdf>.
- [31] B. Vela Sánchez , «MIDAS/DB: Una Metodología basada en Modelos para el Desarrollo de la Dimensión Estructural de Sistemas de Información Web,» 2003. [En línea]. Available: <http://repositorio.esпам.edu.ec/bitstream/42000/58/1/Gema%20Velez-Cristhian%20Ramirez.pdf>.
- [32] D. Mínguez Sanz y E. J. García Morales, «Metodologías para el Desarrollo de Aplicaciones Web: UWE,» [En línea]. Available: <https://jorgeportella.files.wordpress.com/2011/11/analisis-diseo-y-desarrollodeaplicacionesweb.pdf>.
- [33] R. Sperko, Java Persistence for Relational Databases, Apress, 2003.

Anexos y Apéndices

ANEXO 1

Manual de administración del portal web

Para la administración del portal web se debe de ingresar los datos de usuario y contraseña (véase **Figura 70**).



Figura 70 Ingreso a la administración del portal web

Una vez dentro del portal web de administración se podrá tener acceso a las diferentes opciones para la administración de cada una de ellas.

Información Institucional

Aquí podemos modificar la información correspondiente a la institución, para lo que contamos con tres pestañas “Básico”, “De contacto y ubicación”, “Estratégico” (véase **Figura 71**).

The screenshot shows the 'UDT Admin' interface. On the left is a navigation menu with categories like 'Panel de Administración', 'Público', 'Áreas de transferencia', 'Comunicados/Noticias', 'Directorio Cooperativas', 'Institución', 'Links de interés', 'Preguntas frecuentes', 'Glosario de términos', 'Recursos', 'Servicios', 'Tarifas de transporte', 'Transportistas', 'Frecuencias', 'Ingresos', and 'Áreas y equipos'. The main content area is titled 'Panel de Administración > Institución' and contains a form titled 'Datos de la Institución'. The form has three tabs: 'Básico', 'De contacto y ubicación', and 'Estratégico'. The 'De contacto y ubicación' tab is selected. It contains the following fields: 'Teléfono' with the value 'Telefax (03) 2522669 - Admin (03) 2 522367'; 'Correo' with the value 'terminal@ambato.gob.ec'; 'Dirección' with the value 'Av. Colombia 332 y Paraguay'; and 'Ubicación' with the coordinates '-1.235792' and '-78.616466'. A blue 'Guardar' button is located at the bottom of the form.

Figura 71 Administración de información institucional

Consideraciones importantes

La pestaña básico contiene campos para nombre, abreviatura, slogan, horario de atención, logo institucional.

La pestaña de contacto y ubicación contiene campos para teléfono, correo electrónico, dirección, y coordenadas latitud y longitud para que en la parte pública sea posible vincularlo con google maps.

La pestaña estratégico contiene campos para misión, visión, valores institucionales, acerca de, éste último contiene el código html de un mapa interactivo de las instalaciones del terminal (véase **Figura 72**).



Figura 72 Mapa interactivo de las instalaciones del terminal

Áreas de transferencia

Aquí podemos modificar la información correspondiente a las áreas de transferencia, para lo que contamos con los botones de “Agregar Área Transferencia”, “Editar”, “Eliminar” y una caja de texto para filtrar la información por cualquier campo que se muestra en la tabla (véase **Figura 73**).

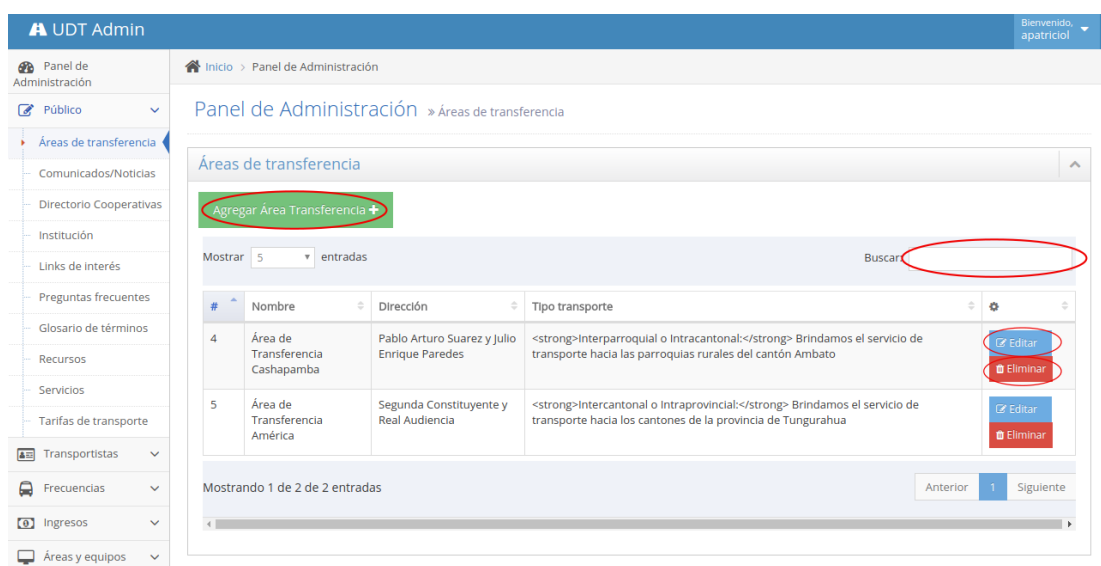


Figura 73 Administración áreas de transferencia

Comunicados y noticias

Aquí podemos modificar la información correspondiente a los comunicados y noticias, para lo que hacemos uso de los botones de “Agregar Comunicado/Noticia” o “Editar” (véase **Figura 74**).

UDT Admin Bienvenido: apatriol

Inicio > Panel de Administración

Panel de Administración > Comunicados y Noticias

Comunicados y Noticias de la Institución

Listado Comunicados y Noticias

Título: TARIFAS DE PARQUEO

Contenido:

Tiempo	Valor
00 - 15 min	Gratis
16 - 30 min	\$ 0.50
31 - 60 min	\$ 3.00

Indicaciones:

- 1.- Pasado los 30 minutos se cobrará \$ 3,00 por hora o fracción hasta las 4 horas.
- 2.- Luego de 4 horas multa por \$ 20,00 y el pago del tiempo respectivo.
- 3.- Una vez registrado su ticket tiene 5 minutos para sacar su vehículo.

Tipo: Comunicado Noticia

Fecha: 2018-03-29

Estado: Publicado No Publicado

Figura 74 Administración de comunicados/noticias

Directorio de cooperativas

Aquí podemos modificar la información correspondiente a las operadoras de transporte, para lo que contamos con los botones de “Agregar Nuevo”, “Editar” (véase **Figura 75**).

Figura 75 Administración directorio de cooperativas

Servicios

Aquí podemos modificar la información correspondiente a los servicios, para lo que contamos con los botones de “Agregar Servicio”, “Editar” (véase **Figura 76**).

Figura 76 Administración servicios

Tarifas de transporte

Aquí podemos modificar la información correspondiente a las tarifas de transporte para destinos tipo interprovinciales, para lo que contamos con los botones de “Agregar tarifa de transporte”, “Editar”, “Eliminar” y una caja de texto para filtrar la información por cualquier campo que se muestra en la tabla (véase **Figura 77**).

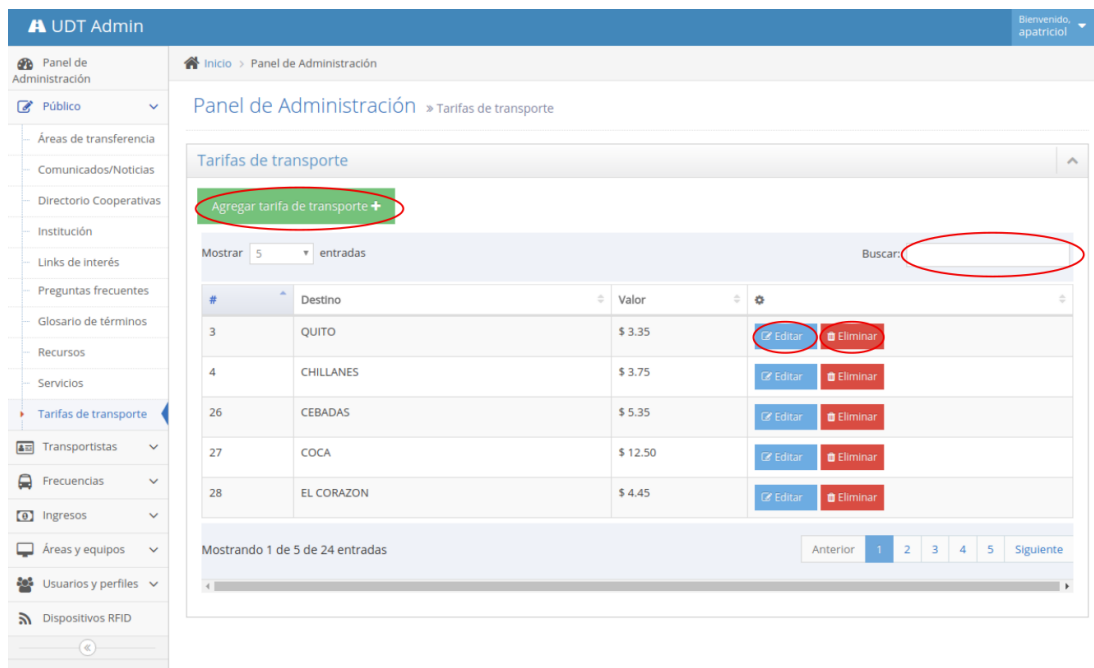


Figura 77 Administración tarifas de transporte

Consideraciones importantes

Los valores de pasajes deben ser en base a las tarifas establecidas por la Agencia Nacional de Tránsito (ANT).

Recursos

Aquí podemos gestionar los recursos del portal web es decir las imágenes, archivos, videos, audios, para lo que contamos con los botones de “Agregar Recurso”, “Editar”, “Eliminar” y una caja de texto para filtrar la información por cualquier campo que se muestra en la tabla (véase **Figura 78**).

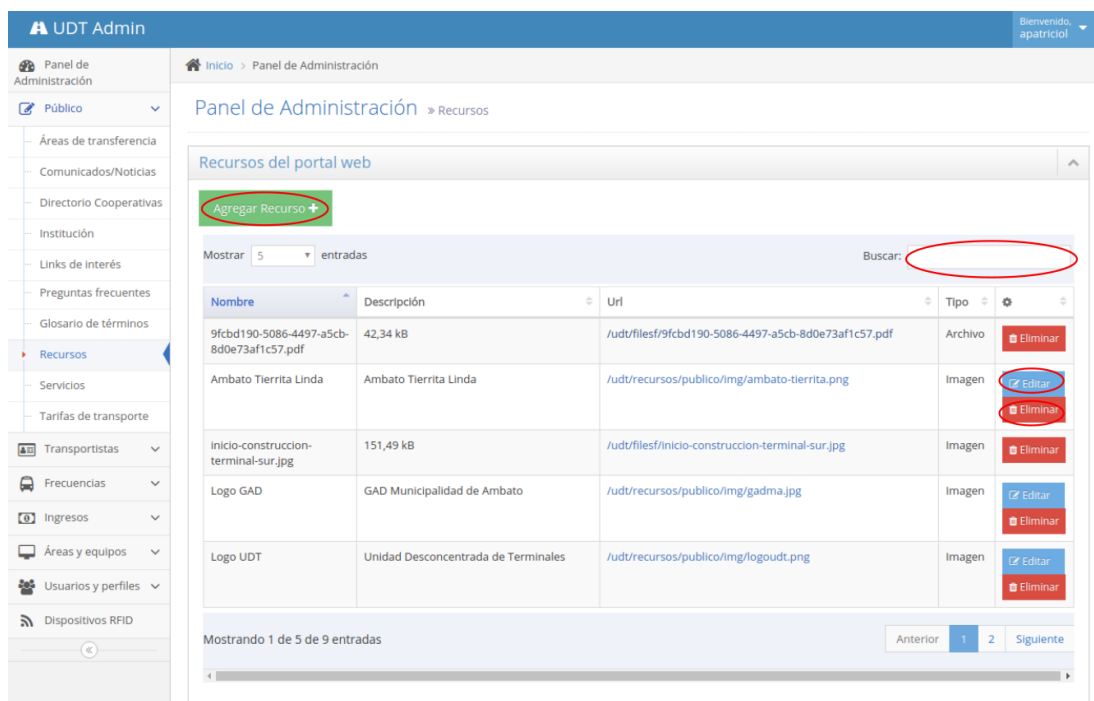


Figura 78 Administración recursos del portal web

Este módulo muestra incluso aquellos recursos que fueron subidos al servidor desde el editor de texto (véase **Figura 79**).

Uso del editor de texto

Para una mejor experiencia de usuario se ha implementado un editor de texto para redactar contenido del portal web del mismo que se detallan algunas de sus funciones.

Insertar imágenes

Para insertar imágenes hacemos uso de la opción cuyo ícono es 

las imágenes pueden provenir desde una url o desde el equipo

local con esta opción las imágenes se suben automáticamente a la carpeta upload del servidor donde reposaran hasta que el usuario decida eliminarlas también podemos seleccionar aquellas imágenes que previamente han sido subidas al servidor.

Los formatos permitidos por este medio son: gif, jpeg, jpg, png, svg, blob.

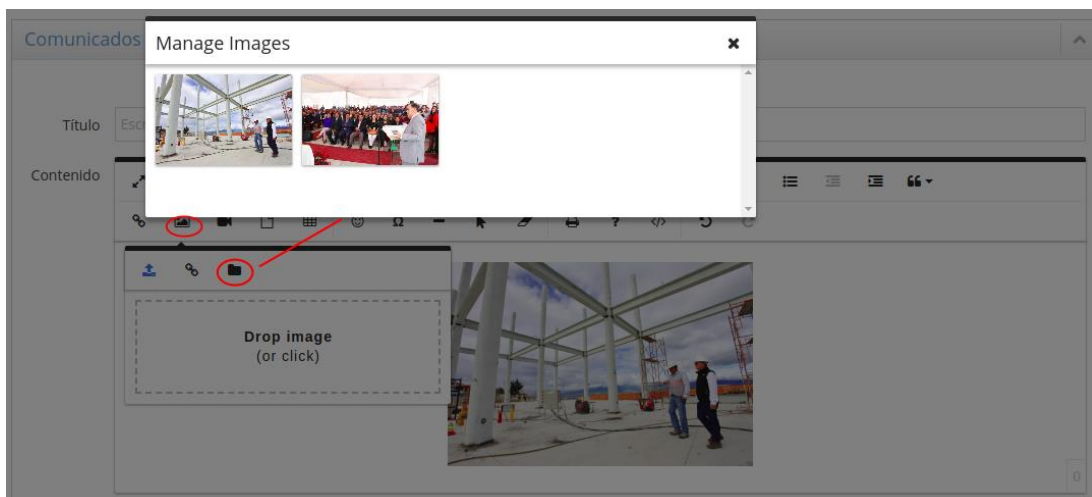


Figura 79 Editor de texto froala – opción imágenes

Esta opción permite una administración completa de imágenes pues permite subir y eliminar imágenes.

Insertar archivos



Para referenciar archivos dentro del editor de texto hacemos uso de la opción cuyo ícono es un archivo, los archivos se suben desde el equipo local con esta opción los archivos se suben automáticamente a la carpeta upload del servidor donde reposaran hasta que el usuario decida eliminarlas.

A diferencia de la opción insertar imágenes que permite una administración total de imágenes ésta únicamente permite subir los archivos al servidor para eliminarlos habrá que hacer uso del módulo recursos.

Los formatos permitidos por este medio son: txt, pdf, doc

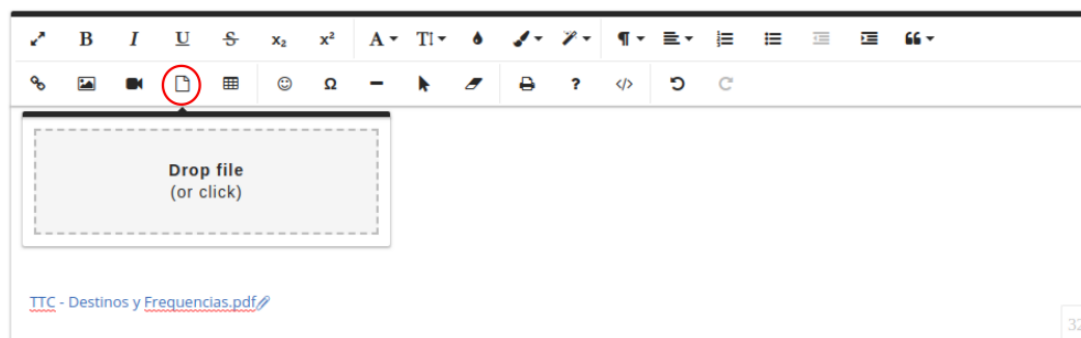



Figura 80 Editor de texto froala – opción archivos

Subir vídeos

Para insertar videos hacemos uso de la opción cuyo ícono es los  videos pueden ser desde una url, código embebido o desde el equipo local con esta opción los videos se suben automáticamente a la carpeta upload del servidor donde reposaran hasta que el usuario decida eliminarlas.

A diferencia de la opción insertar imágenes que permite una administración total de imágenes ésta únicamente permite subir los videos al servidor para eliminarlos habrá que hacer uso del módulo recursos.

Los formatos permitidos por este medio son: mp4, webm, ogg

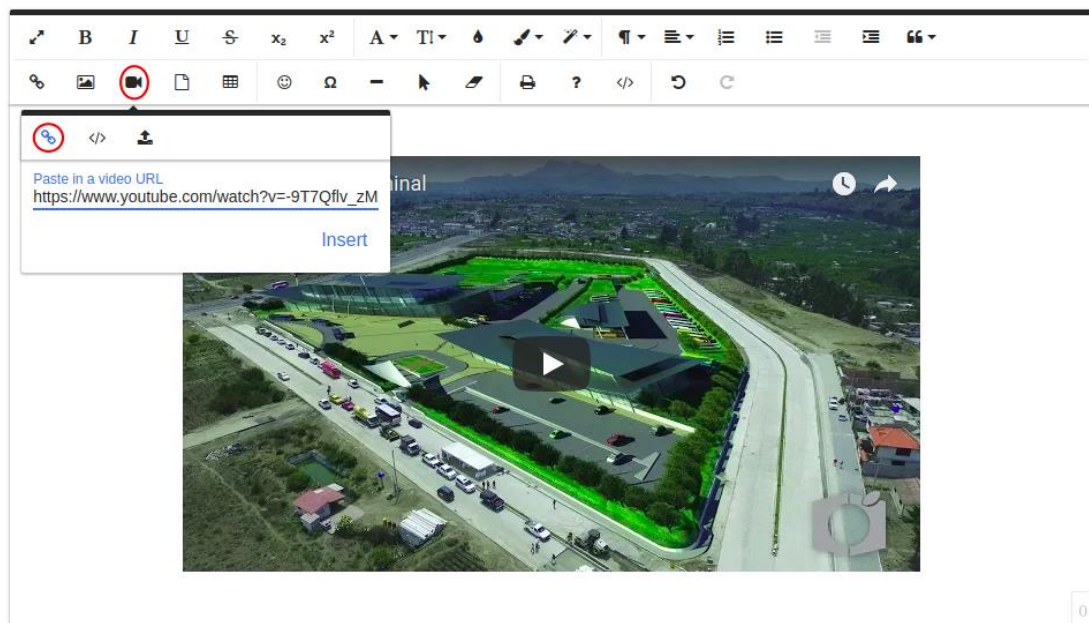


Figura 81 Editor de texto froala – opción videos

Usuarios

Aquí podemos gestionar los usuarios que tienen acceso para administrar el portal web, para lo que contamos con los botones de “Agregar Usuario”, “Editar” (véase **Figura 82**).

UDT Admin Bienvenido, apatriciol

Inicio > Panel de Administración

Panel de Administración > Usuarios

Usuarios

[← Listado Usuarios](#)

CI/RUC: 1803438587

Nombre: PATRICIO LANDA

Usuario: apatriciol

Actualizar contraseña

Contraseña:

Repita Contraseña:

Mostrar contraseña

Perfil: WEB_ADMIN

Habilitado

Figura 82 Administración de usuarios

ANEXO 2

Mapa del sitio (parte pública)

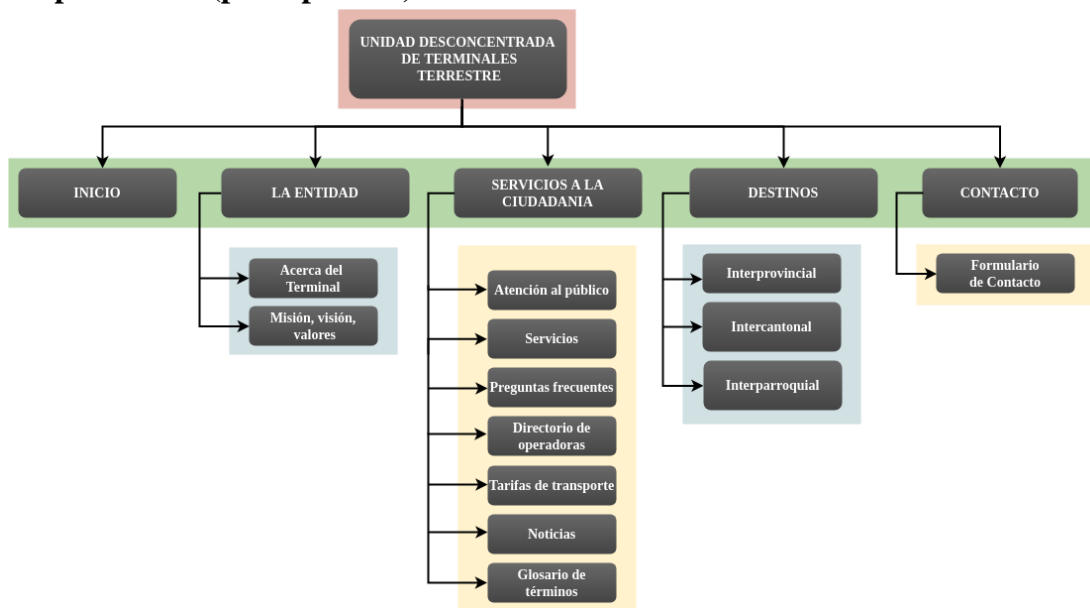


Figura 83 Mapa del sitio (público)

Secciones que integran la parte pública del portal web

Sección inicio

En esta sección se encuentra la página de inicio donde se muestra información relevante a las actividades que realiza la institución como son las operadoras próximas a salir desde el terminal terrestre, las operadoras que prestan sus servicios en el terminal, los comunicados, las noticias más recientes e información relevante de contacto.

Sección de Misión, visión y valores

Aquí los usuarios encontrarán información relacionada con la planificación estratégica que es de primordial importancia ya que esta demuestra identidad, ideología de una institución y permite diferenciarse de otras entidades.

Sección de Acerca del terminal

En esta sección se encuentra un mapa interactivo de las instalaciones de la Unidad Desconcentrada de Terminales este está compuesto por dos pisos los cuales muestran una clara ubicación de los diferentes locales o cubículos con los que cuenta, cada local muestra un detalle acerca de lo que puede encontrar en cada uno de ellos.

Sección de Atención al público

Aquí los usuarios podrán encontrar información de contacto, horarios de atención, ubicación de la institución, así como también de las áreas de transferencia.

Sección de Servicios

Aquí los usuarios encontrarán los diferentes servicios que ofrece la Unidad Desconcentrada de Terminales.

Sección de Preguntas frecuentes

Esta sección del portal web está destinada para aquellas preguntas que comúnmente los usuarios hacen llegar a la administración del terminal.

Sección de Directorio de operadoras

En esta sección los visitantes del portal web podrán encontrar información relacionada con las operadoras de transporte como teléfono, correo electrónico, sitio web, destinos, numero de local y horario de atención para aquellas operadoras que cuentan con locales dentro de las instalaciones del terminal.

Sección de Tarifas de Transporte

Aquí los usuarios podrán encontrar un listado de las tarifas de transporte que la Agencia Nacional de Tránsito ha establecido para destinos interprovinciales, este listado posee una caja de búsqueda para filtrar y ubicar la información de una manera más rápida.

Sección de Noticias

Esta sección del portal web, brinda a los visitantes las noticias que la Unidad Desconcentrada de Terminales ha publicado y quiere da a conocer a la ciudadanía.

Sección de Glosario de Términos

Esta sección del portal web, muestra a los visitantes la definición de ciertos términos que son usados en ciertas partes del portal y con los cuales el usuario no está familiarizado.

Sección de Destinos Interprovincial

En esta sección el usuario podrá consultar las frecuencias interprovinciales con las que cuenta la Unidad Desconcentrada de Terminales previa selección de destino y operadora de transporte.

Sección de Destinos Intercantonal

Aquí los usuarios podrán observar información relacionada con el Área de Transferencia “América” así como también las frecuencias de transporte (destinos, días, horarios) que cubre el área de transferencia mencionado.

Sección de Destinos Interparroquial

Aquí los usuarios podrán observar información relacionada con el Área de Transferencia ubicado en el sector de Cashapamba así como también las frecuencias de transporte (destinos, días, horarios) que cubre el área de transferencia mencionado.

Sección de Contacto

En esta sección el usuario podrá encontrar un formulario por medio cual podrá contactarse de forma directa con la Unidad Desconcentrada de Terminales y expresar alguna inquietud.

Mapa del sitio (parte administrativa)

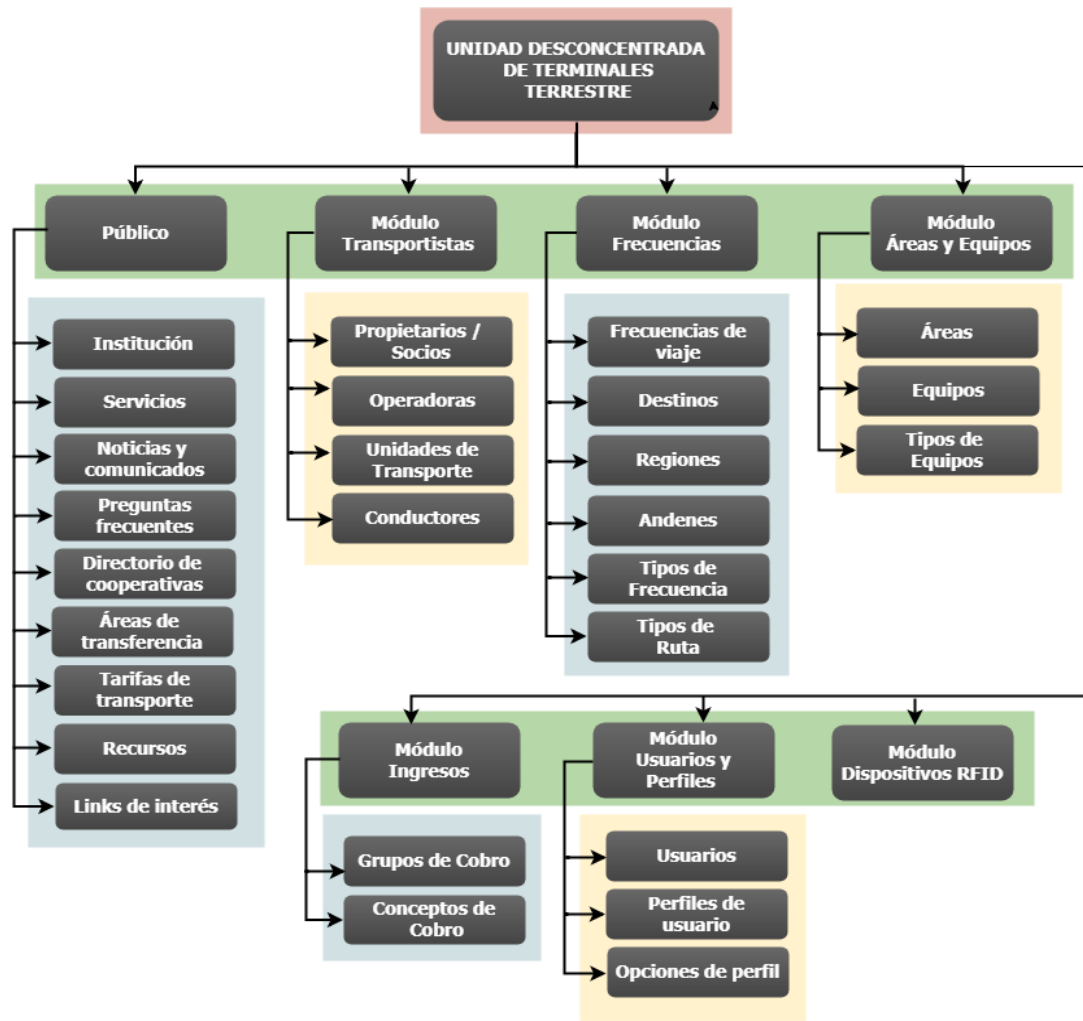


Figura 84 Mapa del sitio (administración)

ANEXO 3

ANÁLISIS DE LAS RESPUESTAS DE LA ENTREVISTA A LA PARTE ADMINISTRATIVA DE LA UDT

Análisis de los Resultados

Para determinar los requerimientos se realizó una entrevista al Administrador y encargado de Recursos Informáticos de la UDT, de lo cual se obtuvo el siguiente resultado:

1.- ¿Qué tipo de servicios ofrecerá el portal web?

Objetivo:

Conocer los servicios que el portal web ofrecerá a la ciudadanía.

Respuesta:

El administrador de la UDT respondió que los servicios que desea ofrecer son netamente de carácter informativos como consulta de tarifas de transporte, un medio de comunicación con el usuario final para la solución de quejas, reclamos o cualquier inquietud, además consulta de horarios, frecuencias.

Análisis:

Los servicios a implantar son los siguientes, consulta de tarifas de transporte, solución de quejas – reclamos – inquietudes por medio de un formulario de contacto, consulta de horarios, frecuencias.

2.- ¿Qué información se colocará en el portal web?

Objetivo:

Conocer el tipo de información que el portal web va a contener.

Respuesta:

El administrador de la UDT respondió que la información que se colocará es acerca de la institución, de contacto, relacionada a noticias, comunicados, directorio de operadoras, mapa de las instalaciones del terminal, preguntas frecuentes, glosario de términos, publicidad, y lo relacionada a cooperativas y próximas salidas de las unidades de transporte.

Análisis:

La información que se colocará en el portal web son los siguientes, acerca de la institución, información de contacto, acerca de las áreas de transferencia, los servicios que brinda la institución a la ciudadanía, directorio de cooperativa, mapa de las instalaciones del terminal, preguntas frecuentes, glosario de términos, próximas salidas, noticias, comunicados y publicidad.

3.- ¿Qué información de tipo administrativa requiere que el portal web permita gestionar?

Objetivo:

Conocer qué tipo de información administrativa a más de la parte pública va a gestionar el portal web.

Respuesta:

El encargado de Recursos Informáticos de la UDT respondió que la información relacionada con el control del terminal terrestre que necesitan que el portal web gestione está relacionada a transportistas, operadoras, unidades de transporte, propietarios, frecuencias de viaje, dispositivos RFID, destinos, andenes, conceptos de cobro, áreas y equipos, usuarios y perfiles.

Análisis:

Las funcionalidades de tipo administrativas a implantar por sugerencia del encargado de Recursos Informáticos son aquellas relacionadas con la información referente a:

- ✓ Propietarios de las unidades de transporte
- ✓ Operadoras de transporte
- ✓ Unidades de transporte
- ✓ Conductores de las unidades de transporte
- ✓ Frecuencias de viaje
- ✓ Tipos de frecuencia de viaje

- ✓ Destinos
- ✓ Regiones del país
- ✓ Andenes de salida de las unidades de transporte
- ✓ Tipos de rutas
- ✓ Grupos de cobro
- ✓ Conceptos de cobro
- ✓ Áreas del terminal terrestre
- ✓ Equipos
- ✓ Tipos de equipos
- ✓ Perfiles de usuario
- ✓ Opciones de perfil
- ✓ Usuarios
- ✓ Dispositivos RFID

4.- ¿Qué beneficios pretende otorgar a los usuarios de este portal web?

Objetivo:

Conocer los beneficios que los usuarios obtendrán con el portal web.

Respuesta:

El encargado de Recursos Informáticos de la UDT respondió en lo referente a la parte administrativa que se pretende otorgar la facilidad para gestionar la información que a diario se maneja dentro de la institución cuya ventaja radica en que la puedan gestionar desde cualquier lugar y no necesariamente desde dentro de las instalaciones como en la actualidad se lo realiza.

El administrador de la UDT además respondió en lo correspondiente a la parte pública que el visitante del portal web tendrá a su disposición información actualizada que debe de considerar si desea viajar desde las instalaciones del terminal terrestre.

Análisis:

La parte pública del portal brindará la información necesaria para informarnos por ejemplo sobre los horarios y tarifas de transporte además permitirá estar informados de lo que pasa en la institución.

Por otro lado, la parte administrativa dará la facilidad y flexibilidad para gestionar la información correspondiente a los procesos dentro de la UDT ya que permitirá hacerlo desde cualquier lugar y hora sin restricciones de ubicación geográfica.

5.- Gráficamente y funcionalmente qué sitios le gustaría que se revise antes de hacer una propuesta y describa que elementos le llaman la atención

Objetivo:

Conocer aquellos sitios web sugeridos por los entrevistados con el fin de determinar aspectos de funcionalidad y diseño.

Respuesta:

El administrador de la UDT respondió que entre los sitios que le gustaría que se visiten están: Terminal terrestre de Guayaquil de lo cual resalta las opciones de búsqueda de destinos, cooperativas y precios. Bolivariano – Colombia de la cual gráficamente resalta la sección de publicidad y el color además funcionalmente la sección de rutas. Terminal de Cali resalta la sección de slider de imágenes y estructura del sitio. Gran Terminal Terrestre – Perú resalta su sencillez y color. Otros sitios web sugeridos quedan a consideración del investigador.

Análisis:

Entre los aspectos que resaltan de los sitios sugeridos por el administrador están el diseño responsivo, colores, tipografía y estructura del sitio.

6.- ¿Si un usuario lo busca por Google por cuáles frases o palabras clave los debería encontrar? Sea lo más específico posible.

Objetivo:

Conocer las palabras clave o frases relacionadas a la entidad con las que los usuarios deberían de buscar el portal en la web.

Respuesta:

El administrador de la UDT respondió que las palabras por las que un usuario debería encontrar el portal web por medio de buscadores serían: Unidad Desconcentrada de Terminales Ambato, Terminal Terrestre Ambato, Áreas de transferencia, UDT,

Destinos ecuador, Cooperativas de transporte, Próximas salidas, Cargas y encomiendas, transportistas.

Análisis:

Las palabras claves sugeridas por el Administrador de la entidad serán usadas como metadatos del portal con el objetivo de que los buscadores como Google indexen al portal de la UDT en sus resultados de búsqueda.

7.- ¿Dispondrá de funcionalidad para bases de datos (generación de contenido dinámico, opciones de búsqueda, personalización/inicio de sesión)? ¿Dispone ya de una base de datos? Descríbalo detalladamente, incluyendo la información específica relativa a los programas y software existentes.

Objetivo:

Conocer si la UDT dispone de algún otro sistema o base de datos y saber si el contenido del portal va a ser dinámico o estático.

Respuesta:

El encargado de Recursos Informáticos de la UDT respondió que el contenido que el portal web va a generar es dinámico por esta razón se debe de proporcionar las facilidades pertinentes para su actualización.

Además, recalca que la parte de la administración del portal debe de proporcionar un inicio de sesión para que no cualquier persona pueda acceder a él puesto que la información que se tiene en la base de datos es importante y no solo la utilizará el portal web también lo hará el sistema local denominado “SICTERT” el cual permite llevar un control del terminal terrestre.

Análisis:

La UDT actualmente posee un sistema llamado SICTERT que trabaja en la intranet y cuya funcionalidad es facilitar y agilizar los procesos en la administración de las tareas diarias en lo referente al procesamiento de información; el mismo que posee también una base de datos la misma que será utilizada para el portal web puesto que el contenido de éste será dinámico.

PREGUNTAS PARA LA ENTREVISTA A LA PARTE ADMINISTRATIVA DE LA UDT

1. ¿Qué tipo de servicios ofrecerá el portal web?

.....
.....
.....

2. ¿Qué información se colocará en el portal web?

.....
.....
.....

3. ¿Qué información de tipo administrativa requiere que el portal web permita gestionar?

.....
.....
.....

4. ¿Qué beneficios pretende otorgar a los usuarios de este portal web?

.....
.....
.....

5. Gráficamente y funcionalmente qué sitios le gustaría que se revise antes de hacer una propuesta y describa que elementos le llaman la atención

.....
.....
.....

6. ¿Si un usuario lo busca por Google por cuáles frases o palabras clave los debería encontrar? Sea lo más específico posible.

.....
.....
.....

7. ¿Dispondrá de funcionalidad para bases de datos (generación de contenido dinámico, opciones de búsqueda, personalización/inicio de sesión)? ¿Dispone ya de una base de datos? Descríbalo detalladamente, incluyendo la información específica relativa a los programas y software existentes.

.....
.....
.....

ANEXO 4

Creación del nuevo esquema “portal”

```
CREATE SCHEMA [portal]
GO
```

Implementación de tablas para el portal web

Tabla AREA_TRANSFERENCIA

```
CREATE TABLE
portal.AREA_TRANSFERENCIA (
    ID_AREA int NOT NULL
    IDENTITY(1,1),
    NOMBRE varchar(100) NOT
    NULL,
    DIRECCION varchar(150)
    NOT NULL,
    TELEFONO varchar(50),
    IMG varchar(900),
    TIPO_TRANSP
    varchar(500),
    HORARIO varchar(100),
    LAT decimal(9,6),
    LON decimal(9,6),
    CONSTRAINT
    PK__AREA_TRA__6E15A1AAF49A16E
    2 PRIMARY KEY (ID_AREA)
);
```

Tabla COMUNICADO_NOTICIA

```
CREATE TABLE
portal.COMUNICADO_NOTICIA (
    ID int NOT NULL
    IDENTITY(1,1),
    TITULO varchar(90) NOT
    NULL,
    CONTENIDO varchar(max)
    NOT NULL,
    TIPO char(1) NOT NULL,
    FECHA date DEFAULT
    (getdate()),
    ESTADO bit DEFAULT
    ((0)),
    CONSTRAINT
    PK__COMUNICA__3214EC2728ACEDD
    7 PRIMARY KEY (ID)
)
CREATE UNIQUE INDEX
UQ__COMUNICA__AC728E50ECD7CA9
B ON
portal.COMUNICADO_NOTICIA
(TITULO);
```

Tabla DETALLE_COOPERATIVA

```
CREATE TABLE
portal.DETALLE_COOPERATIVA (
    ID int NOT NULL
    IDENTITY(1,1),
    ID_COOPERATIVA bigint
    NOT NULL,
```

Tabla INSTITUCION

```
CREATE TABLE
portal.INSTITUCION (
    ID int NOT NULL
    IDENTITY(1,1),
    NOMBRE varchar(120) NOT
    NULL,
```

```

        N_CUBICULO int NOT NULL,
        N_TELEFONO nvarchar(50),
        H_ATENCION
nvarchar(100),
        LOGO nvarchar(250)
DEFAULT ('no-logo'),
        WEBSITE nvarchar(250)
DEFAULT ('no-website'),
        CONSTRAINT
PK__DETALLE__3214EC271E2BDF9
6 PRIMARY KEY (ID),
        CONSTRAINT
FK__DETALLE_COOPERATIVA
FOREIGN KEY (ID_COOPERATIVA)
REFERENCES
dbo.COOPERATIVA(Id_Cooperativ
a) ON DELETE CASCADE ON
UPDATE CASCADE
);

        ABREVIATURA
varchar(20),
        SLOGAN varchar(250),
        DIRECCION varchar(150)
NOT NULL,
        TELEFONO varchar(80)
NOT NULL,
        CORREO_ADMIN
varchar(110) NOT NULL,
        HORARIO_ATENCION
varchar(110) NOT NULL,
        LOGO varchar(150),
        MISION varchar(950),
        VISION varchar(950),
        VALORES varchar(max),
        ACERCA_DE varchar(max),
        LAT decimal(9,6),
        LON decimal(9,6),
        CONSTRAINT
PK__INSTITUC__3214EC27CEEABFE
B PRIMARY KEY (ID)
)
CREATE UNIQUE INDEX
UQ__INSTITUC__B21D0AB90349079
D ON portal.INSTITUCION
(NOMBRE)
CREATE UNIQUE INDEX
UQ__INSTITUC__C074C20C24F6807
D ON portal.INSTITUCION
(ABREVIATURA);

```

Tabla LINK

```

CREATE TABLE portal.LINK (
        ID_LINK int NOT NULL
IDENTITY(1,1),
        NOMBRE varchar(90) NOT
NULL,
        ICONO varchar(max),
        URL varchar(100) NOT
NULL,
        CONSTRAINT
PK__LINK__9BCC952EB251E048
PRIMARY KEY (ID_LINK)
);

```

Tabla PREGUNTA_GLOSARIO

```

CREATE TABLE
portal.PREGUNTA_GLOSARIO (
        ID smallint PRIMARY KEY
NOT NULL IDENTITY(1,1),
        NOMBRE varchar(300),
        DESCRIPCION
varchar(max),
        TIPO char(1) NOT NULL
);

```

Tabla SERVICIO

Tabla RECURSO

```
CREATE TABLE portal.RECURSO (  
    ID int NOT NULL  
    IDENTITY(1,1),  
    NOMBRE varchar(100),  
    URL varchar(max),  
    DESCRIPCION  
    varchar(100),  
    TIPO char(1),  
    CONSTRAINT  
    PK__RECURSO__3214EC27D1F00245  
    PRIMARY KEY (ID)  
);
```

```
CREATE TABLE portal.SERVICIO  
(  
    ID_SERVICIO smallint  
    NOT NULL IDENTITY(1,1),  
    NOMBRE varchar(90) NOT  
    NULL,  
    DESCRIPCION  
    varchar(max),  
    IMAGEN varchar(150),  
    CONSTRAINT  
    PK__SERVICIO__C8BDE0EB78AEE2C  
    6 PRIMARY KEY (ID_SERVICIO)  
);
```

Tabla TARIFAS_DESTINO

```
CREATE TABLE  
portal.TARIFAS_DESTINO (  
    ID int NOT NULL  
    IDENTITY(1,1),  
    ID_DESTINO bigint NOT  
    NULL,  
    VALOR decimal(18,2) NOT  
    NULL,  
    IMG_DESTINO  
    varchar(2000) DEFAULT ('no-  
image'),  
    CONSTRAINT  
    PK__TARIFAS__3214EC273FBAE37  
    7 PRIMARY KEY (ID),  
    CONSTRAINT  
    FK__TARIFAS_D__ID_DE__63AEB14  
    3 FOREIGN KEY (ID_DESTINO)  
    REFERENCES  
    dbo.CIUDAD(Id_Ciudad) ON  
    DELETE CASCADE ON UPDATE  
    CASCADE  
);
```

Glosario Técnico y Acrónimos

BBDD: Base de Datos.

SQL: Structured Query Language (Lenguaje de Consulta Estructurados)

ORM: Object Relational Mapping.

IDE: Integrated Development Environment (Entorno de desarrollo integrado)

JDT: Java Development Tools (Herramientas de desarrollo para Java)

CSS: Cascade Style Sheet (Hoja de estilos en cascada)

MIT: Massachusetts Institute of Technology (Instituto Tecnológico de Massachusetts)

AJAX: Asynchronous Javascript and XML (Javascript y XML Asíncrono)

RIA: Rich Internet applications (Aplicación de Internet enriquecida)

DOM: Document Object Model (Modelo de Objetos del Documento)

DDL: Data Definition Language (Lenguaje de Definición de Datos)

DML: Data Manipulation Language (Lenguaje de Manipulación de Datos)

ANSI: American National Standards Institute (Sistema de codificación de caracteres alfanuméricos)

ISO: International Organization for Standardization (Organización Internacional de Estandarización)

SGBD: Sistemas Gestores de Bases de Datos

ANT: Agencia Nacional de Tránsito

UDT: Unidad Desconcentrada de Terminales

RFID: (Radio Frequency Identification - identificación por radiofrecuencia)

SICTERT: Sistema de Control de Terminales Terrestres

CRUD: (Create, Read, Update and Delete) - Crear, Leer, Actualizar y Borrar

OOHDM: Object Oriented Hypermedia Design Methodology

HDM: Hypertext Design Model

WSDM: Web Site Design Method

UWE: UML-BASED WEB ENGINEERING

JDBC: Java Database Connectivity

JTA: Java Transacción API

JNDI: Java Naming and Directory Interface.

RDBMS: Relational Database Management System (Sistema de gestión de bases de datos relacionales)