

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO – CHILE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**SISTEMA DE GESTIÓN DE INFORMACIÓN VÍA
DISPOSITIVO MÓVIL PARA LE-PUCV**

ANDRÉS MAURICIO SANTOS CONTRERAS
DANILO ANDRÉS VÁSQUEZ SIMONETTI

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

Abril, 2013

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO – CHILE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**SISTEMA DE GESTIÓN DE INFORMACIÓN VÍA
DISPOSITIVO MÓVIL PARA LE-PUCV**

ANDRÉS MAURICIO SANTOS CONTRERAS
DANILO ANDRÉS VASQUEZ SIMONETTI

Profesor Guía: Guillermo Cabrera Guerrero
Profesor Correferente: Iván Mercado Bermúdez

Carrera: **Ingeniería de Ejecución en Informática**

Abril, 2013

Dedicatoria

A mi familia, por el apoyo incondicional, especialmente a mis padres, quiénes han sido mi motivación e impulso principal para el cumplimiento de mi carrera profesional.

A la señorita Karla Pacheco, quién fue el apoyo fundamental en los momentos de flaqueza y adversidad vividos durante todo este ciclo, quien me ayudó siempre a continuar sin decaer y sin perder la Fe y Esperanza en el cumplimiento de los Objetivos trazados en el inicio de esta travesía.

En general a mis compañeros, amigos y a todos quienes desearon que este proceso fuera experimentado y culminado satisfactoriamente.

A mi profesor guía quien proporcionó las herramientas y los conocimientos necesarios para culminar este Proyecto exitosamente

A todos ellos sinceramente muchas gracias.

Andrés Mauricio Santos Contreras.

Dedicatoria

A mi familia, especialmente a mi madre por su apoyo incondicional y por darme ánimo cuando más lo necesitaba.

A mis amigos y compañeros de universidad por ser parte de este proceso.

A los profesores de la Universidad por formarnos como profesionales y prepararnos para realizar este proyecto.

A todos ellos muchas gracias.

Danilo Vásquez Simonetti

ÍNDICE

Resumen	vii
Lista de Figuras	ix
Lista de Tablas	x
<u>CAPÍTULO 1: INTRODUCCIÓN</u>	1
<u>CAPÍTULO 2: DEFINICIÓN DE OBJETIVOS</u>	2
<u>CAPÍTULO 3: ESTADO DEL ARTE</u>	3
3.2.2 Seguridad en servicios web	9
<u>CAPÍTULO 4: SOLUCIÓN PROPUESTA</u>	11
4.1 Metodología.....	15
4.2 Desarrollo	16
4.2.1 Estudio de factibilidad.....	16
4.2.1.1 Factibilidad técnica	16
4.2.1.2 Factibilidad económica	17
4.2.1.3 Factibilidad operativa	19
4.2.1.4 Factibilidad legal.....	19
4.2.2 Gestión de riesgos	20
4.2.2.1 Identificación de riesgos	20
4.2.2.2 Análisis de riesgos	21
4.2.2.3 Planificación de riesgos	22
4.2.3 Enfoque	23
4.2.4 Modelo de Desarrollo.....	24
4.2.5 Herramientas y Tecnologías	27
4.2.6 Arquitectura.....	29
4.2.7 Análisis y definición de requerimientos.....	30
4.2.8 Diseño.....	34
4.2.8.1 Modelamiento de la base de datos	34
4.2.8.2 Diagramas de clases	49
4.2.8.3 Diagramas de secuencia	55
4.2.8.4 Servicio web	57
4.2.8.5 Página web de administración	60
<u>CAPÍTULO 5: INTERFACES DE USUARIO</u>	60
<u>CAPÍTULO 6: VALIDACIÓN Y VERIFICACIÓN</u>	65

6.1	Plan de Pruebas.....	65
6.1.5	Guía de pruebas tipo para elementos de un formulario	69
	<u>CAPÍTULO 7: CONCLUSIONES</u>	70
	<u>CAPÍTULO 8: REFERENCIAS BIBLIOGRÁFICAS</u>	71
8.1	Referencias	71
8.2	Bibliografía.....	71
	 ANEXOS.....	
	A: ESPECIFICACIÓN DE REQUERIMIENTOS.....	
	B: ESPECIFICACIÓN CASOS DE PRUEBA.....	
	C: NORMAS LEGALES.....	

Resumen

Entre los servicios que presta la empresa LE-PUCV se encuentran los controles de calidad que se llevan a cabo a través de distintos ensayos efectuados en terreno. Éstos tienen un proceso en común: los datos que entregan dichos ensayos son anotados en formularios impresos, lo cual, implica posibles errores de lectura y trabajo extra, ya que finalmente la información es traspasada a un computador manualmente estando susceptible a errores humanos.

Para solucionar esta problemática se propone la implementación de un sistema de formularios digitales en un dispositivo móvil, con el fin de automatizar la captura y almacenamiento de la información. El software le permite al usuario ingresar los datos relacionados con el formulario en cuestión y posteriormente enviar dichos datos de forma inalámbrica a un servidor, donde se alojará la información para su eventual procesamiento. La comunicación entre el dispositivo móvil y la base de datos, es permitida por un servicio web. La administración del sistema se lleva a cabo mediante una página web.

Con la implementación de la solución desarrollada se obtienen muchos resultados favorables, entre los cuales destacan: una disminución considerable en los errores humanos como lectura y digitalización, además de una reducción en los tiempos de tramitación, mayor control sobre el personal que opera en terreno y una disponibilidad de la información en tiempo real.

Cada vez es más común que las necesidades de las empresas demanden nuevas tecnologías para mejorar la productividad y eficiencia del trabajo. El software desarrollado satisface la actual demanda, utilizando dispositivos móviles para capturar datos en terreno facilitando el ingreso, envío, almacenamiento y posterior procesamiento de la información.

Palabras claves: dispositivos móviles, construcción, servicios web.

Abstract

Among the services the company provides are the quality control services which are carried out as different tests at the construction sites. These tests have one thing in common, the information gathered is written down on forms, which could lead to misinterpretation and implies extra work since all this information has to be transferred manually to a computer.

To solve this problem the proposition is to implement the same forms digitally on a mobile device allowing us to automate the input and storage of information. The software lets the user enter the data associated to a form and send it wirelessly to a server, where it is stored and processed accordingly. The administration of this software is managed through a web page.

With the implementation of the aforementioned solution come various benefits, for example: human errors should diminish considerably thanks to the input validation of a mobile device, the time needed to get the data ready for processing is reduced thanks to the wireless transfer and the authorization required by the mobile device implies greater control.

It's becoming more and more common for companies to require new technologies to improve their productivity and efficiency. The developed software satisfies this need utilizing mobile devices to capture data at the construction sites making it easier to enter, send, store and process the information.

Key words: mobile devices, construction, web services

Lista de Figuras

Figura 3.1 Servicios web basados en XML.....	7
Figura 4.1 Esquema de implementación	11
Figura 4.2 Diagrama de actividades del negocio.....	12
Figura 4.3 Diagrama de actividades del negocio con dispositivo móvil.....	13
Figura 4.4 Diagrama de flujos factibilidad económica.....	18
Figura 4.5 Metodología de desarrollo.....	25
Figura 4.6 Arquitectura Cliente-Servidor	29
Figura 4.7 Caso de uso. General.....	31
Figura 4.8 Caso de uso. Administrador	32
Figura 4.9 Caso de uso. Usuario en terreno.....	33
Figura 4.10 MER General	34
Figura 4.11 MER Muestreo asfalto	35
Figura 4.12 MER Marshall de control.....	36
Figura 4.13 MER Densidades carpeta asfáltica.....	37
Figura 4.14 MER Control de temperatura.....	38
Figura 4.15 MER Control tasa de riego.....	39
Figura 4.16 MER Hi-lo.....	40
Figura 4.17 MER Densidades con densímetro	41
Figura 4.18 MER Densidades con cono de arena.....	42
Figura 4.19 MER Toma de muestra de suelo	43
Figura 4.20 MER Toma de muestra de hormigón	44
Figura 4.21 MER Muestreo.....	45
Figura 4.22 Modelo Relacional parte 1	46
Figura 4.23 Modelo Relacional parte 2	47
Figura 4.24 Modelo Relacional Sistema de Administración LE-PUCV	48
Figura 4.25 Diagrama de clases. General.....	49
Figura 4.26 Diagrama de clases. Software Dispositivo Móvil.....	50
Figura 4.27 Paquetes	51
Figura 4.28 Usuarios	51
Figura 4.29 Roles.....	52
Figura 4.30 Formularios	52
Figura 4.31 Diagrama de despliegue.....	53
Figura 4.32 Diagrama de componentes	54
Figura 4.33 Diagrama de secuencia. Ingresar orden de trabajo.....	55
Figura 4.34 Diagrama de secuencia. Crear formulario.....	55
Figura 4.35 Diagrama de secuencia. Enviar formularios	56
Figura 4.36 Diagrama de secuencia. Añadir usuario	56
Figura 4.37 Arquitectura de página web.....	60
Figura 5.1 Pantalla Inicial Sw. Disp. Móvil	61
Figura 5.2 Login Sw. Disp. Móvil.....	62
Figura 5.3 Menú Orden de Trabajo Sw. Disp. Móvil	63
Figura 5.4 Menú Formulario Sw. Disp. Móvil	64
Figura 6.1 Ilustración Caja Negra.....	67

Lista de Tablas

Tabla 3.1 Tabla uso de recursos servicio web RESTful.....	9
Tabla 4.1 Análisis de riesgo.....	21
Tabla 4.2 Planificación de riesgos.....	22
Tabla 4.3 Caso de uso. Añadir usuario.....	72
Tabla 4.4 Caso de uso. Eliminar usuario.....	73
Tabla 4.5 Caso de uso. Editar usuario.....	74
Tabla 4.6 Caso de uso. Añadir rol.....	75
Tabla 4.7 Caso de uso. Eliminar rol.....	76
Tabla 4.8 Caso de uso. Editar rol.....	77
Tabla 4.9 Caso de uso. Asociar ensayo a rol.....	78
Tabla 4.10 Caso de uso. Asociar usuario a rol.....	79
Tabla 4.11 Caso de uso. Visualizar formulario.....	80
Tabla 4.12 Caso de uso. Crear informe.....	81
Tabla 4.13 Caso de uso. Eliminar formulario.....	82
Tabla 4.14 Caso de uso. Ingresar orden de trabajo.....	83
Tabla 4.15 Caso de uso. Editar orden de trabajo.....	84
Tabla 4.16 Caso de uso. Crear formulario.....	85
Tabla 4.17 Caso de uso. Crear copia de formulario.....	86
Tabla 4.18 Caso de uso. Editar formulario.....	87
Tabla 4.19 Caso de uso. Enviar formularios.....	88
Tabla 4.20 Caso de uso. Actualizar información en PDA.....	89
Tabla 4.21 Caso de uso. Obtener formularios.....	90
Tabla 4.22 Caso de uso. Regenerar base de datos.....	91
Tabla 5.1 Caso de prueba. Añadir usuario.....	92
Tabla 5.2 Caso de prueba. Eliminar usuario.....	93
Tabla 5.3 Caso de prueba. Crear informe.....	94
Tabla 5.4 Caso de prueba. Ingresar orden de trabajo.....	95
Tabla 5.5 Caso de prueba. Editar orden de trabajo.....	97
Tabla 5.6 Caso de prueba. Crear formulario.....	99
Tabla 5.7 Caso de prueba. Editar formulario.....	101
Tabla 5.8 Caso de prueba. Enviar formularios.....	103
Tabla 5.9 Caso de prueba. Actualizar información en PDA.....	105

Glosario de Términos

Facelets: Framework web de código abierto bajo la licencia Apache y la vista por defecto de la tecnología de controlador JavaServer Faces (JSF). Facelets es compatible con todos los componentes JSF interfaz de usuario y se centra por completo en la construcción del árbol de componentes JSF, lo que refleja el punto de vista de una aplicación JSF.

Feedback: También denominado retroalimentación o realimentación significa "ida y vuelta" es el proceso de compartir observaciones, preocupaciones y sugerencias, con la intención de recabar información, a nivel individual o colectivo, para intentar mejorar el funcionamiento de una organización o de cualquier grupo formado por seres humanos.

Framework: Define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

Glassfish: Servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación.

Java SE 6: Es una colección de APIs del lenguaje de programación Java útiles para muchos programas de la Plataforma Java.

JavaServerPages (JSP): Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Java Persistence API: Más conocida por su sigla JPA, es la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB3. Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos.

Login: Proceso de identificación del usuario en el sistema.

MySQL: Sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

PhpMyAdmin: Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas. Se encuentra disponible bajo la licencia GPL.

Servlets: Son objetos que corren dentro del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad. El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envía el navegador web.

Testing: Pruebas de software correspondientes a los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

Tomcat: (también llamado JakartaTomcat o Apache Tomcat) Funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServerPages (JSP) de Sun Microsystems.

Lista de Abreviaturas o Siglas

CCDL	: Common Development and Distribution License, Licencia Común de Desarrollo y Distribución.
DLL	: Dynamic-Link Library, Biblioteca de Enlace Dinámico.
DVD	: Digital Versatile Disc, Disco Versátil Digital.
GNU/GPL	: General Public License, Licencia Pública General.
HP	: Hewlett-Packard.
HTC	: High Tech Computer.
HTTP	: Hypertext Transfer Protocol, Protocolo de Transferencia de Hipertexto.
J2ME	: Java 2 Micro Edition, Java 2 Edición Micro.
J2EE	: Java Platform, Enterprise Edition o Java EE.
JPA	: Java Persistence API.
JSP	: Java Server Faces.
JSP	: Java Server Pages.
LE-PUCV	: Laboratorio de Ensayos de la Pontificia Universidad Católica de Valparaíso.
MER	: Modelo Entidad Relación.
MID	: Mobile Internet Device, Dispositivo Móvil para conectividad a Internet.
MIME	: Multipurpose Internet Mail Extensions, Extensiones Multipropósito de Correo de Internet.
MVC	: Modelo Vista Controlador.
PDA	: Personal Digital Assistant, Asistente Digital Personal.
PDF	: Portable Document Format, Formato de Documento Portátil
PHP	: PHP Hypertext Pre-processor.
REST	: Representational State Transfer, Transferencia de Estado Representacional.

RPC : Remote Procedure Call, Llamada a Procedimiento Remoto.

SMS : Short Message Service, Servicio de Mensajes Cortos.

SOA : Service-oriented architecture, Arquitectura Orientada a Servicios.

SOAP : Simple Object Access Protocol.

SSL/TLS : Secure Sockets Layer -Protocolo de Capa de Conexión Segura- y Transport Layer Security -Seguridad de la Capa de Transporte-.

SQL : Lenguaje de Consulta Estructurado.

UDDI : Universal Description, Discovery and Integration.

UML : Unified Modeling Language, Lenguaje Unificado de Modelado.

UMPC : Ultra Mobile Personal Computer.

URI : Uniform Resource Identifier, Identificador Uniforme de Recurso.

VAN : Valor Actual Neto.

WAP : Wireless Application Protocol, Protocolo de Aplicaciones Inalámbricas.

WSDL : Web Services Description Language.

WS-Security : Seguridad en Servicios Web.

W3C : World Wide Web Consortium.

XML : eXtensible Markup Language, Lenguaje de MarcaseXtensible.

CAPÍTULO 1: INTRODUCCIÓN

Laboratorio de Ensayos de la Pontificia Universidad Católica de Valparaíso (LE-PUCV) es una sociedad anónima que nació como un proyecto de esta institución universitaria y sigue fuertemente ligada a la Universidad, ya que esta Casa de Estudios es dueña de la mayoría de las acciones de la empresa.

LE-PUCV es una empresa perteneciente al rubro de la construcción, la cual ha sido fundada en Valparaíso a fines de la década de los setenta. Actualmente posee cuatro sedes distribuidas en cuatro ciudades distintas; Copiapó que se encuentra en la tercera región, Vallenar que se sitúa en la cuarta región, Valparaíso que se encuentra en la quinta región y Coyhaique que se sitúa en la décimo primera región.

La empresa presta variados servicios, dentro de los cuales se encuentran las asesorías en distintas áreas como por ejemplo: edificación, control de calidad, geotecnia, proyectos de ingeniería, manejo de residuos sólidos y capacitación, entre otros.

Los controles de calidad se llevan a cabo a través de distintos ensayos, muchos de éstos se efectúan en terreno. Bajo este contexto el presente proyecto informático desarrolla su función. Los distintos ensayos que se llevan a cabo tienen un proceso en común: los datos asociados a dichos ensayos son registrados en formularios de manera manual. Cada ensayo tiene asociado un formulario con un formato específico y dado que en un día se pueden desarrollar decenas de ensayos, la tarea de completar un formulario puede volverse tediosa. Además, esto, implica posibles errores de lectura y trabajo extra, ya que la información al fin y al cabo se traspa a un computador, siendo ingresada manualmente.

La propuesta consiste en reemplazar los formularios impresos por un software implementado en un dispositivo móvil. Los datos se registran en formularios digitales y posteriormente son traspasados a un servidor central. Operando con la solución propuesta a desarrollar, la persona en terreno utilizará un dispositivo móvil, como por ejemplo una PDA, que tendrá instalado el sistema a manipular. En él se debe especificar el área y el tipo de ensayo con que se está trabajando. El software le permitirá ingresar los datos relacionados con el ensayo en cuestión y posteriormente enviar dichos datos de forma inalámbrica a un servidor, donde se alojará la información.

La solución propuesta implica una inversión inicial, al tener que adquirir los dispositivos móviles y capacitar al personal en cuanto a su uso, pero a largo plazo se podrá observar el retorno de la inversión, ya que se obtendría una disminución considerable en los errores humanos y en los tiempos de tramitación entre otras cosas y al mismo tiempo se experimentaría un aumento en la productividad y en general en la eficiencia del trabajo.

CAPÍTULO 2: DEFINICIÓN DE OBJETIVOS

2.1 Objetivo general

- Desarrollar un sistema que permita evitar el uso de formularios físicos al momento de la toma de datos en terreno, minimizando los errores generados por la doble digitación manual y agilizando el control en la gestión de los ensayos.

2.2 Objetivos específicos

- Analizar la forma de trabajo del cliente y entender la problemática existente.
- Investigar las alternativas que existen entre los dispositivos móviles y escoger la más indicada para este proyecto.
- Investigar cómo realizar una transferencia segura de la información desde el dispositivo móvil al servidor y sobre seguridad en general relacionada con dispositivos móviles.
- Desarrollar un software que cumpla cabalmente con los requisitos del cliente.
- Aplicar diversas pruebas a los prototipos para asegurar la construcción de un software robusto.

CAPÍTULO 3: ESTADO DEL ARTE

3.1 Dispositivos móviles

En la actualidad los dispositivos móviles han adquirido un rol importante dentro de las nuevas tecnologías que las empresas han adoptado para aumentar la productividad. En el presente proyecto se incorpora la utilización de los dispositivos móviles para facilitar el desarrollo de las actividades de la empresa LE-PUCV. En base a lo anterior se realiza un estudio de las diferentes alternativas presentes en el mercado aplicables a la solución propuesta.

3.1.1 Definición de dispositivo móvil

El término dispositivo móvil es un concepto bastante amplio y antes de continuar se analizará el significado que se le dará en el resto del documento. Es debido mencionar que un dispositivo móvil posee variadas características, siendo las más generales las que se presentan a continuación:

- Son aparatos relativamente pequeños.
- Tienen una capacidad de procesamiento limitada.
- Están conectados permanente o intermitentemente a una red.
- Tienen memoria limitada.

Además, como el nombre lo indica, un dispositivo móvil es lo suficientemente pequeño para ser trasladado y al mismo tiempo empleado durante el transporte. La industria de los dispositivos móviles es muy variopinta, presentándose a continuación algunos de los equipos existentes:

- Mensáfono.
- Videoconsolas portátiles.
- PDA (Asistente Digital Personal).
- Celulares.
- Teléfonos inteligentes (Smartphone).
- Computadores portátiles.

Para el presente proyecto, es de interés un tipo específico de dispositivos móviles, aquellos capaces de llevar a cabo tareas generales, mediante la previa instalación de software adicional.

En los siguientes párrafos se abordan estos dispositivos con mayor detalle.

Teléfonos celulares: Los teléfonos celulares tienen una pantalla pequeña, basada en texto o con pocos colores, con servicios de transferencia de datos generalmente limitados a SMS y WAP. Se pueden identificar dos grupos. Por un lado están los más básicos, los cuales no tienen la capacidad de agregarles programas adicionales. Por otro lado se encuentran los más avanzados, los que generalmente pueden ejecutar programas J2ME.

Teléfono inteligente: En realidad no existe una definición estándar de lo que es un teléfono inteligente. Una de las definiciones disponibles, y la que se utilizará de aquí en adelante, define a un teléfono inteligente como aquel que incorpora un sistema operativo con una interfaz consistente y una plataforma orientada a desarrolladores de aplicaciones. Además un teléfono inteligente puede tener pantalla sensible al tacto, teclado qwerty y Wifi con navegador web entre otras características.

Existen diversos sistemas operativos disponibles para teléfonos inteligentes, los más conocidos y utilizados son los siguientes^[1]:

- **SymbianFoundation (47.1% del mercado 2008)**
 - Una alianza entre varias compañías como Nokia, Samsung, Sony Ericsson, etc. El código fuente se liberó a principio del año 2010 y es uno de los sistemas operativos más utilizados en teléfonos inteligentes.
- **RIM Blackberry (19.5% del mercado 2008)**
 - Enfocado en que sea sencillo de utilizar y originalmente diseñado para los negocios.
- **Windows Mobile (12.4% del mercado 2008)**
 - El sistema operativo de Microsoft.
- **iPhone OS (10.7% del mercado 2008)**
 - El sistema operativo del iPhone basado en Mac OS X.
- **GNU/Linux (8.4% del mercado 2008)**
 - Utilizado principalmente en China por Motorola y en Japón por DoCoMo. No es un sistema operativo único, más bien GNU/Linux es la base para una variedad de sistemas operativos.
- **Android**
 - El sistema operativo de Google de código abierto, basado en Linux.

PDA: La PDA (asistente personal digital) es un computador de mano originalmente diseñado como agenda electrónica. Han evolucionado hasta el punto de ser capaces de realizar muchas funciones propias de computadores de escritorio. Generalmente tienen pantalla táctil con reconocimiento de escritura para ingresar datos, bluetooth y Wifi.

El mercado de las PDAs, actualmente es muy amplio, algunas de las que se pueden encontrar disponibles hoy en día son^[2]:

- **Dispositivos HTC**
 - Principal competidor del Iphone. Utiliza Windows Mobile.
- **Dispositivos con Palm OS**
 - Actualmente se encuentra casi obsoleto ya que hasta algunos dispositivos Palm usan Windows Mobile.
- **Dispositivos Pocket PC**
 - PDAs que utilizan Windows Mobile, fabricadas por HP, Dell y Acer.
- **Dispositivos con GNU/Linux**
 - Principalmente usado en el modelo Sharp Zaurus que fue muy popular en Japón en los 90. Existe una versión de Linux para los Pocket PC pero es una versión en desarrollo que no tiene la madurez como para competir con Windows Mobile.

MID: Dispositivos concebidos por Nokia, diseñados para la total conectividad y total portabilidad con baja potencia y bajo precio. Tienen un tamaño de pantalla alrededor de 5 pulgadas y generalmente usan Linux como sistema operativo. No han logrado encontrar su nicho en el mercado.

UMPC: Dispositivo presentado por Microsoft en 2006. Usa Windows XP y tiene una pantalla de alrededor de 7 pulgadas. El problema de estos dispositivos siempre ha sido su precio que, aunque fue bajando, se mantuvo muy elevado. Estos dispositivos, al igual que los MID, tampoco lograron encontrar un nicho en el mercado.

Netbook: Los netbooks son dispositivos de bajo costo y reducidas dimensiones con el fin de proveer una mayor movilidad y autonomía. Dado sus prestaciones limitadas, son usados principalmente para navegar por internet y para funciones básicas como procesamiento de texto y hojas de cálculo. Generalmente no tienen unidad de medios ópticos como DVD. Sistemas operativos comunes son Windows XP y variantes de GNU/Linux.

Notebook: Un notebook es un dispositivo con capacidades cercanas a las de un computador de escritorio, pero que pesa entre 1 y 3 kilogramos y por sobre todo es portátil. Los sistemas operativos disponibles son los mismos que para un computador de escritorio.

3.1.2 Seguridad en dispositivos móviles

Los dispositivos móviles se encuentran, por su naturaleza, muy expuestos a robos o pérdidas, por lo que, es de vital importancia restringir el acceso de personas no autorizadas, especialmente si el dispositivo contiene datos sensibles, como es el caso del presente proyecto, dónde el dispositivo móvil será usado como una herramienta de trabajo empresarial.

En general, toda empresa, de cualquier índole, que utilice dispositivos móviles en su forma de trabajo, debe implementar una política que indique cómo se deben usar estos dispositivos de una forma segura.

Es prudente mencionar, que existe una buena costumbre que se debe acatar; siempre mantener el bluetooth y el Wifi apagado si no se está usando, ya que al estar disponible uno de estos medios, el dispositivo se encuentra expuesto a un ataque informático.

Los dispositivos HP Ipaq vienen con un software llamado HP ProtectTools, que se encarga de encriptar los datos almacenados y proteger el dispositivo con contraseña. Activar esta función es muy recomendado.

Otro factor a tener en cuenta son los virus, que afectan desde celulares hasta notebooks. Un virus informático puede causar, entre otras cosas, pérdida de datos, lo cual es muy serio cuando hay un contrato con un cliente de por medio, o cuando la información que se pierde es de suma importancia. Además, estos programas malintencionados se pueden propagar a los computadores de escritorio o servidores de la empresa, al conectarse el dispositivo móvil infectado con dichos equipos, poniendo en riesgo toda la información de la organización.

En el caso de dispositivos móviles con Windows Mobile, existe software antivirus sin costo monetario para instalar, lo cual no deja de ser una recomendación importante para el cliente.

En términos generales, es de suma importancia tomar todas las medidas que sean necesarias para resguardar tanto la disponibilidad de los datos, como la integridad de los mismos y la confidencialidad de la información.

3.1.3 Requisitos mínimos del dispositivo móvil

Los dispositivos móviles, que se quieran utilizar, deben cumplir ciertos requisitos mínimos para ser usados de manera óptima. En este caso particular el dispositivo móvil debe cumplir los siguientes requisitos:

- Tener por lo menos 32 MB de memoria de almacenamiento disponible para el usuario, ya que se necesita almacenar los formularios con sus datos correspondientes, más la información asociada a cada usuario para ingresar al sistema.
- Tener una pantalla de mínimo 240x320 pixeles, debido a que algunos formularios son bastante extensos y sería incómodo ingresar los datos en una pantalla muy pequeña.
- Tener conexión inalámbrica a redes Wifi y ser capaz de utilizar servicios web, para así poder acceder a una base de datos relacional.

Mencionadas las características anteriores, existen dos alternativas a seguir, a continuación se presentan ambas opciones y finalmente la que ha sido elegida por el cliente para la implementación del proyecto.

La primera alternativa es implementar el sistema en una PDA que cumpla con los requisitos mencionados. Entre las ventajas de utilizar una PDA se encuentran, su tamaño reducido, que permite un transporte sencillo por parte del equipo de trabajo y su alta duración de batería, que favorece un tiempo de trabajo prolongado. Entre sus desventajas cabe mencionar, su pantalla de tamaño reducido, que desfavorece la visualización de formularios extensos y su relativamente lento ingreso de datos, que se lleva a cabo mediante un teclado virtual o por reconocimiento de escritura.

La segunda alternativa es usar un netbook, dado que el precio de este equipo está al mismo nivel de una PDA. Una de las ventajas de utilizar un netbook es su monitor de mayor tamaño, lo que significa mejor visualización de la aplicación. Otra ventaja es que tiene un poder de procesamiento y almacenamiento superior al de una PDA.

Sin embargo, al ser presentadas al cliente las alternativas, con sus respectivas ventajas y desventajas, éste respondió que un netbook, por pequeño que sea, no deja de ser un equipo más que tiene que cuidar el laboratorista, ya que para realizar mediciones, muchas veces se utilizan equipos especiales y en fin de cuentas resultaría una preocupación extra. Lo anterior es un punto de consideración para el cliente, debido a que han ocurrido incidentes como robo de implementos en las camionetas de trabajo. Mencionadas las condiciones anteriores, se optó finalmente por la utilización de una PDA, que sin problemas se puede llevar en un bolsillo, no distrayendo al laboratorista de su trabajo, dado que no se incrementa el número de equipos a vigilar.

El software a implementar en el dispositivo móvil será compatible con Windows Mobile 5, por lo que el modelo específico de la PDA puede ser a elección del cliente, basado en presupuesto y otros factores de consideración.

3.2 Servicios web

3.2.1 Definición de servicio web

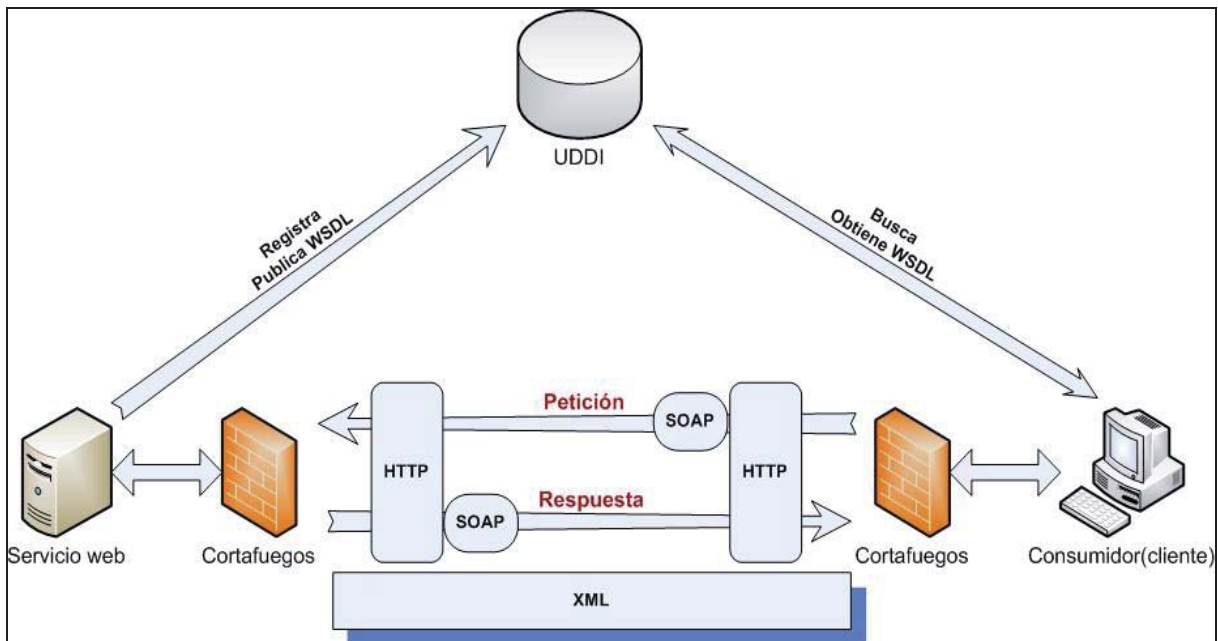


Figura 3.1 Servicios web basados en XML

Un servicio web es, según la definición de la W3C, “un sistema de software diseñado para implementar interacción interoperativa entre máquinas sobre una red.”^[3]. Esta definición es bastante general y puede referirse a muchos sistemas diferentes, pero normalmente se refiere a clientes y servidores que se comunican sobre el protocolo HTTP utilizado en internet. Se puede diferenciar entre los servicios web que se basan en XML y otros que operan directamente sobre HTTP.

Los servicios web basados en XML se comunican a través del protocolo SOAP, ofrecen una descripción de las operaciones que se pueden realizar en el lenguaje WSDL y pueden publicar las operaciones disponibles a través de UDDI que es un registro que provee acceso a documentos WSDL y describe el formato de los mensajes para interactuar con los servicios web. Los servicios web se pueden implementar de distintas formas. Los tres estilos de implementación más comunes son:

1. Llamada a procedimiento remoto (RPC)

Los servicios web RPC presentan una interfaz para llamar a funciones distribuidas. Generalmente la unidad básica de un servicio web RPC es la operación WSDL. Cuando recién surgieron los servicios web se implementaron con este estilo y es por eso que es uno de los estilos más utilizados hoy en día. Sin embargo tiene sus desventajas, siendo la principal que las funciones pueden no tener un acoplamiento débil. En la figura 2.1 se ve un ejemplo de su implementación típica. Los servicios web RPC están basados en XML por lo que, basado en lo mencionado anteriormente, también utiliza SOAP y WSDL.

2. **Arquitectura orientada a servicios (SOA)**

“SOA es en realidad un estilo de diseño arquitectónico que puede o no utilizar servicios web en su implementación.”^[4]. SOA enfatiza la implementación de componentes, como servicios modulares, que los clientes pueden descubrir y utilizar. Los servicios generalmente poseen las siguientes características:

1. Se pueden usar individualmente o integrarse con otros servicios para proveer servicios de más alto nivel. Esto promueve la reutilización de la funcionalidad existente.
2. Los servicios se comunican con sus clientes intercambiando mensajes. Se define por los mensajes, qué se puede recibir y las respuestas que se pueden enviar.
3. Los servicios pueden participar en un flujo de trabajo, en donde el orden de los mensajes afecta el resultado de las operaciones llevadas a cabo por un servicio.
4. Los servicios pueden ser totalmente independientes, o depender de otros servicios, o de algún recurso como una base de datos.
5. Los servicios dan a conocer detalles como sus capacidades, interfaces, políticas y protocolos de comunicación admitidos.

En SOA, la unidad básica de comunicación es el mensaje en vez de una función. Esto implica un acoplamiento débil, ya que el enfoque no está en cómo se implementan las funciones, sino en lo que hacen las funciones. Permitiendo así la creación de sistemas altamente escalables que reflejan el negocio de la organización.

3. **Servicios web RESTFul**

Los servicios web RESTFul no son basados en XML, sino que se basan en el protocolo HTTP actualmente en uso en la web y en REST, un estilo de arquitectura de software. Utilizan las operaciones del lenguaje HTML, como GET, POST, PUT y DELETE para realizar la comunicación y manipular los recursos disponibles. Con frecuencia estas operaciones se comparan con las operaciones CRUD (Crear, obtener, actualizar y borrar) de una base de datos. En los servicios web RESTFul todo es un recurso y todo recurso se accede a través de un URI que es un identificador global único.

La definición de un servicio web RESTFul se puede dividir en tres aspectos:

1. La URI base del servicio web como por ejemplo <http://www.lepucv.cl/recursos/formularios>.
2. El tipo MIME de los datos que puede leer el servicio web. Un ejemplo es XML pero existen muchos otros.
3. El conjunto de operaciones permitidas por el servicio web utilizando operaciones HTTP.

La siguiente tabla muestra un ejemplo que clarifica como se usan los recursos y las operaciones en un servicio web RESTFul^[5].

3.1 Tabla uso de recursos servicio web RESTful

Recurso	GET	PUT	POST	DELETE
URI de colección como http://www.lepucv.cl/recursos/formularios	Muestra los formularios que forman parte del recurso.	Reemplaza el conjunto de formularios por otro.	Crea un nuevo formulario que forma parte del conjunto.	Elimina el conjunto de formularios.
URI de un miembro individual de una colección http://www.lepucv.cl/recursos/formularios/formularioXYZ	Obtiene el miembro individual del conjunto.	Actualiza el miembro del conjunto.	En general no se usa.	Elimina el miembro del conjunto.

3.2.2 Seguridad en servicios web

Los servicios web, por estar disponibles en internet, se encuentran muy expuestos a diversos problemas de seguridad. Éstos pueden, en más de una ocasión, poner en riesgo la disponibilidad, la confidencialidad y la integridad de los datos. Algunas de las amenazas más comunes en torno a los servicios web son:

- **Alteración de mensajes.** Un atacante inserta, remueve o modifica información en un mensaje para engañar al receptor.
- **Pérdida de confidencialidad.** Una persona no autorizada tiene acceso al contenido de un mensaje.
- **Mensajes falsificados.** Mensajes ficticios que el atacante envía para que el destinatario crea que lo envió otra persona o máquina.
- **Man in themiddle.** Un tercero se encuentra interceptando mensajes entre, en este caso, el servicio web y el consumidor. El tercero reenvía los mensajes que recibe sin que lo sepan los participantes. Esto le permite ver y modificar todos los mensajes.
- **Solicitud falsificada.** Un atacante construye un mensaje con credenciales falsas que aparenta ser válido para el que lo recibe.
- **Reenvío de mensaje.** Un atacante envía un mensaje que ya ha sido enviado.
- **Reenvío de partes de un mensaje.** Un atacante envía un mensaje con partes de un mensaje que ya ha sido enviado.
- **Denial of service.** Un atacante envía una gran cantidad de peticiones al servicio web causando que las peticiones válidas no se puedan recibir.

Dados los riesgos mencionados anteriormente, existen algunos métodos para prevenir el impacto de un ataque informático, los cuales se describen de la siguiente manera:

- **Replicar datos y servicios para aumentar la disponibilidad.** Como los servicios web son susceptibles a un ataque DoS, es importante este punto. El hecho de implementar una réplica puede asegurar el acceso a datos importantes en el momento en que ha ocurrido una falla.
- **Guardar información sobre las transacciones en un log para mejorar el no repudio.** En un SOA es complejo implementar un log centralizado pero en un servicio web RPC sólo se debe asegurar que esté activado el log.
- **Usar técnicas de diseño de software seguras para proteger el servicio web de ataques.** Un servicio web mal diseñado puede permitir a una persona hábil realizar un ataque exitoso. Si se utilizan técnicas seguras se minimiza el riesgo de incidentes.

Lo anterior describe cómo proteger el servicio web en sí, pero para evitar varios de los problemas mencionados hay que implementar seguridad a nivel de mensajes. Hay tres formas de implementar seguridad en relación a los mensajes en un servicio web^[6]:

- **SSL/TLS.** TLS es un protocolo de seguridad que proporciona autenticación y privacidad de la información entre extremos sobre internet, mediante el uso de criptografía.
- **Encriptación XML y XML firmado.** Estándares de W3C permiten firmar y encriptar contenido XML. Se puede aplicar a parte de los mensajes SOAP pero no hay forma estándar de informar al que recibe un mensaje sobre cómo se aplicó la encriptación.
- **WS-Security.** Protocolo que aplica la encriptación XML y XML firmado a los mensajes SOAP de manera estandarizada. Provee autenticación, confidencialidad e integridad a los mensajes SOAP.

El problema de utilizar TLS es que si existe un proxy entre los extremos, el servidor vería la petición que llega del proxy, no del cliente. Esto se puede solucionar si el proxy tiene una copia de la clave y el certificado del cliente, pero es una solución algo engorrosa. El cliente se encuentra tras el proxy de la PUCV por lo que tendría que solucionar este problema en caso de aplicar TLS.

WS-Security solo se puede aplicar a servicios web basados en XML, por razones obvias. Por esto es que al utilizar servicios web RESTful sólo se podría aplicar TLS para implementar seguridad. WS-Security pareciera ser la mejor forma de implementar seguridad en un servicio web basado en XML. Sin embargo puede presentar problemas. Puede ser susceptible a ataques de reenvío de mensajes. Este problema se puede mitigar utilizando números de secuencia y proporcionándole tiempo de vida a los mensajes.

CAPÍTULO 4: SOLUCIÓN PROPUESTA

A continuación se muestra un esquema sobre cómo se va a implementar la solución al problema.

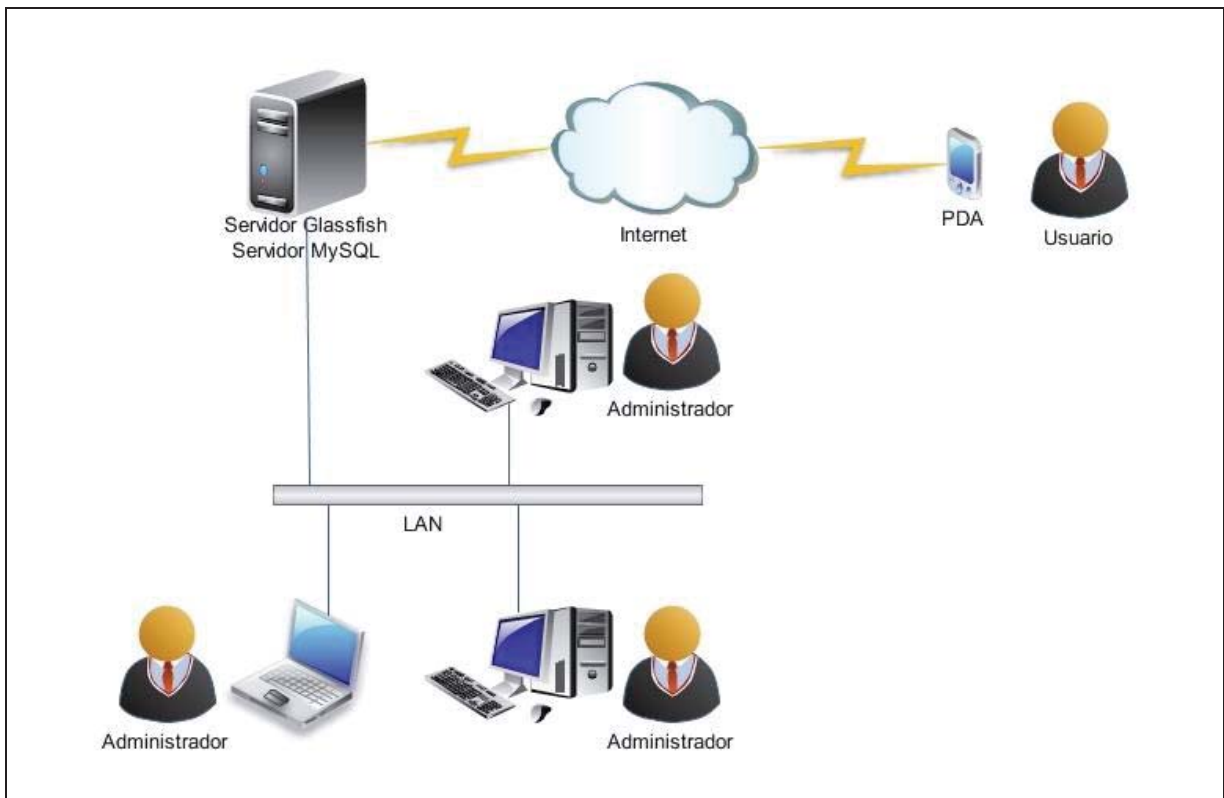


Figura 4.1 Esquema de implementación

La idea es la siguiente, utilizar un servidor que almacene los formularios ingresados en un sistema web. El sistema web se utiliza para la administración de usuarios, creación de informes y visualización de formularios ingresados por los trabajadores. Mediante el dispositivo móvil se puede actualizar tanto la información de usuarios como de órdenes de trabajo, sincronizar información de formularios y se pueden ingresar y enviar formularios, entre otras cosas. Esto gracias a la inmediata conexión que posee el dispositivo móvil con el servidor a través de internet. Para poder utilizar estas funcionalidades, el usuario del dispositivo móvil debe estar previamente registrado en el sistema. Dicho registro debe ser realizado por medio del Administrador en el servidor central. De esta manera, este último es el encargado de agregar o eliminar un usuario del sistema, por consiguiente la primera vez que se utilice el software, dicho usuario debe estar previamente registrado para poder acceder al sistema.

La figura 4.2 muestra la forma en que se lleva a cabo un ensayo desde el momento en que el cliente hace el pedido. El siguiente diagrama corresponde al flujo actual de trabajo, sin la implementación del dispositivo móvil.

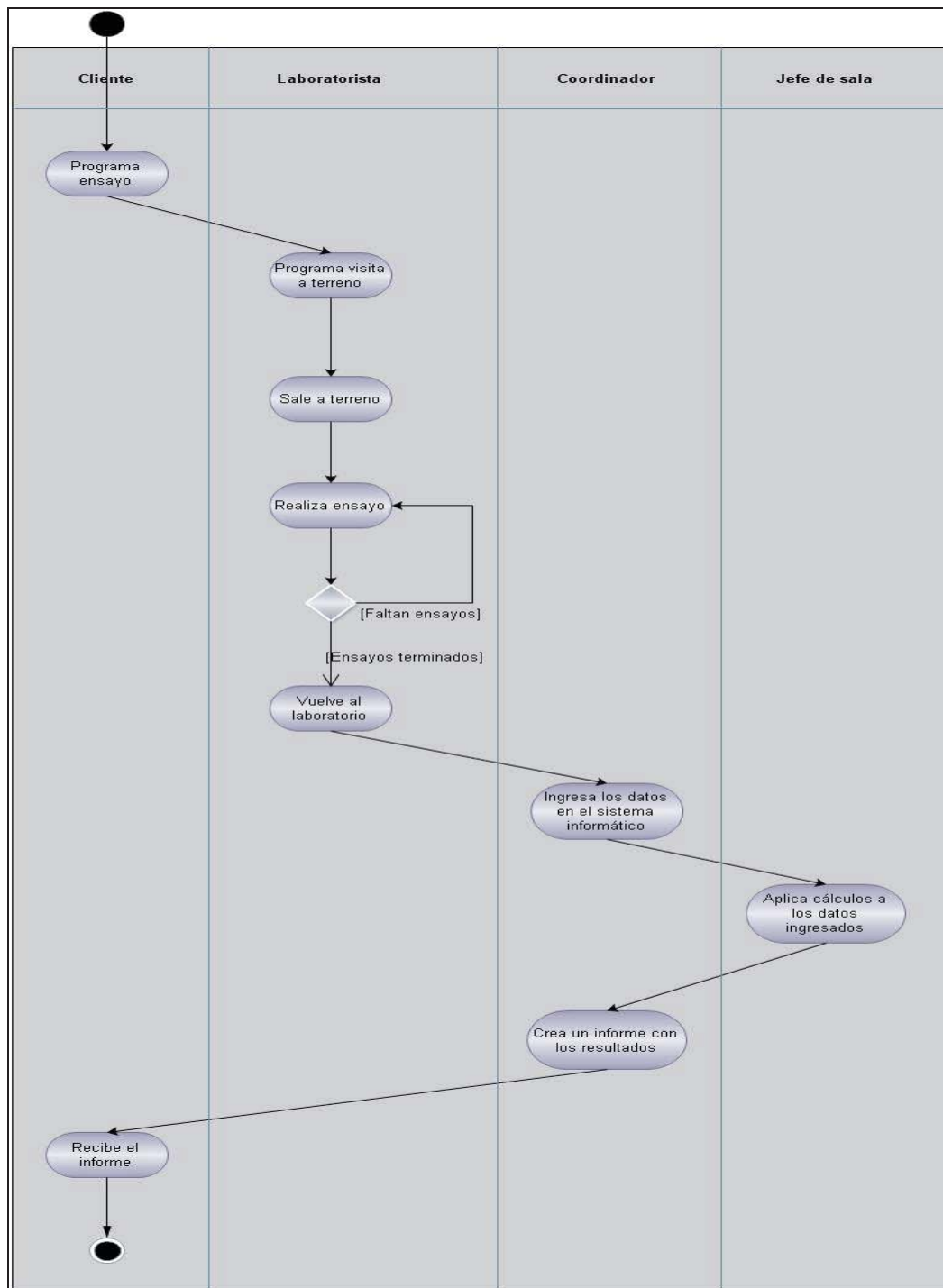


Figura 4.2 Diagrama de actividades del negocio

El cliente contacta a la empresa y solicita un ensayo en terreno. La solicitud se analiza y si se

llega a un acuerdo se programa la ida a terreno asignando el trabajo a un laboratorista. El laboratorista realiza los ensayos solicitados, vuelve al laboratorio con los datos capturados y el coordinador se encarga de ingresarlos en el sistema informático actual. El sistema informático aplica los cálculos correspondientes y el coordinador crea un informe en base a los resultados obtenidos según los datos ingresados.

La figura 4.3 muestra cómo funciona el mismo proceso de la figura 4.2, pero corresponde al flujo de trabajo futuro, es decir con el sistema de formularios digitales implementado.

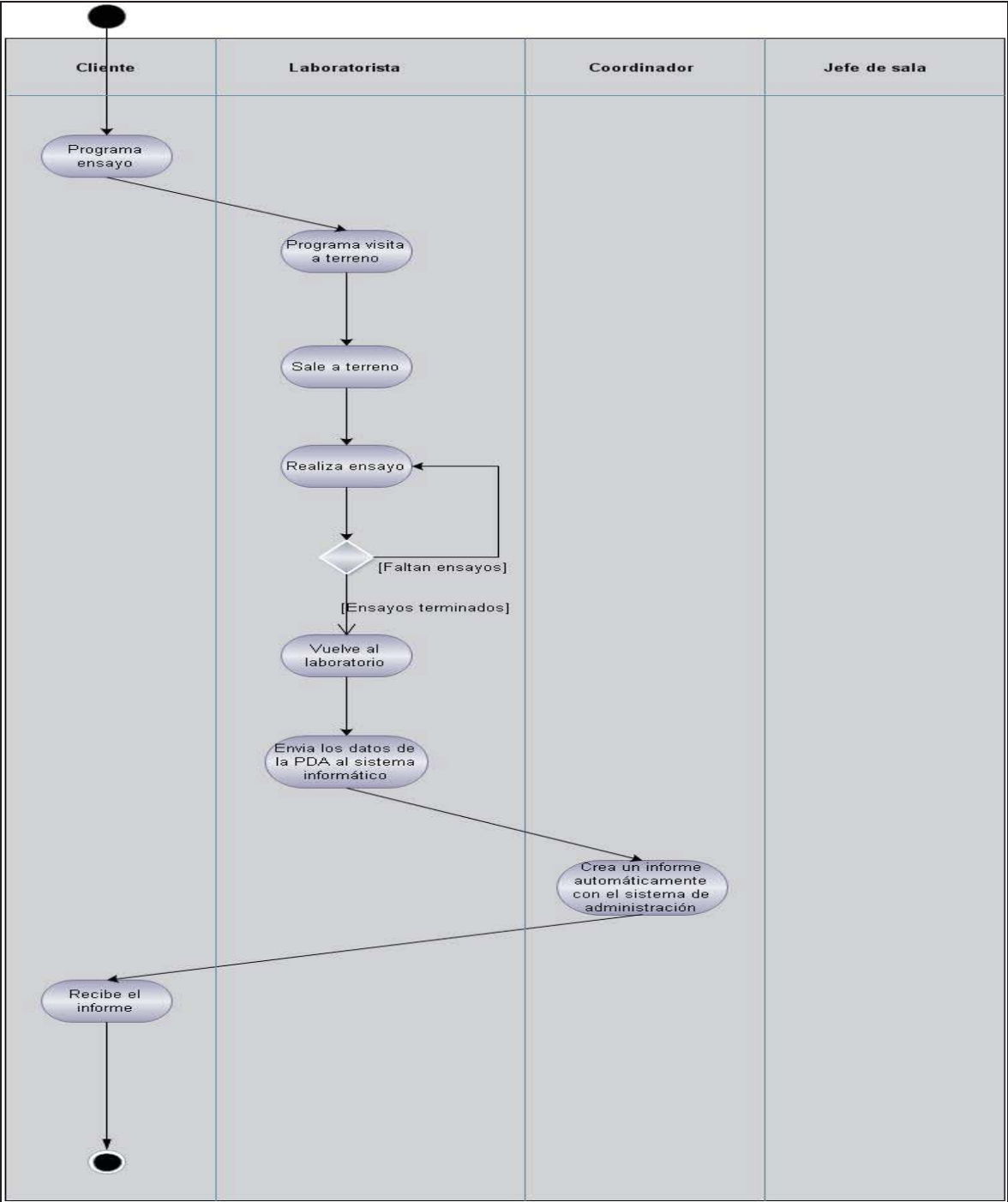


Figura 4.3 Diagrama de actividades del negocio con dispositivo móvil

Para el mejor entendimiento de cómo opera el flujo futuro de trabajo, que se observa en la Figura 4.3, se presenta a continuación, un supuesto, que facilita la explicación con más detalle.

Suponga que es el primer día de trabajo de un laboratorista con el sistema de formularios digitales. El administrador lo ha agregado previamente al sistema, registrándolo a través del software de administración, al que se accede a través de un navegador web. El laboratorista debe actualizar la información de usuarios desde el dispositivo móvil. El software le indica si se llevó a cabo el proceso de manera exitosa y si ese es el caso, el laboratorista puede realizar el proceso de ingreso a la aplicación, identificándose con su RUT y clave, eventualmente éstas coincidan con las registradas en el sistema, el laboratorista se encontrará logueado en el software y podrá acceder a las distintas funcionalidades, permitidas en base a su rol, disponibles en el sistema del dispositivo móvil. En caso que la actualización de usuarios no finalice de manera exitosa, no se podrá ingresar al software implementado en el dispositivo móvil.

Una vez realizado el pedido por el cliente y programado el ensayo, el laboratorista sale a terreno, hace sus mediciones e ingresa los datos en el dispositivo. Lamentablemente no se pudo llegar a una conclusión en cuanto a si una firma escrita en un dispositivo móvil es válida, por lo que el laboratorista tendrá que seguir llenando en papel aquellos documentos que requieran firma como lo son las órdenes de trabajo. Como normalmente no tienen acceso a internet en terreno, la información ingresada será enviada al servidor una vez que se vuelva a la oficina. En el servidor de la empresa se tiene instalado Glassfish, en el cual se levantará un servicio web. Así, cuando el laboratorista vuelva a la oficina e indique en el dispositivo móvil el envío de los datos ingresados al servidor, el dispositivo móvil realiza la transferencia de los datos a través del servicio web y sobre la red de la empresa. De ahí en adelante, el manejo de los datos es automatizado y los cálculos que se deben realizar los lleva a cabo el sistema. El coordinador solo debe pedir el informe pertinente para que se genere automáticamente y se pueda entregar al cliente.

4.1 Metodología

Para lograr los objetivos del presente proyecto, ha sido de vital importancia incorporar un método de trabajo para la mejora del proceso, el cual incorpora una serie de actividades, donde las dos primeras intervienen para lograr los objetivos del sistema, dichas actividades son las siguientes:

1. Análisis del proceso actual de trabajo.
2. Identificación de la mejora.
3. Diseño de la mejora.
4. Implementación del cambio de proceso de trabajo.
5. Afinación del nuevo proceso de trabajo.

Existe una serie de actividades que se debieron desarrollar para alcanzar los objetivos del presente proyecto. A continuación se detallan las actividades más importantes:

- Analizar el funcionamiento actual de trabajo de la empresa LE-PUCV e investigar los inconvenientes que posee en su forma de trabajar.
- Presentar una propuesta al cliente, definiendo básicamente que es lo que se pretende con el proyecto, de qué manera afectará a la empresa, a su personal, a sus clientes y por qué es importante la mejora.
- Luego de estar aceptada la propuesta, se coordinan las reuniones con la gente de LE-PUCV.
- Se realizan citas con la gente de la empresa, dirigidas por el paradigma de desarrollo escogido. En éstas, se recopilan las ideas del cliente, se define qué es lo que se espera del proyecto y por último se captura la información necesaria para una posterior toma de decisiones.
- Una vez claro lo que se espera para el proyecto, se comienza con la toma de requerimientos, definiendo cada uno de ellos detalladamente y estableciendo la estandarización de la información.

Teniendo claro los puntos anteriores se puede dar paso a describir y definir el sistema que se llevará a cabo, indicando así las principales funciones del mismo. Todas las actividades definidas en los puntos anteriores, unido con las actividades y tareas que propone la metodología escogida y que se definen en una sección posterior, completan el análisis que se llevará a cabo para desarrollar este sistema.

4.2 Desarrollo

4.2.1 Estudio de factibilidad

4.2.1.1 Factibilidad técnica

LE-PUCV tiene un computador que se encuentra disponible para ser usado en el contexto del presente proyecto. Las características de dicho computador son las siguientes:

- Procesador Intel Core 2 Quad, 2.4 Ghz.
- 8 GB de memoria RAM.
- Disco duro con capacidad de almacenar 500 GB.
- Sistema operativo Windows Server 2003.
- Grabador de DVD.
- Tarjeta de video GeForce 8400 GS con 256 MB de memoria.

El computador disponible además se encuentra conectado a internet con una conexión de 80 MB/s.

El equipo desarrollador tiene disponible un computador con las características que se denotan a continuación:

- 1 GB de memoria RAM
- Disco duro con capacidad de 80 GB de almacenamiento.
- Procesador AMD Athlon XP 1.2 Ghz.
- Sistema operativo Windows XP SP3.
- Lector/grabador de DVD.

Además del computador mencionado anteriormente se tiene acceso a un notebook Dell con las siguientes características:

- 3 GB de memoria RAM.
- Disco duro con capacidad de 160 GB de almacenamiento.
- Procesador Intel CoreDuo 2.1 Ghz.
- Sistema operativo Windows XP SP3
- Sistema operativo Kubuntu 9.10.
- Lector/grabador de DVD.

Según la página web de Sun^[7] el servidor de aplicaciones Glassfish v3, contemplado a utilizar para el desarrollo de la página web, que va a funcionar como interfaz de usuario para el administrador y también, para gestionar el servicio web con el que se conecta el dispositivo móvil, se puede instalar en Windows 2003 Server con los siguientes requisitos:

- 1 GB de memoria RAM.
- 500 MB de espacio en disco.
- Java SE 6 (JDK 6 update 17 or greater)

Hay requisitos, como por ejemplo velocidad del procesador, que no se especifican en la página oficial, pero dada la gran potencia del computador que se tiene a disposición, está claro que el funcionamiento del servidor Glassfish va a ser óptimo. En cuanto a la velocidad de la conexión a internet, también se puede concluir que es más que suficiente, ya que la carga de conexiones va a ser bastante baja. En un día trabajan alrededor de 5 usuarios que salen a terreno y aunque se conecten todos a la misma vez no van a poder sobrecargar el servidor.

Considerando los requisitos anteriores, se asume un funcionamiento adecuado por parte del servidor en ambos equipos de los desarrolladores.

Según el proveedor del software los requerimientos mínimos para el funcionamiento de Visual Studio 2005 son los siguientes:

- Sistema operativo Windows XP SP2.
- Procesador de 600 Mhz.
- 192 MB RAM.
- 2 GB Libres en el disco duro.
- Lector de DVD.

De esta manera se verifica que Visual Studio 2005 funciona perfectamente en los equipos que se tiene a disposición.

No se han encontrado requisitos mínimos para Mysql, pero se concluye, en base a la experiencia que se posee, que funciona con normalidad en todos los equipos mencionados anteriormente.

Se concluye finalmente, considerando el análisis realizado, que no existe impedimento técnico para poder llevar a cabo el presente proyecto.

4.2.1.2 Factibilidad económica

La factibilidad económica se lleva a cabo para determinar si es rentable o no para la empresa desarrollar un proyecto. Se toman en cuenta los gastos e ingresos y se aplica la ingeniería económica para obtener un resultado definitivo y científico. A continuación se analizan los flujos asociados al proyecto, se crea un diagrama con dichos flujos y se calcula el VAN que ayuda a tomar una decisión sobre la realización del proyecto.

El equipo de trabajo consiste en dos estudiantes de ingeniería informática que le dedicarán tres días por semana y 6 horas diarias al proyecto. El monto que recibe cada ingeniero son \$3000/hora por lo que el costo mensual en sueldos corresponde a \$216000 por persona, lo que da un total de \$432000. El proyecto tiene una duración de 6 meses, desde junio hasta noviembre.

Una de las herramientas que se utilizan para el desarrollo del software es Visual Studio 2005 que, a diferencia de las demás herramientas, no es de uso liberado. El costo de la licencia equivale a \$570100 y se paga una sola vez al inicio del proyecto.

Para realizar el desarrollo del software del dispositivo móvil de una manera más eficiente se decide adquirir un equipo, ya que si bien Visual Studio 2005 viene con un emulador de dicho dispositivo, significa un riesgo no poder verificar que el software funcione correctamente en el equipo para el cual está destinado. El dispositivo móvil escogido es la PDA IPAQ 116 que tiene un costo de \$170000.

El ingreso que percibe la empresa desarrolladora es el costo del software, que equivale a \$5900000. Este monto se paga en dos partes, 50% al iniciar el proyecto y 50% al realizar la entrega final del software.

En resumen, los costos son los siguientes:

- \$432000 mensual en pago de sueldos.
- \$570100 en licencias.
- \$170000 en equipamiento informático.

El ingreso total corresponde a \$5900000.

A continuación se presenta el diagrama de flujos correspondiente al estudio de factibilidad económica.

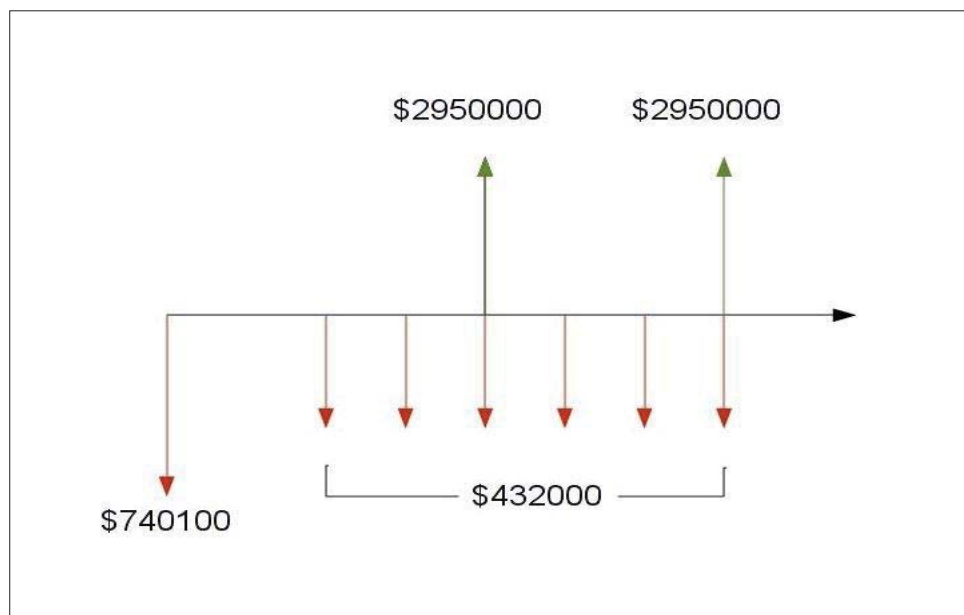


Figura 4.4 Diagrama de flujos factibilidad económica

Teniendo los flujos establecidos se calcula el VAN:

$$\begin{aligned} \text{VAN} &= -740100 - 432000(P/A, 12\%, 6) + 2950000(P/F, 12\%, 3) + 2950000(P/F, 12\%, 6) \\ \text{VAN} &= -740100 - 432000 * 4,111 + 2950000 * 0,712 + 2950000 * 0,507 \\ \text{VAN} &= 1079998 \end{aligned}$$

Con un VAN resultante mayor a cero, se concluye que el proyecto es rentable y se puede llevar a cabo.

4.2.1.3 Factibilidad operativa

El software actualmente instalado en el servidor disponible es el siguiente:

- AppServer: Un software que integra Apache, MySQL, PHP, phpMyAdmin.
- Tomcat 6: Servidor web con soporte de servlets y JSP escrito en Java desarrollado por Apache.

El hecho que esté instalado MySQL es beneficioso, ya que es el SGBD que se tiene considerado a utilizar. En cambio Tomcat 6 utiliza el mismo puerto que Glassfish por defecto y puede causar conflictos. Si es que hay una página web implementada en Tomcat se puede traspasar a Glassfish sin problemas. Otra posible solución es configurar Glassfish para que utilice otro puerto.

En la empresa se encuentra una persona encargada de los sistemas informáticos, a la cual se le entregará la documentación del proyecto, con la cual no debería tener ningún problema en mantener el software en funcionamiento.

4.2.1.4 Factibilidad legal

El software que se va a utilizar en el servidor es Glassfish. Glassfish es un servidor de aplicaciones gratuito y de código libre que se distribuye bajo las licencias CDDL y GNU GPLv2^[8]. Se tiene contemplado que la PDA utilice Windows Mobile como sistema operativo, el cual es propiedad de Microsoft y es software propietario. Microsoft no impone costos a los usuarios por el uso de dicho sistema operativo, sino que el costo recae en los desarrolladores de aplicaciones y en los creadores de dispositivos que usan dicho sistema operativo. Por lo tanto para el cliente no existe impedimento legal para la implementación del software.

4.2.2 Gestión de riesgos

4.2.2.1 Identificación de riesgos

Analizando los problemas que pueden surgir durante el proyecto se han identificado los siguientes riesgos:

- El proyecto no se puede llevar a cabo por falta de presupuesto.
 - Al conversar con la persona encargada de los asuntos económicos de la empresa, se ha comunicado, que no existe un presupuesto para el presente proyecto durante el año 2010. En general, la empresa pasa por un período complicado con un presupuesto limitado, por lo que es un riesgo que hay que tener presente. Si bien puede que el proyecto no se cancele, es posible que haya que abaratar costos, lo que se analizará cuando se sepa más al respecto. Según la persona a cargo del financiamiento para el proyecto, se conversará con autoridades superiores para tomar una decisión final acerca del financiamiento del software.
- La interacción con otros proyectos en desarrollo para el mismo cliente no funciona de manera óptima.
 - En paralelo con el desarrollo del presente sistema, se está llevando a cabo otro proyecto que utilizará los datos ingresados a través de los dispositivos móviles. Este otro proyecto aplicará ciertos cálculos a los datos, los cuales son necesarios para crear el informe que se entrega al cliente. Esto significa que habrá una interacción con un proyecto externo, por lo que es posible que los requerimientos del otro proyecto afecten de alguna forma el diseño del presente proyecto. Esto implica un riesgo, ya que si no existe una coordinación de ambos proyectos, puede que uno de éstos se vea afectado negativamente.
- La transferencia inalámbrica de datos no cumple con los requisitos de seguridad de la empresa.
 - En cuanto a lo que es confidencialidad de la información transmitida por internet, el cliente ha solicitado que los datos se envíen de forma confidencial y segura. Se puede implementar una transferencia segura pero siempre existirán riesgos de pérdida de datos confidenciales, ya sea por mal manejo de los dispositivos por parte de los trabajadores o engaños sofisticados a los usuarios.

4.2.2.2 Análisis de riesgos

En la siguiente tabla se lleva a cabo el análisis de riesgos. La probabilidad se divide en los niveles muy baja (<10%), baja (10-25%), moderada (25-50%), alta (50-75%) o muy alta(>75%) y los efectos pueden ser insignificante, tolerable, serio o catastrófico^[9].

Tabla 4.1 Análisis de riesgo.

Tipo	Riesgo	Probabilidad	Efecto
Organizacional	El proyecto no se puede llevar a cabo por falta de presupuesto.	Moderada	Catastrófico
Tecnología	La interacción con los otros proyectos en desarrollo para el mismo cliente no funciona de manera óptima.	Baja	Serio
Requerimientos	La transferencia inalámbrica de datos no cumple con los requisitos de seguridad de la empresa.	Muy baja	Serio

Mediante el análisis de riesgos, se lleva a cabo el estudio de las causas de las posibles amenazas y los daños que éstas puedan producir. Lo primero a realizar en esta etapa del proyecto es identificar lo que queremos proteger, los eventuales riesgos existentes, así como su probabilidad de ocurrencia y su eventual efecto. Posteriormente es necesario realizar un proceso de gestión de riesgos, que involucre una estrategia de prevención, una estrategia de minimización y por supuesto un plan de contingencia aplicable a cada riesgo identificado.

4.2.2.3 Planificación de riesgos

La siguiente tabla indica las estrategias a llevar a cabo para gestionar los riesgos^[9].

Tabla 4.2 Planificación de riesgos.

Riesgo	Estrategia de prevención	Estrategia de minimización	Plan de contingencia
El proyecto no se puede llevar a cabo por falta de presupuesto.	-	Tener en cuenta alternativas más baratas al cotizar los dispositivos móviles.	Proponer las alternativas más económicas.
La interacción con los otros proyectos en desarrollo para el mismo cliente no funciona de manera óptima.	Mantener contacto con los integrantes de los otros proyectos y en lo posible seguir sus avances.	-	Analizar los problemas en conjunto con los integrantes de los otros proyectos y realizar los cambios necesarios.
La transferencia inalámbrica de datos no cumple con los requisitos de seguridad de la empresa.	Investigar cómo implementar una transferencia segura.	-	Llevar a cabo el proceso de sincronización inalámbrica de la información dentro de la red de la empresa.

Tras terminar una versión nueva de prototipo se debe llevar a cabo la supervisión de riesgos que implica valorar los riesgos descritos anteriormente y restablecer los datos cuando sea necesario, ya sea si han cambiado las probabilidades de ocurrencia y/o si han cambiado sus efectos.

4.2.3 Enfoque

Dentro de las alternativas que se manejan en cuanto a los enfoques de desarrollo de software que más se acomodan al proyecto, se escogió el enfoque orientado a objetos por las razones que se indican a continuación:

- Documenta, sin importar su implementación.
- Abstrae el mundo real, por lo que es más fácilmente entendible.
- Está más orientado a la intersección con el usuario que a procesos con entrada y salida.
- Facilidad de dividir el sistema en varios subsistemas.
- Fomenta la reutilización de componentes.
- Se basa en modelar el sistema mediante los objetos que forman parte de él y las relaciones entre estos objetos, con la finalidad de conseguir modelos que se ajusten mejor al problema real. Esto a partir del conocimiento del dominio del problema, sin pensar en términos de implementar el sistema en un computador, ya que permite pasar directamente del dominio del problema al modelo del sistema.
- Permite una mejor comunicación entre desarrolladores y clientes.
- UML como lenguaje para documentar y especificar.
- Finalmente, los cambios en los requisitos afectan de una menor manera que el análisis del enfoque estructurado.

4.2.4 Modelo de Desarrollo

La elección del modelo de desarrollo de software es, tal vez, una de las decisiones más importantes en un proyecto de sistema informático, y no es para menos, ya que éstos establecen las “actividades necesarias para transformar los requisitos de un usuario en software.”^[9] Es un modelo de proceso, por lo tanto, una “representación abstracta de un proceso de software” y la “utilización de un proceso inadecuado, probablemente reducirá la calidad o la utilidad del producto de software que se va a desarrollar.”^[9]

“Aunque existen muchos procesos diferentes de software, algunas actividades fundamentales son comunes para todos ellos;

1. Especificación del software. Se debe definir la funcionalidad del software y las restricciones en su operación.
2. Diseño e implementación del software. Se debe producir software que cumpla su especificación.
3. Validación del software. Se debe validar el software para asegurar que hace lo que el cliente desea.
4. Evolución del software. El software debe evolucionar para cumplir las necesidades cambiantes del cliente.”^[9]

Muchos autores concuerdan en que no existe un modelo de desarrollo “ideal” ^[9], por lo que se hace necesario buscar aquella alternativa que sea más conveniente y que cumpla de mejor manera con las necesidades del proyecto, o en defecto adaptar una metodología a las necesidades personales.

Antes de escoger el modelo de desarrollo se evaluaron distintas alternativas, seleccionando la más adecuada para el presente proyecto, de esta manera se llegó a la conclusión de utilizar el enfoque de desarrollo evolutivo exploratorio.

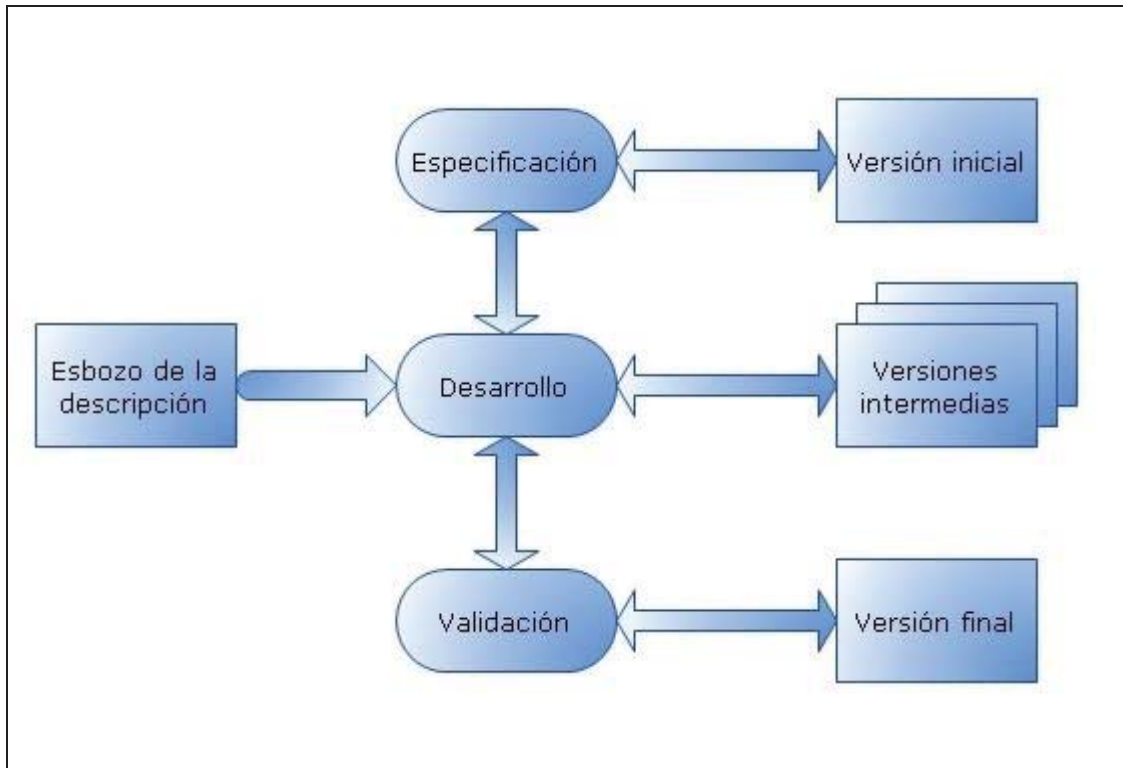


Figura 4.5 Metodología de desarrollo

El modelo de proceso de software guía este proyecto es el modelo evolutivo, utilizando el desarrollo exploratorio. Esto debido a que el cliente no tiene muy claro cuáles son los requisitos del software a desarrollar. En base a esta metodología se presentan periódicamente prototipos al cliente y se obtiene una visión mucho más clara sobre el rumbo que debe tomar el desarrollo del software. Como indica la figura 4.5, las actividades de especificación, desarrollo y validación se llevan a cabo de forma paralela, cobrando distintos niveles de importancia dependiendo de la etapa en que se encuentre el proyecto. Durante este proceso se van creando versiones intermedias del software las que se presentan al cliente, influyendo en el desarrollo de la siguiente versión.

¿Por qué usar el desarrollo evolutivo? Porque es un modelo de proceso del software eficiente cuando se trata de sistemas relativamente pequeños. Si se utiliza para un sistema de gran tamaño es muy factible que los cambios constantes generen una estructura débil y deficiente. Además es probable que no se escriba toda la documentación necesaria, debido a la gran cantidad de documentos que se van generando entre cada versión. Cabe resaltar que el desarrollo evolutivo es bastante útil cuando el cliente no tiene una idea muy clara sobre cómo quiere que el software esté estructurado, situación que sucede en el presente proyecto.

A continuación se puntualizan algunas características y ventajas del modelo de proceso de software escogido.

Las principales características del modelo evolutivo son^[10]:

- Se desarrolla una implementación inicial, exponiéndola a los comentarios del usuario y redefiniéndola a través de las diferentes versiones.
- Las actividades de especificación, desarrollo y validación se llevan a cabo concurrentemente, y tienen realimentación rápida a lo largo del proceso.
- Un primer sistema se desarrolla rápidamente, a partir de especificaciones abstractas, y se refina después, con la ayuda del cliente.
- Un enfoque evolutivo cumple con las necesidades inmediatas del cliente.
- La especificación se puede desarrollar de forma creciente.
- El mejor entendimiento de los usuarios por su problema se reflejará en el sistema software.

En base a lo expuesto anteriormente se pueden identificar algunas ventajas que fueron decisivas a la hora de la elección:

- Permite ir comprendiendo de mejor manera los requerimientos a medidas que se avanza en el proyecto.
- El cliente puede visualizar, en etapas tempranas, las características iniciales que tendrá el software.
- Adecuado para sistemas pequeños.

“En la producción de sistemas, un enfoque evolutivo para el desarrollo de software suele ser más efectivo que el enfoque en cascada, ya que satisface las necesidades inmediatas de los clientes. La ventaja de un proceso del software que se basa en un enfoque evolutivo es que la especificación se puede desarrollar en forma creciente. Tan pronto como los usuarios desarrollen un mejor entendimiento de su problema, éste se puede reflejar en el sistema software.”^[9]

“Para sistemas pequeños y de tamaño medio (hasta 500.000 líneas de código), el enfoque evolutivo de desarrollo es el mejor. Los problemas del desarrollo evolutivo se hacen particularmente agudos para sistemas largos y complejos con un período de vida largo, donde diferentes equipos desarrollan diferentes partes del sistema. Es difícil establecer una arquitectura del sistema estable usando este enfoque, el cual hace difícil integrar las contribuciones de los equipos.”^[9]

4.2.5 Herramientas y Tecnologías

El desarrollo del presente proyecto no sería posible sin el uso adecuado de herramientas y tecnologías existentes, es por esto, que es de vital importancia escoger inequívocamente las herramientas y tecnologías que se utilizarán para llevar a cabo el proyecto. A continuación se describe lo mencionado anteriormente, abordando primeramente las herramientas escogidas y finalizando con las tecnologías utilizadas.

Las herramientas escogidas y utilizadas por el grupo de desarrollo son:

- **Netbeans**
Netbeans es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) escrito en Java que se puede ejecutar en Windows, Linux, Mac y Solaris. En el presente proyecto se utiliza la versión 6.8.
- **Visual Studio**
Visual Studio es un IDE desarrollado por Microsoft. Permite programar en varios lenguajes como por ejemplo Visual C#, Visual C++, Visual J#, ASP.NET, Visual Basic.NET y ahora, en la versión 2010, F#. En Visual Studio se puede desarrollar aplicaciones de escritorio, sitios y aplicaciones web además de servicios web en cualquier entorno que soporte la plataforma .NET. Para el presente proyecto se utiliza la versión Visual Studio 2005.
- **Mysql**
Mysql es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. Mysql es software libre. En el presente proyecto se utiliza la última versión 5.1.
- **Glassfish**
Glassfish es un servidor de aplicaciones de código abierto desarrollado por Sun Microsystems para la plataforma Java EE. La versión 3.0, que se utiliza en el presente proyecto, cumple con el estándar Java EE 6.

Las tecnologías escogidas y utilizadas por el grupo de desarrollo son:

- **Java EE**
Java EE (Java Enterprise Edition) es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación y es considerada informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE.

- **.NET Framework CE**

El .NET Framework CE es una versión más limitada del .NET Framework, diseñado para ejecutarse en dispositivos que utilizan Windows CE como su sistema operativo, como por ejemplo PDAs y celulares. El .NET Framework es un software que se puede instalar en computadores que utilizan Windows como sistema operativo. Incluye una gran librería con soluciones a problemas de programación comunes y una máquina virtual que maneja la ejecución del software hecho para el framework. En el presente proyecto se utiliza la versión 2.0 del .NET Compact Framework.

- **Icefaces**

Icefaces es un framework de código AJAX que permite a desarrolladores de Java EE crear Aplicaciones de Internet Enriquecidas (RIA, por sus siglas en inglés). La ventaja que tiene este framework es que no es necesario programar en javascript para obtener la funcionalidad AJAX. El desarrollador hace todo en Java EE y el framework se encarga del resto.

4.2.6 Arquitectura

A continuación una cita que ayuda a explicar el término arquitectura del software.

“La arquitectura de software de un sistema de programación o computación es la estructura de las estructuras del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos.”^[10].

La arquitectura del software no necesariamente es de un sólo tipo, pueden convivir distintas formas en un sistema, como es el caso del presente proyecto.

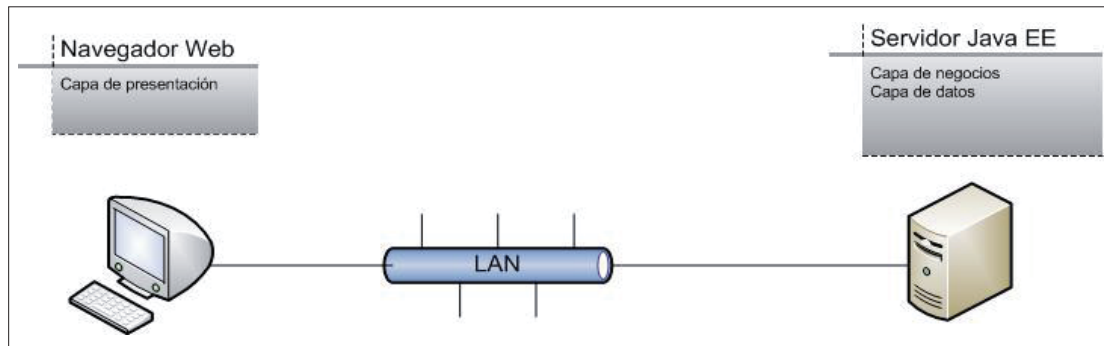


Figura 4.6 Arquitectura Cliente-Servidor

Se puede separar el sistema en desarrollo en dos partes, la página web de administración y el software en la PDA. La página web de administración utiliza la arquitectura cliente-servidor con cliente delgado. Esto significa que la capa de presentación se ubica en el cliente y la capa de negocios y de datos se ubica en el servidor web. Esto tiene como ventaja que al realizar un cambio en la capa de negocios o de datos, el cambio se realiza solamente en el servidor. Si no fuera un cliente delgado, los cambios se tendrían que aplicar en todos los clientes, dificultando la actualización del software y la aplicación de mejoras. La desventaja que tiene es que se utiliza la red en mayor proporción, lo que puede resultar en una respuesta más lenta para el usuario y la posible saturación de la red. Esto se puede agudizar si el cliente se conecta al servidor a través de internet.

En el software de la PDA, se ha implementado una arquitectura de separación de las tres capas; presentación, negocio y datos. Las capas de negocio y datos se implementan como librerías DLL por lo que hay una abstracción del punto de vista de la capa de presentación. La capa de datos se conecta a la base de datos SQL Server CE y se encarga de realizar las consultas que piden las capas superiores.

En la capa de presentación se utiliza una arquitectura centralizada, en donde una clase estática, controla la interfaz de usuario. Las clases que contienen las distintas pantallas, le piden a la clase controladora que cambie a una página específica cuando el usuario así lo indique.

4.2.7 Análisis y definición de requerimientos

Para llevar a cabo el proceso de captura de los requerimientos, se ha recurrido a entrevistas con el cliente, teniendo como objetivo entender exactamente lo que necesita. Con la información que se ha capturado de las diversas reuniones, se han creado diferentes diagramas que han ayudado al entendimiento de los requerimientos sin ambigüedades. Posteriormente se han dispuesto todos los conocimientos y habilidades para el eventual refinamiento de dichos requerimientos, los cuales han dado lugar a los distintos casos de uso que han definido los requisitos del cliente. En las siguientes páginas se pueden apreciar las diferentes tablas y diagramas que denotan los requerimientos mencionados recientemente.

Del diagrama de casos de uso general del sistema, que se puede observar más adelante en la figura 4.7 se pueden identificar los dos actores que interactúan con el sistema, los cuales se describen a continuación:

- **Usuario:** Corresponde al trabajador en terreno, quién utiliza el software del dispositivo móvil para ingresar datos. Mediante el cual, puede gestionar formularios y órdenes de trabajo y posteriormente transferir por medios inalámbricos la información almacenada en el dispositivo móvil hacia el servidor central.
- **Administrador:** Es el encargado de la parte estacionaria del sistema, es decir, la parte que se implementa en el servidor al que se conectan los dispositivos móviles. Gestiona los usuarios que trabajan en terreno, los roles que se pueden asociar a los mismos, puede visualizar los formularios disponibles y también crear informes.

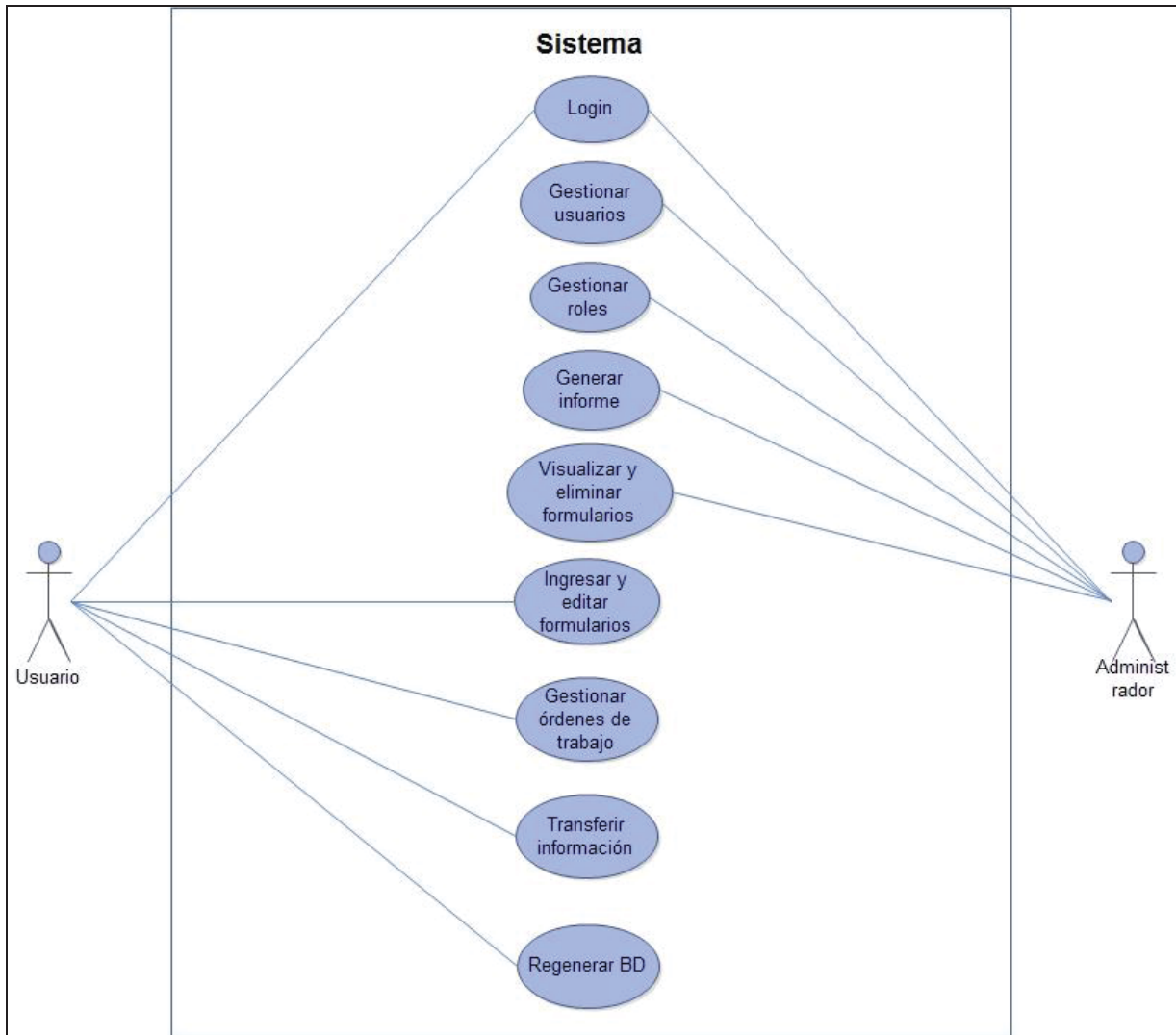


Figura 4.7 Caso de uso. General

La figura 4.7 ilustra a los dos actores que interactúan con el sistema. Se observan también las distintas acciones que tanto Administrador como Usuario llevan a cabo en el sistema.

En la figura 4.8 que se aprecia a continuación, se puede observar el diagrama de casos de uso asociado al administrador.

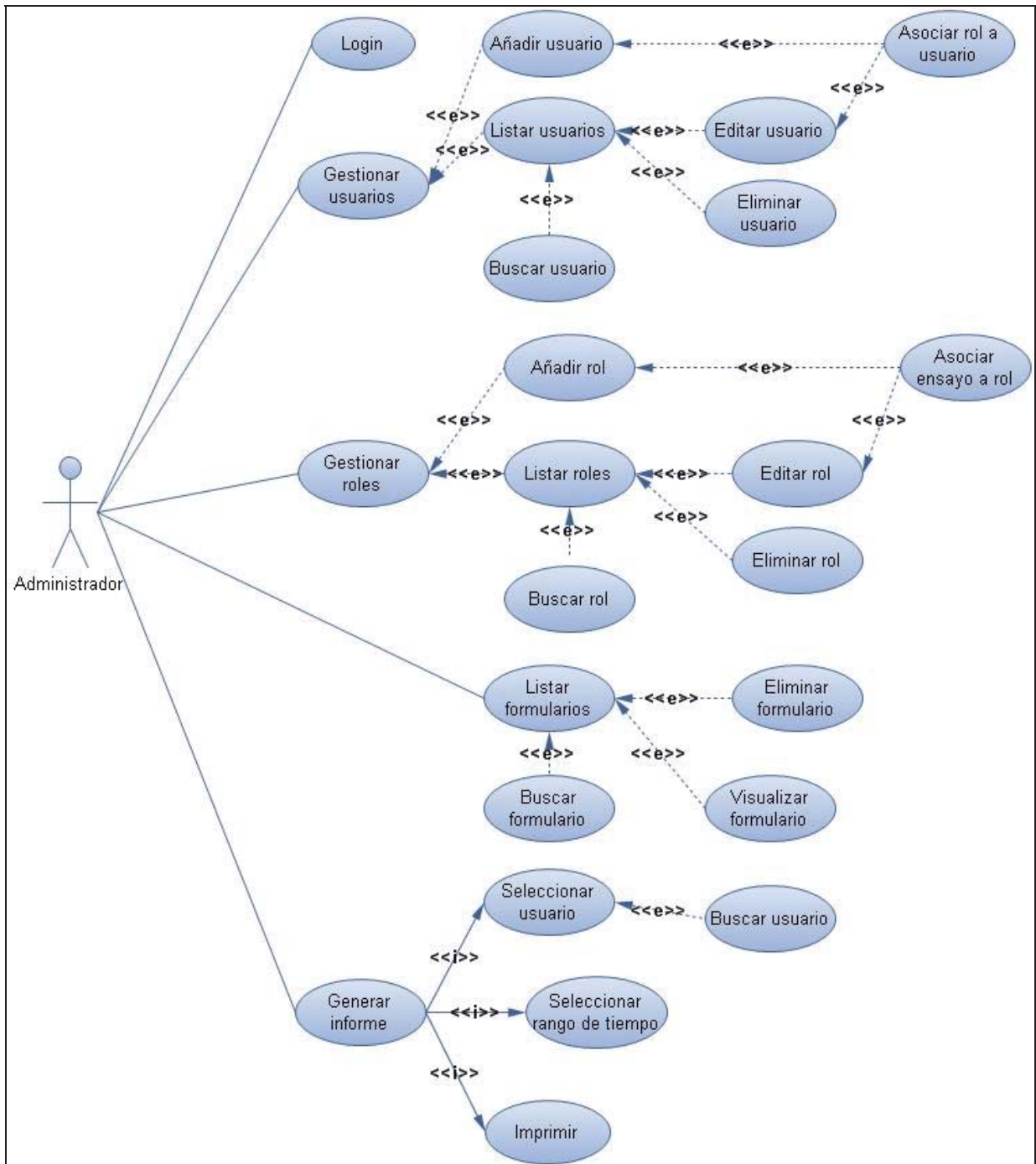


Figura 4.8 Caso de uso. Administrador

Los casos de uso narrativos correspondientes a la especificación del caso de uso de la figura 4.8, se encuentran en la sección ANEXO B: ESPECIFICACIÓN DE REQUERIMIENTOS, al final del documento.

En la siguiente figura se observa el diagrama de casos de uso asociado al usuario en terreno.

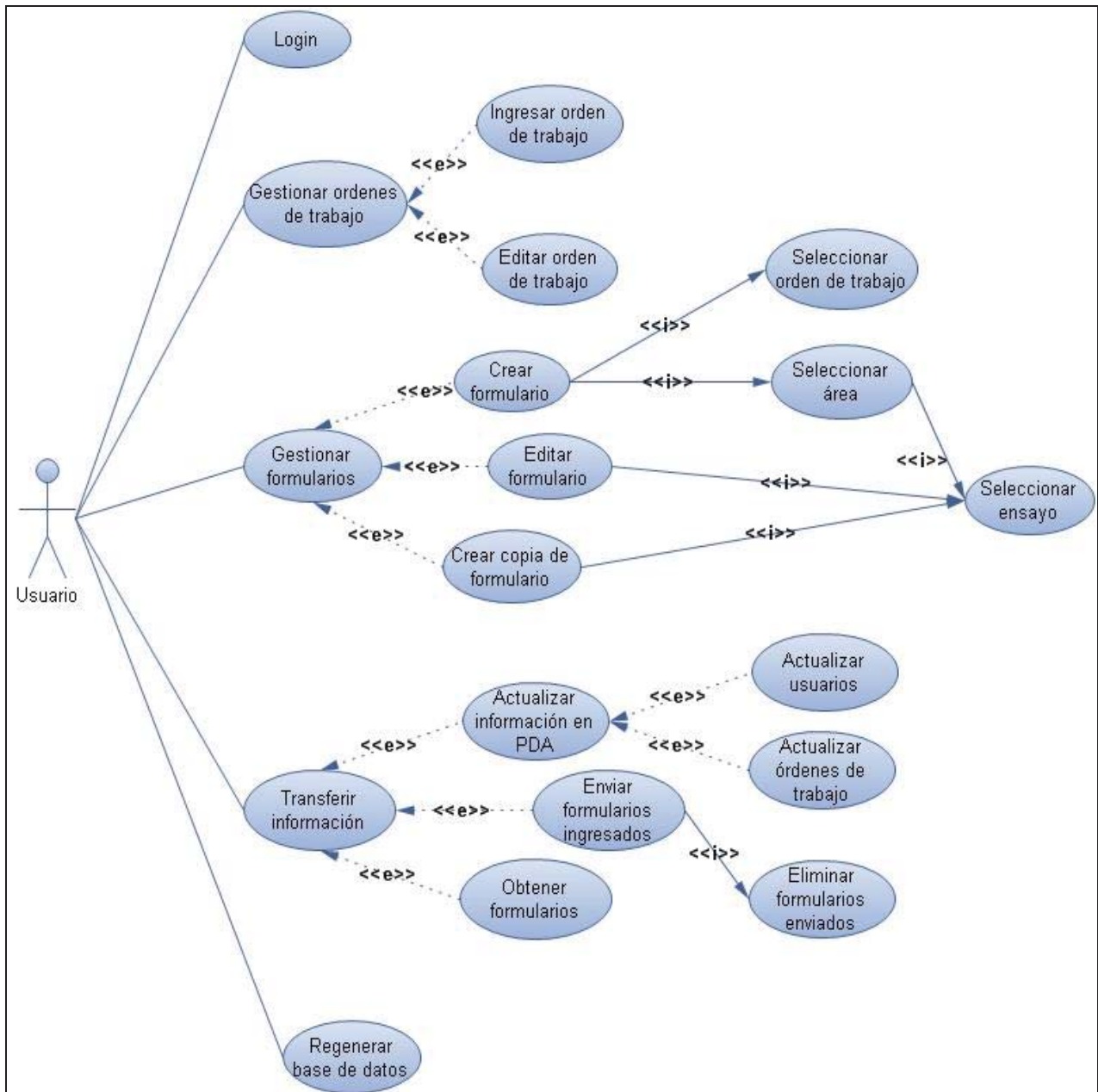


Figura 4.9 Caso de uso. Usuario en terreno

Los casos de usos narrativos correspondientes a la especificación del caso de uso de la figura 4.9, se encuentran en la sección ANEXO B: ESPECIFICACIÓN DE REQUERIMIENTOS, al final del documento.

4.3.8 Diseño

4.2.8.1 Modelamiento de la base de datos

Para almacenar la información ingresada por los usuarios en terreno de forma eficiente y para que dicha información se pueda procesar de manera óptima, es de suma importancia un buen diseño de base de datos. En las siguientes páginas se pueden apreciar los distintos modelos y sus eventuales normalizaciones que se han llevado a cabo, con el fin de obtener el modelo de diseño de base de datos más eficiente y representativo para el proyecto.

A continuación se aprecia el Modelo Entidad-Relación propuesto como solución:

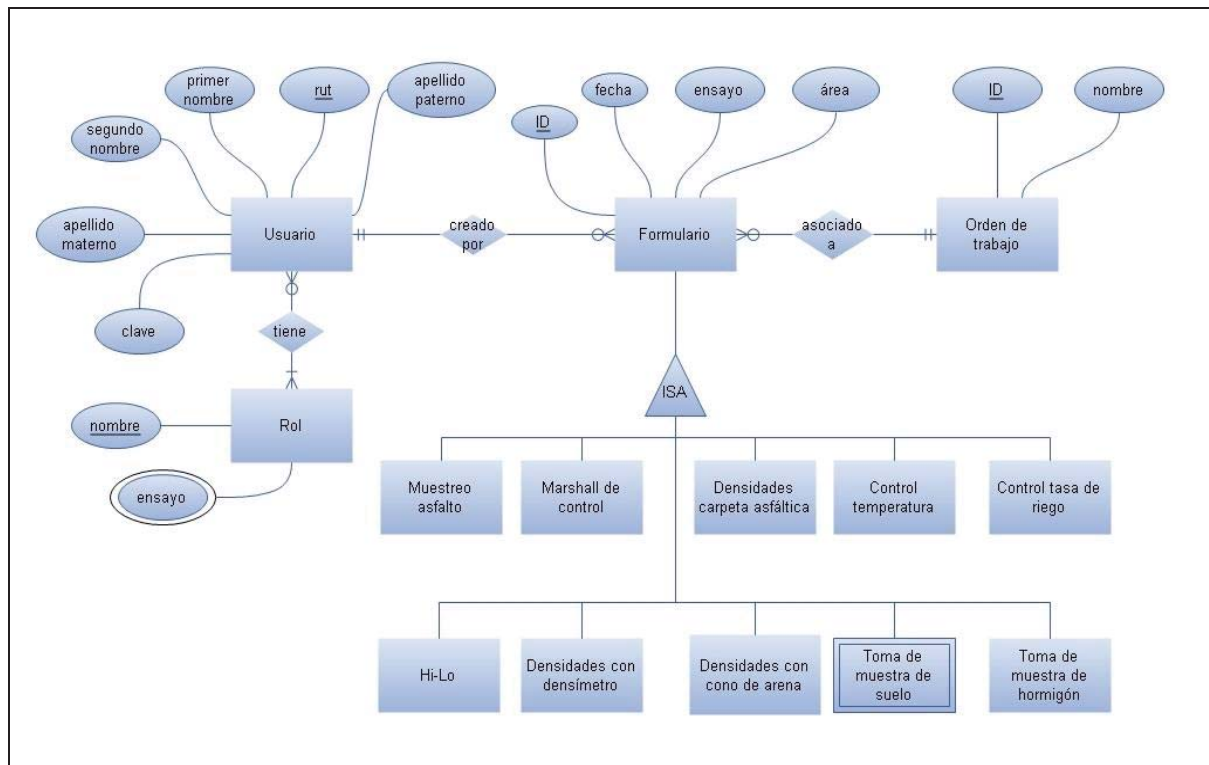


Figura 4.10 MER General

Es prudente mencionar que la figura 4.10 corresponde sólo a un MER General en primera instancia, ya que falta la especificación de cada entidad correspondiente a los formularios, lo que se puede observar en las figuras que se presentan a continuación.

La figura 4.11 presentada a continuación ilustra el formulario “Muestreo asfalto”.

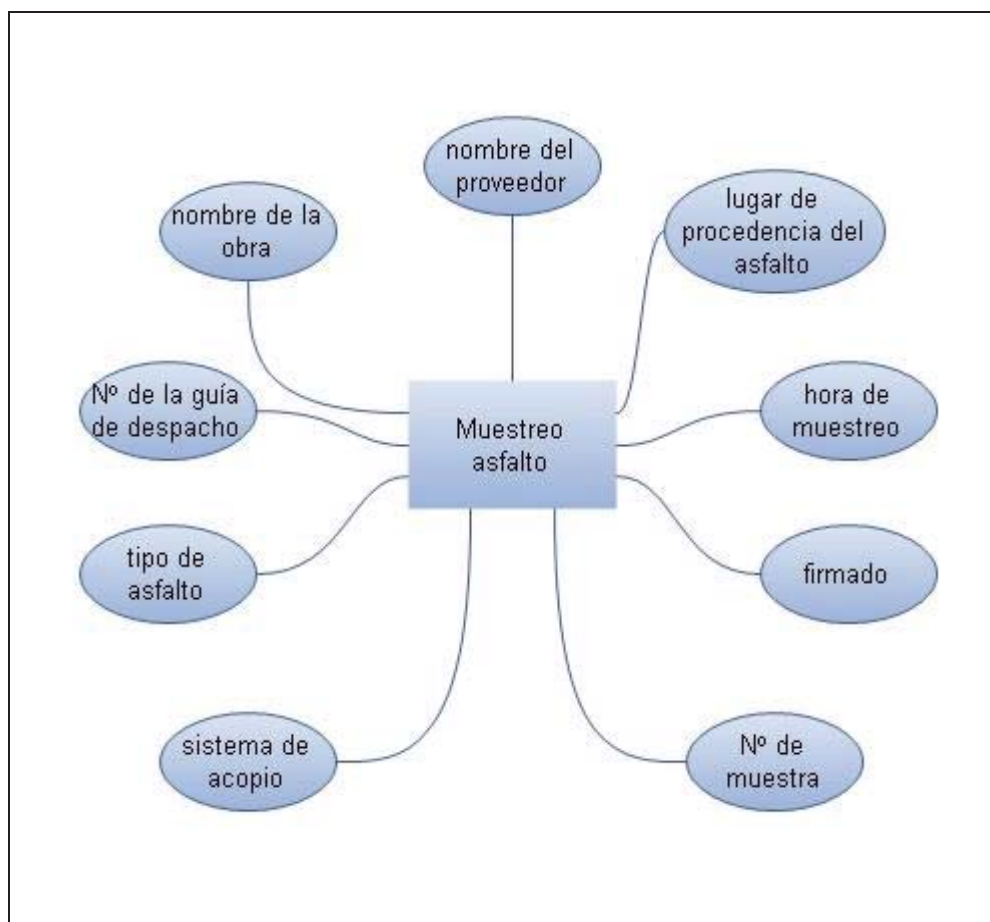


Figura 4.11 MER Muestreo asfalto

Para poder implementar este modelo en un SGBD (Sistema Gestión Base de Datos) como MySQL hace falta convertir el MER resultante en un Modelo Relacional y luego normalizar las nuevas entidades. Al terminar dicho proceso se obtiene el Modelo Relacional, que es una fiel representación de la base de datos que se utilizará en el proyecto.

A continuación se puede observar en la Figura 4.12 el MER del formulario “Marshall de control”.

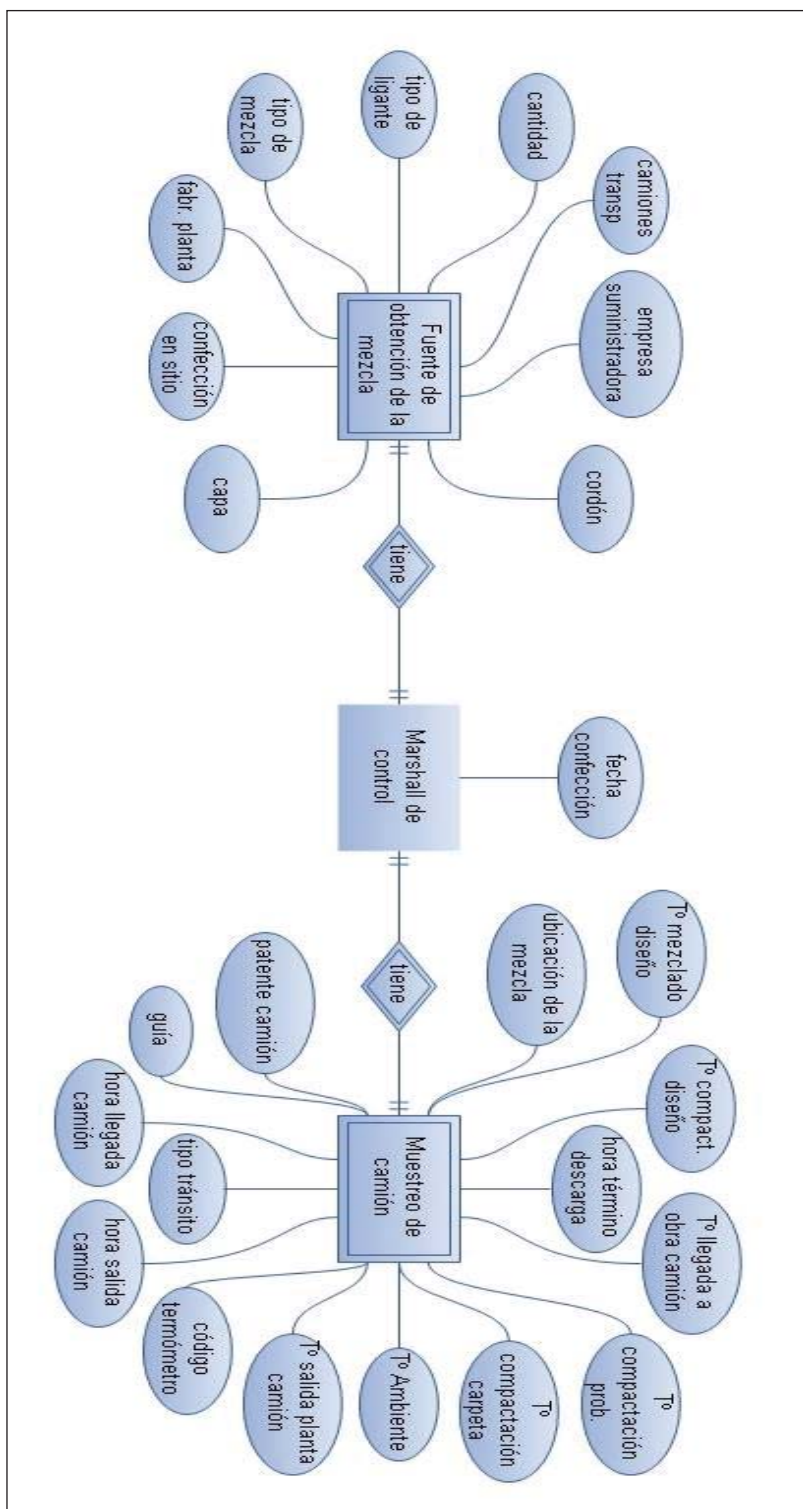


Figura 4.12 MER Marshall de control

La siguiente figura muestra el MER del formulario “Densidades carpeta asfáltica”:

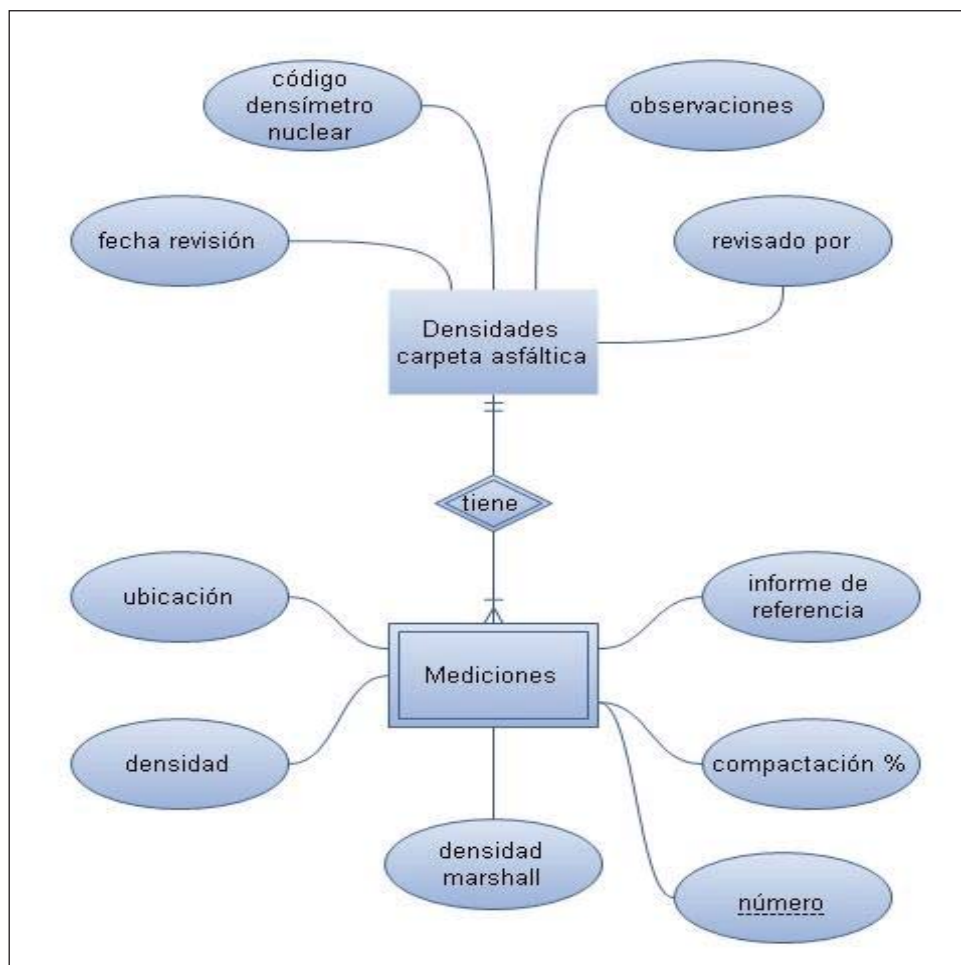


Figura 4.13 MER Densidades carpeta asfáltica

Para poder implementar este modelo en un SGBD (Sistema Gestión Base de Datos) como MySQL hace falta convertir el MER resultante en un Modelo Relacional y luego normalizar las nuevas entidades. Al terminar dicho proceso se obtiene el Modelo Relacional, que es una fiel representación de la base de datos que se utilizará en el proyecto.

El siguiente diagrama muestra el MER del formulario “Control de temperatura”:

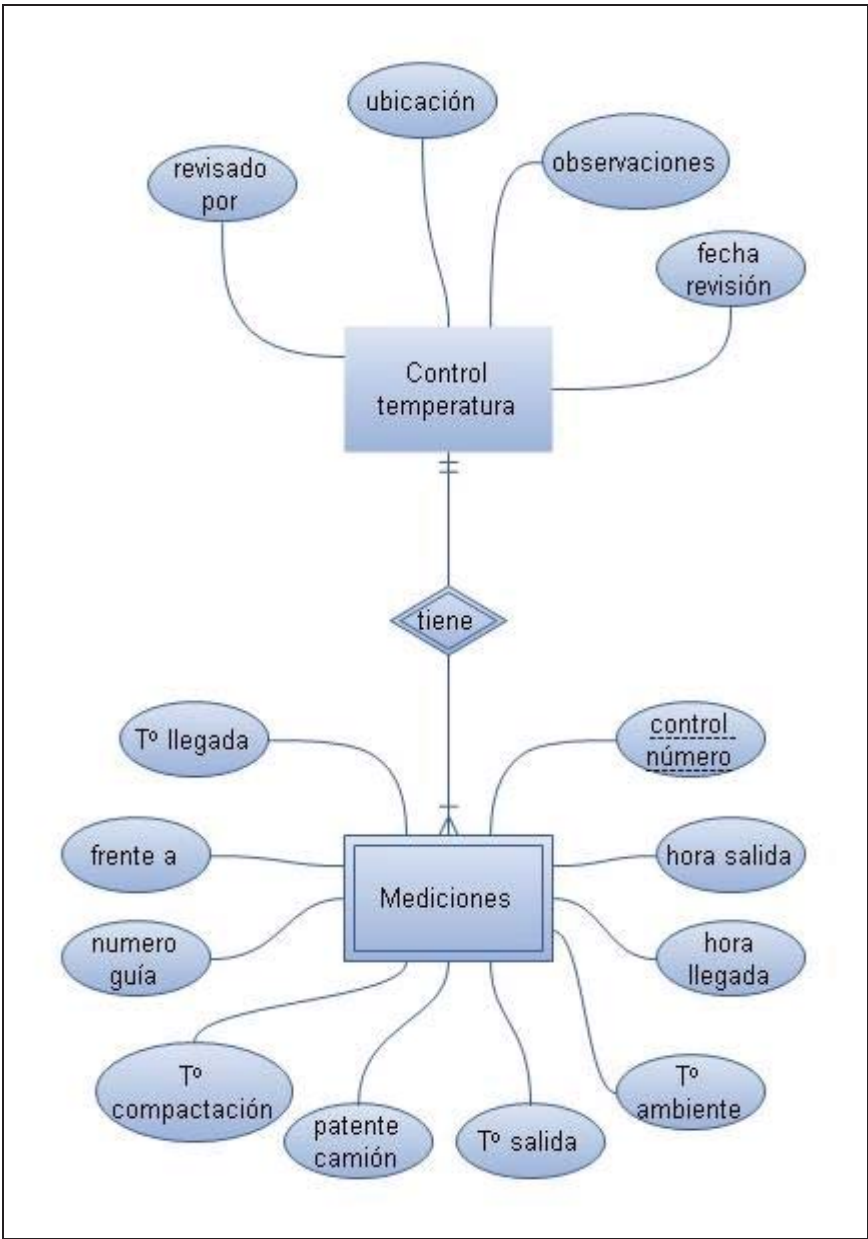


Figura 4.14 MER Control de temperatura

A continuación se presenta el MER del formulario “Control tasa de riego”:

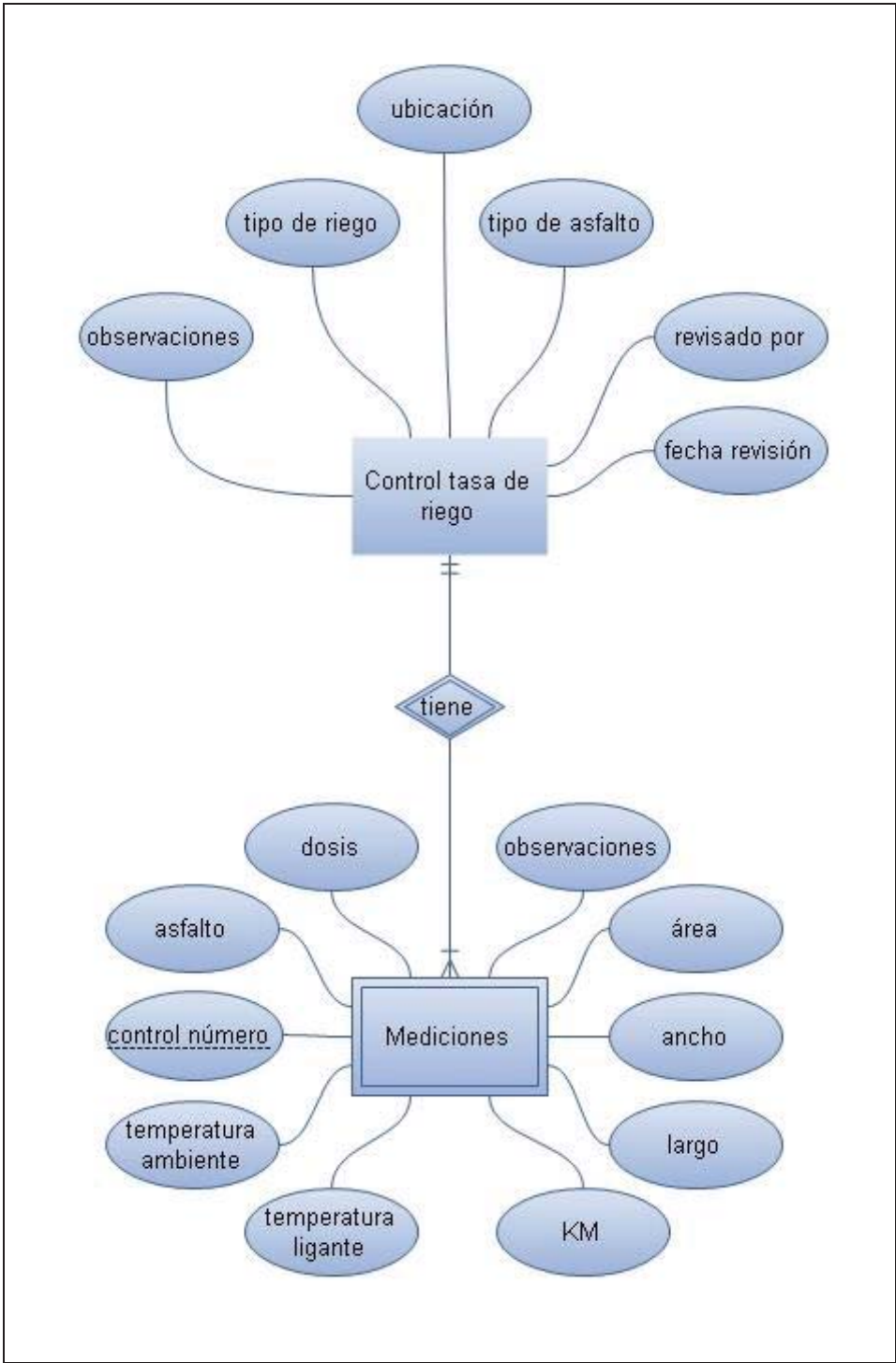


Figura 4.15 MER Control tasa de riego

La siguiente ilustración indica el MER del formulario “Hi-lo”:

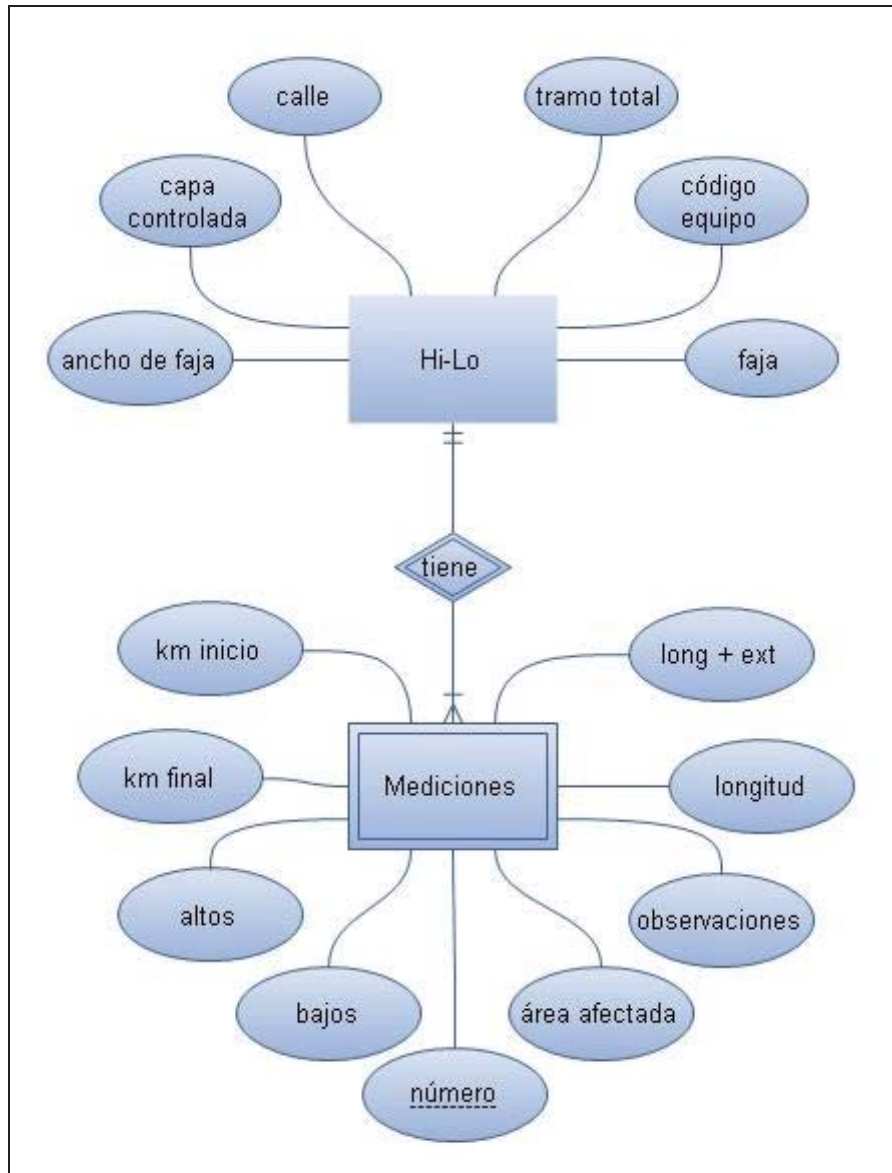


Figura 4.16 MER Hi-lo

Para poder implementar este modelo en un SGBD (Sistema Gestión Base de Datos) como MySQL hace falta convertir el MER resultante en un Modelo Relacional y luego normalizar las nuevas entidades. Al terminar dicho proceso se obtiene el Modelo Relacional, que es una fiel representación de la base de datos que se utilizará en el proyecto.

A continuación se observa el MER del formulario “Densidades con densímetro”:

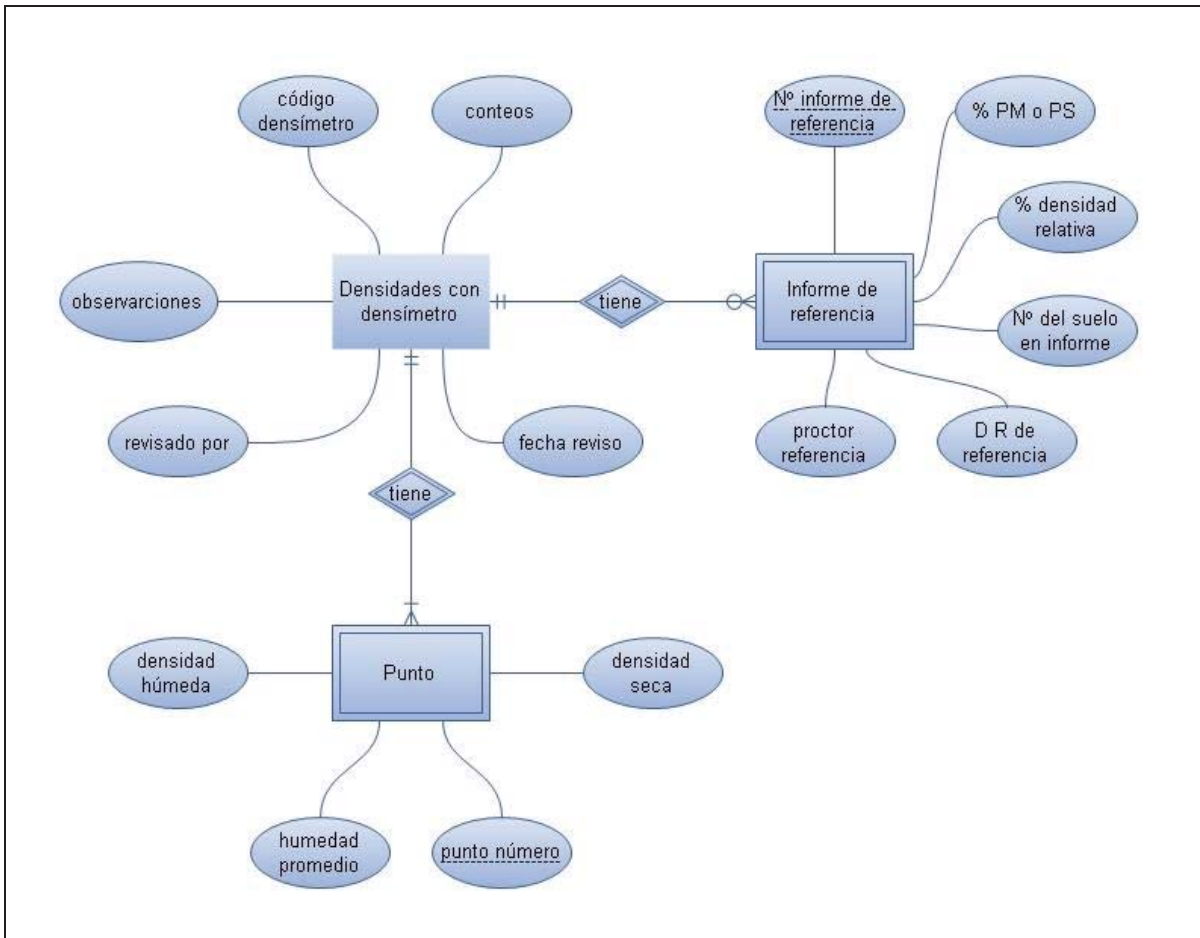


Figura 4.17 MER Densidades con densímetro

Para poder implementar este modelo en un SGBD (Sistema Gestión Base de Datos) como MySQL hace falta convertir el MER resultante en un Modelo Relacional y luego normalizar las nuevas entidades. Al terminar dicho proceso se obtiene el Modelo Relacional, que es una fiel representación de la base de datos que se utilizará en el proyecto.

En la siguiente imagen se aprecia el MER del formulario “Densidades con cono de arena”:

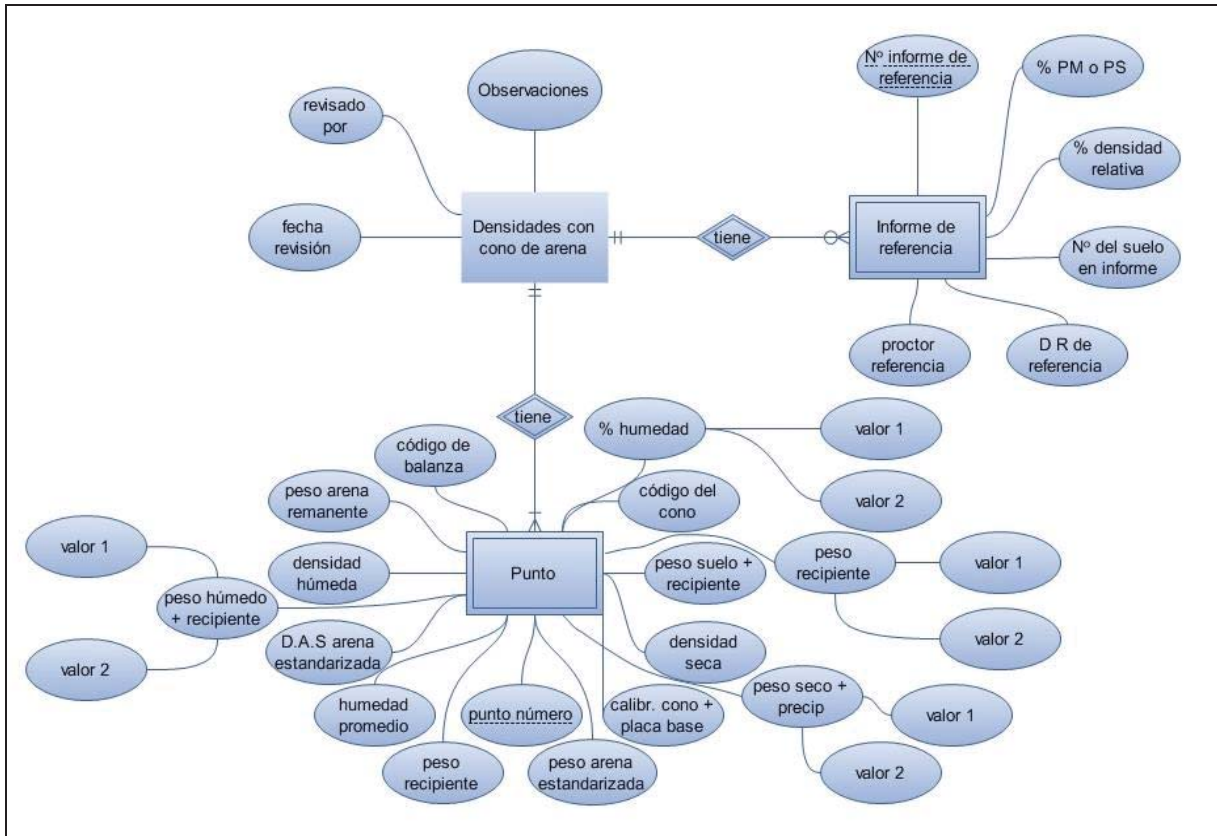


Figura 4.18 MER Densidades con cono de arena

Para poder implementar este modelo en un SGBD (Sistema Gestión Base de Datos) como MySQL hace falta convertir el MER resultante en un Modelo Relacional y luego normalizar las nuevas entidades. Al terminar dicho proceso se obtiene el Modelo Relacional, que es una fiel representación de la base de datos que se utilizará en el proyecto.

En la figura 4.19 se puede observar el MER del formulario “Toma de muestra de suelo”:



Figura 4.19 MER Toma de muestra de suelo

Para poder implementar este modelo en un SGBD (Sistema Gestión Base de Datos) como MySQL hace falta convertir el MER resultante en un Modelo Relacional y luego normalizar las nuevas entidades. Al terminar dicho proceso se obtiene el Modelo Relacional, que es una fiel representación de la base de datos que se utilizará en el proyecto.

A continuación se presenta el MER del formulario “Toma de muestra de hormigón”:

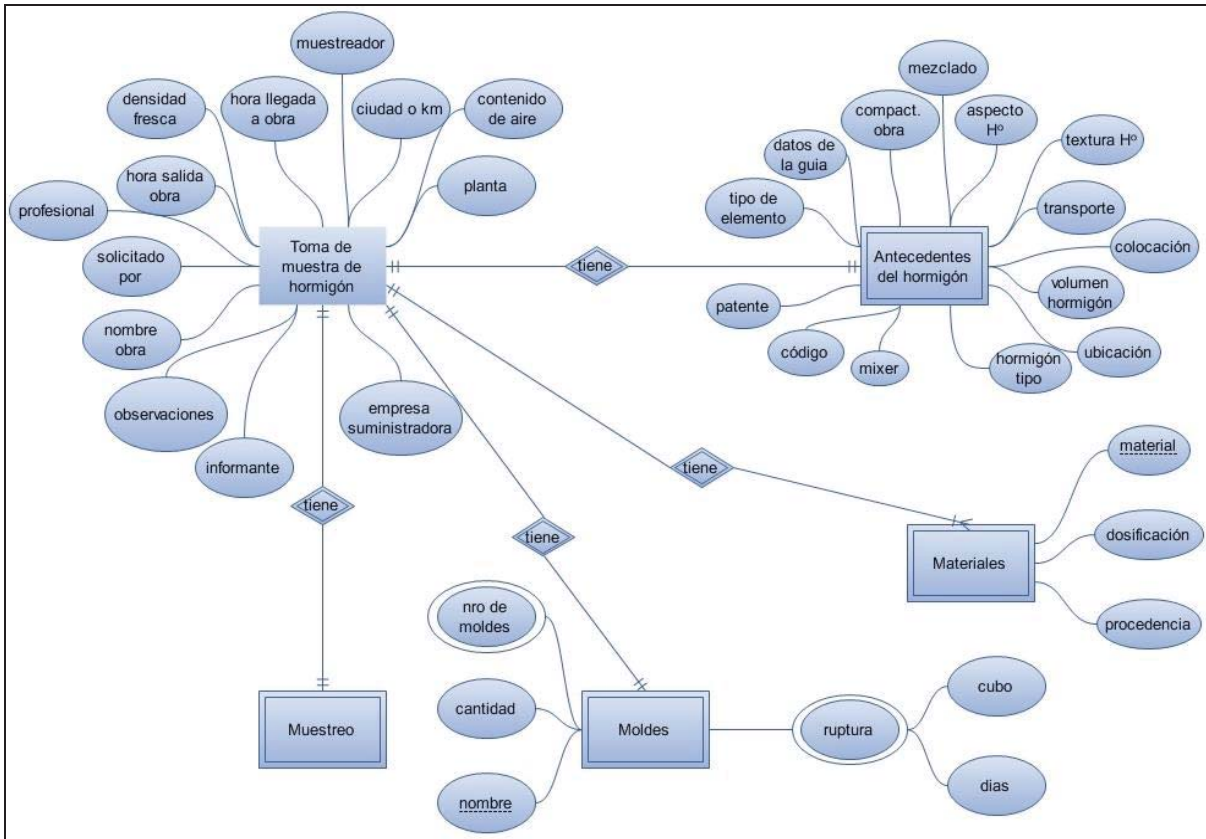


Figura 4.20 MER Toma de muestra de hormigón

Para poder implementar este modelo en un SGBD (Sistema Gestión Base de Datos) como MySQL hace falta convertir el MER resultante en un Modelo Relacional y luego normalizar las nuevas entidades. Al terminar dicho proceso se obtiene el Modelo Relacional, que es una fiel representación de la base de datos que se utilizará en el proyecto.

La siguiente imagen representa el MER de la sección Muestreo perteneciente al formulario “Toma de muestra de hormigón”:

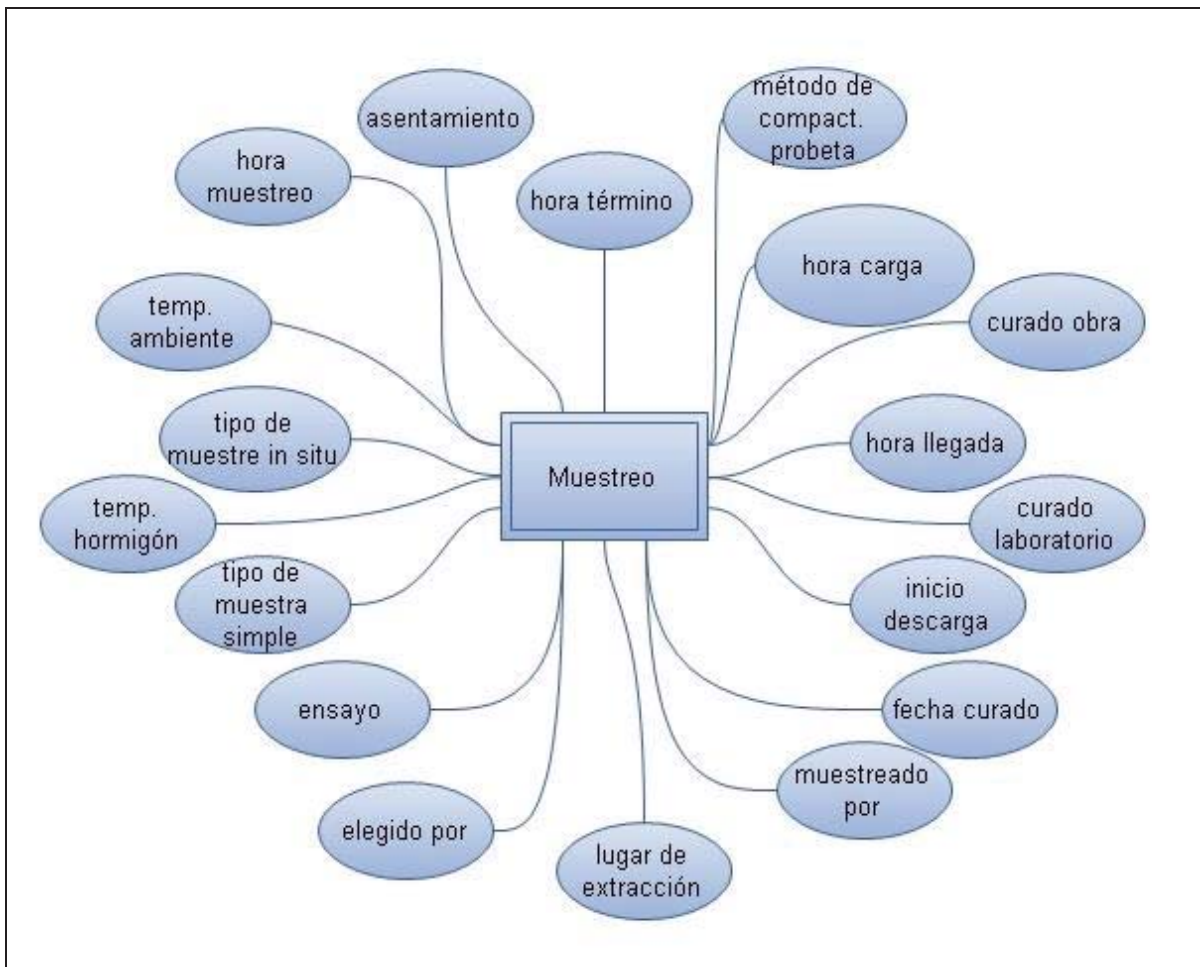


Figura 4.21 MER Muestreo

Para poder implementar este modelo en un SGBD (Sistema Gestión Base de Datos) como MySQL hace falta convertir el MER resultante en un Modelo Relacional y luego normalizar las nuevas entidades. Al terminar dicho proceso se obtiene el Modelo Relacional, que es una fiel representación de la base de datos que se utilizará en el proyecto.

El modelo relacional que se presenta a continuación, está separado en dos imágenes con el objetivo de lograr una mejor visibilidad.

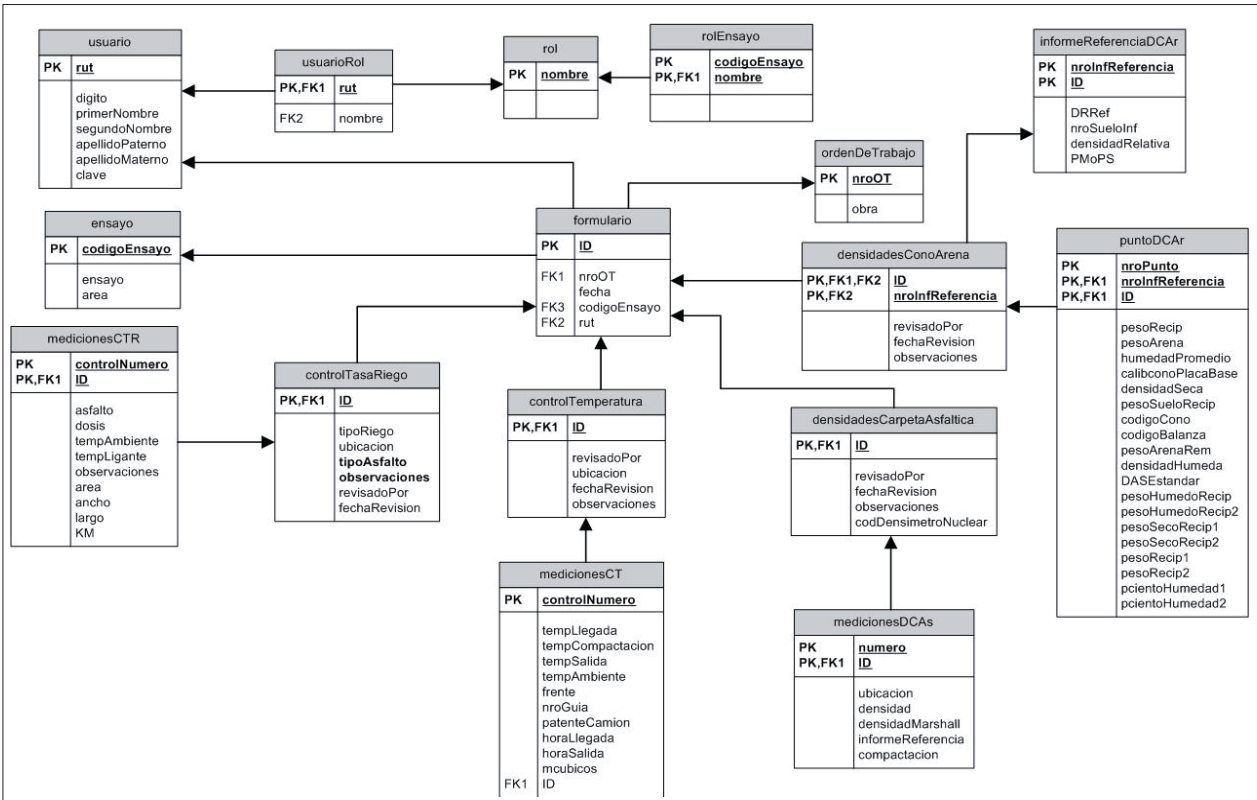


Figura 4.22 Modelo Relacional parte 1

Este modelo será implementado tanto en el servidor como en el dispositivo móvil. Existen ciertas diferencias entre el manejo de datos por parte de MySql y de Sql Server CE pero el servicio web se encargará de llevar a cabo la transición. Algunas diferencias en el manejo de datos son las siguientes:

- Sql Server CE no maneja el tipo de dato Date por lo que se tiene que utilizar Datetime. La diferencia es que Datetime incluye la fecha y la hora pero se puede omitir la hora y usar como Date. Hay que asegurarse que no se visualice la hora al mostrar el dato.
- Distintos nombres para las cadenas de caracteres. MySql las denomina varchar y Sql Server CE las denomina nvarchar.
- Distintos nombres para los tipo de datos auto numéricos. MySql los denomina AUTO_INCREMENT y Sql Server CE los llama Identity.

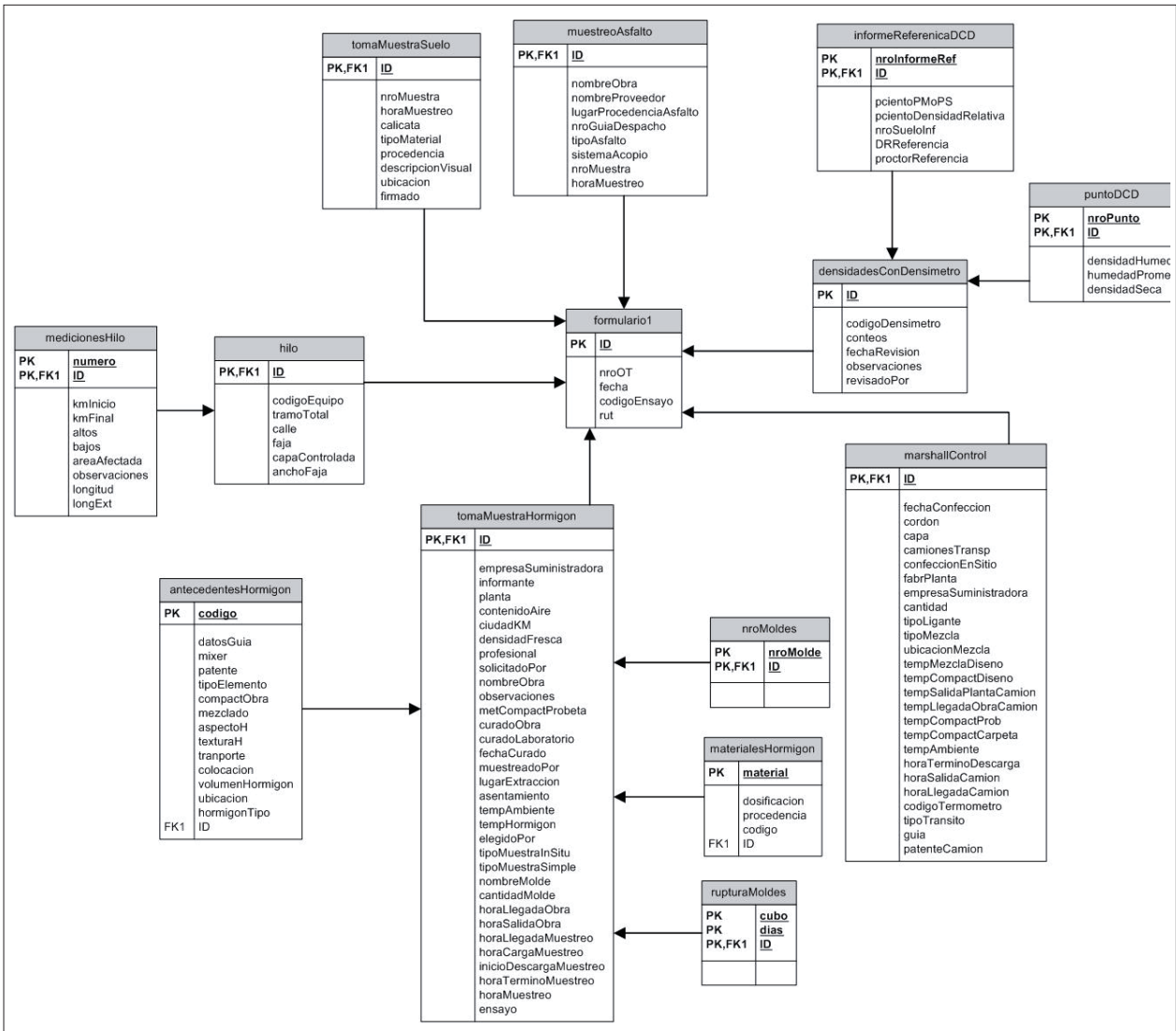


Figura 4.23 Modelo Relacional parte 2

Este modelo será implementado tanto en el servidor como en el dispositivo móvil. Existen ciertas diferencias entre el manejo de datos por parte de MySQL y de SQL Server CE pero el servicio web se encargará de llevar a cabo la transición. Algunas diferencias en el manejo de datos son las siguientes:

- SQL Server CE no maneja el tipo de dato Date por lo que se tiene que utilizar Datetime. La diferencia es que Datetime incluye la fecha y la hora pero se puede omitir la hora y usar como Date. Hay que asegurarse que no se visualice la hora al mostrar el dato.
- Distintos nombres para las cadenas de caracteres. MySQL las denomina varchar y SQL Server CE las denomina nvarchar.
- Distintos nombres para los tipos de datos auto numéricos. MySQL los denomina AUTO_INCREMENT y SQL Server CE los llama Identity.

A continuación se presenta el modelo relacional del sistema de Administración LE-PUCV, que se encuentra actualmente en producción en la empresa. Es importante destacar que el software desarrollado interactúa con dicho sistema, realizando tareas de lectura y escritura en algunas relaciones de la base de datos que se presenta en la siguiente figura.

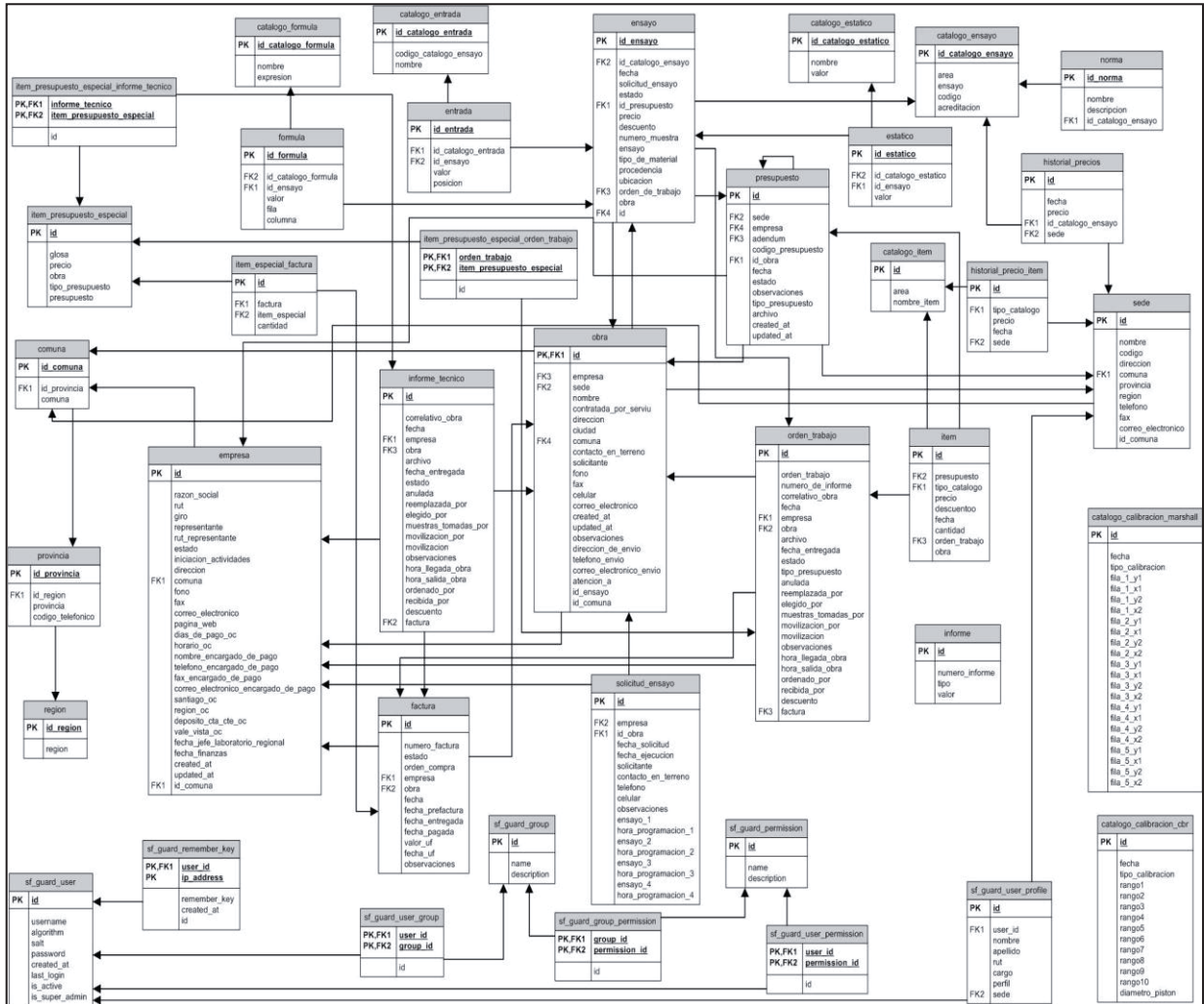


Figura 4.24 Modelo Relacional Sistema de Administración LE-PUCV

Existe una sincronización entre sistemas, que se desata por el usuario del dispositivo móvil al iniciar alguna de las funcionalidades implementadas, permitiendo operar datos de ambos sistemas.

Los datos que se ven involucrados en la transferencia de información fluctúan entre las tablas: usuarios, obras, órdenes de trabajo, empresas y formularios. La figura 4.28 muestra dónde se almacena dicha información.

4.2.8.2 Diagramas de clases

Un diagrama de clases es una vista estática del sistema, sin sentido del tiempo. A continuación en la figura 4.25 se muestra un diagrama de clases general del software implementado en la PDA.

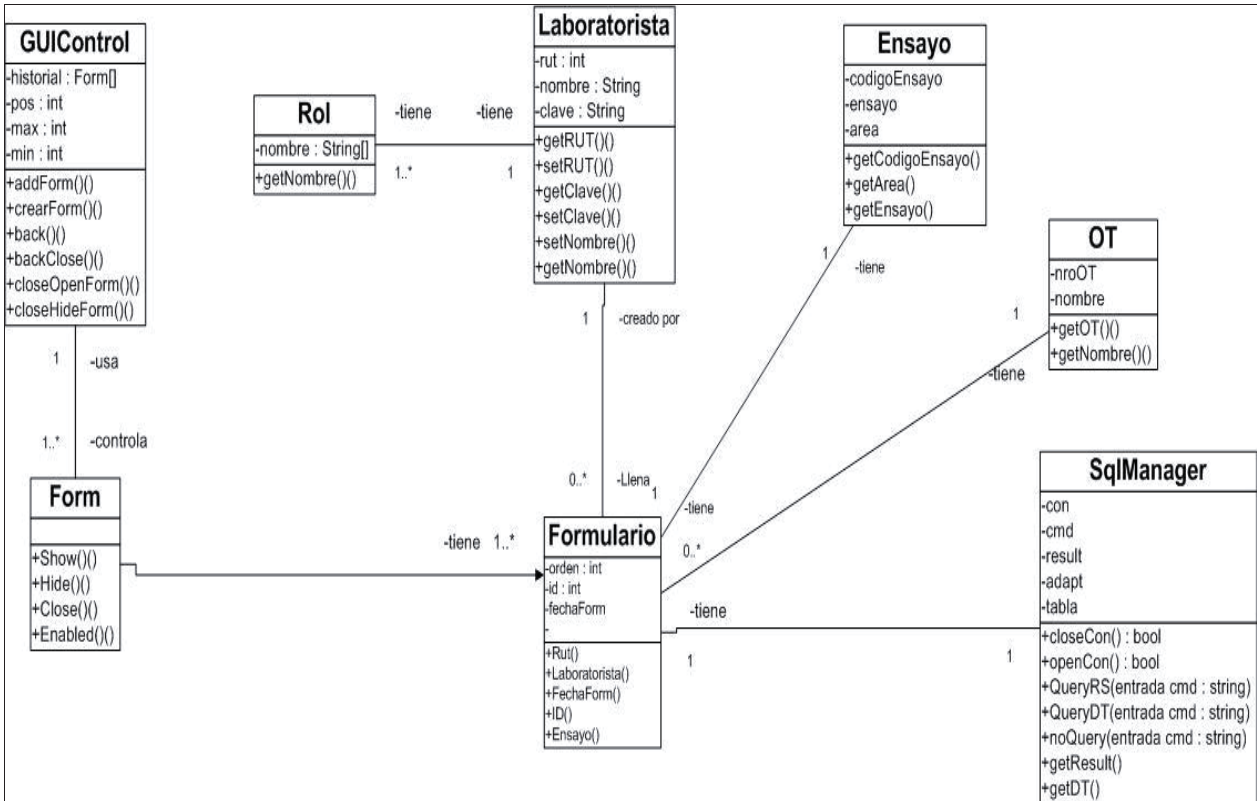


Figura 4.25 Diagrama de clases. General

Los ensayos tienen ciertos datos en común, los cuales heredan de la clase Formulario. Todas las clases de ensayo tienen un método que guarda los datos del objeto en su correspondiente tabla en la base de datos. Por lo tanto Formulario está relacionada con la clase SqlManager que maneja los accesos a la base de datos.

El siguiente diagrama de clases que se presenta a continuación en la figura 4.26, muestra las clases del software del dispositivo móvil en su totalidad. En dicho diagrama se puede apreciar la arquitectura de tres capas utilizada en la construcción del software, en donde se observa de manera independiente la capa de datos, de presentación y de negocio.

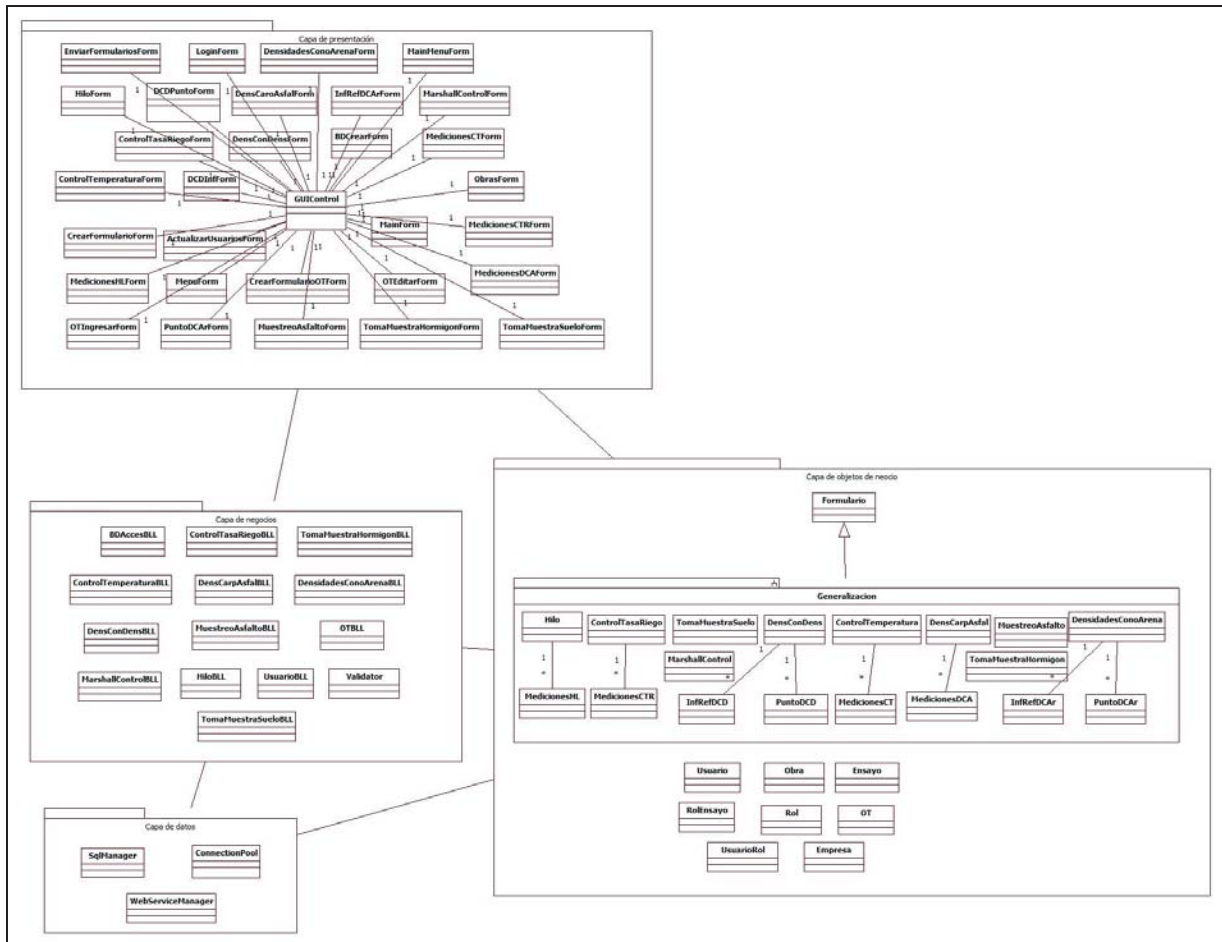


Figura 4.26 Diagrama de clases. Software Dispositivo Móvil

Las siguientes imágenes ilustran los distintos componentes utilizados en la página web, además del manejo de usuarios, roles y formularios. Los detalles de cada componente se omiten por razones de visibilidad.

A continuación, la figura 4.27 muestra los paquetes utilizados en la página web.

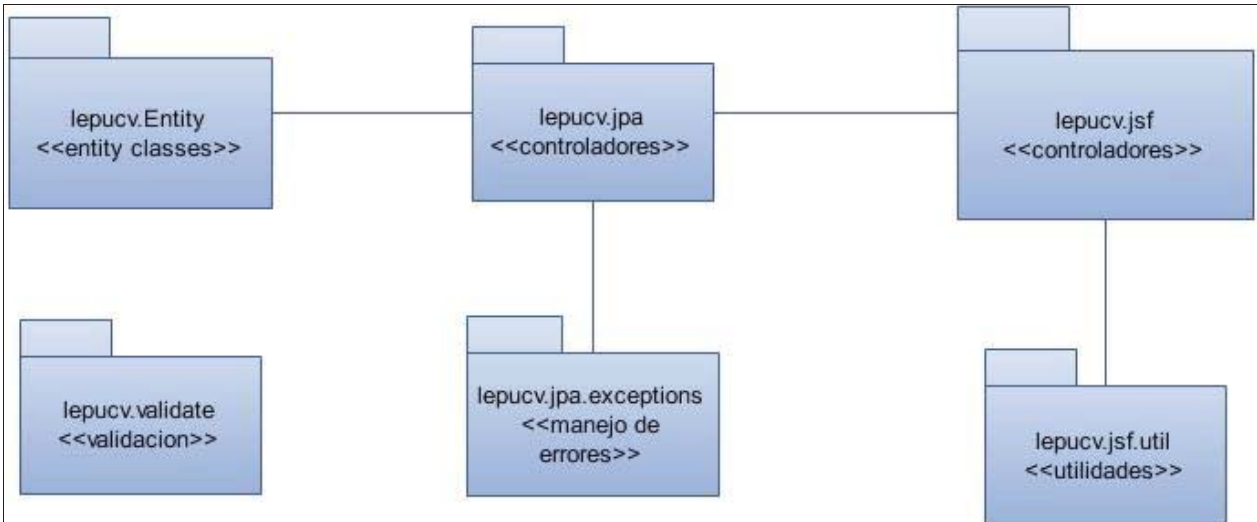


Figura 4.27 Paquetes

La siguiente imagen ilustra lo que es el manejo de usuarios en la página web. Los detalles de cada clase se omiten por razones de visibilidad pero se especifican más adelante.

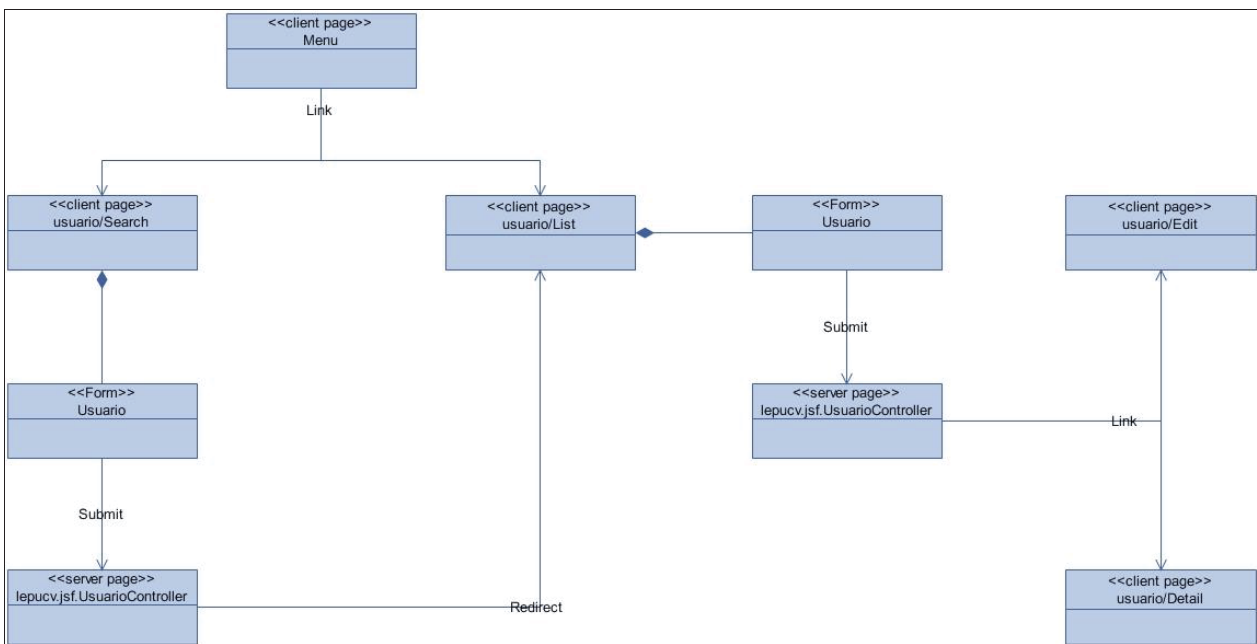


Figura 4.28 Usuarios

La figura 4.29 que se presenta a continuación muestra el manejo de roles.

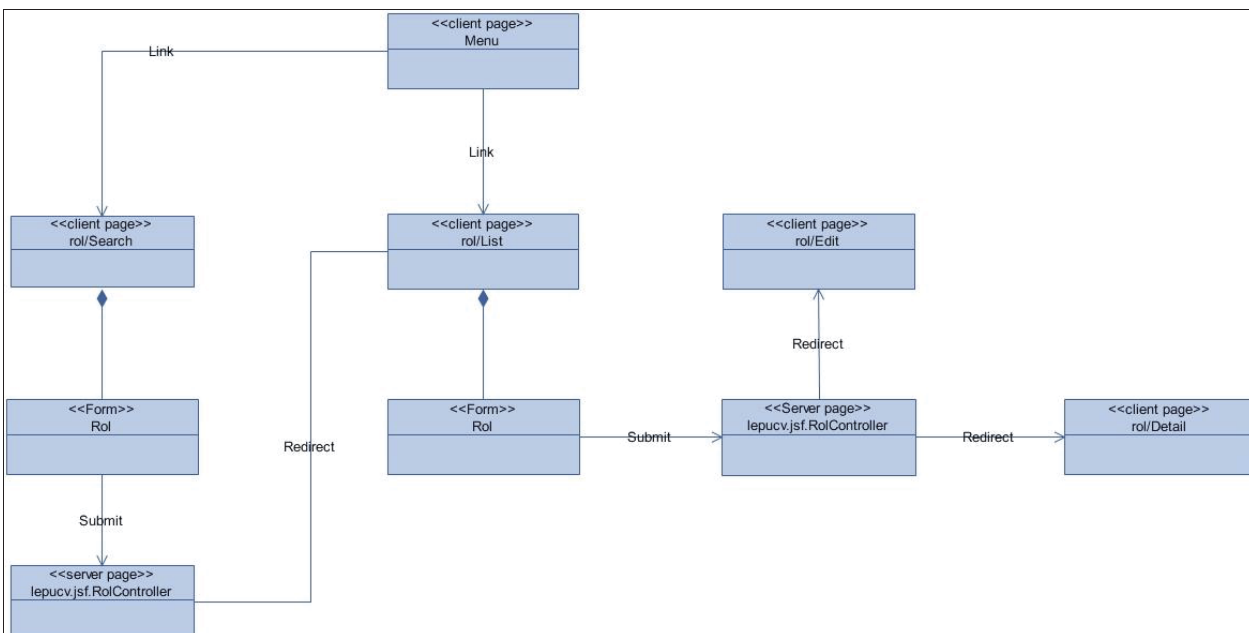


Figura 4.29 Roles

La siguiente imagen muestra cómo funciona el manejo de formularios.

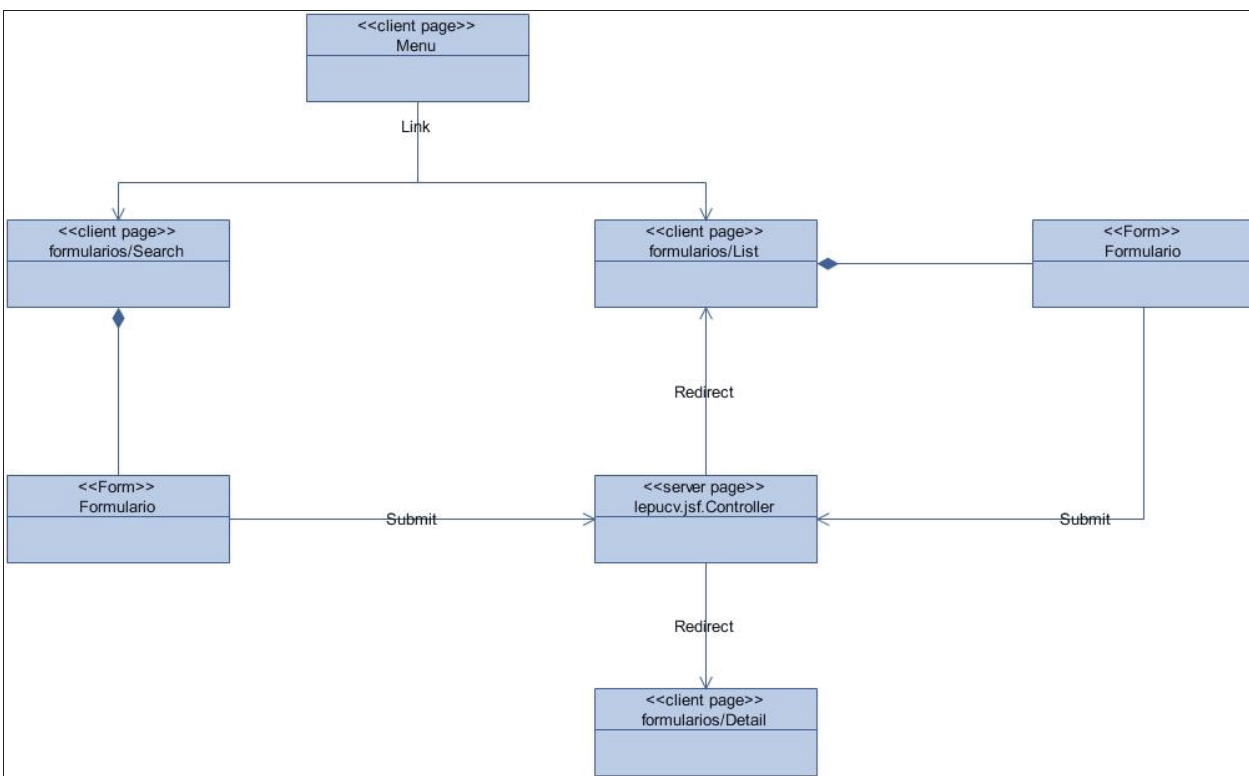


Figura 4.30 Formularios

A continuación, la figura 4.31 muestra el diagrama de despliegue del sistema implementado.

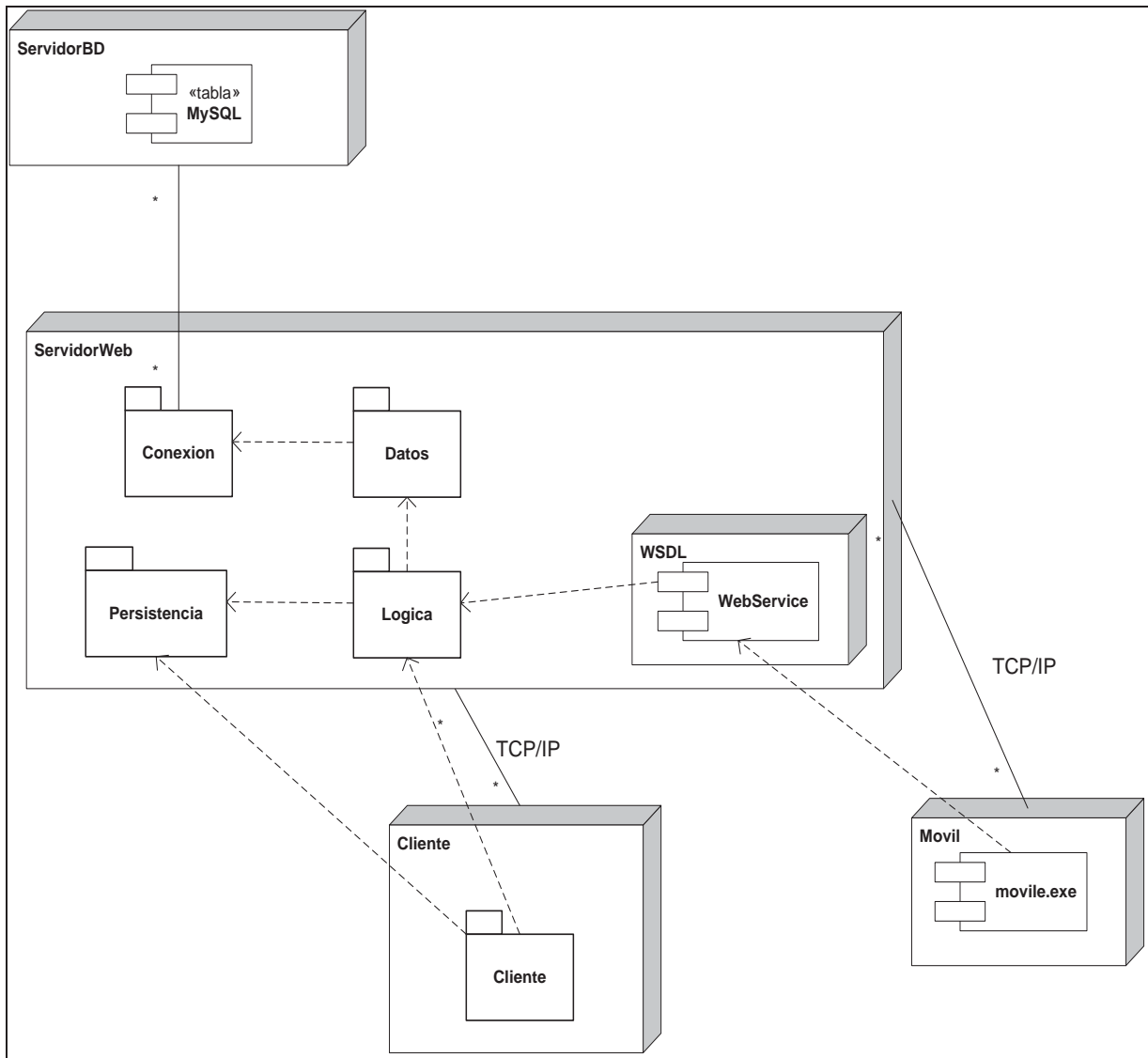


Figura 4.31 Diagrama de despliegue

El diagrama de despliegue ilustra la disposición de las particiones físicas del sistema a implementar y la asignación de los componentes software a dichas particiones. Es decir, visualiza las relaciones físicas entre los componentes software y hardware del sistema a desarrollar.

4.2.8.3 Diagramas de secuencia

Los diagramas de secuencia muestran cómo interactúan las clases en función del tiempo. A continuación se observan unos casos de uso representados mediante diagramas de secuencia.

Caso de uso Ingresar orden de trabajo:

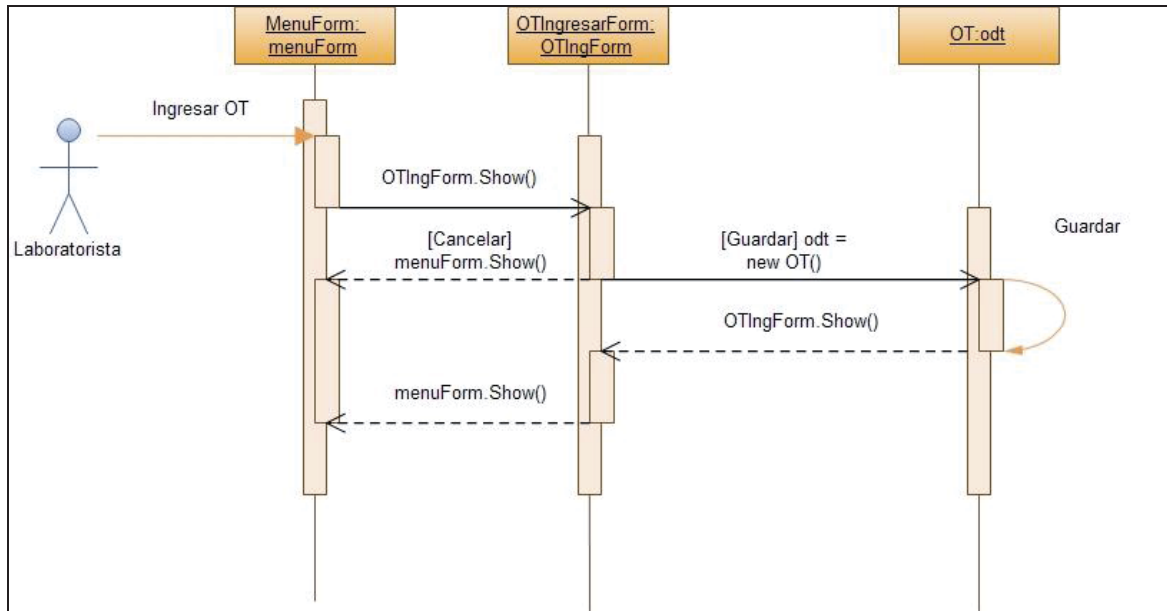


Figura 4.33 Diagrama de secuencia. Ingresar orden de trabajo

Caso de uso Crear formulario:

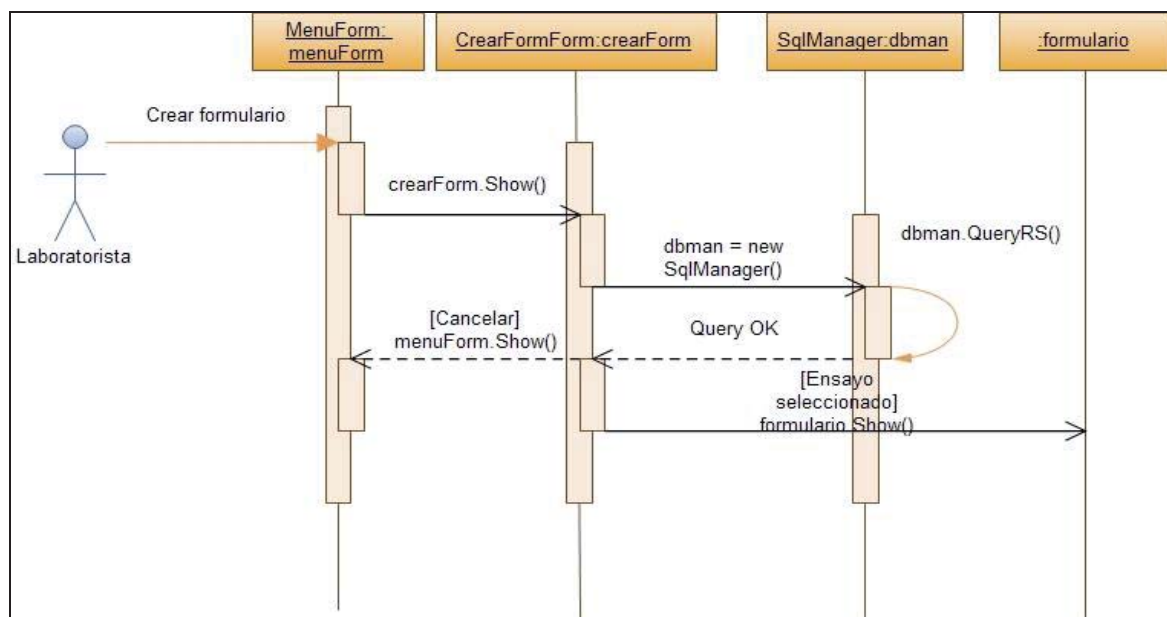


Figura 4.34 Diagrama de secuencia. Crear formulario

Caso de uso Enviar formularios:

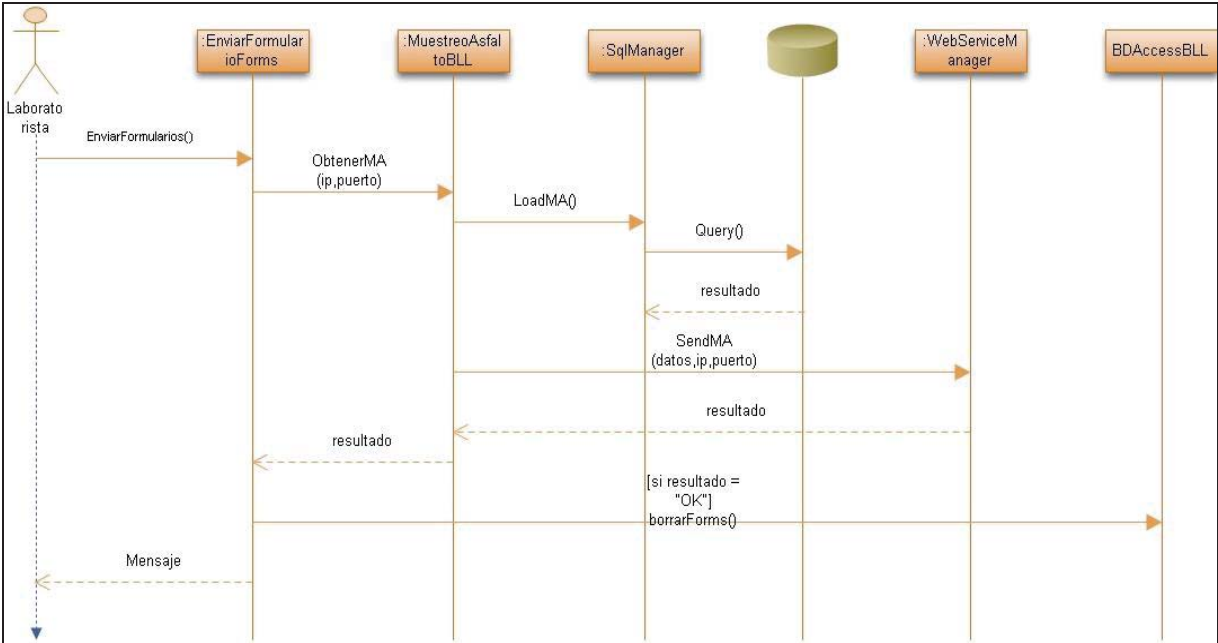


Figura 4.35 Diagrama de secuencia. Enviar formularios

Diagrama de secuencia Añadir usuario:

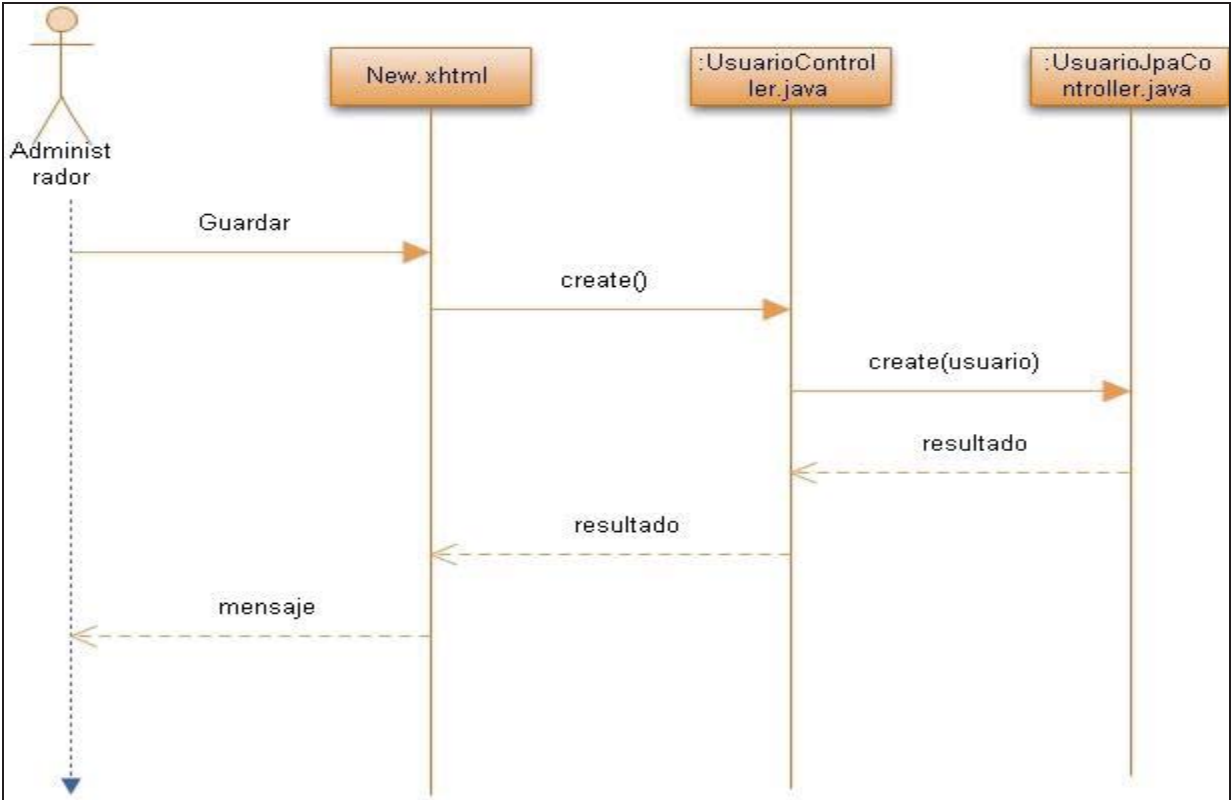


Figura 4.36 Diagrama de secuencia. Añadir usuario

4.2.8.4 Servicio web

El servicio web implementado corresponde al tipo RPC, ya que como se observó anteriormente, RESTFul no cumple con los requisitos de seguridad y SOA es más útil para situaciones más complejas. Para implementar seguridad en los mensajes se utilizará XML encriptado.

El servicio web realiza tres funciones, basadas en sincronización entre el servidor y el dispositivo móvil. Dichas funciones son: recibir los formularios enviados por el laboratorista desde la PDA, enviar información sobre los usuarios, obras y empresas desde el servidor al dispositivo móvil y enviar formularios relacionados con una obra u orden de trabajo desde el servidor a la PDA. Para llevar a cabo estas tareas se implementan varios procedimientos:

getUser(): Obtiene los usuarios registrados en el sistema web y los envía a la PDA. El formato del XML que se envía es el siguiente:

```
<usuarios>
  <usuario>
    <rut></rut>
    <digito></digito>
    <primerNombre></primerNombre>
    <segundoNombre></segundoNombre>
    <apellidoPaterno></apellidoPaterno>
    <apellidoMaterno></apellidoMaterno>
    <clave></clave>
    <roles>
      <rol></rol>
      ...
    </roles>
  </usuario>
  ...
</usuarios>
```

getOrdenesDeTrabajo(): Obtiene las ordenes de trabajo del sistema externo y las envía a la PDA. El formato del XML que se envía es el indicado a continuación:

```
<ordenes>
  <orden>
    <nroOT></nroOT>
    <obra></obra>
    <empresa></empresa>
    <fecha></fecha>
  </orden>
  ...
</ordenes>
```

getEmpresas(): Obtiene las empresas registradas en el sistema externo y las envía a la PDA. El

formato del XML que se envía se detalla a continuación:

```
<empresas>
  <empresa>
    <id></id>
    <razonSocial></razonSocial>
    <rut></rut>
  </empresa>
  ...
</empresas>
```

getFormularios(int obra, string orden): Recibe un id de obra y/o un id de orden como parámetro y envía los formularios asociados que tengan el atributo estado “En espera”. El formato del XML que retorna el método es el siguiente:

```
<formularios>
  <formulario>
    <id></id>
    <fecha></fecha>
    <codigoEnsayo></codigoEnsayo>
    <obra></obra>
    <orden></orden>
  </formulario>
  ...
</formularios>
```

sendControlTasaRiego(string formularios): Recibe los formularios asociados al ensayo Control Tasa Riego como parámetro y las ordenes de trabajo asociadas. Guarda la información en la base de datos del sistema web y en la base de datos externa. El formato del XML que se recibe es el siguiente:

```
<formularios>
  <orden>
    <nroOT></nroOT>
    <obra></obra>
    <estado></estado>
    <fecha></fecha>
  </orden>
  ...
  <formulario>
    <codigoEnsayo></codigoEnsayo>
    <fecha></fecha>
    <nroOT></nroOT>
    <rut></rut>
    <id_externo></id_externo>
  </formulario>
  <mediciones>
    <medicion>
```

```

<asfalto></asfalto>
<dosis></dosis>
<tempAmbiente></tempAmbiente>
<tempLigante></tempLigante>
<observaciones></observaciones>
<area></area>
<largo></largo>
<ancho></ancho>
<KM></KM>
</medición>
...
</mediciones>
...
</formularios>

```

SendControlTemp(string formularios): Recibe los formularios asociados al ensayo Control Tasa Riego como parámetro y las ordenes de trabajo asociadas. Guarda la información en la base de datos del sistema web y en la base de datos externa.

SendDensidadesCarpetaAsf(string formularios): Recibe los formularios asociados al ensayo Densidades Carpeta Asfáltica como parámetro y las ordenes de trabajo asociadas. Guarda la información en la base de datos del sistema web y en la base de datos externa.

SendDensidadesConDensimetro(string formularios): Recibe los formularios asociados al ensayo Densidades con Densímetro como parámetro y las ordenes de trabajo asociadas. Guarda la información en la base de datos del sistema web y en la base de datos externa.

SendDensidadesConoArena(string formularios): Recibe los formularios asociados al ensayo Densidades Cono de Arena como parámetro y las ordenes de trabajo asociadas. Guarda la información en la base de datos del sistema web y en la base de datos externa.

SendHilo(string formularios): Recibe los formularios asociados al ensayo Hi-lo como parámetro y las ordenes de trabajo asociadas. Guarda la información en la base de datos del sistema web y en la base de datos externa.

SendMarshallControl(string formularios): Recibe los formularios asociados al ensayo Marshall Control como parámetro y las ordenes de trabajo asociadas. Guarda la información en la base de datos del sistema web y en la base de datos externa.

SendMuestreoAsfalto(string formularios): Recibe los formularios asociados al ensayo Muestreo Asfalto como parámetro y las ordenes de trabajo asociadas. Guarda la información en la base de datos del sistema web y en la base de datos externa.

SendTomaMuestraHormigon(string formularios): Recibe los formularios asociados al ensayo Toma Muestra Hormigón como parámetro y las ordenes de trabajo asociadas. Guarda la información en la base de datos del sistema web y en la base de datos externa.

4.2.8.5 Página web de administración

Como se mencionó en el comienzo del CAPÍTULO 4 y en los casos de uso, una parte del proyecto es una página web de administración del sistema. Esta página web se encuentra implementada en JSF, bajo glassfish utilizando el framework Java Server Faces. Java Server Faces es un framework que ayuda a implementar aplicaciones web J2EE utilizando la arquitectura MVC (Modelo-Vista-Controlador) que separa la presentación de los datos y la lógica del negocio.

La base de datos es accedida a través de una clase denominada Connection Pool, que es una colección de conexiones abiertas reutilizables al realizarse múltiples consultas a la base de datos. Considerando que LE-PUCV es una empresa pequeña se puede constatar que no va a existir una gran carga sobre la base de datos, pero lo antes mencionado se aplica como medida preventiva en caso de que cambien las condiciones de uso.

Aparte del connection pool se utilizará el Java Persistence API, o JPA, para acceder a la base de datos. JPA provee un mapeo objeto-relacional, es decir, por cada tabla en la base de datos relacional se crea una clase correspondiente y a través de esa clase se manipulan los datos.

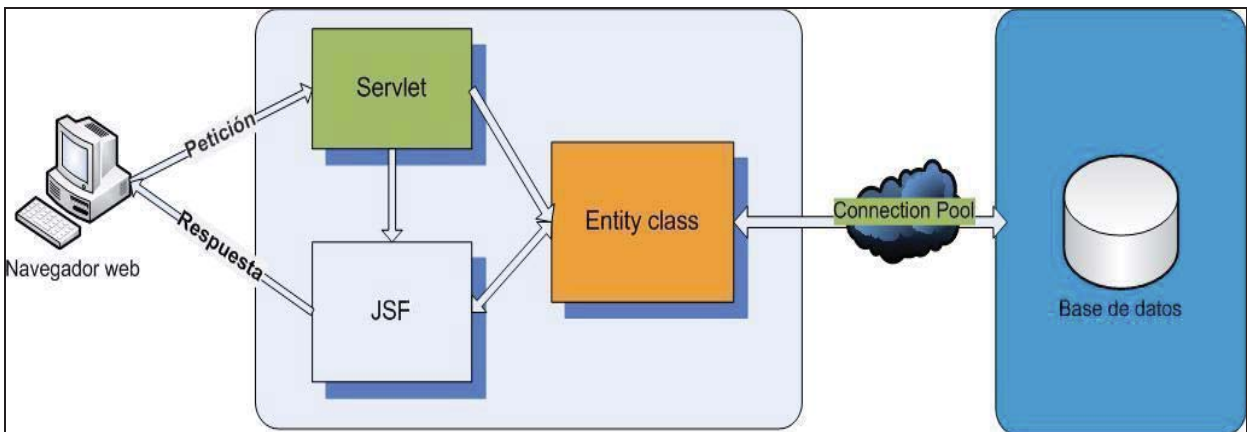


Figura 4.37 Arquitectura de página web

CAPÍTULO 5: INTERFACES DE USUARIO

A continuación se presentan algunas capturas por pantalla que fueron obtenidas mientras se ejecutaba el software implementado en el dispositivo móvil. Dichas capturas corresponden a ciertas interfaces de usuario correspondientes al software desarrollado para la PDA.

Pantalla Inicial



Figura 5.1 Pantalla Inicial Sw. Disp. Móvil

La Figura 5.1 ilustra la pantalla de Bienvenida del Software del Dispositivo Móvil, en ella se aprecia un texto de bienvenida junto a una breve descripción de cómo acceder al Sistema implementado en la PDA.

Login



Figura 5.2 Login Sw. Disp. Móvil

Se aprecia en la Figura 5.2 la pantalla de Login del Software del Dispositivo Móvil. En ella el usuario, previamente registrado, debe ingresar su RUT y Contraseña, una vez validados estos datos se ingresa al Sistema de la PDA.

Menú Orden de Trabajo



Figura 5.3 Menú Orden de Trabajo Sw. Disp. Móvil

La Figura 5.3 ilustra el Menú de Orden de Trabajo del Software del Dispositivo Móvil, también se aprecian las Órdenes de Trabajo almacenadas en la PDA. En esta pantalla además se puede Ingresar una nueva Orden de Trabajo, así como también Editar alguna Orden de Trabajo existente.

Menú Formulario



Figura 5.4 Menú Formulario Sw. Disp. Móvil

En la Figura 5.4 se observa el Menú de Formulario del Software implementado en el Dispositivo Móvil. En esta pantalla se observan los Formularios almacenados en la PDA, así como también las opciones de Crear un nuevo formulario, Editar un formulario existente y Crea una Copia de algún Formulario.

CAPÍTULO 6: VALIDACIÓN Y VERIFICACIÓN

La validación y verificación son parte importante del desarrollo de software. Al documentar dicha actividad se puede presentar al cliente y demostrar que el software es estable y que cumple con sus requerimientos. En el caso del presente proyecto, se han realizado pruebas dentro de cada versión intermedia del software a diferentes partes del mismo. La inspección de código también se ha realizado pero de manera menos formal, ya que cuando se saca mayor provecho de la inspección es cuando una persona que no escribió el código lo revisa. En este caso eso no era posible y simplemente se han realizado inspecciones tras terminar un módulo.

6.1 Plan de Pruebas

“Las pruebas de software se aplican como una etapa más del proceso de desarrollo de software, su objetivo es asegurar que el software cumpla con las especificaciones requeridas y eliminar los posibles defectos que este pudiera tener.”^[10]

Las Pruebas de software es el proceso que permite verificar y revelar la calidad de un producto software.

Las Pruebas de software se integran dentro de las diferentes fases del Ciclo del software dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene.

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una versión final de las especificaciones, del diseño y de la codificación.

6.1.1 Objetivo de las pruebas

“Glen Myers establece varias normas que pueden servir acertadamente como objetivos de las pruebas:

1. La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
2. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
3. Una prueba tiene éxito si descubre un error no detectado hasta entonces.”^[11]

Finalmente se puede decir que cuando se lleva a cabo un plan de pruebas, se buscan los siguientes objetivos:

- Planificar las pruebas necesarias.
- Diseñar e implementar las pruebas creando casos de prueba que especifican qué probar.
- Realizar las diferentes pruebas y manejar los resultados de cada una sistemáticamente.

En base al objetivo de las pruebas, se deben diseñar pruebas que tengan una alta probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible.

Existen varios flujos de trabajo para realizar las pruebas, de los que destacan:

1. **Modelo de Prueba:** Éste describe cómo se prueban los distintos componentes del sistema.
2. **Caso de Prueba:** Éstos ayudan a validar que el software desarrollado realice las funciones para las que ha sido creado en base a los requerimientos del cliente, esto indica que por lo menos deberá existir un caso de prueba por cada requerimiento que el software deba cumplir. Entre los casos de prueba más común podemos mencionar:
 - Pruebas unitarias
 - Pruebas funcionales
 - Pruebas de Integración
 - Pruebas de validación
 - Pruebas de sistema
 - Caja blanca (sistemas)
 - Caja negra (sistemas)
 - Pruebas de Aceptación.
 - Pruebas de Regresión.
 - Pruebas de Cargas.
 - Pruebas de Prestaciones.
 - Prueba de Recorrido.

El principal objetivo del diseño de casos de prueba es obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del software. Para llevar a cabo este objetivo, se usan dos categorías diferentes de técnicas de diseño de casos de prueba: prueba de caja blanca y prueba de caja negra.

Caja Blanca

“La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que:

1. Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
2. Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
3. Ejecuten todos los bucles en sus límites y con sus límites operacionales.
4. Ejerciten las estructuras internas de datos para asegurar su validez.”^[10]

Caja Negra

“Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca.

La prueba de caja negra intenta encontrar errores de las siguientes categorías:

1. Funciones incorrectas o Ausentes.
2. Errores de interfaz.
3. Errores en estructuras de datos o en accesos a bases de datos externas.
4. Errores de rendimiento.
5. Errores de inicialización y de terminación.»^[10]

El enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas.

Al realizar pruebas funcionales lo que se pretende es ponerse en los pies del usuario, su objetivo será encontrar alguna posible debilidad y si la llega a ubicar se esforzará por que deje de ser pequeña y posiblemente se convertirá en un gran error, cada error encontrado por el analista de pruebas es un éxito.

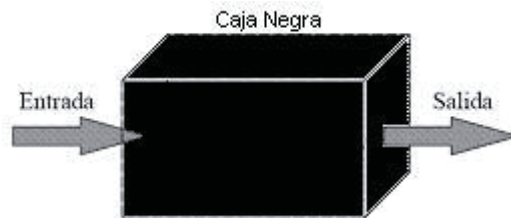


Figura 6.1 Ilustración Caja Negra

6.1.2 Propósito

Este Plan de Pruebas desarrollado para el Sistema de Gestión de información vía dispositivo móvil para LE-PUCV, contempla dentro de sus objetivos lo siguiente:

Proveer un set de planillas de revisión que cubran los procesos implementados en el sistema con tal de poder comprobar su comportamiento esperado, según lo expresado en los documentos de especificación de requerimientos.

6.1.3 Alcance

El alcance del Plan de Pruebas se circunscribe al ámbito del Proyecto de Gestión de información vía dispositivo móvil para LE-PUCV y en los documentos considerados parte de él, es decir, concentrados en la sección Análisis y definición de requerimientos, más específicamente en los casos de uso.

Este Plan de Pruebas se refiere única y exclusivamente a este proyecto y no tiene relación con ningún otro proyecto.

6.1.4 Casos de Prueba

Para el desarrollo de los casos de prueba se ha establecido una plantilla compuesta de los siguientes campos:

- Id de caso de prueba: Indica un número de identificación del caso de prueba.
- Datos de prueba: Indica los datos que se utilizan para el caso de prueba.
- Descripción de la prueba: Indica la acción sometida a la prueba.
- Resultado esperado: Es lo que se espera que ocurra al finalizar el caso de prueba.
- Observación: Este campo posee información en el caso que ocurra un comportamiento no correspondiente (incidente) al esperado.

Previa a la plantilla mencionada anteriormente, se aprecia una tabla en la que se indica información relacionada al caso de prueba en cuestión.

La especificación de los casos de prueba considerada para el desarrollo y testing del Sistema informático de Gestión de información vía dispositivo móvil para LE-PUCV, junto con los resultados esperados de acuerdo a lo descrito en la especificación de requerimientos del software, se encuentra al final del documento, en la sección ANEXO C: ESPECIFICACIÓN CASOS DE PRUEBA.

6.1.5 Guía de pruebas tipo para elementos de un formulario

La siguiente lista tiene el objetivo de apoyar la revisión de los campos que pudiera contener un formulario (<form...>).

Cadenas

- cadena vacía.
- cadena consistente de únicamente un espacio vacío.
- cadena que empieza o termina en un espacio en blanco, en algunos casos.
- sintácticamente correcta: valores cortos y largos.
- sintácticamente correcta: valores semánticamente válidos e inválidos.
- valor sintácticamente incorrecto: caracteres o combinaciones ilegales.
- caracteres especiales como #, ", ', &, y <.

Números

- cadena vacía, en algunos casos si es posible 0.
- pequeños y largos en rangos positivos.
- fuera del rango de positivos y/o negativos si es que son posibles.
- sintácticamente inválidos (por ejemplo, que incluya letras).

Identificadores

- cadena vacía.
- valor sintácticamente correcto.
- sintácticamente correcto: referencia a un ID existente, referencia inválida.
- valor sintácticamente incorrecto.

Botones de opción (radio)

- un objeto seleccionado.
- nada seleccionado, si es posible.

Botones de opción múltiple

- seleccionados.
- sin seleccionar.

Menús de pestaña (combobox)

- seleccione un elemento en cada turno.

Listas con Scroll

- no seleccione ningún elemento, si es posible.
- seleccione un elemento en cada turno.
- seleccione combinaciones de elementos, si es posible seleccione todos los elementos.

CAPÍTULO 7: CONCLUSIONES

En este trabajo se ha presentado el desarrollo de un sistema informático para la empresa de construcción LE-PUCV, Laboratorio de Ensayos de la Pontificia Universidad Católica de Valparaíso, conformado por un software de formularios digitales implementado en un dispositivo móvil y un sistema web que administra el software en general, todo esto con el objetivo principal de automatizar la captura y almacenamiento de la información.

Respecto a la problemática se ha dado énfasis en analizar la forma de trabajo que posee el cliente; lo cual se llevó a cabo a través de entrevistas. A partir de ello, se plantearon alternativas que pudieran optimizar el modo de trabajo, éstas fueron discutidas con el cliente con el fin de aprovechar este feedback y realizar los ajustes correspondientes.

A su vez, se realizaron las investigaciones necesarias para poder enfocar los puntos críticos y lograr una solución óptima en la implementación del proyecto.

Cabe resaltar, que un factor fundamental en el desarrollo del software fue la planificación del proyecto, ya que su cumplimiento permitió presentar los avances al cliente en las fechas estipuladas y al mismo tiempo finalizar el software en el plazo estimado.

En el transcurso del proyecto se ha podido constatar que los requerimientos no han cambiado en gran medida durante el desarrollo y quizás, desde ese punto de vista, el desarrollo evolutivo no ha sido el mejor modelo escogido para este software en particular. Sin embargo la flexibilidad que da dicho modelo ha sido una ventaja considerable, ya que para la aplicación de modelos más estrictos es fundamental contar con una gran experiencia, lo cual no es el caso de los desarrolladores del presente proyecto.

Para el último período de desarrollo del proyecto se puso en práctica parte de la planificación de riesgos, aumentando las horas hombre destinadas al trabajo diario. Se confeccionó con gran énfasis la etapa de validación y verificación, dándole gran relevancia al plan de pruebas, el cual fue ejecutado con éxito.

Finalmente se logró construir el sistema en su totalidad, tanto en el software del dispositivo móvil, como en el sistema de administración web.

Con la implementación de la solución desarrollada, se obtienen muchos resultados favorables, orientados específicamente al aumento de la productividad y eficiencia del trabajo. Entre dichos resultados destacan; una disminución considerable en los errores humanos como lectura y digitalización, además de una reducción en los tiempos de tramitación, mayor control sobre el personal que opera en terreno y una disponibilidad de la información en tiempo real.

En la actualidad cada vez es más común que las necesidades de las empresas demanden nuevas tecnologías, para mejorar la productividad y eficiencia del trabajo. El software desarrollado satisface la actual demanda, utilizando dispositivos móviles para capturar datos en terreno, tecnología que está emergiendo y que permite, entre otras cosas, un mayor dinamismo en el ingreso, envío y almacenamiento de los datos, facilitando el posterior procesamiento de la información y otorgando resultados más fidedignos.

De esta manera finaliza la construcción del Proyecto “Sistema de Gestión de información vía dispositivo móvil para LE-PUCV”, culminando con un producto de software finalizado, en condiciones óptimas para su entrega y puesta en producción.

Finalmente, se concluye la real importancia que tiene un constante estudio de las nuevas tecnologías, y la favorable consecuencia que puede ocasionar su puesta en práctica facilitando el desarrollo y crecimiento de cualquier organización, manteniéndolas activas en un mercado competitivo.

CAPÍTULO 8: REFERENCIAS BIBLIOGRÁFICAS

8.1 Referencias

- [1] “La distribución de sistemas operativos móviles en el mercado”; <http://www.expansion.com/2012/08/08/empresas/digitech/1344423861.html>
- [2] “Laptop, Notebook’s, NetBooks y Tablets. Cuál Comprar?”; <http://www.poderpda.com/noticias/laptop-notebooks-netbooks-y-tablets-cual-comprar/>
- [3] “Web Services Infraestructure”. The global utility for real time business. A white paper by Phil Wainewright.
- [42] “An Overview of Service-oriented Architecture, Web Services and Grid Computing”, Latha Srinivasan and Jem Treadwell, HP Software Global Business Unit.
- [5] Fielding, Roy Thomas (2000), *Architectural Styles and the Design of Network-based Software Architectures*, Doctoral dissertation, University of California, Irvine.
- [6] “Guide to Secure Web, Services Recommendations of the National Institute of Standards and Technology”, Anoop Singhal, Theodore Winograd, Karen Scarfone.
- [7] Sun Glassfish Enterprise Server 2.1 Release notes: <http://docs.sun.com/app/docs/doc/820-4329/abpak?a=view>.
- [8] Common development and distribution licence (CDDL) version 1.0 <https://glassfish.dev.java.net/public/CDDL+GPL.html>.
- [9] Sommerville, Ian. (2002). *Ingeniería en Software*. Ciudad de México: Pearson.
- [10] “Modelos evolutivos”; <http://tema3isoftware.blogspot.com/p/modelos-de-desarrollo-tecnicas-y.html> [11] Pressman, Roger. *Ingeniería en Software*. Editorial McGraw Hill.

8.2 Bibliografía

- Ingeniería del software, Ian Sommerville, ed. Prentice Hall, 7ª edición 2005.
- Ingeniería del software, Roger Pressman, ed. McGraw-Hill.
- Aprendiendo UML en 24 horas, Joseph Schmuller, ed. Editorial división computación, México.

ANEXO A. ESPECIFICACIÓN DE REQUERIMIENTOS

Especificación Casos de Uso Administrador

Las siguientes tablas representan los casos de uso narrativos, que corresponden a una especificación de los casos de uso asociados a la figura 4.8 del CAPITULO 4.

Tabla 4.3 Caso de uso. Añadir usuario

Caso de uso	Añadir usuario
Actor primario	Administrador
Pre-condiciones	El administrador se ha identificado mediante el login.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none">1. El administrador solicita agregar un usuario al sistema.2. El sistema despliega una página con los campos a ingresar por el administrador.3. El administrador ingresa nombres, apellidos y RUT del usuario.4. El sistema muestra un mensaje indicando que el usuario se ha agregado correctamente.
Flujo alternativo	<ol style="list-style-type: none">3.a El administrador ingresa un RUT que ya está registrado en el sistema.<ol style="list-style-type: none">i) El sistema muestra un mensaje indicando que el RUT ya está registrado.ii). El sistema vuelve a la página de ingreso de usuario.3.b El administrador ingresa un RUT no válido.<ol style="list-style-type: none">i) El sistema muestra un mensaje indicando que el RUT no es válido.ii) El sistema vuelve a la página de ingreso de usuario.

Tabla 4.4 Caso de uso. Eliminar usuario

Caso de uso	Eliminar usuario
Actor primario	Administrador
Pre-condiciones	<ul style="list-style-type: none"> - El administrador se ha identificado mediante el login. - Existe al menos un usuario registrado en el sistema.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El administrador solicita eliminar un usuario al sistema. 2. El sistema despliega una página con los usuarios registrados. 3. El administrador selecciona un usuario y procede a eliminarlo. 4. El sistema despliega un mensaje solicitando la confirmación de la eliminación. 5. El administrador confirma la acción de eliminación. 6. El sistema muestra un mensaje indicando que el usuario se ha eliminado exitosamente.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El administrador decide buscar un usuario ingresando su RUT o parte de su nombre completo. <ol style="list-style-type: none"> i) El sistema muestra los usuarios que coinciden con la búsqueda. ii) Continúa el flujo en el paso 3.

Tabla 4.5 Caso de uso. Editar usuario

Caso de uso	Editar usuario
Actor primario	Administrador
Pre-condiciones	<ul style="list-style-type: none"> - El administrador se ha identificado mediante el login. - Existe al menos un usuario registrado en el sistema.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El administrador solicita editar un usuario. 2. El sistema despliega una página con los usuarios registrados. 3. El administrador selecciona un usuario. 4. El sistema muestra una página con los datos asociados al usuario. 5. El administrador cambia uno o más datos. 6. El sistema despliega un mensaje indicando que los datos se guardaron de manera satisfactoria.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El administrador decide buscar un usuario ingresando su RUT o parte de su nombre completo. <ol style="list-style-type: none"> i) El sistema muestra los usuarios que coinciden con la búsqueda. ii) Continúa el flujo en el paso 3.

Tabla 4.6 Caso de uso. Añadir rol

Caso de uso	Añadir rol
Actor primario	Administrador
Pre-condiciones	El administrador se ha identificado mediante el login.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El administrador solicita agregar un rol. 2. El sistema despliega una página con los roles registrados. 3. El administrador ingresa el nombre del rol. 4. El sistema despliega un mensaje indicando que el rol fue agregado exitosamente.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El administrador ingresa un rol que ya existe. <ol style="list-style-type: none"> i) El sistema indica que el rol ya existe. ii) El usuario puede cancelar el ingreso del rol o ingresar otro nombre.

Tabla 4.7 Caso de uso. Eliminar rol

Caso de uso	Eliminar rol
Actor primario	Administrador
Pre-condiciones	<ul style="list-style-type: none"> - El administrador se ha identificado mediante el login. - Existe al menos dos roles registrados en el sistema.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El administrador solicita eliminar un rol. 2. El sistema despliega una página con los roles registrados. 3. El administrador selecciona el rol y procede a eliminarlo. 4. El sistema despliega un mensaje solicitando la confirmación de la eliminación. 5. El administrador confirma la acción de eliminación. 6. El sistema despliega un mensaje indicando que el rol fue eliminado exitosamente.
Flujo alternativo	-

Tabla 4.8 Caso de uso. Editar rol

Caso de uso	Editar rol
Actor primario	Administrador
Pre-condiciones	<ul style="list-style-type: none"> - El administrador se ha identificado mediante el login. - Existe al menos un rol registrado en el sistema.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El administrador solicita editar un rol. 2. El sistema despliega una página con los roles registrados. 3. El administrador selecciona el rol a editar. 4. El sistema permite editar el nombre del rol. 5. El usuario cambia el nombre del rol. 6. El sistema indica que el cambio de nombre fue exitoso.
Flujo alternativo	<ol style="list-style-type: none"> 5.a El usuario cambia el nombre del rol a un nombre que ya existe. <ol style="list-style-type: none"> i) El sistema indica que el nombre ya se encuentra asociado a otro rol. ii) El usuario cambia el nombre a uno que no existe o cancela la edición del rol.

Tabla 4.9 Caso de uso. Asociar ensayo a rol

Caso de uso	Asociar ensayo a rol
Actor primario	Administrador
Pre-condiciones	<ul style="list-style-type: none"> - El administrador se ha identificado mediante el login. - Existe al menos un rol registrado en el sistema.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El administrador solicita asociar un ensayo a un rol. 2. El sistema despliega una página con los roles registrados. 3. El administrador selecciona el rol. 4. El sistema muestra los ensayos asociados al rol. 5. El administrador selecciona los ensayos asociados al rol elegido. 6. El sistema indica que el cambio de ensayos asociados al rol fue exitoso.
Flujo alternativo	-

Tabla 4.10 Caso de uso. Asociar usuario a rol

Caso de uso	Asociar usuario a rol
Actor primario	Administrador
Pre-condiciones	<ul style="list-style-type: none"> - El administrador se ha identificado mediante el login. - Existe al menos un rol y un usuario registrados en el sistema.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El administrador solicita asociar un usuario a un rol. 2. El sistema despliega una página con los usuarios registrados. 3. El administrador selecciona un usuario. 4. El sistema muestra los roles asociados al usuario. 5. El administrador selecciona los roles asociados al usuario especificado. 6. El sistema indica que el cambio de roles asociados al usuario fue exitoso.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El administrador decide buscar un usuario ingresando su RUT o parte de su nombre completo. <ol style="list-style-type: none"> i) El sistema muestra los usuarios que coinciden con la búsqueda. ii) Continúa el flujo en el paso 3.

Tabla 4.11 Caso de uso. Visualizar formulario

Caso de uso	Visualizar formulario
Actor primario	Administrador
Pre-condiciones	<ul style="list-style-type: none"> - El administrador se ha identificado mediante el login. - Existe al menos un formulario guardado en el sistema.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El administrador solicita visualizar un formulario. 2. El sistema despliega una página de filtros/búsqueda. 3. El administrador selecciona un formulario. 4. El sistema muestra el formulario con los datos ingresados. 5. El administrador cierra el formulario. 6. El sistema vuelve a la página principal.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El administrador elige mostrar sólo formularios de un área y/o tipo de ensayo específico. <ol style="list-style-type: none"> i) El sistema despliega los formularios asociados al área y/o tipo de ensayo seleccionado. ii) Continúa el flujo en el punto 3. 3.b El administrador busca los formularios asociados a un usuario específico ingresando nombre o RUT. <ol style="list-style-type: none"> i) El sistema muestra los formularios asociados. ii) Continúa el flujo en el punto 3.

Tabla 4.12 Caso de uso. Crear informe

Caso de uso	Crear informe
Actor primario	Administrador
Pre-condiciones	- El administrador se ha identificado mediante el login.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El administrador solicita crear un informe. 2. El administrador escoge un usuario, una fecha de inicio y una fecha final. 3. El sistema despliega una página con los formularios ingresados por dicho usuario en ese rango de tiempo. 4. El sistema vuelva a la página principal.
Flujo alternativo	<ol style="list-style-type: none"> 2.a El administrador ingresa un usuario y fechas que no coinciden con ningún formulario. <ol style="list-style-type: none"> i) El sistema despliega un mensaje indicando que no coincide ningún formulario con los parámetros ingresados. ii) Continúa el flujo en el punto 2. 4.a El administrador escoge generar un archivo PDF con la información. <ol style="list-style-type: none"> i) El sistema genera el archivo y lo abre. ii) El administrador imprime el archivo. iii) Continúa el flujo en el punto 4.

Tabla 4.13 Caso de uso. Eliminar formulario

Caso de uso	Eliminar formulario
Actor primario	Administrador
Pre-condiciones	<ul style="list-style-type: none"> - El administrador se ha identificado mediante el login. - Existe al menos un formulario guardado en el sistema.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El administrador solicita visualizar formularios. 2. El administrador escoge un formulario para eliminar. 3. El sistema despliega un mensaje de alerta. 4. El administrador confirma la eliminación. 5. El sistema elimina el usuario.
Flujo alternativo	<ol style="list-style-type: none"> 4.a El administrador decide no confirmar la eliminación. <ol style="list-style-type: none"> i) El sistema cancela la operación.

Especificación Casos de Uso Usuario

Las siguientes tablas representan los casos de uso narrativos, que corresponden a una especificación de los casos de uso asociados a la figura 4.9 del CAPITULO 4.

Tabla 4.15 Caso de uso. Editar orden de trabajo

Caso de uso	Editar orden de trabajo
Actor primario	Usuario
Pre-condiciones	<ul style="list-style-type: none"> - El usuario se ha identificado mediante el login. - Existe por lo menos una orden de trabajo registrada.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El usuario se encuentra en el menú de órdenes de trabajo. 2. El sistema despliega una página con las órdenes de trabajo registradas. 3. El usuario selecciona una orden de trabajo y solicita editarla. 4. El sistema permite editar la empresa y la obra de la orden de trabajo seleccionada. 5. El usuario realiza los cambios. 6. El sistema indica que los cambios se guardaron exitosamente.
Flujo alternativo	<ol style="list-style-type: none"> 5.a El usuario ingresa una empresa inexistente. <ol style="list-style-type: none"> i) El sistema indica que hubo un error al intentar guardar los datos. ii) El flujo continúa en el punto 5. 5.b El usuario ingresa una obra inexistente. <ol style="list-style-type: none"> i) El sistema indica que hubo un error al intentar guardar los datos. ii) El flujo continúa en el punto 5.

Tabla 4.14 Caso de uso. Ingresar orden de trabajo

Caso de uso	Ingresar orden de trabajo
Actor primario	Usuario
Pre-condiciones	<ul style="list-style-type: none"> - El usuario se ha identificado mediante el login. - El usuario ha actualizado la información de órdenes de trabajo.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El usuario se encuentra en el menú de órdenes de trabajo. 2. El usuario solicita ingresar una orden de trabajo. 3. El sistema despliega una página para llevar a cabo el ingreso de una orden de trabajo. 4. El usuario ingresa número identificador de la orden de trabajo y selecciona una empresa y una obra para la nueva orden de trabajo. 5. El sistema despliega un mensaje indicando que la orden de trabajo se ingresó exitosamente.
Flujo alternativo	<p>2.a El sistema no posee información para crear una orden de trabajo.</p> <p>i) El sistema indica que es necesario actualizar la información de órdenes de trabajo desde el menú principal para poder ingresar una.</p> <p>ii) El flujo continúa en el punto 1.</p> <p>4.a El usuario ingresa un número identificador que ya está registrado.</p> <p>i) El sistema indica que el número identificador ya está registrado.</p> <p>ii) El flujo continúa en el punto 4.</p> <p>4.b El usuario ingresa una empresa inexistente.</p> <p>i) El sistema indica que hubo un error al intentar guardar los datos.</p> <p>ii) El flujo continúa en el punto 4.</p> <p>4.c El usuario ingresa una obra inexistente.</p> <p>i) El sistema indica que hubo un error al intentar guardar los datos.</p> <p>ii) El flujo continúa en el punto 4.</p>

Tabla 4.16 Caso de uso. Crear formulario

Caso de uso	Crear formulario
Actor primario	Usuario
Pre-condiciones	<ul style="list-style-type: none"> - El usuario se ha identificado mediante el login. - Existe por lo menos una orden de trabajo registrada. - El usuario tiene asociado como mínimo un rol con permisos a lo menos para crear un formulario.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El usuario se encuentra en el menú de formularios. 2. El usuario solicita crear un formulario. 3. El sistema despliega una página con las órdenes de trabajo registradas. 4. El usuario selecciona una orden de trabajo a la cual asociar el nuevo formulario. 5. El usuario selecciona área y luego ensayo. 6. El sistema despliega el formulario. 7. El usuario ingresa los datos. 8. El sistema indica que el formulario ha sido guardado exitosamente.
Flujo alternativo	<ol style="list-style-type: none"> 2.a El usuario no ha registrado una orden de trabajo. <ol style="list-style-type: none"> i) El sistema indica que no hay una orden de trabajo registrada a la cual asociar el formulario. ii) El flujo continúa en el punto 1. 2.b El usuario no posee asociado un rol con permisos. <ol style="list-style-type: none"> i) El sistema indica que no tiene permisos para crear un formulario. ii) El flujo continúa en el punto 1.

Tabla 4.17 Caso de uso. Crear copia de formulario

Caso de uso	Crear copia de formulario
Actor primario	Usuario
Pre-condiciones	<ul style="list-style-type: none"> - El usuario se ha identificado mediante el login. - Existe por lo menos un formulario almacenado.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El usuario se encuentra en el menú de formularios. 2. El usuario selecciona un formulario y solicita crear una copia. 3. El sistema despliega el formulario seleccionado junto a su información. 4. El usuario selecciona guardar. 5. El sistema indica que el formulario ha sido guardado exitosamente.
Flujo alternativo	<ol style="list-style-type: none"> 4.a El usuario selecciona cancelar. <ol style="list-style-type: none"> i) El sistema cancela la copia del formulario y cierra el ensayo sin guardarlo. ii) El flujo continúa en el punto 1.

Tabla 4.18 Caso de uso. Editar formulario

Caso de uso	Editar formulario
Actor primario	Usuario
Pre-condiciones	<ul style="list-style-type: none"> - El usuario se ha identificado mediante el login. - Existe por lo menos un formulario almacenado.
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El usuario se encuentra en el menú de formularios. 2. El sistema muestra los formularios almacenados. 3. El usuario selecciona el formulario y solicita editarlo. 4. El sistema despliega el formulario junto a sus campos a editar. 5. El usuario cambia los datos. 6. El sistema indica que el formulario ha sido guardado exitosamente.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El usuario solicita cancelar. <ol style="list-style-type: none"> i) El sistema cancela la edición del formulario y cierra el ensayo sin aplicar los cambios. ii) El flujo continúa en el punto 1.

Tabla 4.19 Caso de uso. Enviar formularios

Caso de uso	Enviar formularios
Actor primario	Usuario
Pre-condiciones	<ul style="list-style-type: none"> - El usuario se ha identificado mediante el login. - Existe por lo menos un formulario almacenado con estado realizado.
Post-condiciones	<ul style="list-style-type: none"> - Se eliminan los formularios enviados del dispositivo móvil.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario solicita en el menú principal enviar los formularios almacenados. 2. El sistema despliega la página de sincronización con la información registrada del servidor y puerto de destino. 3. El usuario solicita enviar los formularios. 4. El sistema se conecta al servidor y envía los formularios que posean estado realizado. 5. El sistema informa que los formularios se enviaron exitosamente.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El usuario solicita volver. <ol style="list-style-type: none"> i) El sistema cancela el envío y vuelve al menú principal. 4.a El dispositivo no logra conectarse al servidor. <ol style="list-style-type: none"> i) El sistema da aviso que no se pudo conectar al servidor. ii) El usuario puede intentar de nuevo o volver al menú principal.

Tabla 4.20 Caso de uso. Actualizar información en PDA

Caso de uso	Actualizar información en PDA
Actor primario	Usuario
Pre-condiciones	-
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El usuario solicita en el menú principal actualizar la información de la PDA. 2. El sistema despliega la página de sincronización con la información registrada del servidor y puerto de destino. 3. El usuario solicita actualizar información de órdenes de trabajo o de usuarios. 4. El sistema se conecta al servidor y actualiza los datos de órdenes de trabajo o usuarios. 5. El sistema informa que la actualización se llevó a cabo exitosamente.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El usuario solicita volver. <ol style="list-style-type: none"> i) El sistema cancela la actualización y vuelve al menú principal. 4.a El dispositivo no logra conectarse al servidor. <ol style="list-style-type: none"> i) El sistema indica que no se pudo conectar al servidor. ii) El usuario puede intentar de nuevo o volver al menú principal.

Tabla 4.21 Caso de uso. Obtener formularios

Caso de uso	Obtener formularios
Actor primario	Usuario
Pre-condiciones	-
Post-condiciones	-
Flujo principal	<ol style="list-style-type: none"> 1. El usuario solicita en el menú principal obtener formularios desde el servidor. 2. El sistema despliega las órdenes de trabajo y obras registradas. 3. El usuario selecciona una orden de trabajo y/u obra para descargar sus formularios asociados. 4. El sistema despliega la página de sincronización con la información registrada del servidor y puerto de destino. 5. El usuario solicita actualizar información. 6. El sistema se conecta al servidor y descarga los formularios asociados que posean estado pendiente. 5. El sistema informa que los formularios se guardaron exitosamente.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El usuario solicita cancelar. <ol style="list-style-type: none"> i) El sistema cancela la actualización y vuelve al menú principal. 4.a El usuario solicita volver. <ol style="list-style-type: none"> i) El flujo continúa en el punto 2. 6.a El dispositivo no logra conectarse al servidor. <ol style="list-style-type: none"> i) El sistema indica que no se pudo conectar al servidor. ii) El usuario puede intentar de nuevo o volver al menú principal. 6.b El sistema no descarga formularios. <ol style="list-style-type: none"> i) El sistema indica que no se encontraron formularios asociados a los datos ingresados. ii) El flujo continúa en el punto 4.

Tabla 4.22 Caso de uso. Regenerar base de datos

Caso de uso	Regenerar base de datos
Actor primario	Usuario
Pre-condiciones	-
Post-condiciones	- Se elimina toda la información almacenada en el dispositivo móvil.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario solicita en el menú de la página de bienvenida regenerar base de datos. 2. El sistema despliega la página para regenerar la base de datos. 3. El usuario escoge regenerar. 4. El sistema despliega un mensaje solicitando la confirmación de la acción. 5. El usuario confirma la acción de regenerar la base de datos. 6. El sistema muestra un mensaje indicando que el proceso finalizó exitosamente.
Flujo alternativo	<ol style="list-style-type: none"> 2.a El usuario solicita volver. <ol style="list-style-type: none"> i) El sistema vuelve a la página de bienvenida. 5.a El usuario no confirma la acción de regenerar la base de datos. <ol style="list-style-type: none"> i) El sistema cancela la acción y vuelve a la página de bienvenida.

ANEXO B. ESPECIFICACIÓN CASOS DE PRUEBA

Especificación Casos de Prueba Administrador

Las siguientes tablas representan los casos de prueba correspondientes a las acciones realizadas por el administrador, los cuales están enfocadas a los casos de uso del mismo actor.

Tabla 5.1 Caso de prueba. Añadir usuario

ID Caso de Prueba	CP-1
Nombre de Caso	Añadir usuario
Tipo de Prueba	Funcional
Objetivo	Añadir al sistema un registro de usuario.
Interfaz	
Precondiciones	El administrador se ha identificado mediante el login.
Caso de Uso relacionado	Añadir usuario

ID	Datos de Prueba	Descripción de la prueba	Resultado esperado	Observación
1		Seleccionar “ingresar usuario” en el menú	El sistema despliega la página de ingreso de usuarios.	
2	Datos de prueba en blanco	Se selecciona guardar sin ingresar datos.	El sistema despliega un mensaje por cada campo no ingresado indicando que se debe ingresar un valor.	
3	Datos de prueba erróneos	Se ingresan los datos no válidos asociados al usuario: nombres, apellidos, rut y clave.	El sistema despliega un mensaje cuando un campo con valor no válido pierde el foco.	
4	Datos de prueba correctos	Se ingresan los datos válidos asociados al usuario: nombres, apellidos, rut y clave.	El sistema guarda el usuario y se despliega un mensaje de éxito.	

Tabla 5.2 Caso de prueba. Eliminar usuario

ID Caso de Prueba	CP-2
Nombre de Caso	Eliminar usuario
Tipo de Prueba	Funcional
Objetivo	Eliminar del sistema un registro de usuario.
Interfaz	
Precondiciones	- El administrador se ha identificado mediante el login. - Existe al menos un usuario registrado en el sistema.
Caso de Uso relacionado	Eliminar usuario

ID	Datos de Prueba	Descripción de la prueba	Resultado esperado	Observación
1		Se selecciona un conjunto de usuarios mediante búsqueda o listando todos los usuarios.	El sistema muestra los usuarios permitiendo eliminar, editar o mostrar detalles de un usuario específico.	
2	Datos de prueba correctos	Se selecciona eliminar usuario.	El sistema pide confirmación y luego elimina el usuario indicado.	Si el usuario es el último usuario con rol Administrador no se permite eliminarlo.

Tabla 5.3 Caso de prueba. Crear informe

ID Caso de Prueba	CP-3
Nombre de Caso	Crear informe
Tipo de Prueba	Funcional
Objetivo	Crear un informe sobre las actividades de un usuario.
Interfaz	
Precondiciones	- El administrador se ha identificado mediante el login. - Existe al menos un usuario registrado en el sistema.
Caso de Uso relacionado	Generar informe

ID	Datos de Prueba	Descripción de la prueba	Resultado esperado	Observación
1		Se selecciona “Crear informe” en el menú. Se realiza una búsqueda de usuarios y se selecciona uno específico.	El sistema muestra la página de creación de informe.	
2	Datos de prueba en blanco	Se dejan los datos fecha de inicio, fecha final en blanco.	El sistema informa al usuario que tiene que ingresar una fecha de inicio y una fecha de término.	
3	Datos de prueba erróneos	Se ingresa una fecha de inicio posterior a la fecha de término.	El sistema despliega un mensaje indicando que la fecha de inicio tiene que ser anterior a la fecha de término.	
4	Datos de prueba correctos	Se ingresan dos fechas válidas.	El sistema genera el informe en formato pdf y el usuario lo puede descargar en el menú.	

Especificación Casos de Prueba Usuario

Las siguientes tablas representan los casos de prueba correspondientes a las acciones realizadas por el usuario, los cuales están enfocadas a los casos de uso del mismo actor.

Tabla 5.4 Caso de prueba. Ingresar orden de trabajo

ID Caso de Prueba	CP-4
Nombre de Caso	Ingresar orden de trabajo
Tipo de Prueba	Funcional
Objetivo	Añadir al software de la PDA una orden de trabajo.
Interfaz	
Precondiciones	- El usuario se ha identificado mediante el login.
Caso de Uso relacionado	Ingresar orden de trabajo

ID	Datos de Prueba	Descripción de la prueba	Resultado esperado	Observación
1		Presionar Ingresar orden de trabajo.	Sistema despliega el campo para ingresar el número de la nueva orden de trabajo, también muestra la tabla con las empresas a elegir y la lista con los nombres de las obras a seleccionar. El sistema permite filtrar las empresas y las obras por nombre.	
2	Datos de prueba en blanco	El usuario ingresa los datos en los distintos ítems: N° OT, Obra, Empresa. Presiona OK.	El sistema valida que se ingresen todos los datos en cada uno de los campos. Si no se encuentra alguno de los datos el sistema despliega el mensaje “Ingrese N° OT” o “Seleccione Obra” o “Seleccione Empresa” según corresponda.	
3	Datos de prueba erróneos	El usuario ingresa los datos en los distintos ítems:	El sistema valida que todos los campos ingresados	

		<p>N° OT, Obra, Empresa.</p> <p>Presiona OK.</p>	<p>correspondan a los definidos en el objeto de negocio.</p> <p>Si el sistema detecta que algunos de los datos ingresados son erróneos, el sistema muestra un mensaje en el campo que fue ingresado erróneamente.</p>	
4	Datos de prueba correctos	<p>El usuario ingresa los datos en los distintos ítems:</p> <p>N° OT, Obra, Empresa.</p> <p>Presiona OK.</p>	<p>El sistema valida que los datos ingresados sean correctos</p> <p>Además valida que se cumplan las siguientes reglas de negocio:</p> <ul style="list-style-type: none"> - N° OT no esté repetido. - Campo Obra no esté en blanco. - Campo Empresa no esté en blanco. <p>En caso de no cumplirse alguna regla el sistema muestra un mensaje advirtiendo lo ocurrido.</p>	

Tabla 5.5 Caso de prueba. Editar orden de trabajo

ID Caso de Prueba	CP-5
Nombre de Caso	Editar orden de trabajo
Tipo de Prueba	Funcional
Objetivo	Editar en el software de la PDA una orden de trabajo.
Interfaz	
Precondiciones	- El usuario se ha identificado mediante el login. - Existe por lo menos una orden de trabajo registrada.
Caso de Uso relacionado	Editar orden de trabajo

ID	Datos de Prueba	Descripción de la prueba	Resultado esperado	Observación
1	Datos de prueba en blanco	Presionar Editar orden de trabajo.	<p>Sistema valida que esté seleccionada la orden de trabajo a editar, desde la tabla que contiene las órdenes de trabajo.</p> <p>Si el sistema detecta que no hay orden de trabajo escogida, se despliega el mensaje “Escoja una OT para editar”</p>	
2	Datos de prueba correctos	Presionar Editar orden de trabajo.	<p>Sistema valida que esté seleccionada la orden de trabajo a editar, desde la tabla que contiene las órdenes de trabajo.</p> <p>El sistema despliega una página con los datos asociados a dicha orden. También muestra la tabla con las empresas a elegir y la lista con los nombres de las obras a seleccionar.</p> <p>El sistema permite filtrar las empresas y</p>	

			las obras por nombre.	
3	Datos de prueba en blanco	El usuario ingresa los datos en los distintos ítems: Obra, Empresa. Presiona OK.	El sistema valida que se ingresen todos los datos en cada uno de los campos a editar. Si no se encuentra alguno de los datos el sistema despliega el mensaje “Seleccione Obra” o “Seleccione Empresa” según corresponda.	
4	Datos de prueba erróneos	El usuario ingresa los datos en los distintos ítems: Obra, Empresa. Presiona OK.	El sistema valida que todos los campos ingresados correspondan a los definidos en el objeto de negocio. Si el sistema detecta que algunos de los datos ingresados son erróneos, el sistema muestra un mensaje indicando el campo que fue ingresado erróneamente.	
5	Datos de prueba correctos	El usuario ingresa los datos en los distintos ítems: Obra, Empresa. Presiona OK.	El sistema valida que los datos ingresados sean correctos Además valida que se cumplan las siguientes reglas de negocio: - Campo Obra no esté en blanco. - Campo Empresa no esté en blanco. En caso de no cumplirse alguna regla el sistema muestra un mensaje advirtiendo lo ocurrido.	

Tabla 5.6 Caso de prueba. Crear formulario

ID Caso de Prueba	CP-6
Nombre de Caso	Crear formulario
Tipo de Prueba	Funcional
Objetivo	Ingresar un formulario en el software de la PDA.
Interfaz	
Precondiciones	- El usuario se ha identificado mediante el login. - Existe por lo menos una orden de trabajo registrada.
Caso de Uso relacionado	Crear formulario

ID	Datos de Prueba	Descripción de la prueba	Resultado esperado	Observación
1		Presionar Crear formulario.	Sistema despliega una pantalla con todas las órdenes de trabajo registradas. Sistema indica que se debe seleccionar una orden de trabajo y luego escoger el área y el formulario a crear.	
2		El usuario selecciona la orden de trabajo y escoge el área y el formulario a ingresar.	Sistema despliega el formulario y sus campos a llenar separados por páginas además de un botón guardar.	
3	Datos de prueba en Blanco	El usuario ingresa los datos en los distintos ítems. Presiona OK.	El sistema valida que se ingresen todos los datos en cada uno de los campos a editar. Si no se encuentra alguno de los datos el sistema despliega el mensaje “Ingrese un valor válido en el campo...”, indicando el nombre del campo según corresponda.	
4	Datos de prueba erróneos	El usuario ingresa los datos en los distintos ítems.	El sistema valida que todos los campos ingresados	

		<p>Presiona OK.</p>	<p>correspondan a los definidos en el objeto de negocio.</p> <p>Si el sistema detecta que algunos de los datos ingresados son erróneos, el sistema muestra un mensaje indicando el campo que fue ingresado erróneamente.</p>	
5	Datos de prueba correctos	<p>El usuario ingresa los datos en los distintos ítems.</p> <p>Presiona OK.</p>	<p>El sistema valida que los datos ingresados sean correctos</p> <p>Además valida que se cumplan las siguientes reglas de negocio:</p> <ul style="list-style-type: none"> - No exista ningún campo en blanco. - Se ingresen los tipos de datos correspondientes a los tipos de datos de los campos. - Los datos ingresados no excedan el tamaño máximo soportado o establecido <p>En caso de no cumplirse alguna regla el sistema muestra un mensaje advirtiendo lo ocurrido.</p>	

Tabla 5.7 Caso de prueba. Editar formulario

ID Caso de Prueba	CP-7
Nombre de Caso	Editar formulario
Tipo de Prueba	Funcional
Objetivo	Editar un formulario almacenado en la PDA.
Interfaz	
Precondiciones	- El usuario se ha identificado mediante el login. - Existe por lo menos un formulario registrado.
Caso de Uso relacionado	Editar formulario

ID	Datos de Prueba	Descripción de la prueba	Resultado esperado	Observación
1			Sistema despliega una pantalla con todos formularios registrados.	
2		El usuario selecciona el formulario a editar desde la tabla que muestra los formularios. Presionar Editar formulario.	Sistema despliega el formulario seleccionado y los sus campos con los respectivos valores almacenados y se encuentra listo para editar.	
3	Datos de prueba en Blanco	El usuario edita los datos de los distintos ítems. Presiona OK.	El sistema valida que se ingresen todos los datos en cada uno de los campos a editar. Si no se encuentra alguno de los datos el sistema despliega el mensaje “Ingrese un valor válido en el campo...”, indicando el nombre del campo según corresponda.	
4	Datos de prueba erróneos	El usuario edita los datos en los distintos ítems. Presiona OK.	El sistema valida que todos los campos editados correspondan a los definidos en el objeto de negocio. Si el sistema detecta	

			que algunos de los datos editados son erróneos, el sistema muestra un mensaje indicando el campo que fue editado erróneamente.	
5	Datos de prueba correctos	El usuario edita los datos en los distintos ítems. Presiona OK.	El sistema valida que los datos editados sean correctos Además valida que se cumplan las siguientes reglas de negocio: <ul style="list-style-type: none"> - No exista ningún campo en blanco. - Se ingresen los tipos de datos correspondientes a los tipos de datos de los campos. - Los datos editados no excedan el tamaño máximo soportado o establecido En caso de no cumplirse alguna regla el sistema muestra un mensaje advirtiendo lo ocurrido.	

Tabla 5.8 Caso de prueba. Enviar formularios

ID Caso de Prueba	CP-8
Nombre de Caso	Enviar formularios
Tipo de Prueba	Funcional
Objetivo	Enviar a la base de datos centralizada los formularios almacenados en la PDA.
Interfaz	
Precondiciones	- El usuario se ha identificado mediante el login. - Existe por lo menos un formulario registrado.
Caso de Uso relacionado	Enviar formularios ingresados.

ID	Datos de Prueba	Descripción de la prueba	Resultado esperado	Observación
1		Presionar formularios. Enviar	Sistema despliega una pantalla junto a dos textbox, uno para ingresar la dirección ip del servidor y otro para ingresar el puerto del servicio web al que se enviarán los datos.	
2	Datos de prueba en Blanco	El usuario ingresa los datos. Presiona Transferir información	El sistema valida que se ingresen todos los datos en cada uno de los textbox. Si no se encuentra alguno de los datos el sistema despliega el mensaje “Ingrese dirección IP” o “Ingrese Puerto” según corresponda.	
3	Datos de prueba erróneos	El usuario ingresa los datos. Presiona Transferir información	El sistema valida que todos los campos ingresados correspondan a los definidos en el objeto de negocio. Si el sistema detecta que algunos de los datos ingresados son erróneos, el sistema muestra un mensaje en	

			el campo que fue ingresado erróneamente.	
4	Datos de prueba correctos	El usuario ingresa los datos. Presiona Transferir información	El sistema valida que los datos ingresados sean correctos Luego solicita el servicio desplegando el mensaje “Procesando información”, seguido del mensaje “Transfiriendo información”. En caso de recibir respuesta positiva por parte del servicio el sistema indica el mensaje “Formularios enviados correctamente”. En caso de recibir respuesta negativa por parte del servicio el sistema indica el mensaje “Los formularios no fueron recibidos correctamente”. En caso de no recibir respuesta por parte del servicio se muestra el mensaje “Error al transferir los datos”, “Revise la dirección IP y el Puerto e intente nuevamente, en caso contrario contacte al administrador del sistema”.	

Tabla 5.9 Caso de prueba. Actualizar información en PDA

ID Caso de Prueba	CP-9
Nombre de Caso	Actualizar información en PDA
Tipo de Prueba	Funcional
Objetivo	Descargar información desde la base de datos centralizada y almacenarla en la PDA.
Interfaz	
Precondiciones	
Caso de Uso relacionado	Actualizar información en PDA.

ID	Datos de Prueba	Descripción de la prueba	Resultado esperado	Observación
1		Presionar Actualizar información.	Sistema despliega una pantalla junto a un botón de confirmación y al mensaje “Desea actualizar la información de la PDA”	
2		El usuario selecciona si en el botón de confirmación.	<p>El sistema muestra el mensaje “Procesando información”, seguido del mensaje “Descargando información”.</p> <p>En caso de recibir respuesta por parte del servicio y recibir toda la información en forma correcta indica el mensaje “Información descargada correctamente”.</p> <p>En caso de recibir respuesta por parte del servicio pero no recibir toda la información en forma correcta indica el mensaje “No se descargó la información de forma correcta”.</p> <p>En caso de no recibir respuesta por parte del servicio se muestra el</p>	

			mensaje "Error al descargar los datos", seguido del mensaje, "Revise la dirección IP y el Puerto e intente nuevamente, en caso contrario contacte al administrador del sistema".	
--	--	--	--	--

ANEXO C. NORMAS LEGALES

Biblioteca del Congreso Nacional

Identificación de la Norma: LEY-19223

Fecha de Publicación: 07.06.1993

Fecha de Promulgación: 28.05.1993

Organismo: MINISTERIO DE JUSTICIA

TIPIFICA FIGURAS PENALES RELATIVAS A LA INFORMATICA

Teniendo presente que el H. Congreso Nacional ha dado su aprobación al siguiente Proyecto de Ley:

Artículo 1°.- El que maliciosamente destruya o inutilice un sistema de tratamiento de información o sus partes o componentes, o impida, obstaculice o modifique su funcionamiento, sufrirá la pena de presidio menor en su grado medio a máximo.

Si como consecuencia de estas conductas se afectaren los datos contenidos en el sistema, se aplicará la pena señalada en el inciso anterior, en su grado máximo.

Artículo 2°.- El que con el ánimo de apoderarse, usar o conocer indebidamente de la información contenida en un sistema de tratamiento de la misma, lo intercepte, interfiera o acceda a él, será castigado con presidio menor en su grado mínimo a medio.

Artículo 3°.- El que maliciosamente altere, dañe o destruya los datos contenidos en un sistema de tratamiento de información, será castigado con presidio menor en su grado medio.

Artículo 4°.- El que maliciosamente revele o difunda los datos contenidos en un sistema de información, sufrirá la pena de presidio menor en su grado medio. Si quien incurre en estas conductas es el responsable del sistema de información, la pena se aumentará en un grado.

Y por cuanto he tenido a bien aprobarlo y sancionarlo; por tanto promúlguese y llévese a efecto como Ley de la República.