

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERIA DE SISTEMAS

DESARROLLO E IMPLANTACIÓN DEL SISTEMA DE GESTIÓN DE ANÁLISIS FÍSICO-QUÍMICO Y MICROBIOLÓGICO DE AGUAS, SUELOS Y LODOS PARA EL CENTRO DE INVESTIGACIÓN DE CONTROL AMBIENTAL (CICAM) DE LA EPN

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

LILIANA FERNANDA CHACHA BURGOS

email: lilifer120@gmail.com

ELIANA JEANETH VISCARRA PRUNA

email: elyjvp_25@yahoo.com

DIRECTOR: ING. FRANCISCO HORACIO VILLAVIVENCIO OTAÑEZ

email: francisco.villavicencio@epn.edu.ec

Quito, Julio 2010

DECLARACIÓN

Nosotras, Liliana Fernanda Chacha Burgos y Eliana Jeaneth Viscarra Pruna, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Liliana Fernanda Chacha Burgos

Eliana Jeaneth Viscarra Pruna

CERTIFICADO

Certifico que el presente trabajo fue realizado por las señoritas Chacha Burgos Liliana Fernanda y Viscarra Pruna Eliana Jeaneth, bajo mi supervisión.

Ing. Francisco Villavicencio
Director de Tesis

CONTENIDO

RESUMEN	XIII
CAPÍTULO 1: PLANTEAMIENTO DEL PROBLEMA	1
1.1. DESCRIPCIÓN DEL CICAM.....	1
1.1.1. ORGANIGRAMA DEL CICAM.....	1
1.1.2. DESCRIPCIÓN GENERAL DE LOS CARGOS.....	1
1.2. DESCRIPCIÓN DEL PROBLEMA.....	2
1.3. SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO.....	4
1.3.1. JUSTIFICACIÓN DE SELECCIÓN DE LA METODOLOGÍA RUP.....	4
1.3.1.1. Fase De Inicio.....	5
1.3.1.2. Fase De Elaboración.....	5
1.3.1.3. Fase De Construcción.....	5
1.3.1.4. Fase De Transición.....	6
1.3.1.5. Documentación.....	6
1.4. JUSTIFICACIÓN DE HERRAMIENTAS DE DESARROLLO.....	6
1.4.1. SOFTWARE LIBRE.....	6
1.4.2. J2EE.....	8
1.4.3. ECLIPSE.....	11
1.4.4. POSTGRESQL.....	13
1.4.5. SERVIDOR DE APLICACIONES JBOSS.....	15
1.4.6. POWER DESIGNER.....	17
1.4.7. RATIONAL ROSE.....	19
CAPITULO2: ANÁLISIS Y DISEÑO	21
2.1. ESPECIFICACIÓN DE REQUERIMIENTOS.....	21
2.1.1. ADMINISTRACIÓN DEL PROYECTO.....	21
2.1.1.1. Plan de Desarrollo de Software.....	22
2.1.1.1.1. Introducción.....	22
2.1.1.1.2. Propósito del Plan de Desarrollo de Software.....	22
2.1.1.1.3. Alcance del Plan de Desarrollo de Software.....	23
2.1.1.1.4. Resumen.....	23
2.1.1.1.5. Vista general del proyecto.....	24
2.1.1.1.6. Organización del proyecto.....	29
2.1.1.1.7. Gestión del proceso.....	30

2.1.1.2.	Modelamiento del Negocio.....	34
2.1.1.2.1.	Modelo de Casos de Uso del Negocio.....	35
2.1.1.2.2.	Descripción de Casos de Uso del Negocio.....	36
2.1.1.2.3.	Modelo del Dominio	39
2.1.1.3.	Análisis de Riesgos.....	40
2.1.1.3.1.	Riesgos del Proyecto	40
2.1.1.3.2.	Riesgos del Producto.....	41
2.1.1.3.3.	Riesgo técnico	41
2.1.2.	REQUISITOS	42
2.1.2.1.	Visión.....	43
2.1.2.1.1.	Alcance	43
2.1.2.1.2.	Posicionamiento del Producto	43
2.1.2.1.3.	Descripción Global del Producto.....	45
2.1.2.2.	Requisitos Específicos.....	45
2.1.2.2.1.	Alcance	45
2.1.2.2.2.	Perspectiva del Producto	46
2.1.2.2.3.	Funcionalidad del Producto	46
2.1.2.2.4.	Restricciones	47
2.1.2.2.5.	Requisitos Funcionales.....	47
2.1.2.2.6.	Requisitos no Funcionales.....	48
2.1.2.3.	Casos de Uso	49
2.1.2.3.1.	Subsistemas.....	49
2.1.2.3.2.	Diagrama de casos de uso del sistema.....	50
2.1.2.3.3.	Especificación de Casos de Usos	52
2.2.	ANÁLISIS	78
2.2.1.	DIAGRAMAS DE SECUENCIA.....	78
2.2.1.1.	Diagrama de Secuencia Ingresar al Sistema.....	80
2.2.1.2.	Diagrama de Secuencia Registrar Unidad	80
2.2.1.3.	Diagrama de Secuencia Registrar Tipo Cliente.....	80
2.2.1.4.	Diagrama de Secuencia Registrar Tipo Servicio	81
2.2.1.5.	Diagrama de Secuencia Registrar Tipo Producto	81
2.2.1.6.	Diagrama de Secuencia Registrar Tipo Personal.....	81
2.2.1.7.	Diagrama de Secuencia Registrar Cargo Personal.....	82
2.2.1.8.	Diagrama de Secuencia Registrar Laboratorio.....	82

2.2.1.9.	Diagrama de Secuencia Registrar Personal.....	83
2.2.1.10.	Diagrama de Secuencia Registrar Cliente.....	83
2.2.1.11.	Diagrama de Secuencia Registrar Unidad de Medida.....	84
2.2.1.12.	Diagrama de Secuencia Registrar Norma.....	84
2.2.1.13.	Diagrama de Secuencia Registrar Producto.....	84
2.2.1.14.	Diagrama de Secuencia Registrar Servicio.....	85
2.2.1.15.	Diagrama de Secuencia Registrar Método.....	85
2.2.1.16.	Diagrama de Secuencia Generar Proforma.....	86
2.2.1.17.	Diagrama de Secuencia Registrar Muestra.....	87
2.2.1.18.	Diagrama de Secuencia Consultar Facturas Canceladas.....	87
2.2.1.19.	Diagrama de Secuencia Generar Orden de Trabajo.....	88
2.2.1.20.	Diagrama de Secuencia Asignar Carga Trabajo.....	88
2.2.1.21.	Diagrama de Secuencia Generar Informe de Resultados.....	89
2.2.1.22.	Diagrama de Secuencia Consultar Proformas Aceptadas.....	89
2.2.1.23.	Diagrama de Secuencia Generar Factura.....	90
2.2.2.	DIAGRAMAS DE ACTIVIDAD.....	90
2.2.2.1.	Diagrama de Actividad Ingresar al Sistema.....	91
2.2.2.2.	Diagrama de Actividad Registrar Unidad.....	91
2.2.2.3.	Diagrama de Actividad Registrar Tipo Cliente.....	92
2.2.2.4.	Diagrama de Actividad Registrar Tipo Servicio.....	92
2.2.2.5.	Diagrama de Actividad Registrar Tipo Producto.....	93
2.2.2.6.	Diagrama de Actividad Registrar Tipo Personal.....	93
2.2.2.7.	Diagrama de Actividad Registrar Cargo Personal.....	94
2.2.2.8.	Diagrama de Actividad Registrar Laboratorio.....	94
2.2.2.9.	Diagrama de Actividad Registrar Personal.....	95
2.2.2.10.	Diagrama de Actividad Registrar Cliente.....	95
2.2.2.11.	Diagrama de Actividad Registrar Unidad Medida.....	96
2.2.2.12.	Diagrama de Actividad Registrar Norma.....	96
2.2.2.13.	Diagrama de Actividad Registrar Producto.....	97
2.2.2.14.	Diagrama de Actividad Registrar Servicio.....	97
2.2.2.15.	Diagrama de Actividad Registrar Método.....	98
2.2.2.16.	Diagrama de Actividad Registrar Muestra.....	98
2.2.2.17.	Diagrama de Actividad Generar Proforma.....	99
2.2.2.18.	Diagrama de Actividad Consultar Facturas Canceladas.....	99

2.2.2.19.	Diagrama de Actividad Generara Orden Trabajo	100
2.2.2.20.	Diagrama de Actividad Asignar Carga Trabajo	100
2.2.2.21.	Diagrama de Actividad Generar Informe Resultados.....	101
2.2.2.22.	Diagrama de Actividad Consultar Proformas Aceptadas	101
2.2.2.23.	Diagrama de Actividad Generar Factura	101
2.3.	DISEÑO	102
2.3.1.	ESQUEMA DE FUNCIONAMIENTO Y ARQUITECTURA DEL SISTEMA	102
2.3.1.1.	Esquema de Funcionamiento del Sistema	102
2.3.1.2.	Arquitectura del Sistema.....	103
2.3.2.	DIAGRAMA DE CLASES	104
2.3.3.	MODELO DE DATOS RELACIONAL	107
CAPITULO 3: IMPLEMENTACIÓN Y PRUEBAS		109
3.1.	IMPLEMENTACIÓN	109
3.1.1.	PROTOTIPO DE INTERFACES DE USUARIO.....	109
3.1.1.1.	Pantalla Normal.....	109
3.1.1.2.	Pantalla con Encabezado y Detalle.....	111
3.1.2.	COMPONENTES Y DESPLIEGUE	112
3.1.2.1.	Diagrama de Componentes.....	112
3.1.2.2.	Diagrama Global de Paquetes	113
3.1.2.3.	Diagrama de Despliegue.....	115
3.1.3.	CÓDIGO E INTERFACES.....	116
3.1.3.1.	Modelo Vista Controlador en JSF	116
3.1.3.2.	Organización del Código de acuerdo al Patrón MVC en JSF.....	118
3.2.	PRUEBAS.....	130
3.2.1.	CASOS DE PRUEBA.....	130
3.2.1.1.	Caso de Prueba Ingresar al Sistema (CP01).....	130
3.2.1.2.	Caso de Prueba Registrar unidad (CP02).....	131
3.2.1.3.	Caso de Prueba Registrar Tipo Cliente (CP03).....	131
3.2.1.4.	Caso de Prueba Registrar Tipo Servicio (CP04)	132
3.2.1.5.	Caso de Prueba Registrar Tipo Producto (CP05)	132
3.2.1.6.	Caso de Prueba Registrar Tipo Personal (CP06).....	132
3.2.1.7.	Caso de Prueba Registrar Cargo Personal (CP07).....	133
3.2.1.8.	Caso de Prueba Registrar Laboratorio (CP08)	133
3.2.1.9.	Caso de Prueba Registrar Tipo Personal (CP09).....	134

3.2.1.10.	Caso de Prueba Registrar Cliente (CP10)	135
3.2.1.11.	Caso de Prueba Registrar Unidad Medida (CP11).....	135
3.2.1.12.	Caso de Prueba Registrar Norma (CP12).....	136
3.2.1.13.	Caso de Prueba Registrar Producto (CP13)	136
3.2.1.14.	Caso de Prueba Registrar Servicio (CP14)	136
3.2.1.15.	Caso de Prueba Registrar Método (CP15).....	137
3.2.1.16.	Caso de Prueba Generar Proforma (CP16).....	137
3.2.1.17.	Caso de Prueba Registrar Muestra (CP17)	138
3.2.1.18.	Caso de Prueba Generar Orden Trabajo (CP19).....	139
3.2.1.19.	Caso de Prueba Asignar Carga Trabajo (CP20).....	139
3.2.1.20.	Caso de Prueba Generar Informe Resultados (CP21).....	140
3.2.1.21.	Caso de Prueba Consultar Proformas Aceptadas (CP22)	141
3.2.1.22.	Caso de Prueba Generar Factura (CP23).....	141
CAPITULO 4: IMPLANTACIÓN Y EVALUACIÓN DE RESULTADO.....		142
4.1.	IMPLANTACIÓN	142
4.1.1.	DEFINICIÓN DEL AMBIENTE DE PRUEBAS	142
4.1.1.1.	Perfil de Usuario	142
4.1.1.2.	Usuarios.....	143
4.1.1.3.	Hardware recomendado	143
4.1.1.4.	Software recomendado.....	143
4.2.	EVALUACIÓN DE RESULTADOS.....	144
4.2.1.	RESULTADOS: USUARIO ADMINISTRADOR	144
4.2.2.	RESULTADOS: USUARIO SECRETARIA.....	145
4.2.3.	RESULTADOS: USUARIO TESORERA.....	146
4.2.4.	RESULTADOS: USUARIO ANALISTA.....	146
4.2.5.	RESUMEN DE LA EVALUACIÓN DE RESULTADOS	147
CAPITULO 5: CONCLUSIONES Y RECOMENDACIONES.....		148
5.1.	CONCLUSIONES	148
5.2.	RECOMENDACIONES	149
GLOSARIO		151
BIBLIOGRAFÍA.....		156
ANEXOS		161

INDICE DE TABLAS

CAPITULO 2

Tabla 2. 1 Roles y Responsabilidades	29
Tabla 2. 2 Plan de Fases	30
Tabla 2. 3 Fases de Proyecto	31
Tabla 2. 4 Calendario del Proyecto	32
Tabla 2. 5 Costos del Proyecto	33
Tabla 2. 6 Caso de Uso del Negocio Solicitar Análisis	37
Tabla 2. 7 Caso de Uso del Negocio Registrar Cliente	37
Tabla 2. 8 Caso de Uso del Negocio Ingresar Servicio	37
Tabla 2. 9 Caso de Uso del Negocio Generar Proforma	37
Tabla 2. 10 Caso de Uso del Negocio Generar Factura	38
Tabla 2. 11 Caso de Uso del Negocio Generar Orden trabajo	38
Tabla 2. 12 Caso de Uso del Negocio Consultar Orden Trabajo	38
Tabla 2. 13 Caso de Uso del Negocio Generar Informe Resultados	38
Tabla 2. 14 Caso de Uso del Negocio Elegir Metodo Aplicar	39
Tabla 2. 15 Caso de Uso del Negocio Realizar Análisis	39

CAPITULO 3

Tabla 3. 1 Descripción Botones	130
Tabla 3. 2 Descripción de Links	130
Tabla 3. 3 Caso de Prueba Ingresar al Sistema	131
Tabla 3. 4 Caso de Prueba Registrar Unidad	131
Tabla 3. 5 Caso de Prueba Registrar Tipo Cliente	132
Tabla 3. 6 Caso de Prueba Registrar Tipo Servicio	132
Tabla 3. 7 Caso de Prueba Registrar Tipo Producto	132
Tabla 3. 8 Caso de Prueba Registrar Tipo Personal	133
Tabla 3. 9 Caso de Prueba Registrar Cargo Personal	133
Tabla 3. 10 Caso de Prueba Registrar Laboratorio	134
Tabla 3. 11 Caso de Prueba Registrar Personal	134
Tabla 3. 12 Caso de Prueba Registrar Cliente	135
Tabla 3. 13 Caso de Prueba Registrar Unidad Medida	135
Tabla 3. 14 Caso de Prueba Registrar Norma	136
Tabla 3. 15 Caso de Prueba Registrar Producto	136
Tabla 3. 16 Caso de Prueba Registrar Servicio	137
Tabla 3. 17 Caso de Prueba Registrar Método	137
Tabla 3. 18 Caso de Prueba Generar Proforma	138
Tabla 3. 19 Caso de Prueba Registrar Muestra	139
Tabla 3. 20 Caso de Prueba Generar Orden Trabajo	139
Tabla 3. 21 Caso de Prueba Asignar Carga Trabajo	140
Tabla 3. 22 Caso de Prueba Generar Informe Resultados	140
Tabla 3. 23 Caso de Prueba Consultar Proformas Aceptadas	141
Tabla 3. 24 Caso de Prueba Generar Factura	141

CAPITULO 4

Tabla 4. 1 Perfiles de Usuario Sugeridos para el uso del sistema	143
--	-----

Tabla 4. 2 Usuario del sistema.....	143
Tabla 4. 3 Hardware recomendado para el Servidor.....	143
Tabla 4. 4 Hardware recomendado para los Terminales.....	143
Tabla 4. 5 Software recomendado para los Terminales.....	144
Tabla 4. 6 Formato de Presentación Evaluación.....	144
Tabla 4. 7 Resultados Evaluación Usuario Administrador.....	144
Tabla 4. 8 Resultados Evaluación Usuario Secretaria.....	145
Tabla 4. 9 Resultados Evaluación Usuario Tesorera.....	146
Tabla 4. 10 Resultados Evaluación Usuario Analista.....	147
Tabla 4. 11 Resultados Evaluación Usuario.....	147

INDICE DE FIGURAS

CAPITULO 1

Figura1. 1 Organigrama del CICAM [4].....	1
Figura1. 2 Plataforma J2EE [37].....	11
Figura1. 3 Entorno Eclipse.....	13
Figura1. 4 PostgreSQL.....	14
Figura1. 5 Servidor de Aplicaciones JBoss [1].....	16
Figura1. 6 JBoss.....	17
Figura1. 7 Power Designer.....	18
Figura1. 8 Interfaz de Usuario.....	20

CAPITULO 2

Figura 2. 1 Diagrama de Metodología RUP [5].....	31
Figura 2. 2 Diagrama de casos de uso del Negocio.....	36
Figura 2. 3 Modelo del Dominio.....	40
Figura 2. 6 Subsistemas.....	50
Figura 2. 7 Módulo Administrativo.....	50
Figura 2. 8 Administración de Servicios.....	51
Figura 2. 9 Administración de Análisis.....	51
Figura 2. 10 Facturación.....	52
Figura 2. 11 Diagrama de Secuencia Ingresar al Sistema.....	80
Figura 2. 12 Diagrama de Secuencia Registrar Unidad.....	80
Figura 2. 13 Diagrama de Secuencia Registrar Tipo Cliente.....	80
Figura 2. 14 Diagrama de Secuencia Registrar Tipo Servicio.....	81
Figura 2. 15 Diagrama de Secuencia Registrar Tipo Producto.....	81
Figura 2. 16 Diagrama de Secuencia Registrar Tipo Personal.....	81
Figura 2. 17 Diagrama de Secuencia Registrar Cargo Personal.....	82
Figura 2. 18 Diagrama de Secuencia Registrar Laboratorio.....	82
Figura 2. 19 Diagrama de Secuencia Registrar Personal.....	83
Figura 2. 20 Diagrama de Secuencia Registrar Cliente.....	83
Figura 2. 21 Diagrama de Secuencia Registrar Unidad Medida.....	84
Figura 2. 22 Diagrama de Secuencia Registrar Norma.....	84
Figura 2. 23 Diagrama de Secuencia Registrar Producto.....	84

Figura 2. 24 Diagrama de Secuencia Registrar Servicio 85

Figura 2. 25 Diagrama de Secuencia Registrar Método 85

Figura 2. 26 Diagrama de Secuencia Generar Proforma 86

Figura 2. 27 Diagrama de Secuencia Registrar Muestra 87

Figura 2. 28 Diagrama de Secuencia Consultar Facturas Canceladas 87

Figura 2. 29 Diagrama de Secuencia Generar Orden Trabajo 88

Figura 2. 30 Diagrama de Secuencia Asignar Carga Trabajo 88

Figura 2. 31 Diagrama de Secuencia Generar Informe Resultados 89

Figura 2. 32 Diagrama de Secuencia Consultar Proformas Aceptadas 89

Figura 2. 33 Diagrama de Secuencia Generar Factura 90

Figura 2. 34 Diagrama de Actividad Ingresar al Sistema 91

Figura 2. 35 Diagrama de Actividad Registrar Unidad 91

Figura 2. 36 Diagrama de Actividad Registrar Tipo Cliente 92

Figura 2. 37 Diagrama de Actividad Registrar Tipo Servicio 92

Figura 2. 38 Diagrama de Actividad Registrar Tipo Producto 93

Figura 2. 39 Diagrama de Actividad Registrar Tipo Personal 93

Figura 2. 40 Diagrama de Actividad Registrar Cargo Personal 94

Figura 2. 41 Diagrama de Actividad Registrar Laboratorio 94

Figura 2. 42 Diagrama de Actividad Registrar Personal 95

Figura 2. 43 Diagrama de Actividad Registrar Cliente 95

Figura 2. 44 Diagrama de Actividad Registrar Unidad Medida 96

Figura 2. 45 Diagrama de Actividad Registrar Norma 96

Figura 2. 46 Diagrama de Actividad Registrar Producto 97

Figura 2. 47 Diagrama de Actividad Registrar Servicio 97

Figura 2. 48 Diagrama de Actividad Registrar Método 98

Figura 2. 50 Diagrama de Actividad Registrar Muestra 98

Figura 2. 49 Diagrama de Actividad Generar Proforma 99

Figura 2. 51 Consultar Facturas Canceladas 99

Figura 2. 52 Diagrama de Actividad Generar Orden Trabajo 100

Figura 2. 53 Diagrama de Actividad Asignar Carga Trabajo 100

Figura 2. 54 Diagrama de Actividad Generar Informe Resultados 101

Figura 2. 55 Diagrama de Actividad Consultar Proformas Aceptadas 101

Figura 2. 56 Diagrama de Actividad Generar Factura 101

Figura 2. 4 Esquema de funcionamiento del sistema 103

Figura 2. 5 Arquitectura del Sistema 104

Figura 2. 57 Diagrama de Clases 106

Figura 2. 58 Modelo de Datos 108

CAPITULO 3

Figura 3. 1 Prototipo de Interfaz para ingreso de datos 109

Figura 3. 2 Prototipo de Interfaz para consultar datos 110

Figura 3. 3 Prototipo de Interfaz para eliminar datos 110

Figura 3. 4 Prototipo de Interfaz para editar datos 111

Figura 3. 5 Prototipo de Interfaz para ingresar datos con encabezado y detalle 111

Figura 3. 6 Diagrama Global de Paquetes 113

Figura 3. 7 Diagrama de Componentes Comunes	113
Figura 3. 8 Diagrama de Componentes Módulo Administrativo.....	114
Figura 3. 9 Diagrama de Componentes Administración de Servicios	114
Figura 3. 10 Diagrama de Componentes Manejo de Flujos de Trabajo.....	115
Figura 3. 11 Diagrama de Componentes Facturación	115
Figura 3. 12 Diagrama de Despliegue.....	116
Figura 3. 13 Organización de componentes de la aplicación	117
Figura 3. 14 Pantalla Consultar Unidad	127
Figura 3. 15 Pantalla Ingresar Unidad.....	127
Figura 3. 16 Pantalla Editar Unidad	128
Figura 3. 17 Pantalla Eliminar Unidad.....	128
Figura 3. 18 Pantalla para Ingresar registros con Encabezado y Detalle	129
Figura 3. 19 Pantalla Ingresar Detalle.....	129

RESUMEN

El presente proyecto de Titulación tiene por objetivo el desarrollo e implantación del Sistema de Gestión de análisis físico-químico y microbiológico de aguas, suelos y lodos para el Centro de Investigación de Control Ambiental (CICAM) de la EPN, al cual se le denomina SISTEMA DE ADMINISTRACION CENTRALIZADA DE LABORATORIOS EN LA ESCUELA POLITECNICA NACIONAL.

El Centro de Investigación de Control Ambiental es uno de los treinta laboratorios de la Escuela Politécnica Nacional, dedicados a la elaboración de análisis químicos, que actualmente no cuenta con un sistema que permita automatizar los procesos que se desarrollan durante el análisis de muestras. Es por ello que el CICAM nos solicita la elaboración de un sistema que le permita desarrollar sus procesos de análisis de manera más eficiente.

La Escuela Politécnica Nacional a través de la Unidad de Gestión Informática (UGI), detecta esta necesidad de los laboratorios químicos, de contar con un sistema informático que logre automatizar los procesos que desarrollan de manera general, de modo que el sistema pueda satisfacer las necesidades habituales de las treinta unidades de laboratorio, tomando como base los requerimientos generales del CICAM.

Como se menciona el objetivo del presente Proyecto de Tesis es la elaboración de un sistema informático para el CICAM, y dado que el sistema planteado por la UGI basará sus requerimientos en el mismo laboratorio; nos unimos a éste departamento como parte del equipo de trabajo, encargándonos de los módulos que involucran procesos específicos que cubren el objetivo de nuestro proyecto. Es así que de acuerdo al alcance del plan de tesis, nosotras desarrollamos los siguientes módulos: Módulo Administrativo, Administración de Servicios, Administración de Análisis y Facturación.

Previo al desarrollo del sistema es necesario definir la metodología y herramientas de desarrollo a usar. La Unidad de Gestión Informática basándose en su experiencia en el desarrollo de sistemas informáticos decidió escoger la metodología de desarrollo RUP, a la cual asentimos, puesto que después de nuestro análisis, concluimos que es la que mejor se adapta a este tipo de desarrollo y a la vez permite realizar un número de iteraciones de mejoramiento continuo del producto, intentando reproducir el ciclo de vida en cascada a menor escala.

La metodología RUP se basa en un conjunto de flujos de trabajo del proceso, los mismos que serán puestos en práctica de acuerdo a sus fases de inicio, elaboración, construcción, transición y documentación.

Para la definición de las herramientas de desarrollo a usar, se toma en cuenta que la Escuela Politécnica Nacional es una Institución del Estado y con el afán de promover software libre, se decide desarrollar el sistema utilizando algunas herramientas de software libre tales como; J2EE como plataforma de desarrollo y Java Eclipse como entorno de desarrollo, Postgres como motor de base de datos y Jboss como servidor de aplicaciones J2EE. Mientras que para el modelamiento se usará las herramientas de licenciamiento Power Designer y Rational Rose.

Posterior a la definición de las herramientas y la metodología a usar, se procede al desarrollo del sistema, iniciando con la etapa de Análisis y Diseño en el Capítulo 2 del documento, para lo cual se toman en cuenta las fases definidas para RUP.

Como primer punto dentro del capítulo 2 se tiene la especificación de requerimientos, sección que es dividida en dos partes: Administración del Proyecto y Requisitos dentro de las cuales se encuentran los respectivos artefactos que nos permiten determinar de forma precisa los requisitos del sistema.

La administración de proyectos se enfoca en coordinar todas las etapas del proyecto de desarrollo de software cuyos objetivos se establecen explícitamente en términos de tiempo, costos y parámetros de realización. Por lo cual es

necesario establecer el Plan de desarrollo de Software, de tal manera que se pueda mantener el control del proyecto durante cada una de sus etapas.

Técnicamente, el Plan de Desarrollo de Software es un compendio de los distintos planes que son necesarios para ejecutar un proceso de desarrollo, y que a su vez proporciona una visión global del enfoque de desarrollo propuesto para un proyecto. Posteriormente desarrollamos el Modelamiento del Negocio y El Análisis de Riesgos.

Como siguiente punto dentro del capítulo 2, se tiene Requisitos, proceso por el cual se describirá los servicios que brinda el sistema y las restricciones asociadas a su funcionamiento. Es aquí donde se define el artefacto Visión que muestra de manera general las características y necesidades del Sistema, luego de lo cual se definen los Requisitos Específicos del Sistema definiendo requisitos funcionales o no funcionales. A continuación se define el Esquema de Funcionamiento y Arquitectura del Sistema que define la accesibilidad a la aplicación y donde se encuentra alojada, además de cómo los usuarios acceden al Sistema. Finalmente se tiene el Modelado de casos de uso, donde se diseñan los diagramas de casos de uso de acuerdo a los subsistemas planteados y se realiza la especificación de cada caso de uso.

Continuando con el capítulo 2 y la fase de elaboración de RUP, se desarrollan las etapas de análisis y diseño; encargadas de dar una solución técnica, puesto que es aquí donde se transforman los requisitos al diseño del sistema. Para llevar a cabo esta tarea se desarrolla una arquitectura para el sistema y se adapta el diseño para que sea consistente con el entorno de implementación.

Durante la etapa de análisis se obtiene una visión del sistema, preocupándonos de ver qué hace, es decir que nos interesamos por los requisitos funcionales para determina los objetivos y límites del sistema, para lo cual se elaboran diagramas de secuencia y actividad

Los diagramas de secuencia se crean con el objetivo de ver gráficamente las interacciones de los distintos actores con los objetos del sistema a través de un intercambio de mensajes en un momento dado. Mientras que los diagramas de actividad demuestran la serie de actividades que se van a realizar en cada caso

de uso, representando los flujos de trabajo paso a paso que se van desencadenando, de modo que se pueda entender como es utilizado el sistema y cómo reaccionar en determinados eventos.

La etapa de diseño consiste en el refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos por ello se define la arquitectura del sistema, tomando en cuenta que se adapte al sistema de información de la organización, que tenga flexibilidad para aceptar modificaciones futuras, que ofrezca seguridad contra el uso no autorizado. En esta etapa se define el diagrama de clases y el modelo de datos.

El diagrama de clases se lo desarrollo para entender la estructura de los objetos del proyecto, los atributos propios de cada clase y las relaciones que existen entre ellas. En cambio el modelo de datos realizado nos permite describir las estructuras de datos de la base, las relaciones que existen entre ellos, la semántica, las restricciones de integridad y las operaciones de manipulación de los datos.

A continuación tenemos el capítulo 3 y la fase de construcción donde nos centramos en la implementación del sistema (producto) mediante el uso de artefactos, diagramas y modelos obtenidos en las dos fases anteriores. Finalmente se desarrollaran las correspondientes pruebas del sistema.

En la etapa de implementación se define el prototipo de interfaces del usuario diseñadas de forma general para las operaciones de ingreso, actualización, consulta y eliminación, las cuales serán la base para el diseño de la aplicación, de esta manera se le permite al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema.

Se especifica el diagrama de paquetes para mostrar como el sistema está dividido en agrupaciones lógicas, mostrando las dependencias entre estas agrupaciones. De acuerdo al diagrama de paquetes se realizan los diferentes diagramas de componentes los cuales son fragmentos de software que conforma el sistema y sus relaciones. Se define el diagrama de despliegue para mostrar la disposición física de los distintos nodos que componen el sistema y el reparto de los

componentes sobre dichos nodos. Por último se tiene el código del sistema el cual se lo distribuye de acuerdo al patrón de diseño Modelo-Vista-Controlador de modo que la lógica del negocio pueda ser separada del almacenamiento de datos, permitiendo así que la aplicación con el tiempo pueda ser modificada a las condiciones que establezcan las nuevas reglas de negocio.

Durante la etapa de pruebas se genera los diferentes casos de pruebas a partir de los casos de uso. Los casos de prueba consisten de un conjunto de condiciones de ejecución, las entradas de la prueba y los resultados esperados, además de una descripción, los cuales nos ayudaron a determinar si los requisitos del sistema se cumplen de forma satisfactoria.

Por último tenemos el capítulo 4 y la fase de transición donde se define la implantación del sistema y la evaluación de resultados. En la implantación se especifica el ambiente real de pruebas, para la cual se tiene la especificación de Perfiles de Usuario. Se define los usuarios reales del Sistema con el respectivo cargo en el laboratorio y correspondiente perfil en el sistema. Por último se especifica el software y el hardware recomendado para el servidor y para el terminal.

Una vez que el sistema cuenta con la funcionalidad requerida y el ambiente de prueba es preparado, se procede a la evaluación real del sistema considerando el formato de prueba y los rangos de evaluación. Después de probado el sistema se evalúan los resultados de las pruebas realizadas a cada uno de los usuarios usando los casos de prueba correspondientes. Una vez termina la evaluación a todos los usuarios se concluye que el sistema obtiene un calificación satisfactoria, lo que significa que el sistema cumple con la funcionalidad requerida, sin embargo el usuario requiere de aprendizaje para utilizar el sistema de manera correcta.

PRESENTACIÓN

El presente documento constituye la documentación primaria del trabajo realizado para cumplir con el objetivo de desarrollar el Sistema de Gestión de análisis físico-químico y microbiológico de aguas, suelos y lodos para el Centro de Investigación de Control Ambiental (CICAM) de la EPN, el mismo que permite la administración centralizada de laboratorios y el control de los procesos de forma óptima.

Para la elaboración del sistema mencionado; se uso en lo posible herramientas de software libre y se empleo la metodología RUP, que permite desarrollar el sistema de forma iterativa incremental de acuerdo a los flujos de trabajo que maneja; con las actividades y tareas correspondientes en cada una de las etapas de desarrollos de un sistema de software.

Esperamos que este documento sea del agrado del lector y permita comprender claramente los pasos realizados para alcanzar el fin propuesto.

CAPÍTULO 1: PLANTEAMIENTO DEL PROBLEMA

1.1. DESCRIPCIÓN DEL CICAM

El Centro de Investigaciones y Control Ambiental (CICAM) de la Escuela Politécnica Nacional tiene como misión brindar servicios de análisis, soporte técnico, y capacitación a la investigación e industria.

1.1.1. ORGANIGRAMA DEL CICAM

La descripción de la organización del CICAM se especifica en el siguiente organigrama donde se encuentran definidos los respectivos puestos.

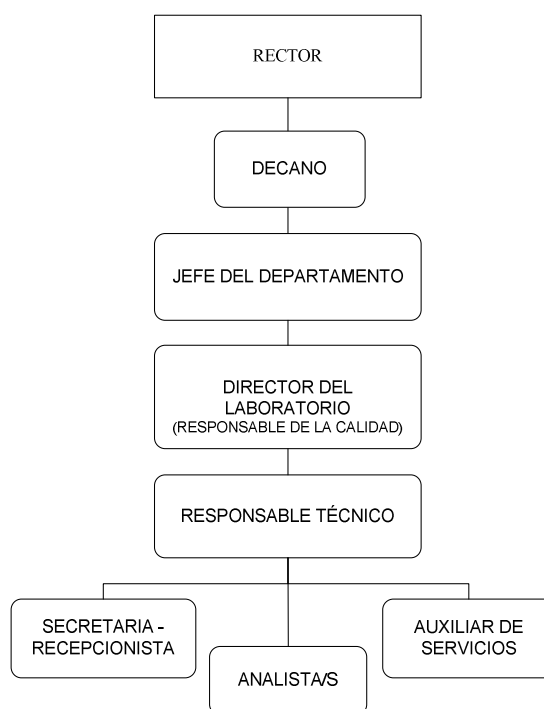


Figura1. 1 Organigrama del CICAM [4]

1.1.2. DESCRIPCIÓN GENERAL DE LOS CARGOS

- *Rector de la Escuela Politécnica Nacional:*
El Rector es la máxima autoridad ejecutiva de la Escuela Politécnica Nacional y su Representante Legal.
- *Decano (D)*
El Decano es la máxima autoridad ejecutiva de la Facultad y su representante.

- *Jefe de Departamento (JD)*
Es la máxima autoridad del DICA-EPN. Sus funciones específicas están relacionadas con la gestión de recursos administrativos del CICAM.
- *Director del Laboratorio (DL)*
Es la máxima autoridad ejecutiva del CICAM y desempeña las funciones de Responsable de la Calidad.
- *Responsable Técnico (RT)*
Es la máxima autoridad técnica del CICAM. Como tal, asume la responsabilidad de que todas las actividades de ensayo sean realizadas de conformidad a lo establecido en las normas NTE-INEN ISO/IEC 17025.
- *ANALISTA /s (AN /s)*
Son los responsables de la ejecución de los ensayos.
- *Auxiliar de Servicio(AS)*
Es la persona responsable de la limpieza y orden del CICAM.
- *Secretaria – Recepcionista*
Es la persona encargada de la Secretaría y Recepción del CICAM. ^[4]

1.2. DESCRIPCIÓN DEL PROBLEMA

El Centro de Investigación y Control ambiental en vista de la necesidad de automatizar los servicios que presta la Unidad de Laboratorio, requiere el desarrollo e implantación de un Sistema de Gestión de análisis físico-químico y microbiológico de aguas, suelos y lodos para el Centro de Investigación de Control Ambiental (CICAM) de la EPN.

Al mismo tiempo La Escuela Politécnica Nacional por medio de la Unidad de Gestión de Información (UGI), con el objetivo de unificar los procesos utilizados en las diferentes unidades de laboratorios existentes y avanzar tecnológicamente; diseñan un plan de desarrollo que permita la administración centralizada de laboratorios y el control de los procesos de forma óptima, al cual denominan SISTEMA DE ADMINISTRACIÓN CENTRALIZADA DE LABORATORIOS EN LA ESCUELA POLITÉCNICA NACIONAL.

[4] CENTRO DE INVESTIGACIÓN Y CONTROL AMBIENTAL, Manual de Calidad del CICAM, fecha: 28-08-2009

Para el desarrollo del sistema mencionado, se decidió tomar como base los requerimientos generales del CICAM. Cabe señalar que el presente proyecto será lanzado como una primera versión, dejando abierta la posibilidad de lanzar nuevas versiones hacia la adaptación de los requerimientos del resto de laboratorios con los que cuenta la institución. [44]

Dado que nuestro Proyecto de Tesis es para el CICAM, y el sistema planteado por la UGI basará sus requerimientos en el CICAM, nos unimos como parte del equipo de trabajo, encargándonos de módulos que involucran procesos específicos.

Los módulos que se establecieron para la conformación del sistema son:

1. Administración de Usuarios
2. Módulo Administrativo
3. Administración de Servicios
4. Administración de Análisis
5. Facturación
6. Control de Inventarios
7. Información Gerencial

De acuerdo al alcance del plan de tesis, nosotras desarrollares los siguientes módulos:

1. Módulo Administrativo
2. Administración de Servicios
3. Administración de Análisis
4. Facturación

UNIDAD DE GESTIÓN DE INFORMACIÓN, Proyecto para la implementación de un sistema de administración centralizada de laboratorios en la Escuela Politécnica Nacional, fecha: 28-09-2009

[44] UNIDAD DE GESTIÓN DE INFORMACIÓN, Proyecto para la implementación de un sistema de administración centralizada de laboratorios en la Escuela Politécnica Nacional

1.3. SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO

La Unidad de Gestión Informática basándose en su experiencia en el desarrollo de sistemas informáticos decidió escoger la metodología de desarrollo RUP, a la cual asentimos, dado que es la que mejor se adapta a este tipo de desarrollo y a su vez, esta permite realizar un número de iteraciones de mejoramiento continuo del producto, intentando reproducir el ciclo de vida en cascada a menor escala.
[21] [36]

1.3.1. JUSTIFICACIÓN DE SELECCIÓN DE LA METODOLOGÍA RUP

RUP es un proceso de ingeniería de software, que provee un acercamiento disciplinario para asignar tareas y responsabilidades dentro de una organización de desarrollo. Lo que asegura un producto de software de alta calidad, al cubrir las necesidades de sus usuarios finales con un calendario y costos predecibles.
[35]

RUP es un proceso de ingeniería de software orientado a objetos que captura muchas de las mejores prácticas durante el desarrollo de un software moderno, por lo que ha llegado a constituir la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Las 6 mejores prácticas de desarrollo que aplica RUP tenemos

- Desarrollo de software en forma iterativa
- Gestión de requerimientos
- Uso de arquitecturas basadas en componentes
- Modelización visual del software
- Verificación de calidad del software
- Control de cambios

Es así que se puede decir que RUP es una metodología de desarrollo de software que intenta integrar todos los aspectos a tomar en cuenta durante el ciclo de vida

[21] JACOBSON, I., BOOCH, G., RUMBAUGH, J. (2000). El proceso unificado de desarrollo de software. Addison-Wesley, Madrid

[36] PRESSMAN, R. (2001). Ingeniería del software. Un enfoque práctico", 5a ed. McGraw-Hill.

[35] PHILIPPE KRUCHTEN, THE RATIONAL UNIFIED PROCESSES AN INTRODUCTION, THIRD EDITION, and EDITORIAL: Addison-Wesley

de software, adaptando el proceso a las características propias de un proyecto u organización. [5] [58]

Dentro de las características de RUP es importante resaltar que se centra en la producción y mantenimiento de modelos del sistema usando el lenguaje unificado de modelado UML de forma más efectiva, además divide el proceso de desarrollo en ciclos y al final de cada uno de estos se obtiene un producto. Cada ciclo se divide en cuatro fases:

- Inicio
- Elaboración
- Construcción
- Transición

1.3.1.1.Fase De Inicio

Durante la fase de inicio las iteraciones se centran con mayor énfasis en las actividades de modelamiento de la empresa y en sus requerimientos. Se definen los principales casos de uso y el alcance del proyecto.

1.3.1.2.Fase De Elaboración

Durante esta fase, las iteraciones se centran al desarrollo del diseño de la base, se completan los casos de uso y se eliminan los riesgos y una parte de implementación es orientada a la base de la construcción.

1.3.1.3.Fase De Construcción

Durante esta fase, se lleva a cabo la construcción del producto por medio de una serie de iteraciones, de las cuales se seleccionan algunos Casos de Uso, se redefine su análisis y diseño y se procede a su implantación y pruebas.

En esta fase se realiza una pequeña cascada para cada ciclo, se realizan tantas iteraciones hasta que se termine la nueva implementación del producto.

[5] CIBERTEC, La ingeniería de Software y RUP

[58] WIKIPEDIA, Proceso Unificado de Rational

1.3.1.4.Fase De Transición

Durante esta fase, se busca garantizar que se tiene un producto preparado para ser entregado al usuario.

1.3.1.5.Documentación

El proceso de desarrollo será debidamente documentado a lo largo de las diferentes fases de la metodología RUP. [9]

1.4. JUSTIFICACIÓN DE HERRAMIENTAS DE DESARROLLO

En vista de que la Escuela Politécnica Nacional es una Institución del Estado y con el afán de promover software libre, ha decidido desarrollar el sistema utilizando algunas herramientas de software libre tales como; J2EE como estándar de desarrollo y Java Eclipse como entorno de desarrollo, Postgres como motor de base de datos y Jboss como servidor de aplicaciones J2EE. Mientras que para el modelamiento se usará la herramienta de licenciamiento Power Designer y Rational Rose.

1.4.1. SOFTWARE LIBRE

Es la denominación utilizada para referirse al software que respeta la libertad de los usuarios sobre su producto adquirido, de manera que el software pueda ser ejecutado, copiado, distribuido, estudiado, cambiado y mejorado.

De manera más precisa, software libre hace referencia a la posibilidad que tienen los usuarios de programas de software, de cuatro libertades esenciales:

- Libertad 0: la libertad de usar el programa, con cualquier propósito.
- Libertad 1: la libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades.
- Libertad 2: la libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.
- Libertad 3: la libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

[9] COMUNIDAD DE PROFESIONALES DE IT, Que es RUP

Las libertades 1 y 3 requieren acceso al código fuente porque estudiar y modificar software sin su código fuente es muy poco viable.

Es así que un programa es considerado como software libre si los usuarios cuentan con todas las libertades descritas anteriormente, bajo una única restricción, el copyleft, que establece que cualquiera que redistribuya el software, con o sin cambios, debe dar las mismas libertades de antes. De este modo se permitirá el acceso al código fuente, imprescindible para ejercer las libertades 1 y 3.

Ventajas del Software Libre

- *Económico*

El bajo o nulo costo de los productos libres permiten proporcionar servicios y ampliar las infraestructuras de las empresas informáticas, sin que se vean mermados sus intentos de crecimiento por no poder hacer frente al pago de grandes cantidades en licencias.

- *Libertad de uso y redistribución*

Las licencias de software libre disponibles, permiten la instalación del software tantas veces y en tantas máquinas como el usuario lo requiera, logrando al mismo tiempo reducir significativamente los costos, al hacer uso de una de las libertades que promulga el software libre.

- *Independencia tecnológica*

Con software libre, el acceso al código fuente es posible, por lo cual se permite el desarrollo de nuevos productos sin la necesidad de desarrollar todo el proceso partiendo de cero. El secretismo tecnológico es uno de los grandes frenos y desequilibrios existentes para el desarrollo dentro del modelo de propiedad intelectual.

- *Independencia del proveedor:*

Al disponer del código fuente, cualquier persona puede continuar ofreciendo soporte, desarrollo u otro tipo de servicios para el software, esto no ocurre con el

software propietario, pues con este nos vemos sometidos a las condiciones del mercado de nuestro proveedor, es decir que si se va del mercado porque no le conviene y discontinúa el soporte, nosotros no podemos contratar a otra persona.

Desventajas de Software Libre

- *Dificultad en el intercambio de archivos*

Esto se da mayormente en los documentos de texto (generalmente creados con Microsoft Word), ya que si los queremos abrir con un Software Libre (p/ ej. Open Office o LaTeX) nos da error o se pierden datos. Pero está claro que si Microsoft Word creara sus documentos con un formato abierto (o público) esto no sucedería.

- *El usuario debe tener nociones de programación*

Es importante que se tenga nociones de programación puesto que la administración del sistema recae mucho en la automatización de tareas y esto se logra utilizando, en muchas ocasiones, lenguajes de guiones (perl, python, shell, etc).

- *La diversidad de distribuciones*

La gran cantidad de distribuciones, métodos de empaquetamiento, licencias de uso, herramientas creadas con un mismo fin, entre otras, pueden crear confusión en cierto número de personas. [10] [15] [47] [60]

1.4.2. J2EE

J2EE es la plataforma creada por Sun Microsystem en el año 1997 ofrece una perspectivas de desarrollo para empresas que quieran basar su arquitectura en productos de software libre.

Dentro de las principales ventajas que ofrece J2EE se tiene:

- Soporte de múltiples sistemas operativos
- Cuenta con un organismo de control (JCP)

[10] CULEBRO JUÁREZ, GÓMEZ HERRERA, TORRES SÁNCHEZ, Software Libre vs Software Propietario

[15] GNU OPERATING SYSTEM, La Definición de Software Libre

[47] WAPEDIA MOBI, Software Libre

[60] WIKIPEDIA, Software Libre

- Competitivo en el mercado por su rendimiento y costo.
- Maduro con alrededor de trece años en el mercado y proyectos importantes detrás de esta plataforma.
- Ofrece la posibilidad de crear arquitecturas basadas completamente en software libre.
- Generación de componentes reutilizables dentro del ámbito de los distintos sistemas del proyecto, reduciendo así los tiempos de desarrollo proyectados para funcionalidades similares.

J2EE no es un producto concreto que puede ser descargado e instalado, por lo que es calificado de manera informal como un estándar debido a que los proveedores deben cumplir con ciertos requisitos de conformidad para poder declarar que sus productos son acordes a Java2EE. [7] [42]

Cuando una persona o entidad cree que es necesaria la presencia de una determinada tecnología dentro de las plataformas basadas en Java, lo que hace es crear un JSR (Java Specification Request) y lo presenta para su aprobación. Un JSR no es más que un documento donde se justifica porque es necesaria la integración de dicha tecnología.

Con la intención de posibilitar que todo el que desee participe en la evolución de java y de las plataformas que se basan en esta, se crea el JSP (Java Community Process), organismo formado por alrededor de 500 empresas, asociaciones y particulares con el objetivo de permite a las partes interesadas involucrarse en la definición de futuras versiones y características de la plataforma java. [52] [53]

La plataforma J2EE provee un modelo de aplicación distribuido, a servidores y clientes, dándoles soporte para desarrollar aplicaciones n-capas. La Arquitectura del Modelo de Aplicación puede dividir el software en capas lógicas para ofrecer escalabilidad y manejabilidad, estas divisiones lógicas son:

- a) Capa de presentación o capa del cliente: contiene componentes para manejar las interfaces de usuario (HTML, Applet, aplicaciones Java, etc).

[7] COMPUTO ACADÉMICO UNAM, J2EE. Una plataforma para el cómputo empresarial

[42] RUEDA CHACÓN JULIO CÉSAR, Aplicación de la Metodología RUP para el desarrollo rápido de aplicaciones basado en el estándar j2ee

[52] WIKIPEDIA, Java Community Process

[53] WIKIPEDIA, Java EE

- b) Capa de lógica de negocio o Capa intermedia: contiene subcapas que solucionan problemas de negocios (Contenedor Web y el contenedor EJB).
- c) Capa Sistema de Información Empresarial o capa de datos: (contiene 1 ó más bases de datos)

Dentro de la arquitectura J2EE existe un concepto clave denominado contenedor, que dicho de forma genérica no es más que un entorno de ejecución estandarizado que ofrece ciertos servicios por medio de componentes. Los componentes externos al contenedor tienen una forma estándar de acceder a los servicios de dicho contenedor, con independencia del fabricante.

Dentro de los tipos de contenedores que maneja J2EE tenemos:

Contenedor Web: también denominado contenedor Servlet/JSP, maneja la ejecución de los servlets y páginas JSP. Estos componentes se ejecutan sobre un servidor Enterprise Edition.

Contenedor Enterprise JavaBeans: que gestiona la ejecución de los *EJB*. Esta ejecución requiere de un Server Enterprise Edition.

Los contenedores incluyen descriptores de despliegue, que son archivos XML que nos sirven para configurar el entorno de ejecución: rutas de acceso a aplicaciones, control de transacciones, parámetros de inicialización, etc. [6]

La plataforma JEE incluye APIs para el acceso a sistemas empresariales:

- JDBC es el API para acceso a GBDR desde Java.
- Java Transaction API (JTA) es el API para manejo de transacciones a través de sistemas heterogéneos.
- Java Naming and Directory Interface (JNDI) es el API para acceso a servicios de nombres y directorios.
- Java Message Service (JMS) es el API para el envío y recepción de mensajes por medio de sistemas de mensajería empresarial como IBM MQ Series.
- JavaMail es el API para envío y recepción de email.

[6] CODE JAVA, J2EE 5.0 – Java 2 Enterprise Edition 5.0

- Java IDL es el API para llamar a servicios CORBA.

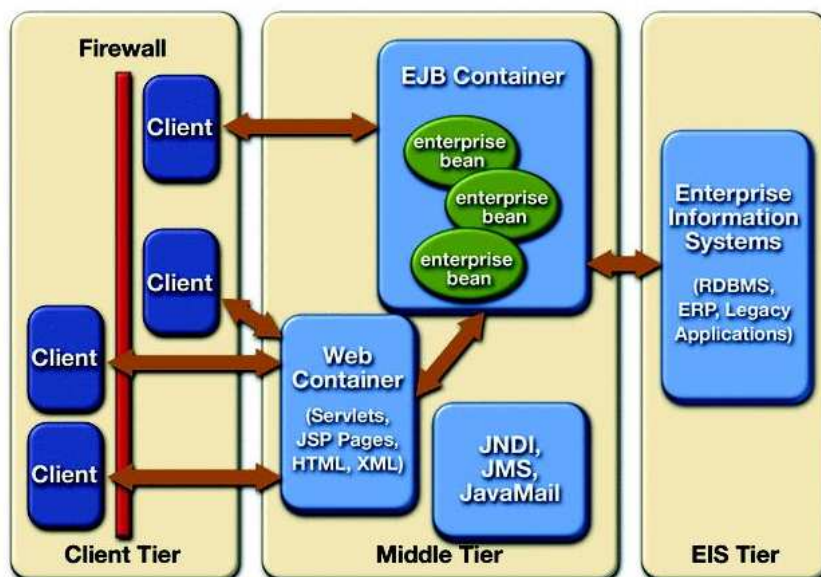


Figura1. 2 Plataforma J2EE [37]

1.4.3. ECLIPSE

Eclipse es un entorno de desarrollo integrado (del inglés IDE) de código abierto multiplataforma basado en Java. Es un desarrollo de IBM, como el sucesor de su familia de herramientas para VisualAge, más tarde paso a ser desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (plug-in), para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido".

Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent Azureus.

En el entorno de desarrollo Eclipse todo archivo se almacena dentro de un proyecto. Esto quiere decir que todo documento, carpeta, archivo de código fuente (.java) y código compilado (.class) está contenido dentro de un proyecto. Así pues, el primer paso antes de usar Eclipse para programar en Java es comprender la estructura de proyectos de Eclipse.

Eclipse contiene una serie de perspectivas. Cada perspectiva proporciona una serie de funcionalidades para el desarrollo de un tipo específico de tarea. Por ejemplo la perspectiva Java combina un conjunto de vistas que permiten ver información útil cuando se está escribiendo código fuente, mientras que la perspectiva de depuración contiene vistas que muestran información útil para la depuración de los programas Java. [8] [12] [51]

Hay varios tipos de proyectos de Eclipse que pueden ser creados:

- Para crear o editar programas Java, un "Java Project" debería ser creado. Notar que dentro de un proyecto Java se puede almacenar toda la información relacionada con el proyecto (la cual no tiene por qué reducirse rigurosamente a código fuente, sino que también puede contener documentación y otros archivos relacionados).
- Los "Simple Project" sólo deberían crearse para almacenar documentos y otros archivos, pero no código Java que se desee compilar. Por ello, siempre que se creen archivos ".java" sería recomendable crear un "Java Project".
- Los "Plug-in Development Project" se usan para añadir nuevos módulos y funciones al entorno Eclipse. Estos proyectos sólo son necesarios si se es un desarrollador de Eclipse.
- Los proyectos de "EMF" se utilizan para crear modelos de análisis y diseño.

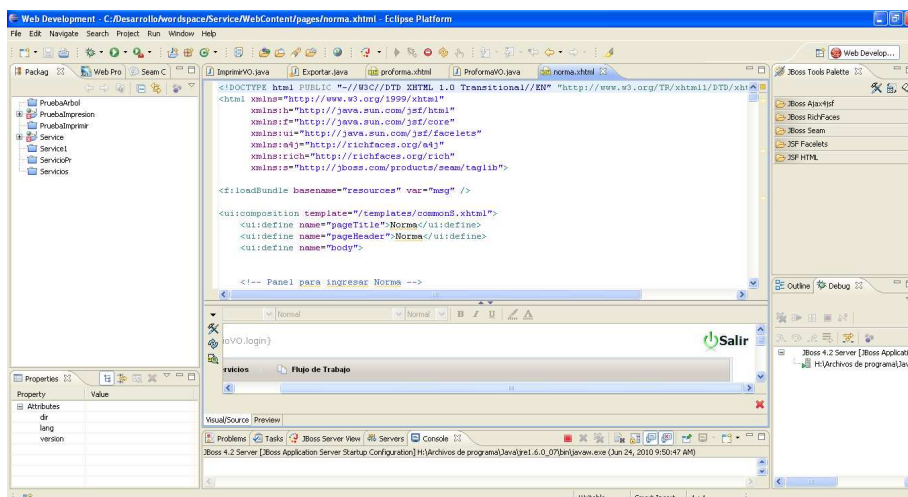


Figura1. 3 Entorno Eclipse

1.4.4. POSTGRESQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, que ha sido desarrollado de varias formas desde la década de 1980.

Esto quiere decir que el código fuente del programa está disponible a cualquier persona libre de cargos directos, permitiendo a cualquiera colaborar con el desarrollo del proyecto o modificar el sistema para ajustarlo a sus necesidades. PostgreSQL está bajo licencia BSD.

Un sistema de base de datos relacionales es un sistema que permite la manipulación de acuerdo con las reglas del álgebra relacional. Los datos se almacenan en tablas de columnas y renglones. Con el uso de llaves, esas tablas se pueden relacionar unas con otras.

Dentro de las principales características con las que cuenta Postgresql tenemos:

- *Ahorros considerables en costos de operación*

PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.

- *Multiplataforma*

PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y ahora en versión nativa para Windows.2

- *Transacciones*

Permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.

- *Integridad referencial*

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos, PK y FK.

- *Bloqueos de tabla y filas*

Postgres ofrece varios modos de bloqueo para controlar el acceso concurrente a los datos en tablas. Algunos de estos modos de bloqueo los adquiere PostgreSQL automáticamente antes de la ejecución de una declaración, mientras que otros son proporcionados para ser usados por las aplicaciones.

- *Constraints y triggers*

Tienen la función de mantener la integridad y consistencia en la BD.

Ejecución de acciones antes o después de un evento de BD. [16] [26]

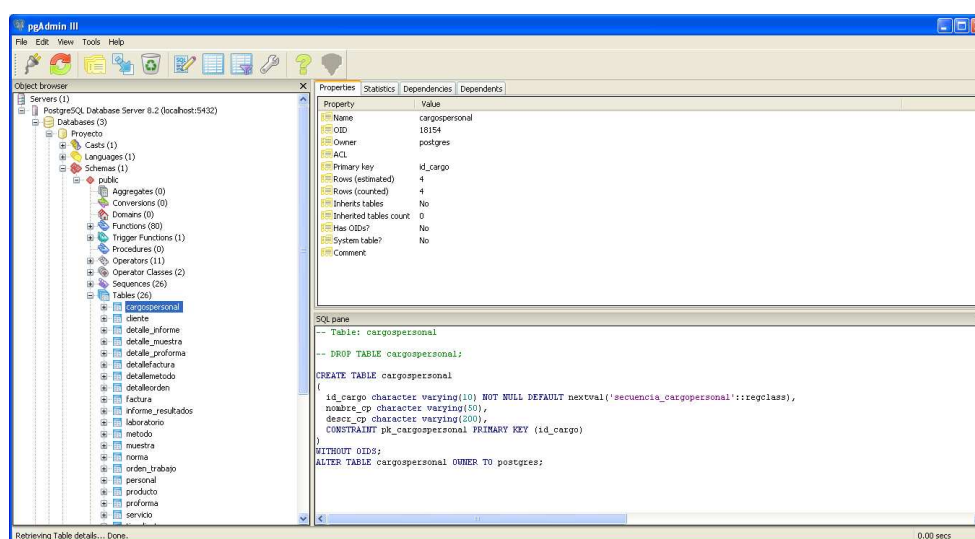


Figura1. 4 PostgreSQL

[16] HUMBERTO ESPINOZA, PostgreSQL

[26] MARC GIBERT GINESTÀ, OSCAR PÉREZ MORA, Base de datos en PostgreSQL

1.4.5. SERVIDOR DE APLICACIONES JBOSS

JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte.

JBoss surge de una implementación de un Contenedor EJB, mediante el cual es posible llevar a cabo un desarrollo con EJBs. Los Contenedores EJB, generalmente no son distribuidos como producto individual y por esta razón se le pudiera considerar a JBoss como un producto diferente más no único.

La gran gama de productos J2EE que han sido comercializados como Servidores de Aplicaciones Java, se encuentra compuestos por dos partes: un *Servlet Engine* y un *EJB Engine*, dentro del *Servlet Engine* se ejecutan exclusivamente las clásicas aplicaciones de Servidor (*JSP* y *Servlets*), mientras el *EJB Engine* (Container) es reservado para aplicaciones desarrolladas alrededor de EJBs.

Casi todos los Servidores de Aplicaciones en el mercado son conocidos como Fully J2EE Compliant, término que implica que se cumplen todas las especificaciones J2EE definidas por Sun y es aquí donde es notable la diferencia con JBoss.

Cuando se usa un Servidor de Aplicaciones que implementan esta característica, no existe una clara distinción entre el Contenedor Web y Contenedor EJB, es decir que es posible ejecutar tanto JSP/Servlets así como EJBs, sin embargo, el ambiente se encuentra altamente integrado para que la comunicación entre JSP/Servlets y EJBs sea transparente, al menos para el programador final.

Mientras que JBoss es únicamente un Contenedor EJB y es por esto que generalmente se utiliza en conjunción con un Contenedor Web. Jboss incluye por defecto Tomcat proporcionado como el Contenedor Web, aunque lo anterior no restringe a JBoss para operar con otro contenedor Web. [34]

[34] OSMOSIS LATINA, Guía de Jboss

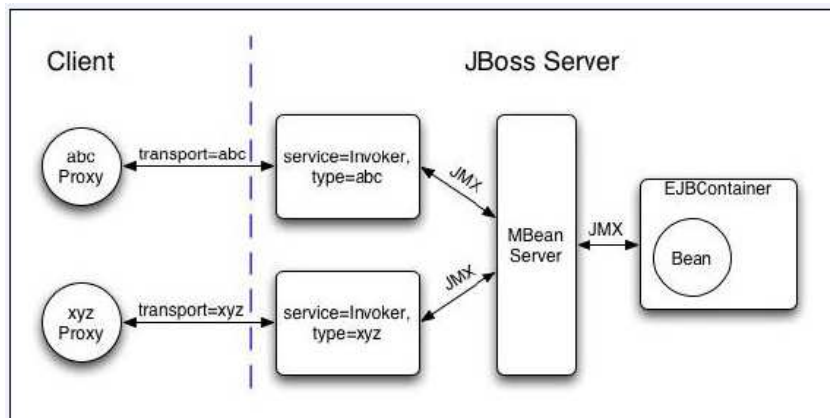


Figura1. 5 Servidor de Aplicaciones JBoss [1]

Dentro de las principales características con las que cuenta JBoss se tiene:

- *Mayor flexibilidad*

JBoss es la plataforma de desarrollo líder de Java J2EE. Por este motivo resultará fácil integrar las aplicaciones Java, ya sean de terceros o internas. Así podrá Vd. beneficiarse del innovador dinamismo de la comunidad Open Source y podrá adaptar fácilmente nuevas aplicaciones a sus necesidades específicas.

- *Costes más bajos*

Ahórrese el pago de licencias y reduzca drásticamente sus costes operativos y de mantenimiento mediante una suscripción integral y asequible.

- *Más recursos y más bajos costes de personal*

Reduzca sus costes de mantenimiento y soporte, abrevie los plazos de implementación y permita a su personal que pueda dedicarse a proyectos informáticos nuevos y estratégicos.

- *Un mejor soporte técnico*

Los programadores que crearon JBoss gozan del prestigio de estar entre los mejores de la comunidad Open Source y de suministrar un soporte técnico industrial líder.

- *Independencia de proveedores*

Elija la solución de JBoss que mejor se adapta a sus necesidades sin tener que vincularse a ningún proveedor. [22] [39]

- *Confiable a nivel de empresa.*
 - Orientado a arquitectura de servicios.
 - Flexibilidad consistente.
 - Servicios del middleware para cualquier objeto de Java
 - Ayuda profesional 24x7 de la fuente.
 - Soporte completo para JMX.

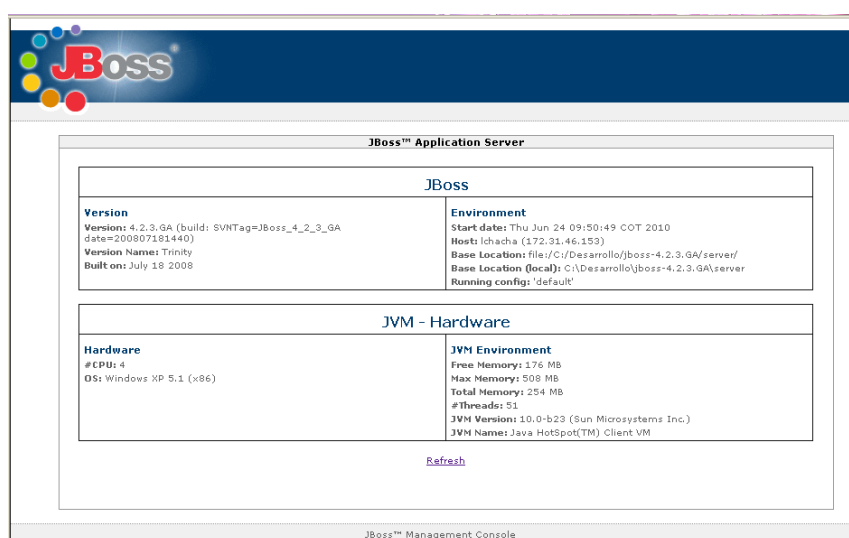


Figura1. 6 JBoss

1.4.6. POWER DESIGNER

Es una herramienta de modelamiento que combina de forma única numerosas técnicas estándar de modelado (UML, modelado de procesos de negocio y el modelado de datos líder del mercado) junto con las herramientas de desarrollo más conocidas del mercado como aon .NET, Sybase WorkSpace, Sybase Powerbuilder, Java y Eclipse, con el fin de ofrecer soluciones de análisis empresarial y diseño de bases de datos formales al ciclo de vida tradicional para el desarrollo de software.

Power Designer corre sobre Microsoft Windows como aplicación nativa, y sobre Eclipse a través de un plugin. Además funciona con más de 60 sistemas de administración de bases de datos relacionales. Esta herramienta soporta un

[22] JAVIER TUREGANO MOLINA, Administrando JBoss

[39] RED HAT, JBOSS, Plataforma de Aplicaciones JBoss Enterprise

arquitectura basada en modelos para diseño de software y usa el formato de archivo .pdm.

PowerDesigner combina varias técnicas estándar de modelamiento con herramientas líder de desarrollo, como .NET, Sybase WorkSpace, Sybase Powerbuilder, Java y Eclipse, para darle a las empresas soluciones de análisis de negocio y de diseño formal de base de datos. [31] [57]

Dentro de las principales características con las que cuenta Power Designer se tiene:

- Alinea la tecnología de información con los procesos del negocio para mejorar la productividad.
- Brinda soporte abierto y compatibilidad con un gran número de ambientes heterogéneos de todas clases.
- Es altamente personalizable, permitiendo acogerse y cumplir con los estándares y regulaciones existentes.
- Facilita la arquitectura empresarial, documentando los sistemas existentes.
- Aumenta la agilidad del negocio por medio de la tecnología “Link & Sync” y análisis de impacto.

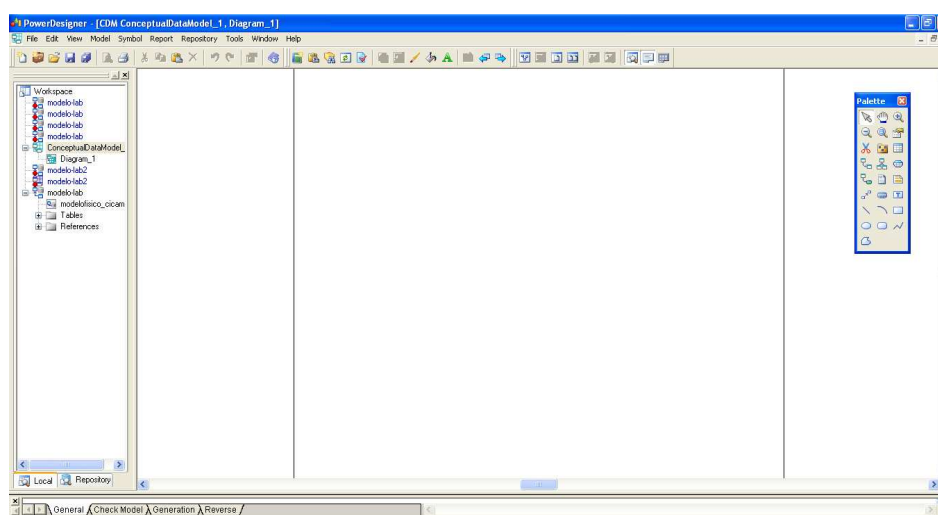


Figura1. 7 Power Designer

[31] MTBASE SYBASE DE COLOMBIA, Power Designer

[57] WIKIPEDIA, Power Designer

1.4.7. RATIONAL ROSE

Rational Rose es una de las más poderosas herramientas de modelo visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de construirlo, cubriendo todo el ciclo de vida de un proyecto

Con Rational rose se puede crear:

- Diagramas de secuencia
- Diagramas de colaboración
- Diagramas de estado
- Diagramas de actividad
- Diagramas de implementación:
- Diagramas de componente
- Diagramas de despliegue

Crear uno a uno por separado es relativamente fácil, el problema está en que todos estos diagramas, deben estar relacionados de alguna manera, la única relación que puede hacer es al generar el diagrama de clase lógico, el cual se convierte en modelo físico (diagrama de clase) y este último convertirlo en una base de datos. [17] [38] [41] [59]

Dentro de las principales características con las que cuenta Rational Rose se tiene:

- Permite a los diseñadores tomar ventaja de desarrollo iterativo porque en la aplicación se pueden crear nuevas etapas con la salida de una iteración en la entrada a la siguiente.
- Es extensible porque se puede descargar complementos y aplicaciones de socios de las partes. Es compatible con COM/DCOM (ActiveX), JavaBeans, y las normas de componentes CORBA.
- Soporte a modelos de análisis

[17] IBM, Rational Software

[38] RATIONAL SOFTWARE CORPORATION, RATIONAL ROSE

[41] RUBÉN GONZÁLEZ BLANCO, SERGIO PÉREZ TOBALINA, UNIVERSIDAD POLITÉCNICA DE CATALUNYA, Introducción a Rational Rose

[59] WIKIPEDIA, Rational Rose

CAPITULO2: ANÁLISIS Y DISEÑO

De acuerdo con la metodología RUP y las fases que se establece en la misma; previamente a la ejecución del proyecto de desarrollo, y específicamente a la realización de la fase de inicio y elaboración, es necesario definir las actividades a realizar en cada una de estas etapas.

En la fase de inicio nos centraremos en las actividades específicas de esta etapa, como son la administración del proyecto, el modelamiento de la empresa y de sus requisitos definiendo el alcance del proyecto y diseñando los principales casos de uso.

En la fase de elaboración se desarrollan las etapas de análisis y diseño; durante la etapa de análisis se elaboran los diagramas de actividad y secuencia, mientras que durante la etapa de diseño se elabora el diagrama de clases y el modelo de datos.

2.1. ESPECIFICACIÓN DE REQUERIMIENTOS

La presente sección será dividida en dos partes: Administración del Proyecto y Requisitos dentro de las cuales se encuentran los respectivos artefactos que nos ayudaran a determinar de forma precisa los requisitos del sistema.

2.1.1. ADMINISTRACIÓN DEL PROYECTO

La administración de proyectos se enfoca en coordinar todas las etapas del proyecto de desarrollo de software cuyos objetivos se establecen explícitamente en términos de tiempo, costos y parámetros de realización.

Por lo mencionado anteriormente, previo a la ejecución del proyecto de desarrollo, es necesario establecer el Plan de desarrollo de Software, de tal manera que se pueda mantener el control del proyecto durante cada una de sus etapas.

Técnicamente, el Plan de Desarrollo de Software es un compendio de los distintos planes que son necesarios para ejecutar un proceso de desarrollo, y que a su vez proporciona una visión global del enfoque de desarrollo propuesto para un proyecto.

Es así, que se realiza el plan de desarrollo para el “Sistema de Administración Centralizada de Laboratorios” enfocada en los módulos que tiene a cargo el

presente proyecto. Este Plan de Desarrollo del Software, es elaborado previo al desarrollo de la fase inicial del proceso RUP.

Una vez establecido el plan de proyecto y durante esta fase de inicio, nos centraremos en las actividades específicas, como son el Modelamiento del Negocio y El Análisis de Riesgos. [2] [25]

2.1.1.1. Plan de Desarrollo de Software

2.1.1.1.1. Introducción

El Plan de Desarrollo del Software es una versión preliminar desarrollada como parte del proyecto de titulación de los módulos: Administrativo, Administración de Laboratorio, Administración de Servicios, y Facturación; pertenecientes al Sistema de Administración Centralizada de laboratorios planteado para la Escuela Politécnica Nacional. Este documento proporciona una visión global del enfoque de desarrollo propuesto.

El desarrollo del sistema centralizado de laboratorios presenta un enfoque con un lineamiento a los procesos de RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes del proyecto, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de RUP.

2.1.1.1.2. Propósito del Plan de Desarrollo de Software

El propósito del Plan de Desarrollo de Software es obtener la información necesaria, de modo que se pueda mantener el control del proyecto durante todo el proceso de desarrollo, al mismo tiempo que nos permita tomar decisiones favorables si llegará a presentarse algún contratiempo.

Los usuarios del Plan de Desarrollo del Software son:

- El jefe del proyecto, quién lo utilizará para organizar la agenda, conocer las necesidades de recursos, y para realizar su seguimiento.
- Los miembros del equipo de desarrollo, quienes lo usarán para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.

[2] ALEXANDER AGUILAR VARGAS, Administración de Proyectos de Software

[25] JOSÉ DE JESÚS RODRIGUEZ VELEZ, Administración de proyectos de desarrollo de sistemas de información

2.1.1.1.3. Alcance del Plan de Desarrollo de Software

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del “Sistema de Administración Centralizada de laboratorios en la Escuela Politécnica Nacional”. Este plan ha sido desarrollado basándonos en la captura de requisitos por medio del stakeholder jefe técnico de laboratorio del Centro de Investigaciones y Control Ambiental (CICAM), de modo que esto nos permitirá hacer una estimación aproximada del plan de desarrollo de software.

Una vez comenzado el proceso de desarrollo y durante la fase de Inicio se generará el artefacto “Especificación de Requerimientos de Software” que definen las características del producto a desarrollar, lo cual constituye la base para la planificación de las iteraciones. El detalle de las iteraciones individuales que se realizarán a lo largo del desarrollo, se describe en los planes de cada iteración, documentos que se presentan en forma separada.

El artefacto “Visión”, se utilizará para refinar este documento. Posteriormente, el progreso del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

2.1.1.1.4. Resumen

Después de esta introducción, el Plan de desarrollo de software está organizado en las siguientes secciones:

Vista General del Proyecto: proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.

Organización del Proyecto: describe la estructura organizacional del equipo de desarrollo.

Gestión del Proceso: define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.

Planes y Guías de aplicación: proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

2.1.1.1.5. Vista general del proyecto

Propósito, Alcance y Objetivos

La información que a continuación se incluye ha sido extraída de las diferentes reuniones que se han celebrado con el stakeholder del laboratorio CICAM al inicio del proyecto, Ing. Carola Fierro, y posteriormente de la información proporcionada por la Unidad de Gestión de la Información de la Escuela Politécnica Nacional.

Inicialmente el proyecto de desarrollo fue planteado para el laboratorio Centro de Investigaciones y Control Ambiental (CICAM), que presta el servicio de análisis de aguas, suelos y lodos a nivel nacional. Sin embargo los procesos que se desarrollan en el mismo actualmente no se encuentran automatizados, lo que resta productividad en la prestación de sus servicios, motivo por el cual, se considero necesario el desarrollo de un nuevo sistema de administración del laboratorio CICAM.

Paralelamente al planteamiento de este proyecto, la Unidad de Gestión de la Información de la Escuela Politécnica Nacional, plantea el desarrollo de un Sistema Completo de Administración de Laboratorios, al cual se deberán someter todos los laboratorios de análisis de la institución; basándose inicialmente en uno de ellos.

Es por esto que se requiere un sistema de administración centralizada de los laboratorios en la EPN, que permita regular y controlar los procesos que se maneja cada uno de los laboratorios, y al mismo tiempo que se logre automatizar dichos procesos.

Por lo cual el presente proyecto debe proveer una propuesta para el desarrollo de los subsistemas implicados en la administración de laboratorios, que se describen en el tema de tesis basándonos en el Centro de Investigación de Control Ambiental.

Los subsistemas que se manejarán en este proyecto se clasifican en cuatro módulos, detallados a continuación:

1. *Módulo Administrativo*
 - Registrar Unidad

- Registrar Tipo de Cliente
 - Registrar Tipo de Servicio
 - Registrar Tipo de Producto
 - Registrar Tipo de Personal
 - Registrar Cargo Personal
2. *Administración de Servicios*
- Registrar Laboratorio
 - Registrar de Cliente
 - Registrar Personal
 - Registrar Producto
 - Registrar de Servicio
 - Registrar de Método
 - Registrar de Norma
 - Registrar de Unidad de Medida
3. *Administración de Análisis*
- Generar Proforma
 - Registrar de Muestra
 - Generar Orden de Trabajo
 - Generar Informe de Resultados
4. *Facturación*
- Generar Factura

Suposiciones y Restricciones

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de las entrevistas con el stakeholder de la empresa son:

Debe contemplarse las implicaciones de los siguientes puntos críticos:

- El subsistema Administrativo debe ser manejado por un único responsable especializado en el área, el cual tendrá la facultad de realizar operaciones en este módulo.
- El subsistema Administración de Servicios, debe contar con seguridad transaccional.
- El subsistema Administración de Laboratorio, la administración del método a aplicar debe dar la posibilidad de ajustarse a los parámetros que establece la legislación vigente.
- El subsistema Facturación debe ser diseñado de manera centralizada, de modo que el departamento financiero, sea el único ente responsable del cobro de los servicios prestados por los diferentes laboratorios de la Escuela Politécnica Nacional.

Entregables del proyecto

Cada uno de los entregables o artefactos que se proponen para ser generados y utilizados por el proyecto, son presentados y descritos de acuerdo a los lineamientos que presenta la metodología RUP desde su perspectiva de artefactos.

Es importante notar que de acuerdo a RUP y todo el proceso iterativo e incremental que propone, todos los artefactos se encuentran sujetos a modificaciones a lo largo del proceso de desarrollo, por lo cual, sólo al término del proceso de desarrollo podríamos tener una versión definitiva y completa de cada uno de ellos.

1) *Plan de Desarrollo del Software*

Es el presente documento.

2) *Modelo de Casos de Uso del Negocio*

Es uno de los artefactos, que tiene por objetivo permitir el modelamiento de casos de uso del negocio y los actores externos, de modo que se describa cómo es usado el Negocio por sus socios y clientes. Este modelo describe al Negocio en términos de un Diagrama de Casos de Uso del Negocio, que corresponden con los generalmente denominados "Procesos", usando estereotipos específicos para este modelo.

3) *Análisis de Riesgos*

Es un artefacto que nos sirve para conocer los riesgos más habituales y algunos métodos para mantener el control

4) *Visión*

El propósito de éste documento es recoger, analizar y definir las necesidades de alto nivel y las características del sistema de centralizado para Laboratorios. El documento se centra en la funcionalidad requerida por los participantes en el proyecto y los usuarios finales.

5) *Modelo de Casos de Uso*

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

6) *Modelo de Análisis*

Es un modelo que describe la realización de cada caso de uso, estableciendo los actores internos, la información que se manipula en términos generales y los flujos de trabajo (workflows) asociados al caso de uso. Para la representación de este modelo se utilizan Diagramas de Secuencia (para mostrar actores externos, internos y las entidades (información) que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones, y Diagramas de Actividad para mostrar los flujos de trabajo.

7) *Modelo de Diseño*

Este modelo establece la realización de los casos de uso en clases, sin incluir aspectos de implementación, de modo que se obtenga un diseño que se oriente hacia el entorno de implementación de acuerdo al avance del proyecto.

8) *Modelo de Datos*

De acuerdo a la permanencia de la información del sistema, este será soportado por una base de datos relacional. Este modelo permite representar la información del mundo real de una manera intuitiva, introduciendo conceptos cotidianos y fáciles de entender por cualquier inexperto, puesto que se basa en la lógica de predicados y la teoría de conjuntos. Para expresar este modelo se utilizará un Diagrama de Clases (para lo cual se utilizará una herramienta UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.)

9) *Prototipos de Interfaces de Usuario*

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados. Asimismo, este artefacto, será desechado en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

10) Modelo de Implementación

Este modelo es un compendio de componentes y subsistemas que los contiene. Los componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es sólo una versión preliminar al final de la fase de Elaboración, posteriormente se tiene bastante refinamiento).

11) Modelo de Despliegue

Este modelo muestra los diagramas de despliegue que representan los nodos del sistema y sus relaciones, en estos nodos se hará el despliegue de los componentes. Generalmente, los nodos son conectados por asociaciones de comunicación como enlaces de red, conexiones TCP/IP, microondas, entre otros.

12) Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

13) Glosario

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología adecuada.

14) Material de Apoyo al Usuario Final

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación, Guías de Mantenimiento.

15) Producto

Los ficheros del producto empaquetados y almacenadas en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva release al final de cada iteración.

Los artefactos 10, 11 y 15 se generarán a partir de la fase de Construcción, con lo cual se han incluido aquí sólo para dar una visión global de todos los artefactos que se generarán en el proceso de desarrollo.

Evolución del Plan de Desarrollo del Software

El Plan de Desarrollo del Software se revisará semanalmente y se refinará antes del comienzo de cada iteración.

2.1.1.1.6. Organización del proyecto

Roles y Responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto.
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.
Programador	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario
Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.
Asesor de Proyecto	Controla y evalúa los artefactos generados en el transcurso del proyecto.

Tabla 2. 1 Roles y Responsabilidades

Participantes en el Proyecto

El personal del proyecto encargado del desarrollo del sistema, y considerando las fases de Inicio, Elaboración, Construcción, estará formado por los siguientes puestos de trabajo y personal asociado:

Jefe de Proyecto: labor a cargo de la Ing. Geovanna Saltos, encargada del departamento de desarrollo de la Unidad de Gestión Informática.

Analista de Sistemas: labor a cargo de las Srtas. Liliana Chacha y Eliana Viscarra, estudiantes de la Escuela Politécnica Nacional.

Analistas – Programadores: labor a cargo de las Srtas. Liliana Chacha y Eliana Viscarra, estudiantes de la Escuela Politécnica Nacional.

Asesor de Proyecto: labor a cargo del Ing. Francisco Villavicencio, profesor de Facultad de Ingeniería en Sistemas de la Escuela Politécnica Nacional.

2.1.1.1.7. Gestión del proceso

Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

Plan de las Fases

El desarrollo del proyecto se llevará a cabo en base a las fases que propone RUP, con una o más iteraciones en cada una de ellas. La siguiente tabla muestra la distribución de tiempos y el número de iteraciones de cada fase.

Flujo	Nro. Iteraciones	Duración
Fase de Inicio	1	1 mes
Fase de Elaboración	1	1.5 meses
Fase de Construcción	1	4 meses
Fase de Transición	1	1.5 mes

Tabla 2. 2 Plan de Fases

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Descripción	Hito
Fase de Inicio	Durante esta fase de inicio las iteraciones se centran con mayor énfasis en las actividades de modelamiento de la empresa y en sus requerimientos. Se definen los principales casos de uso y el alcance del proyecto.
Fase de Elaboración	Durante esta fase, las iteraciones se centran al desarrollo

	del diseño de la base, se completan los casos de uso y se eliminan los riesgos y una parte de implementación es orientada a la base de la construcción.
Fase de Construcción	Durante esta fase, se lleva a cabo la construcción del producto por medio de una serie de iteraciones las cuales se seleccionan algunos Casos de Uso, se redefine su análisis y diseño y se procede a su implantación y pruebas. En esta fase se realiza una pequeña cascada para cada ciclo, se realizan tantas iteraciones hasta que se termine la nueva implementación del producto.
Fase de Transición	En esta fase se asegura una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. Durante esta fase, se busca garantizar que se tiene un producto preparado para ser entregado al usuario.

Tabla 2. 3 Fases de Proyecto

Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo las fases de Inicio, Elaboración, Construcción y Transición. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto. La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina en un momento determinado del desarrollo.

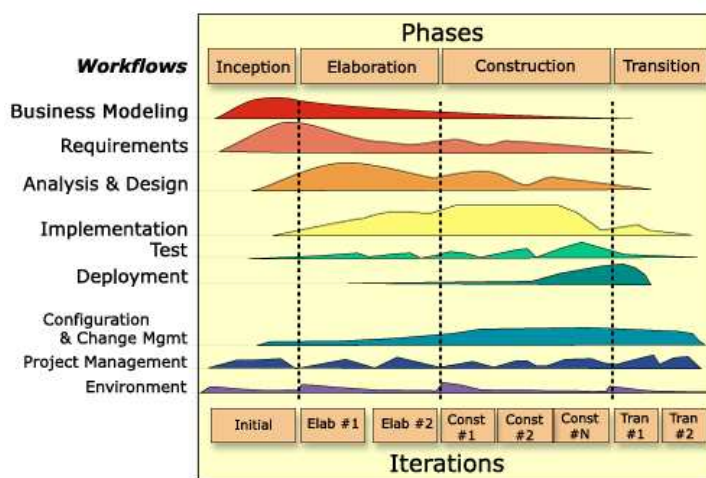


Figura 2. 1 Diagrama de Metodología RUP [5]

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el entregable en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

Disciplinas/Artefactos generados	Inicio	Elaboración	Construcción	Transición	Aprobación
MODELADO DEL NEGOCIO					
Modelo de Casos de Uso del Negocio Modelo de Objetos del Negocio	Semana 1 05/10 – 09/10	Semana 4 02/11 – 06/11			Semana 4 02/11 – 06/11
REQUISITOS					
SRS	Semana 2 12/10 – 16/10	Semana 4 02/11 – 06/11			Semana 4 02/11 – 06/11
Modelo de Casos de Uso	Semana 2 12/10 – 16/10	Semana 4 02/11 – 06/11			Semana 4 02/11 – 06/11
Especificación de Casos de Uso	Semana 2 12/10 – 16/10	Semana 4 02/11 – 06/11			Semana 4 02/11 – 06/11
Especificaciones Adicionales	Semana 2 12/10 – 16/10	Semana 4 02/11 – 06/11			Semana 4 02/11 – 06/11
ANÁLISIS / DISEÑO					
Modelo de Análisis / Diseño	Semana 4 02/11 – 06/11	Semana 5 09/11 – 13/11			Semana 5 09/11 – 13/11
Modelo de Datos	Semana 4 02/11 – 06/11	Semana 5 09/11 – 13/11			Semana 5 09/11 – 13/11
IMPLEMENTACIÓN					
Prototipos de Interfaces de Usuario		Semana 5 09/11 – 13/11	Semana 9 14/12 – 18/12		Semana 9 14/12 – 18/12
Modelo de Implementación			Semana 9 14/12 – 18/12	Semana 20 08/03 – 12/03	Semana 20 08/03 – 12/03
PRUEBAS					
Casos de Pruebas Funcionales			Semana 20 08/03 – 12/03	Semana 24 05/04 – 09/04	Semana 24 05/04 – 09/04
DESPLIEGUE					
Modelo de Despliegue			Semana 25 12/04 – 16/04	Semana 25 12/04 – 16/04	Semana 25 12/04 – 16/04
GESTIÓN DE CAMBIOS Y CONFIGURACIÓN	Durante todo el proyecto	Durante todo el proyecto	Durante todo el proyecto	Durante todo el proyecto	
AMBIENTE	Durante todo el proyecto	Durante todo el proyecto	Durante todo el proyecto	Durante todo el proyecto	

Tabla 2. 4 Calendario del Proyecto

Costos del Proyecto

Los costos del proyecto informático serán cubiertos por la Escuela Politécnica Nacional y por los desarrolladores. A continuación se muestra una tabla detallando el monto de los costos de desarrollo.

Recurso	Costo Unitario	Costo Total
Recursos Humanos	Los desarrolladores a lo largo del proyecto se dedican 12 horas * \$3 USD/h (valor que cobra un becario de la EPN)	\$1 800 USD
Recursos de Hardware	Servidor	\$ 1 500 USD
	Estaciones de Trabajo	\$ 2 000 USD
	Equipo de Pruebas	\$ 1 000 USD
Recursos de Software	No se invierte en Software	\$ 0.00 USD
Recursos Logísticos	Insumos de oficina (papel Bond A4, CD-R, DVD-R, carpetas, impresiones, etc)	\$ 50 USD
	Movilización	\$ 200 USD
	Servicios (energía eléctrica, teléfono, internet)	\$ 400 USD
Total		\$ 5 950 USD

Tabla 2. 5 Costos del Proyecto

Seguimiento y Control del Proyecto

Gestión de Requisitos

Los requisitos del sistema son especificados en el entregable SRS. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el jefe de proyecto y por el tutor de la tesis.

Control de Calidad

Los defectos detectados en las revisiones tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias Para la revisión de

cada entregable y su correspondiente garantía de calidad se utilizarán las guías de revisión y listas de verificación incluidas en RUP.

Gestión de Configuración

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que éstas produzcan, informando y publicando dichos cambios para que sean accesibles a todo los participantes en el proyecto. Al final de cada iteración se establecerá un registro del estado de cada entregable, estableciendo una versión, la cual podrá ser modificada sólo por una Solicitud de Cambio aprobada.

Ver Anexo: Plantilla: Plan de Desarrollo de Software

2.1.1.2. Modelamiento del Negocio

El modelamiento del negocio es el mecanismo por el cual un negocio trata de generar ingresos y beneficios, para identificar con facilidad donde se encuentran sus problemas u oportunidades de crecimiento y mejora. La necesidad de esta etapa surge ante el hecho de que muchos de los productos software que se desarrollan, automatizan algunos o todos los procesos existentes en un negocio, y desde la perspectiva de los sistemas no es posible automatizar procesos que no se encuentren claramente definidos, estudiando las implicaciones de los cambios producidos por la adopción de estos productos.

El Modelado del negocio tiene como objetivos:

1. Entender los problemas actuales en la organización o empresa para identificar los aspectos a mejorar.
2. Comprender la estructura y el dinamismo de la organización o empresa para la cual se va a desarrollar el sistema software.
3. Estudiar el impacto que pueden producir los cambios a nivel organizativo.
4. Asegurar que los clientes, usuarios finales, desarrolladores y otros involucrados tienen una visión común de la organización considerada.
5. Obtener los requisitos del sistema software.
6. Entender como el sistema software encaja en la organización.

Es importante modelar el negocio antes de modelar el sistema, puesto que el sistema a desarrollar manejará la información perteneciente al negocio. Este modelo será utilizado en la organización para ejecutar procesos del negocio cada vez más automatizables, de modo que se podrá adaptar a la organización que lo usará. [29] [54] [61]

2.1.1.2.1. Modelo de Casos de Uso del Negocio

El modelado del negocio se basa en dos diagramas principales el modelo de casos de uso del negocio y el modelo del dominio.

Un modelo de casos de uso del negocio describe los procesos de un negocio, vinculados al campo de acción, y cómo se benefician e interactúan los socios y clientes en estos procesos.

El diagrama de la Figura 2.3 muestra el modelo de casos de uso del negocio, es decir cómo interactúan los elementos externos e internos del negocio.

La empresa interactúa con el cliente como elemento externo, quién solicita la realización de análisis. Como elementos internos se tiene:

- Al administrador quién se encarga del registro de unidad, laboratorio, tipo de cliente, tipo de servicio y productos.
- La recepcionista quien registra a los clientes, genera proformas y órdenes de trabajo.
- El analista el cual recibe orden de trabajo, elige método a aplicar, verifica la receta, realiza análisis, registra resultados y genera informe de resultados.
 - Y por último tenemos al jefe de una Unidad de laboratorio el cual valida el informe de resultados.[23]

[29] MSC. MANUEL SANCHEZ CHERO, Desarrollo de un Sistema de Gestión

[54] WIKIPEDIA, Modelo del Negocio

[61] YAMIL RAMOS, PILAR STRONGUILO, Modelo del Negocio con RUP y UML

[23] JESÚS GARCÍA MOLINA, M. JOSÉ ORTÍN, BEGOÑA MOROS, JOAQUÍN NICOLÁS, AMBROSIO TOVAL, De los Procesos del Negocio a los Casos de Uso

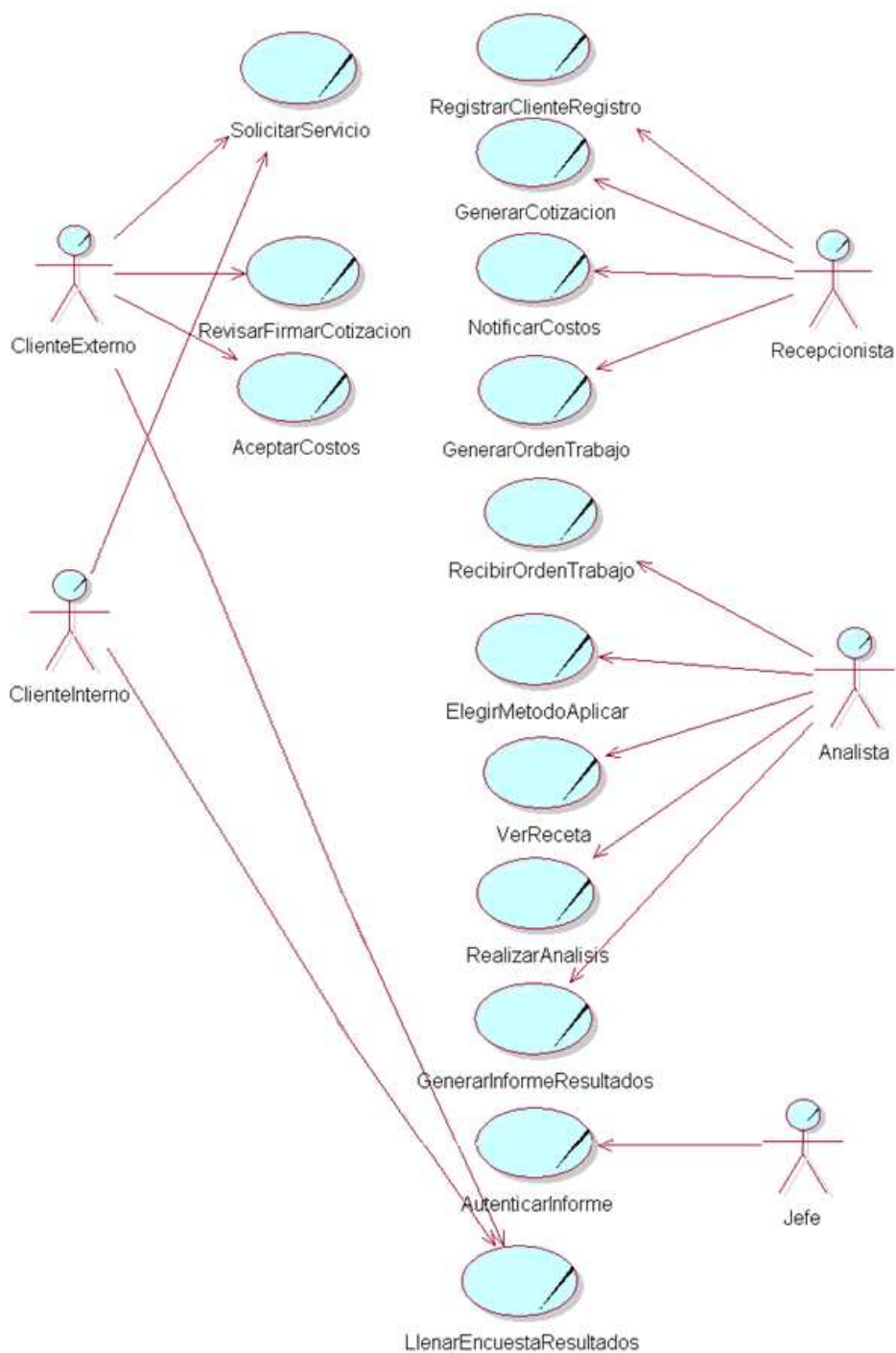


Figura 2. 2Diagrama de casos de uso del Negocio

2.1.1.2.2. Descripción de Casos de Uso del Negocio

No.	1
Proceso del Negocio	Solicitar Análisis
Objetivo	El cliente al llegar al laboratorio debe realizar la solicitud del respectivo(s) análisis.
Descripción	El Cliente del Laboratorio realiza las siguientes actividades: <ul style="list-style-type: none"> • Pregunta sobre los Servicios que brinda el Laboratorio • Solicita un(os) Servicio(s).

	<ul style="list-style-type: none"> • Socita una Proforma para saber los precios.
Tiempo de Ejecución	Al momento que el Cliente llega al Laboratorio.

Tabla 2. 6 Caso de Uso del Negocio Solicitar Análisis

No.	2
Proceso del Negocio	Registrar Cliente
Objetivo	Una vez que llega el Cliente se lo debe registrar como Cliente del Laboratorio si no existe como tal.
Descripción	La Secretaria/Recepcionista del Laboratorio realiza las siguientes actividades: <ul style="list-style-type: none"> • Selecciona un tipo de Cliente. • Ingresa el RUC o la cédula del Cliente, su nombre y dirección de forma obligatoria. • Ingresa datos adicionales como teléfono, fax, email, contacto.
Tiempo de Ejecución	Una vez que el Cliente solicita el Análisis.

Tabla 2. 7 Caso de Uso del Negocio Registrar Cliente

No.	3
Proceso del Negocio	Ingresar Servicio
Objetivo	Los servicios que brinda cada Laboratorio debe ser ingresado para posteriormente utilizado en las respectivas Proformas.
Descripción	La Secretaria/Recepcionista del Laboratorio realiza las siguientes actividades: <ul style="list-style-type: none"> • Selecciona un tipo de Servicio. • Ingresa el nombre, el laboratorio y el precio como datos necesarios. • Ingresa datos adicionales como dirección.
Tiempo de Ejecución	Una vez que el Laboratorio tiene un Servicio que brindar

Tabla 2. 8 Caso de Uso del Negocio Ingresar Servicio

No.	4
Proceso del Negocio	Generar Proforma
Objetivo	Establecer una lista de servicios con el respectivo método para al final tener el costo de los exámenes a realizar.
Descripción	La secretaria/Recepcionista del Laboratorio realiza las siguientes actividades: <ul style="list-style-type: none"> • Selecciona el Cliente al cual va dirigida la Proforma • Ingresa el o los servicios que se desean realizar. • Selecciona el método del servicio • Ingresa la cantidad • Guarda la Proforma, la imprime y la entrega al Cliente.
Tiempo de Ejecución	Al momento que el Cliente solicita un análisis

Tabla 2. 9 Caso de Uso del Negocio Generar Proforma

No.	5
Proceso del Negocio	Generar Factura
Objetivo	Ingresa un proforma se puede generar la respectiva Factura para cobrar el monto que se acepto en la Proforma.

Descripción	La Tesorera realiza las siguientes actividades: <ul style="list-style-type: none"> • Selecciona una Proforma • Ingresa el código de la Factura • Guarda la Factura y se cobra el monto.
Tiempo de Ejecución	Una vez aceptada la Proforma

Tabla 2. 10 Caso de Uso del Negocio Generar Factura

No.	6
Proceso del Negocio	Generar Orden Trabajo
Objetivo	Una vez pagada la Factura se debe generar una Orden de Trabajo para poder empezar a realizar los análisis de las diferentes muestras entregadas.
Descripción	La secretaria/Recepcionista del Laboratorio realiza las siguientes actividades: <ul style="list-style-type: none"> • Selecciona una Factura de la cual se va a generar Orden de Trabajo • Ingresa las fechas de recepción y de entrega del Informe de Resultados. • Ingresa en número de muestras • Ingresa el Responsable de realizar cada análisis, con su respectivo tiempo invertido.
Tiempo de Ejecución	Una vez cancelada la Factura

Tabla 2. 11 Caso de Uso del Negocio Generar Orden trabajo

No.	7
Proceso del Negocio	Consultar Orden Trabajo
Objetivo	Si el Cliente solicita el cambio del método para realizar el Servicio, se debe Consultar la Respectiva Orden de Trabajo a ser modificada.
Descripción	El Analista del Laboratorio realiza las siguientes actividades: Selecciona Orden de Trabajo Selecciona la Orden de Trabajo a ser modificada dando clic en el botón editar.
Tiempo de Ejecución	Cuando se solicita el cambio de método de un Servicio a realizar.

Tabla 2. 12 Caso de Uso del Negocio Consultar Orden Trabajo

No.	8
Proceso del Negocio	Generar Informe de Resultados
Objetivo	Se debe generar una Informe de Resultado por cada una de las muestras registradas en la Orden de Trabajo generada.
Descripción	El analista del Laboratorio realiza las siguientes actividades: <ul style="list-style-type: none"> • Selecciona la Orden de Trabajo • Selecciona una de las Muestra de la Orden de Trabajo seleccionada. • Ingresa el resultado de cada servicio realizado a la Muestra. • Puede modificar el resultado
Tiempo de Ejecución	Se lo ejecuta una vez realizado los respectivos análisis

Tabla 2. 13 Caso de Uso del Negocio Generar Informe Resultados

No.	9
Proceso del Negocio	Elegir Método a Aplicar
Objetivo	Si el Cliente sugiere el cambio del método con el cual se va a realizar

	el servicio el analista puede modificar el método establecido en la Orden de Trabajo.
Descripción	El analista del Laboratorio realiza las siguientes actividades: <ul style="list-style-type: none"> • Selecciona la Orden de Trabajo • Modifica el método del servicio necesario
Tiempo de Ejecución	Se lo ejecuta cuando el Cliente sugiere otro método.

Tabla 2. 14 Caso de Uso del Negocio Elegir Método Aplicar

No.	10
Proceso del Negocio	Realizar Análisis
Objetivo	El Analista una vez establecido el método con el cual se realiza el análisis de una muestra entregada por el Cliente.
Descripción	El analista del Laboratorio realiza las siguientes actividades: <ul style="list-style-type: none"> • Selecciona la Orden de Trabajo de la cual debe realizar los análisis • Verifica el método con el cual se va a realizar el Servicio. • Realiza el Análisis.
Tiempo de Ejecución	Una vez que se definan los métodos de cada Servicio a realizar.

Tabla 2. 15 Caso de Uso del Negocio Realizar Análisis

2.1.1.2.3. Modelo del Dominio

El modelo de dominio es el encargado de representar los conceptos (objetos) importantes en el dominio del problema.

El diseño de un modelo de dominio, constituye una actividad clásica del análisis orientado a objetos. Cuando nos hablamos de objetos, nos referimos a objetos del dominio (diccionario visual) no a objetos de software. ^[30] ^[56]

Un modelo de dominio incluye:

- Identificar las clases de objetos de interés en el dominio
- Identificar los atributos de las clases de objetos, y
- Las relaciones entre las clases de objetos.

[30] MSC. MARTHA D. DELGADO DAPENA, Definición del modelo del negocio y del dominio utilizando Razonamiento Basado en Casos

[56] VERÓNICA MACÍAS MENDOZA, Modelamiento Basado en el Dominio

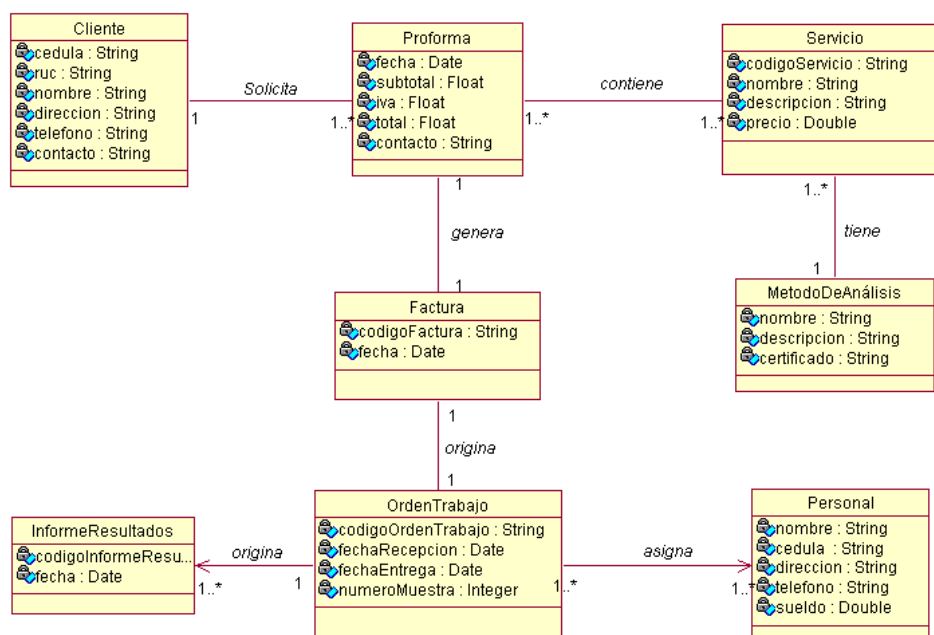


Figura 2. 3 Modelo del Dominio

2.1.1.3. Análisis de Riesgos

2.1.1.3.1. Riesgos del Proyecto

Los riesgos encontrados que pueden afectar el calendario establecido para el desarrollo del sistema son:

- La no disponibilidad del hardware o recursos necesarios.
- Cambios en la Gestión.

Magnitud del riesgo

Los riesgos del proyecto se los considera de gran magnitud, puesto que al involucrarse en la calendarización puede afectar la calidad, así como también podría provocar retrasos en la entrega del producto final.

Estrategia de Mitigación

Para garantizar la disponibilidad de recursos se realiza una planificación de los recursos necesarios con anticipación, de esta manera el proyecto no quedará parado.

Plan de Contingencia

En caso de que se presenten los riesgos mencionados, se tratará de utilizar los recursos presentes de una forma óptima para evitar contratiempos.

2.1.1.3.2.Riesgos del Producto

Los riesgos encontrados que pueden afectar la calidad o el rendimiento del software son:

- Cambios en los requerimientos
- Retraso en las especificaciones
- Malas estimaciones en cuanto a tiempo y personal

Magnitud del riesgo

Los riesgos del Producto son riesgos de gran magnitud, puesto que afectan la calidad y afectan también el rendimiento del software a diferencia de los riesgos del proyecto, estos no afectarán la calendarización del proyecto.

Estrategia de Mitigación

Se mantiene un control del software mediante presentación de prototipos al cliente, para poder ir validando la funcionalidad y rendimiento del software.

Plan de Contingencia

Si se presentan fallas en cuanto a la calidad se evaluará las áreas de conflicto para minimizar el impacto y si existe bajo rendimiento en la aplicación se harán optimizaciones de código para mejorar el rendimiento.

2.1.1.3.3.Riesgo técnico

Los riesgos técnicos afectan la calidad del producto y también la planificación del proyecto. Si se presentan riesgos técnicos en la etapa de implementación sería muy dificultoso o no se podría llevar a cabo.

Magnitud del riesgo

Los riesgos técnicos se los considera de magnitud alta, debido a su impacto en fechas calendarizadas y en la planificación del proyecto.

Estrategia de Mitigación

Se hace un listado o inventario de todos los recursos técnicos a usar y prever con anticipación si hace falta algo.

Plan de Contingencia

Si se presentan fallos se debe llegar a acuerdos para solventar lo más rápido posible las fallas y minimizar los riesgos. [14] [24]

VER ANEXO: Plantilla Análisis de Riesgos

2.1.2. REQUISITOS

Iniciamos con la sección de Identificación de Requisitos, que es el proceso por el cual se describirá los servicios que brindará el sistema y las restricciones asociadas a su funcionamiento. Estos requisitos pueden ser funcionales o no funcionales. Los requisitos funcionales nos muestran como interacciona el sistema con su entorno y cuál va a ser su funcionalidad, en cambio los requisitos no funcionales nos indican las restricciones sobre el espacio de posibles soluciones, es decir cómo debe ser el sistema, definiendo así los principales casos de uso y el alcance del proyecto.

Dentro de la Identificación de Requisitos tenemos el artefacto visión que tiene como propósito la recolección, el análisis y definición de las necesidades generales de la empresa que son percibidos a primera vista. Posteriormente se definen los Requisitos Específicos del Sistema, donde se describen de forma clara y precisa la funcionalidad que el sistema va a contener.

A continuación definimos el Esquema de Funcionamiento y Arquitectura del Sistema de modo que se entienda como se implantará el sistema para su funcionamiento en producción y como los accederán usuarios al sistema.

Una vez establecidos los requisitos del sistema y su esquema de funcionamiento, se procede al Modelamiento de Casos de Uso del Sistema, definiendo los Subsistemas, diseñando el Diagrama de Casos de Uso y desarrollando la Especificación de Casos de Uso. [43]

[14] FAIRLEY, R. "Risk Management for Software Development"

[24] JONES, C. Assessment and Control of Software Risks

[43] SOLUCIONES RACIONALES, RUP/EASY guía metodológica de desarrollo de sistemas

2.1.2.1. Visión

2.1.2.1.1. Alcance

El propósito de la visión es recoger, analizar y definir las necesidades de alto nivel y las características del sistema de gestión de Laboratorios. El documento se centra en la funcionalidad requerida por los participantes en el proyecto y los usuarios finales.

Esta funcionalidad se basa principalmente en la gestión de los Laboratorios de la Escuela Politécnica Nacional, de forma que dichos laboratorios sean capaces de atender las distintas solicitudes de análisis que son requeridas por parte de los clientes.

2.1.2.1.2. Posicionamiento del Producto

Oportunidad de Negocio

Este sistema permitirá a los laboratorios el control de todas sus actividades (gestión de clientes, de proformas, de servicios, etc.), lo cual supondrá un acceso rápido y sencillo a los datos, gracias a interfaces gráficas sencillas y amigables. Además, los datos accedidos estarán siempre actualizados, lo cual es un factor muy importante para poder llevar un control centralizado de los distintos laboratorios.

Sentencia que define el problema

El problema de	<p>Registrar los Clientes que solicitan la realización de un análisis.</p> <p>Generar las proformas que son generadas con los servicios solicitados por el cliente.</p> <p>Gestionar la facturación por cada proforma aceptada por el Cliente.</p> <p>Registrar los Servicios que brinda el laboratorio.</p> <p>Registrar los productos que tiene el laboratorio.</p> <p>Registrar el personal con el cual cuenta el laboratorio.</p> <p>Generar las Órdenes de Trabajo donde se registran las muestras entregadas por los Clientes.</p> <p>Generar Informes de Resultados generado por cada una de las muestras analizadas.</p>
----------------	--

afecta a	Laboratorios, Jefes de cada Laboratorio, Analistas del Laboratorio, Secretaria del laboratorio, Clientes de cada laboratorio, Departamento de contabilidad / facturación.
El impacto asociado es	Almacenar toda la información referente a los clientes, proformas, ordenes, facturas, informes de Resultados, servicios, productos, personal y que esta información esté al instante accesible y actualizada en lugares físicamente muy distantes es un proceso prácticamente imposible de realizar en el caso de que no esté informatizado.
Una solución adecuada sería	Informatizar el proceso, usando una intranet con una base de datos accesible desde los distintos nodos de la red y generar interfaces amigables y sencillas con las que se puede acceder a dicha base de datos.

Tabla 2. 16 Definición del Problema

Sentencia que define la posición del Producto

para	Laboratorios, Jefes de cada Laboratorio, Analistas del Laboratorio, Secretaria del laboratorio, Clientes de cada laboratorio, Departamento de contabilidad / facturación.
quienes	Controlan el ingreso de Clientes, proformas, las órdenes de trabajo, la facturación, los servicios, los productos, el personal, las muestras, los métodos y los informes de resultados.
El nombre del producto	Es una herramienta software.
que	Almacena la información necesaria para gestionar laboratorios.
no como	El sistema actual.
Nuestro producto	Permite administrar las distintas actividades de los laboratorios mediante una interfaz gráfica sencilla y amigable. Además proporciona un acceso rápido y actualizado a la información desde cualquier punto que tenga acceso a la base de datos.

Tabla 2. 17 Posición del Producto

2.1.2.1.3.Descripción Global del Producto

Perspectiva del producto

El producto a desarrollar es un sistema global para laboratorios, con la intención de agilizar su funcionamiento. Las áreas a tratar por el sistema son: solicitud de análisis, generación de proformas, generación de órdenes de trabajo, facturación y generación de informes de resultados.

Resumen de características

A continuación se mostrará un listado con los beneficios que obtendrá el cliente a partir del producto:

Beneficio del cliente	Características que lo apoyan
Mayor agilidad en los análisis.	Aplicación web desde la cual se puede registrar la solicitud.
Automatización de la generación de proformas y órdenes de trabajo.	Sistema automatiza la emisión de proformas y órdenes de trabajo.
Mayor administración.	Sistema automatiza el ingreso de servicios, métodos, productos, personal, también la modificación y eliminación de los mismos si estos los requieren y si es permitido.
Gestión automatizada del proceso de facturación	Sistema de optimización de generación de Facturas con el respectivo pago.
Automatización del sistema de nóminas	Sistema automático de generación de nóminas.
Mayor Facilidad al momento de generar Informes de resultados	Sistema genera Informe de Resultados de cada muestra analizada que se registro en la Orden de Trabajo.

Tabla 2. 18 Características del Producto

VER ANEXO: PLANTILLA DOCUMENTO VISION

2.1.2.2.Requisitos Específicos

2.1.2.2.1.Alcance

El producto a desarrollar se lo conocerá como SISTEMA DE ADMINISTRACIÓN CENTRALIZADA DE LABORATORIOS EN LA ESCUELA POLITÉCNICA NACIONAL.

El sistema beneficiará a los diferentes laboratorios de análisis, de modo que pueda brindar mayor comodidad y facilidad en el proceso de realización de análisis. Además, facilitará la recaudación del dinero por medio de tesorería.

EL SISTEMA DE ADMINISTRACIÓN CENTRALIZADA DE LABORATORIOS EN LA ESCUELA POLITÉCNICA NACIONAL tiene como objetivo llenar las expectativas de los usuarios, esto implica que haya una buena aceptación por parte de ellos y facilidad de uso para que interactúen con el mismo.

Este sistema tiene como meta alcanzar un alto nivel de calidad, eficacia y optimización en lo que se refiere al cumplimiento de los requerimientos de los usuarios.

2.1.2.2.2.Perspectiva del Producto

Este software dará respuesta a los requerimientos de los 30 laboratorio que se encuentran en la Escuela Politécnica Nacional pero podría ser utilizado en otras universidades del país o grupos de instituciones con características similares. Su objetivo es que la red de la universidad cuente con una herramienta Web que gestione las actividades de las respectivas Unidades de laboratorio.

2.1.2.2.3.Funcionalidad del Producto

Entre las funciones que debe realizar el sistema se encuentran las siguientes:

- Registro de Unidades de Laboratorio.
- Ingreso de Laboratorios pertenecientes a cada Unidad.
- Ingreso de Tipos de Cliente, establecidos por las autoridades de la EPN.
- Registro de clientes, con datos fundamentales para procesos posteriores.
- Ingreso de Tipos de Servicio, establecidos por las autoridades de la EPN.
- Registro de Métodos de acuerdo a cada servicio.
- Ingreso de Productos usados por los diferentes laboratorios.
- Emisión de proformas de los análisis solicitados por un cliente, y posterior aprobación.
- Generación de factura una vez aprobada la proforma.
- Registro de Muestras a ser analizadas.
- Elaboración de la Orden de Trabajo una vez cancela la factura.

- Distribución de trabajo para cada analista.
- Realización del informe de resultados.

2.1.2.2.4. Restricciones

El desarrollo del proyecto se basará en la metodología de desarrollo RUP, las herramientas de desarrollo a utilizar son:

- Java Eclipse como plataforma de desarrollo.
- Postgres como motor de base de datos.
- Jboss como servidor de aplicaciones J2EE.
- Mientras que para el modelamiento se usará la herramienta con licenciamiento Power Designer y Rational Rose.

Las máquinas que se utilizarán para el respectivo diseño y desarrollo del sistema tienen las siguientes características:

- Procesador mínimo Pentium 4.
- Memoria RAM mínimo 2 GB.
- Unidad de DVD.
- La máquinas se encuentran con sistema operativo Windows XP

El tiempo estimado para el desarrollo del sistema en su primera versión es de 8 meses a partir del mes de octubre de 2009.

2.1.2.2.5. Requisitos Funcionales

Administración de Clientes: En el sistema se almacena los datos más importantes de cada uno de los clientes que se acerquen al respectivo laboratorio a solicitar la realización de un análisis. Adicionalmente los datos de los clientes registrados pueden ser modificados y/o eliminados.

Elaboración de proforma de los análisis solicitados por un cliente: Se genera una cotización con los precios correspondientes a cada análisis a realizar a la respectiva muestra. La cotización puede ser aceptada el mismo instante de la emisión o enviada por e-mail para una posterior aceptación.

Generar Factura: Una vez que el cliente acepta los costos mostrados en la proforma el sistema emite la respectiva Factura con los costos que deben ser cancelados en Tesorería por parte del cliente.

Generación de Orden de Trabajo: Una vez que el cliente ha cancelado el valor correspondiente del análisis a realizar, se genera la orden de trabajo.

Asignar carga de trabajo a los analistas: Se asignará una orden de trabajo a cada analista de manera q ninguno se quede sin realizar análisis y por otro lado no se le acumulen una serie de análisis a un mismo analista.

Recepción de Órdenes de Trabajo: una vez que se genera una orden de trabajo el analista revisa que ordenes de trabajo le fueron asignadas y que análisis (servicios) debe realizar y con qué método.

Registro de resultados: Una vez realizados los análisis, cada analista debe registrar los resultados obtenidos, una vez realizado el análisis correspondiente.

Generación de informe: Una vez que se ingresan los resultados el sistema genera un informe el cual será enviado al jefe del respectivo laboratorio.

2.1.2.2.6.Requisitos no Funcionales

Rendimiento: El sistema será construido para trabajar vía web, dentro del campus politécnico, por lo cual el número de terminales que se conecten simultáneamente al sistema variará con el tiempo. Por lo tanto el tiempo de respuesta a los usuarios, dependerá del ancho de banda que se maneja en la red, que en condiciones normales es de 2MB por segundo.

Así, se estima que el 95% de transacciones solicitadas al sistema sean atendidas en un segundo. El 95% de los usuarios sean atendidos tras solicitar una conexión simultánea a la red en un segundo.

Seguridad: Con el fin de evitar infiltraciones maliciosas al sistema y por ende a la información, se han decido manejar perfiles de usuario para el acceso al sistema, de tal modo que, de acuerdo al tipo de usuario que ingrese, este puede acceder a determinadas funcionalidades del sistema y restringirle el acceso a aquellas que no son de su interés.

Fiabilidad: Se estima que el sistema soportará un total de incidentes permisibles del 5% dado que la congestión en la red podría provocar retrasos que podrían afectar al usuario.

Disponibilidad: El sistema se encontrará accesible para los usuarios en un 99,99%, las veinticuatro horas del día, los 365 días del año.

Mantenibilidad: En cuanto a mantenibilidad, es importante realizar un mantenimiento predictivo en busca de posibles errores, que impidan el uso adecuado del sistema.

El sistema deberá ser mantenido por los desarrolladores del sistema, al tratarse de un sistema web, más aún al ser la primera versión que será lanzada en base a dos laboratorios, y que posteriormente se ampliará a un total de treinta laboratorios.

Portabilidad: El sistema será desarrollado siguiendo una arquitectura cliente-servidor, y los atributos que presenta este en cuanto a herramientas y plataformas se describe a continuación:

- El porcentaje de componentes dependientes del servidor es del 85%
- El porcentaje de código dependiente del servidor es del 85%.
- La plataforma de desarrollo que se maneja es JAVA ECLIPSE.
- Se maneja un servidor de aplicaciones JBOSS
- Como sistema operativo se tiene WINDOWS XP.

2.1.2.3.Casos de Uso

Para el modelamiento del Sistema se realizan diagramas de casos de uso planteados para cada uno de los subsistemas definidos para los laboratorios (Ver Plan de Desarrollo).

2.1.2.3.1.Subsistemas

- Módulo Administrativo
- Administración de Laboratorio
- Administración de Servicios
- Facturación

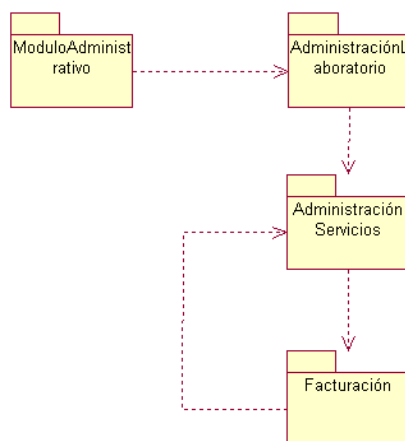


Figura 2. 4 Subsistemas

2.1.2.3.2. Diagrama de casos de uso del sistema

Módulo Administrativo

Este Subsistema será manejado por una sola persona la cual tiene permisos privilegiados, este se encargará de la información sensible que todos los Laboratorios utilizarán como es el ingreso, modificación y eliminación de Unidad, Tipos de Cliente, Tipo de Servicio, Tipo de Producto, Tipo Personal, Cargo Personal.

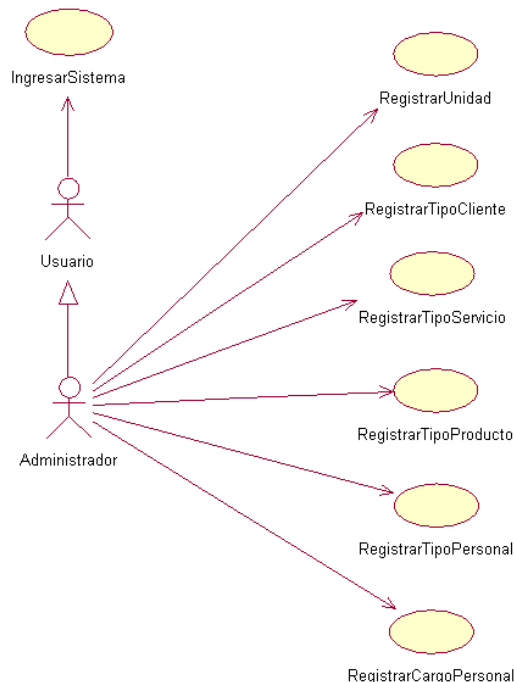


Figura 2. 5 Módulo Administrativo

Administración de Laboratorio

En el subsistema del Laboratorio se administran los Laboratorios, Cliente, Personal, Unidad Medida, Norma, los Servicios y los Métodos con los que cuenta cada laboratorio para brindar a los Clientes.

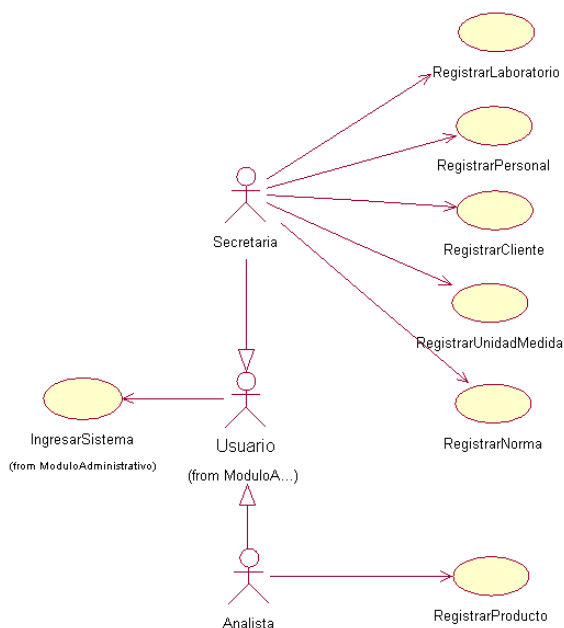


Figura 2. 6 Administración de Servicios

Administración de Servicios

En el subsistema Administración de Servicios se emiten, modifican y si no existen datos relacionados se pueden eliminar Proformas, Muestras, Órdenes de Trabajo e Informes de Resultados todos relacionados con los Servicios solicitados por el Cliente.

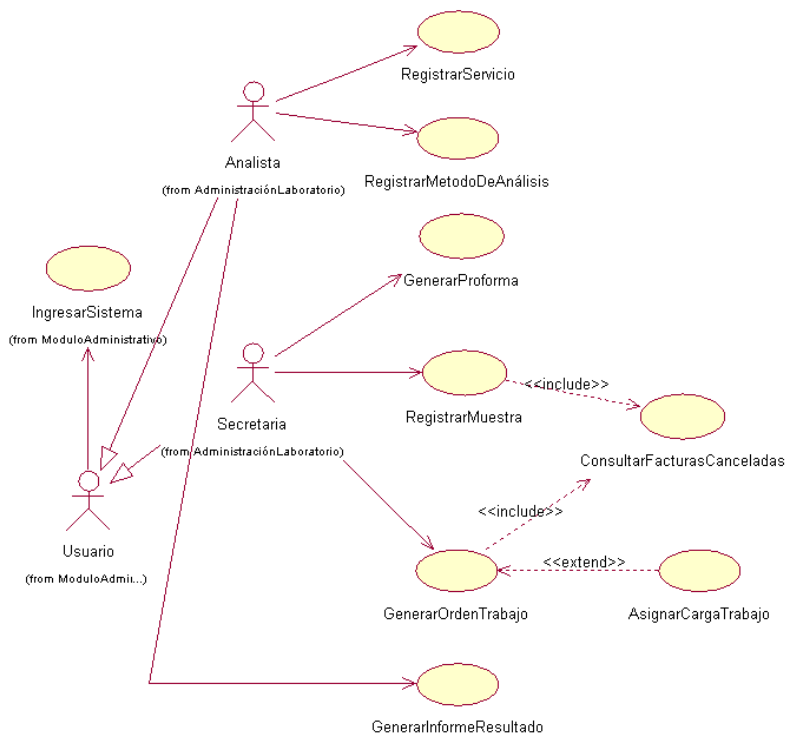


Figura 2. 7 Administración de Análisis

Facturación

La gestión de facturación tiene como objetivo para generar facturas de las proformas aceptadas por el Cliente y el cobro de las mismas.

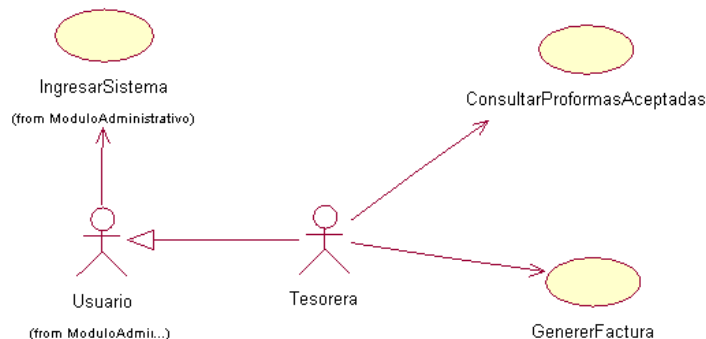


Figura 2. 8 Facturación

2.1.2.3.3. Especificación de Casos de Usos

Ingresar al Sistema (Caso de uso CU01)

1. Descripción

Este caso de uso especifica el Ingreso al Sistema por parte de los Usuarios, que tiene por objetivo permitir que un usuario previamente registrado en la nómina de personal pueda tener acceso a las operaciones que le son conferidas de acuerdo a su perfil.

2. Flujo de Eventos

2.1 Flujo Básico

1. El Usuario ingresa a la pantalla principal del Sistema, donde se muestra la página que le permite loguearse.
2. El usuario selecciona el perfil al que pertenece
3. El usuario ingresa su respectivo login y password.
4. El usuario selecciona el botón 'Aceptar'.
5. El Sistema procede a validar el login y el password del usuario de acuerdo a su perfil.
 - a. El login es válido, se procede a validar el password.
 - b. El password es válido
6. El caso de uso termina cuando el usuario ingresa al sistema.

2.2 Flujos Alternativos

1.2.1 En el punto 5.a.- El login no es válido, se muestra un mensaje de error: "LOS DATOS INGRESADOS NO SON VÁLIDOS".

1.2.2 En el punto 5.b.- El password no es válido, se muestra un mensaje de error: "LOS DATOS INGRESADOS NO SON VÁLIDOS".

3. Precondiciones

3.1 El Usuario debe ser registrado por el Administrador en la base de datos de Personal.

4. Poscondiciones

El usuario visualizará las opciones que tiene disponible en el menú de acuerdo a su perfil.

Registrar Unidad (Caso de uso CU02)

1. Descripción

Este caso de uso especifica el registro de una Unidad, que tiene por objetivo ingresar los datos de una Unidad.

El Administrador es la persona encargada de registrar las Unidades de Laboratorio que se encuentran en la EPN. El código de la Unidad se autogenerará al momento de almacenar la misma.

2. Flujo de Eventos

2.1 Flujo Básico

1. El Administrador ingresa los datos de la Unidad (requeridos: nombre, jefe; opcionales: descripción).
2. El Administrador selecciona el botón guardar Unidad.
 - a. El sistema valida que todos los campos requeridos estén llenos.
 - b. El sistema valida que todos los datos ingresados sean correctos.
 - c. El sistema guarda la Unidad y emite un mensaje de éxito "Datos almacenados con éxito".

2.2 Flujos Alternativos

2.2.1 En el punto 2.a.- El sistema coloca una X en los datos requeridos faltantes.

2.2.2 En el punto 2.b.- Si los datos ingresados son incorrectos el sistema emite un mensaje de error indicando el dato que es incorrecto.

3. Precondiciones

3.1 El Administrador ha ingresado correctamente el login y password para ingresar al sistema.

3.2 El Administrador ha seleccionado en el menú correspondiente la opción "Unidad".

4. Poscondiciones

La Unidad ingresada está disponible para ser utilizada en el registro de Laboratorios, Productos.

Caso de uso Registrar Tipo Cliente (Caso de uso CU03)

1. Descripción

Este caso especifica el registro de un tipo cliente, cuyo objetivo es definir los tipos de clientes que se van a manejar dentro de la institución, de modo que los clientes puedan ser clasificados de acuerdo al tipo de cliente que se seleccione.

El Administrador es la persona encargada de registrar en el sistema los tipos de clientes. El código de cada tipo de cliente se autogenerará el momento de almacenarlo.

2. Flujo de Eventos

1. **Flujo Básico**

1. El Administrador ingresar los campos (requeridos: nombre, opcionales: descripción) en la interfaz Tipo Cliente.
3. El Administrador selecciona la opción guardar Tipo Cliente.
 - a. El Sistema valida que los campos requeridos hayan sido ingresados.
 - b. El Sistema valida que el tipo de dato ingresado sea correcto.
4. El Sistema guarda Tipo Cliente y emite un mensaje de éxito "Datos almacenados con éxito."

2. **Flujos Alternativos**

2.2.1 En el punto 2.a.- El sistema señala con una x los campos que no han sido ingresados; si estos están señalados como requeridos.

2.2.2 En el punto 2.b.- El sistema emite un mensaje de error que señala que el tipo de dato es incorrecto.

3. Precondiciones

1. El Administrador ha ingresado correctamente el login y password para ingresar al sistema.
2. El Administrador ha seleccionado la interfaz gráfica "Tipo Cliente" de su menú correspondiente.

4. Poscondiciones

1. El tipo de cliente ingresado estará disponible para el registro de un cliente.

Registrar Tipo Servicio (Caso de uso CU014)

1. Descripción

Este caso de uso se especifica el registro de un Tipo Servicio, que tiene por objetivo ingresar los datos de un Tipo Servicio que asocia a Servicios. El Administrador es la persona encargada de registrar los Tipo Servicios. El código del Tipo Servicio se autogenerará al momento de almacenar el mismo.

2. Flujo de Eventos

1. **Flujo Básico**

1. El Administrador ingresa los datos del Tipo Servicio (requeridos: nombre; opcionales: descripción).
2. El Administrador selecciona el botón guardar Tipo Servicio.
 - a. El sistema valida que todos los campos requeridos estén llenos.
 - b. El sistema valida que todos los datos ingresados sean correctos.
 - c. El sistema guarda el Tipo Servicio y emite un mensaje de éxito "Datos almacenados con éxito".

1. **Flujos Alternativos**

1. En el punto 2.a.- El sistema coloca una X en los datos requeridos faltantes.

2. En el punto 2.b.- Si los datos ingresados son incorrectos el sistema emite un mensaje de error indicando el dato que es incorrecto.
3. Precondiciones
 1. El Administrador ha ingresado correctamente el login y password para ingresar al sistema.
 2. El Administrador ha seleccionado en el menú correspondiente la opción "Tipo Servicio".
4. Poscondiciones
 1. El Tipo Servicio ingresado está disponible para ser utilizado en el registro de servicios.

Caso de uso Registrar Tipo Producto (Caso de uso CU05)

1. Descripción

Este caso especifica el registro de un tipo producto, cuyo objetivo es definir los tipos de producto que se van a manejar dentro de la institución, de modo que los productos ingresados puedan ser clasificados de acuerdo al tipo de producto que seleccione.

El Administrador es la persona encargada de registrar en el sistema los tipos de producto. El código de cada tipo de producto se autogenerará el momento de almacenarlo.
2. Flujo de Eventos
 1. **Flujo Básico**
 1. El Administrador ingresar los campos requeridos en la interfaz "Tipo Producto".
 2. El Administrador selecciona la opción guardar Tipo Producto.
 - a. El Sistema valida que los campos requeridos hayan sido ingresados.
 - b. El Sistema valida que el tipo de dato ingresado sea correcto.
 2. El Sistema guarda Tipo Producto y emite un mensaje de éxito "Datos almacenados con éxito."
 2. **Flujos Alternativos**
 1. En el punto 2.a.- El sistema señala con una x los campos que no han sido ingresados; si estos están señalados como requeridos.

2. En el punto 2.b.- El sistema emite un mensaje de error que señala que el tipo de dato es incorrecto.
3. Precondiciones
 1. El Administrador ha ingresado correctamente el login y password para ingresar al sistema.
 2. El Administrador ha seleccionado la interfaz gráfica “Tipo Producto” de su menú correspondiente.
4. Poscondiciones
 1. El tipo de producto ingresado estará disponible para el registro de un Producto.

Caso de uso Registrar Tipo Personal (Caso de uso CU06)

1. Descripción

Este caso especifica el registro de un tipo personal, cuyo objetivo es definir los tipos de personal que se van a manejar dentro de la institución, de modo que al ingresar un personal, este pueda ser clasificado de acuerdo al tipo de personal que seleccione.

El Administrador es la persona encargada de registrar en el sistema los tipos de personal. El código de cada tipo de personal se autogenerará el momento de almacenarlo.
2. Flujo de Eventos
 1. **Flujo Básico**
 1. El Administrador ingresar los campos (requeridos: nombre, opcionales: descripción) en la interfaz “Tipo Personal”.
 2. El Administrador selecciona la opción guardar Tipo Personal.
 - a. El Sistema valida que los campos requeridos hayan sido ingresados.
 - b. El Sistema valida que el tipo de dato ingresado sea correcto.
 3. El Sistema guarda Tipo Personal y emite un mensaje de éxito “Datos almacenados con éxito.”
 1. **Flujos Alternativos**
 1. En el punto 2.a.- El sistema señala con una x los campos que no han sido ingresados; si estos están señalados como requeridos.

2. En el punto 2.b.- El sistema emite un mensaje de error que señala que el tipo de dato es incorrecto.
3. Precondiciones
 1. El Administrador ha ingresado correctamente el login y password para ingresar al sistema.
 2. El Administrador ha seleccionado la interfaz gráfica "Tipo Personal" de su menú correspondiente.
4. Poscondiciones
 1. El tipo de personal ingresado estará disponible para el registro de un Personal.

Caso de uso Registrar Cargo Personal (Caso de uso CU07)

1. Descripción

Este caso especifica el registro de un cargo personal, cuyo objetivo es definir los cargos de personal que se van a manejar dentro de la institución, de modo que al ingresar un personal, este pueda ser clasificado de acuerdo al cargo de personal que seleccione.

El Administrador es la persona encargada de registrar en el sistema los cargos de personal. El código de cada cargo de personal se autogenerará el momento de almacenarlo.
2. Flujo de Eventos
 1. **Flujo Básico**
 1. El Administrador ingresar los campos (requeridos: nombre, opcionales: descripción) en la interfaz "Cargo Personal".
 2. El Administrador selecciona la opción guardar Cargo Personal.
 - a. El Sistema valida que los campos requeridos hayan sido ingresados.
 - b. El Sistema valida que el cargo de dato ingresado sea correcto.
 2. El Sistema guarda Cargo Personal y emite un mensaje de éxito "Datos almacenados con éxito."
 2. **Flujos Alternativos**
 1. En el punto 2.a.- El sistema señala con una x los campos que no han sido ingresados; si estos están señalados como requeridos.

2. En el punto 2.b.- El sistema emite un mensaje de error que señala que el cargo de dato es incorrecto.
3. Precondiciones
 1. El Administrador ha ingresado correctamente el login y password para ingresar al sistema.
 2. El Administrador ha seleccionado la interfaz gráfica "Cargo Personal" de su menú correspondiente.
4. Poscondiciones
 1. El cargo de personal ingresado estará disponible para el registro de un Personal.

Registrar Laboratorio (Caso de uso CU08)

1. Descripción

Este caso de uso especifica el registro de un Laboratorio, que tiene por objetivo ingresar los datos de un Laboratorio que pertenece a una Unidad. La Secretaria es la persona encargada de registrar los Laboratorios que las diferentes Unidades tienen. El código del Laboratorio se autogenerará al momento de almacenar el mismo.
2. Flujo de Eventos
 1. **Flujo Básico**
 1. La secretaria selecciona la Unidad en la va a pertenecer el laboratorio.
 2. El sistema presenta la Unidad seleccionada.
 3. La secretaria ingresa los datos requeridos del Laboratorio.
 3. La secretaria selecciona el botón guardar Laboratorio.
 - c. El sistema valida que todos los campos requeridos que estén llenos.
 - d. El sistema valida que todos los datos ingresados sean correctos.
 - e. El sistema guarda el laboratorio y emite un mensaje de éxito "Datos almacenados con éxito".
 2. **Flujos Alternativos**
 1. En el punto 4.a.- El sistema coloca una X en los campos requeridos faltantes.

2. En el punto 4.b.- Si los campos ingresados son incorrectos el sistema emite un mensaje de error indicando el dato que es incorrecto.
3. Precondiciones
 2. La secretaria ha ingresado correctamente el login y password para ingresar al sistema.
 3. La secretaria ha seleccionado en el menú correspondiente la opción "Laboratorio".
 4. Deben existir Unidades ingresadas en la base de datos.
4. Poscondiciones

Caso de uso Registrar Personal (Caso de uso CU09)

1. Descripción

Este caso especifica el registro de personal, cuyo objetivo es registrar al personal que pertenece a una Unidad de Laboratorio prestando sus servicios profesionales.

El registro de un personal debe estar asociado de manera obligatoria a un Tipo Personal y a un Personal, de modo que un personal pueda ser clasificado por medio de estos dos parámetros. Adicionalmente debe seleccionar el Laboratorio al cual prestará sus servicios.

La Secretaria es la persona encargada de registrar en el sistema al personal que ingresa a un Laboratorio. El código de cada de personal se autogenerará el momento de almacenarlo, con un formato establecido por el usuario para Personal.
2. Flujo de Eventos
 1. **Flujo Básico**
 1. La Secretaria selecciona el Laboratorio del combo que se presenta con todos los Laboratorios registrados para esa Unidad de Laboratorio.
 2. El sistema presenta el Laboratorio seleccionado.
 3. La Secretaria selecciona el Tipo Personal del combo que se presenta con todos los Tipos de Personal disponibles.
 4. El sistema presenta el tipo de Personal seleccionado.

4. La Secretaria selecciona el Cargo Personal del combo que se presenta con todos los Cargos de Personal disponibles.
5. El sistema presenta el Cargo de Personal seleccionado.
6. La Secretaria ingresa los campos (requeridos: fecha Ingreso, fecha Fin, nombre, cédula, sueldo, nombre de usuario y clave; opcionales: dirección, teléfono, email) en la interfaz "Personal".
7. La Secretaria selecciona la opción guardar Personal.
 - f. El Sistema valida que los campos requeridos hayan sido ingresados.
 - g. El Sistema valida que los datos ingresados sean correctos.
8. El Sistema guarda Personal y emite un mensaje de éxito: "Datos almacenados con éxito."

2. **Flujos Alternativos**

1. En el punto 8.a.- El sistema señala con una x los campos que no han sido ingresados; si estos están señalados como requeridos.
2. En el punto 8.b.- El sistema emite un mensaje de error que señala que los datos son incorrectos.

2. Precondiciones

2. La Secretaria ha ingresado correctamente el login y password para ingresar al sistema.
3. La Secretaria ha seleccionado la interfaz gráfica "Personal" de su menú correspondiente.
4. Deben existir Tipos Personal ingresados en la base de datos.
5. Deben existir Cargos Personal ingresados en la base de datos.

3. Poscondiciones

2. El personal ingresado estará disponible para la asignación de Carga de Personal durante la generación de una orden de trabajo..

Caso de uso Registrar Cliente (Caso de uso CU10)

1. Descripción

Este caso especifica el registro de clientes, cuyo objetivo es ingresar un cliente que requiera los servicios de una Unidad de laboratorio.

El registro de un cliente debe estar asociado de manera obligatoria a un Tipo Cliente, de modo que el cliente pueda ser clasificado por medio del Tipo Cliente seleccionado.

La Secretaria es la persona encargada de registrar en el sistema al Cliente. El código de cada de cliente se autogenerará el momento de almacenarlo, con un formato establecido por el usuario, para Cliente.

2. Flujo de Eventos

1. **Flujo Básico**

1. La Secretaria selecciona el Tipo Cliente del combo que se presenta con todos los Tipos de Cliente disponibles.
2. El sistema presenta el tipo de Cliente seleccionado.
3. La Secretaria ingresa los campos (Requeridos: nombre, dirección, cédula o ruc; Opcionales: teléfono Convencional, teléfono Cedular, otro teléfono, fax, email, contacto) en la interfaz “Cliente”.
4. La Secretaria selecciona la opción guardar Cliente.
 - a. El Sistema valida que los campos requeridos hayan sido ingresados.
 - b. El Sistema valida que los datos ingresados sean correctos.
5. El Sistema guarda Cliente y emite un mensaje de éxito: “Datos almacenados con éxito.”

2. **Flujos Alternativos**

1. En el punto 4.a.- El sistema señala con una x los campos que no han sido ingresados; si estos están señalados como requeridos.
2. En el punto 4.b.- El sistema emite un mensaje de error que señala que los datos son incorrectos.

3. Precondiciones

1. La Secretaria ha ingresado correctamente el login y password para ingresar al sistema.
2. La Secretaria ha seleccionado la interfaz gráfica “Cliente” de su menú correspondiente.
3. Deben existir Tipos Cliente ingresados en la base de datos.

4. Poscondiciones

1. El Cliente ingresado estará disponible para la generación de Proformas.

Registrar Unidad Medida (Caso de uso CU11)

1. Descripción

Este caso de uso específica el registro de una Unidad de Medida, que tiene por objetivo ingresar los datos de una Unidad de Medida que se usará al crear un Producto.

La Secretaria es la persona encargada de registrar las Unidades de Medida. El código de la Unidad de Medida se autogenerará al momento de almacenar la misma.

2. Flujo de Eventos

1. **Flujo Básico**

1. La Secretaria ingresa los datos de la Unidad de Medida (requeridos: nombre, sigla; opcionales: descripción)
2. La Secretaria selecciona el botón guardar Unidad de Medida.
 - c. El sistema valida que todos los campos requeridos estén llenos.
 - d. El sistema valida que todos los datos ingresados sean correctos.
 - e. El sistema guarda la Unidad de Medida y emite un mensaje de éxito “Datos almacenados con éxito”.

2. **Flujos Alternativos**

1. En el punto 2.a.- El sistema coloca una X en los datos requeridos faltantes.
2. En el punto 2.b.- Si los datos ingresados son incorrectos el sistema emite un mensaje de error indicando el dato que es incorrecto.

3. Precondiciones

1. La Secretaria ha ingresado correctamente el login y password para ingresar al sistema.
2. La Secretaria ha seleccionado en el menú correspondiente la opción “Unidad de Medida”.

4. Poscondiciones

1. La Unidad de Medida ingresada está disponible para ser utilizada en el registro de Productos y Resultados.

Caso de uso Registrar Norma (Caso de uso CU12)

1. Descripción

Este caso especifica el registro de una norma, cuyo objetivo es registrar los nombres de las normativas que se encuentran disponibles para la realización de un servicio.

La Secretaria es la persona encargada de registrar en el sistema los nombres de las normativas. El código de cada norma se autogenerará el momento de almacenarlo.

2. Flujo de Eventos

1. **Flujo Básico**

1. La Secretaria ingresar los campos (requeridos: nombre, opcionales: descripción) en la interfaz Norma.
2. La Secretaria selecciona la opción guardar Norma.
 1. El Sistema valida que los campos requeridos hayan sido ingresados.
 2. El Sistema valida que el tipo de dato ingresado sea correcto.
3. El Sistema guarda Norma y emite un mensaje de éxito "Datos almacenados con éxito."

3. **Flujos Alternativos**

1. En el punto 2.a.- El sistema señala con una x los campos que no han sido ingresados; si estos están señalados como requeridos.
2. En el punto 2.b.- El sistema emite un mensaje de error que señala que el tipo de dato es incorrecto.

3. Precondiciones

1. La Secretaria ha ingresado correctamente el login y password para ingresar al sistema.
2. La Secretaria ha seleccionado la interfaz gráfica "Norma" de su menú correspondiente.

4. Poscondiciones

1. La norma ingresada estará disponible durante la generación de un informe de resultados.

Registrar Producto (Caso de uso CU013)

1. Descripción

Este caso de uso especifica el registro de un Producto, que tiene por objetivo ingresar los datos de un Producto que pertenecen a una Unidad.

El Analista es la persona encargada de registrar los Productos que la Unidad tiene. El código del Producto se autogenerará al momento de almacenar el mismo.

2. Flujo de Eventos

1. **Flujo Básico**

1. El Analista selecciona la Unidad a la que va a pertenecer el Producto.
2. El sistema presenta la Unidad seleccionada.
3. El Analista selecciona el Tipo de Producto al que va a pertenecer el Producto.
4. El sistema presenta el Tipo de Producto seleccionado.
5. El Analista ingresa los datos del Producto (requeridos: ; opcionales:).
6. El Analista selecciona el botón guardar Producto.
 - a. El sistema valida que todos los campos requeridos estén llenos.
 - b. El sistema valida que todos los datos ingresados sean correctos.
 - c. El sistema guarda el Producto y emite un mensaje de éxito "Datos almacenados con éxito".

2. **Flujos Alternativos**

1. En el punto 6.a.- El sistema coloca una X en los datos requeridos faltantes.
2. En el punto 6.b.- Si los datos ingresados son incorrectos el sistema emite un mensaje de error indicando el dato que es incorrecto.

3. Precondiciones

1. El Analista ha ingresado correctamente el login y password para ingresar al sistema.
2. El Analista ha seleccionado en el menú correspondiente la opción "Producto".
3. Deben existir Unidades ingresadas en la base de datos.
4. Deben existir Tipo de Productos ingresados en la base de datos.

4. Poscondiciones

1. El Producto ingresado está disponible para ser utilizado en el registro de métodos.

Registrar Servicio (Caso de uso CU014)

1. Descripción

Este caso de uso se especifica el registro de un Servicio, que tiene por objetivo ingresar los datos de un Servicio que pertenecen a un Laboratorio y a un Tipo de Servicio.

El Analista es la persona encargada de registrar los Servicios que el Laboratorio brinda. El código del Servicio se autogenerará al momento de almacenar el mismo.

2. Flujo de Eventos

1. **Flujo Básico**

1. El Analista selecciona el Laboratorio al cual va a pertenecer el Servicio.
 2. El sistema presenta el Laboratorio seleccionado.
 3. El Analista selecciona el Tipo de Servicio al que va a pertenecer el Servicio.
 4. El sistema presenta el Tipo de Servicio seleccionado.
 5. El Analista ingresa los datos del Servicio (requeridos: ; opcionales:).
 6. El Analista selecciona el botón guardar Servicio.
 - a. El sistema valida que todos los campos requeridos estén llenos.
 - b. El sistema valida que todos los datos ingresados sean correctos.
 - c. El sistema guarda el Servicio y emite un mensaje de éxito "Datos almacenados con éxito".

2. **Flujos Alternativos**

1. En el punto 6.a.- El sistema coloca una X en los datos requeridos faltantes.
 2. En el punto 6.b.- Si los datos ingresados son incorrectos el sistema emite un mensaje de error indicando el dato que es incorrecto.

3. Precondiciones

1. El Analista ha ingresado correctamente el login y password para ingresar al sistema.
2. El Analista ha seleccionado en el menú correspondiente la opción "Servicio".
3. Deben existir Unidades ingresadas en la base de datos.
4. Deben existir Tipo de Servicios ingresados en la base de datos.
4. Poscondiciones
 1. El Servicio ingresado está disponible para ser utilizado en el registro de métodos y generación de proformas.

Registrar Método (Caso de uso CU15)

1. Descripción

Este caso de uso especifica el registro de un método, que tiene por objetivo el ingreso de los productos con sus respectivas cantidades que se usarán en dicho método, el cual está asociado con un servicio.

El analista es la persona encargada de registrar en el sistema, los métodos relacionados con un servicio. El código de cada Método se autogenerará el momento de almacenar el mismo.

2. Flujo de Eventos

1. **Flujo Básico**

1. El analista selecciona un Servicio al cuál pertenecerá el Método.
2. El sistema despliega el Servicio seleccionado.
3. El analista ingresa los datos (requeridos: nombre, opcionales: descripción, certificado).
4. El analista selecciona la opción Agregar Detalle.
5. El sistema muestra la ventana, el analista ingresa los datos (requeridos: nombre del Producto, cantidad de Producto y la unidad de medida).
6. El analista selecciona la opción aceptar.
 - a. El sistema valida que todos los campos requeridos estén llenos.
 - b. El sistema valida que todos los datos ingresados sean correctos.
 - c. El sistema guarda el producto ingresado en una lista.
7. El analista selecciona la opción guardar Método.
 - a. El sistema valida que todos los campos requeridos estén llenos.

- b. El sistema valida que el método ingresado no exista.
- c. El sistema valida que el método tenga productos ingresados en su Detalle.
- d. El sistema valida que todos los datos ingresados sean correctos.
- e. El sistema guarda el Método y emite un mensaje de éxito “Datos almacenados con éxito”.

2. **Flujos Alternativos**

- 1. En el punto 6.a.- El sistema coloca una X en los datos requeridos faltantes.
- 2. En el punto 6.a.- Si los datos ingresados son incorrectos el sistema emite un mensaje de error indicando el dato que es incorrecto
- 3. En el punto 7.a.- El sistema coloca una X en los datos requeridos faltantes.
- 4. En el punto 7.b.- El sistema emite un mensaje de error “Método ya existe”.
- 5. En el punto 7.b.- El sistema emite un mensaje de error “Ingrese Detalle de Método ”
- 6. En el punto 7.d.- Si los datos ingresados son incorrectos el sistema emite un mensaje de error indicando el dato que es incorrecto.

3. Precondiciones

- 1. El analista ha ingresado correctamente el login y password para ingresar al sistema.
- 2. El analista ha seleccionado en el menú correspondiente la opción “Método”.
- 3. Deben existir Servicios ingresados en la base de datos.
- 4. Deben existir Productos ingresados en la base de datos.

4. Poscondiciones

- 1. Los métodos registrados por servicios se utilizaran para la generación de una proforma y la realización del análisis.

Generar Proforma (Caso de uso CU16)

1. Descripción

Este caso de uso especifica la generación de Proforma, que tiene por objetivo el registro de los datos en una Proforma emitida por una de las Unidades de Laboratorio de EPN, al cliente que haya solicitado la realización de análisis de acuerdo a uno o varios servicios.

La Secretaria es la persona encargada de registrar en el sistema, las Proformas solicitadas por un cliente. El código de cada Proforma se autogenerará el momento de almacenar la misma, con un formato establecido por el usuario.

2. Flujo de Eventos

1. **Flujo Básico**

1. La Secretaria selecciona el nombre del Cliente que solicita la realización de un análisis y envía una consulta de los datos de cliente mediante el nombre seleccionado.
2. El sistema devuelve los datos del cliente correspondiente, y almacena códigoCliente en la proforma a generar.
3. La Secretaria ingresa el campo representante, con lo cual se termina de llenar la sección cabecera de la proforma.
4. La Secretaria selecciona agregar servicios.
5. El Sistema despliega el panel para ingresar servicios.
6. La Secretaria selecciona un servicio de la Lista de Servicios.
7. El Sistema despliega los datos requeridos del servicio seleccionado, como son nombre del servicio, nombre del laboratorio, costo del servicio.
8. El sistema solicita la búsqueda de los métodos asociados al servicio seleccionado mediante el código del servicio.
 - f. El sistema despliega nombres de Métodos asociados al servicio.
9. La Secretaria selecciona el método a usar en la realización de los servicios solicitado.
10. El sistema consulta el código del método de acuerdo al nombre seleccionado.
11. La Secretaria ingresa la cantidad de servicios que requiere el cliente.
12. El sistema calcula el total del servicio.

13. La Secretaria seleccionar la opción guardar.
14. El sistema almacena los datos del servicio ingresado en el detalle Proforma.
15. El sistema calcula el subtotal de la proforma sumando el total del servicio de todos los servicios agregados a detalle.
16. El sistema calcula el IVA de la proforma de acuerdo al subtotal obtenido.
17. El sistema calcula el total de la proforma sumando el subtotal e IVA de la proforma.
18. La Secretaria selecciona la opción guardar proforma
 - a. El sistema valida que hayan servicios agregados al detalle proforma.
 - b. El Sistema valida que los campos requeridos hayan sido ingresados.
 - c. El Sistema valida el tipo de dato ingresado en representante.
19. El Sistema guarda Proforma y emite un mensaje de éxito "Datos almacenados con éxito."

2. **Flujos Alternativos**

1. En el punto 8.a.- El sistema no despliega los nombres de Métodos asociados al servicio seleccionado, porque no se han registrado métodos para ese servicio. El caso de uso termina porque método es un campo obligatorio.
2. En el punto 18.b.- El sistema emite un mensaje de error "Ingrese detalle", si no se ha ingresado servicios en el detalle proforma.
3. En el punto 18.c.- El sistema emite un mensaje de error "El representante no es una cadena de caracteres", en caso de que el nombre del representante ingresado no sea una cadena de caracteres válidos.

3. Precondiciones

1. La Secretaria ha ingresado correctamente el login y password para ingresar al sistema.
2. La Secretaria ha seleccionado la interfaz gráfica "Proforma" de su menú correspondiente.
3. Deben existir Clientes ingresados en la base de datos.
4. Deben existir Servicios ingresados en la base de datos.

5. Deben existir Métodos ingresados y asociados al Servicio seleccionado en la base de datos.
4. Poscondiciones
 1. El estado de la proforma debe ser modificado a “Aceptada” para la generación de facturas, puesto que al momento de registrar una proforma el estado ingresa como “Pendiente” por default.

Registrar Muestra (Caso de uso CU17)

1. Descripción

Este caso de uso especifica el registro de una Muestra, que tiene por objetivo el ingreso de los servicios con sus respectivos métodos con los cuales se analizará la muestra.

La Secretaria es la persona encargada de registrar en el sistema, las Muestras con sus respectivos servicios. El código de cada Muestra se autogenerará el momento de almacenar el mismo.

2. Flujo de Eventos

1. **Flujo Básico**

1. La Secretaria selecciona una Factura Cancelada.
2. El sistema consulta los Servicios que tiene la Factura seleccionada.
3. La Secretaria ingresa los datos (requeridos: nombre, fecha; opcionales: descripción, tipo de muestra, origen, tipo de envase, preservante, tipo de preservante, refrigeración).
4. La Secretaria selecciona la opción Agregar Detalle.
5. El sistema muestra la ventana con los servicios consultados, La Secretaria selecciona los servicios que se realizaran a la muestra.
6. La Secretaria selecciona la opción guardar Muestra.
 - a. El sistema valida que todos los campos requeridos estén llenos.
 - b. El sistema valida que la Muestra tenga servicios ingresados en su Detalle.
 - c. El sistema valida que todos los datos ingresados sean correctos.
 - d. El sistema guarda el Muestra y emite un mensaje de éxito “Datos almacenados con éxito”.

2. **Flujos Alternativos**

1. En el punto 6.a.- El sistema coloca una X en los datos requeridos faltantes.
 2. En el punto 6.b.- El sistema emite un mensaje de error “Ingrese Detalle de Muestra ”
 3. En el punto 6.c.- Si los datos ingresados son incorrectos el sistema emite un mensaje de error indicando el dato que es incorrecto.
3. Precondiciones
1. La Secretaria ha ingresado correctamente el login y password para ingresar al sistema.
 2. La Secretaria ha seleccionado en el menú correspondiente la opción “Muestra”.
 3. Deben existir Facturas con estado Cancelada ingresadas en la base de datos.
4. Poscondiciones
1. Las muestras registradas se utilizaran al generar una Orden de Trabajo.

Caso de uso Consultar Facturas Canceladas (Caso de uso CU18)

1. Descripción

Este caso especifica la consulta de las facturas que han sido canceladas por el cliente, de modo que esto permitirá la generación de Órdenes de Trabajo.

La Secretaria es la persona encargada de consultar las facturas q han sido Facturadas.
2. Flujo de Eventos
 1. **Flujo Básico**
 1. La Secretaria consulta las facturas que han sido canceladas.
 2. **Flujos Alternativos**
 1. Ninguno
3. Precondiciones
 1. La Secretaria ha ingresado correctamente el login y password para ingresar al sistema.

2. La Secretaria ha seleccionado la interfaz gráfica “Orden Trabajo” de su menú correspondiente.
4. Poscondiciones
 1. Las facturas consultas podrán ser usadas para la generación de Ordenes de Trabajo.

Generar Informe Resultados (Caso de uso CU19)

1. Descripción

Este caso de uso especifica la generación de la Orden de Trabajo, que tiene por objetivo el registro de las muestras con el respectivo servicio y método a ser realizado.

La Secretaria es la persona encargada de registrar en el sistema, las órdenes de trabajo por cada factura cancelada y que contenga muestras asociadas. El código de cada Orden de Trabajo se autogenerará el momento de almacenar el mismo.

2. Flujo de Eventos

1. **Flujo Básico**

1. La Secretaria selecciona una Factura cancelada de la cual se va a generar la Orden de Trabajo.
 - a. El sistema muestra los datos del Cliente al cual va dirigida y las muestras con sus respectivos servicios a realizar.
2. La secretaria ingresa la fecha de emisión de la Orden de Trabajo, la fecha de entrega de los Informes de Resultado y el número de muestras que se registran en la orden.
3. Una vez asignada el responsable por muestra (CU20), la Secretaria selecciona la opción guardar Orden Trabajo.
 - a. El sistema valida que exista el detalle de la Orden de Trabajo.
 - b. El Sistema valida que los campos requeridos hayan sido ingresados.
 - c. El Sistema guarda Orden de Trabajo y emite un mensaje de éxito “Datos almacenados con éxito.”

2. **Flujos Alternativos**

1. En el punto 1.a.- Si la Factura seleccionada ya tiene una Orden Generada el sistema despliega un mensaje de error “Factura ya tiene Orden de Trabajo”.
 2. En el punto 3.a.- Si no existen muestras ingresadas el sistema emite el mensaje “Ingrese Detalle Orden Trabajo”.
 3. En el punto 3.b.- El sistema señala con una X los campos requeridos que no están llenos.
3. Precondiciones
1. La Secretaria ha ingresado correctamente el login y password para ingresar al sistema.
 2. La Secretaria ha seleccionado en el menú correspondiente la opción “Orden de Trabajo”.
 3. Deben existir Facturas canceladas con muestras asociadas ingresadas en la base de datos.
4. Poscondiciones
1. Se deben generar informes de resultados por cada muestra de la orden de trabajo seleccionada.

Asignar Carga Trabajo (Caso de uso CU20)

1. Descripción

Este caso de uso especifica la asignación de Carga de Trabajo, que tiene por objetivo asignar un analista para la realización de un servicio a una muestra dada.

La Secretaria es la persona encargada de registrar en la Orden de Trabajo, los analistas que deben realizar el análisis con su respectiva hora de duración con una fecha de inicio, otra de finalización de la realización del análisis.
2. Flujo de Eventos
 1. **Flujo Básico**
 1. La secretaria selecciona una muestra a la cual le va a asignar un analista para ello selecciona el link asignar responsable.
 2. La Secretaria selecciona el analista quien va a realizar el análisis de la respectiva muestra.

3. El sistema muestra el Analista seleccionado.
4. La secretaria ingresa los datos (requeridos: horas/t, fecha de inicio, fecha fin).
5. La secretaria selecciona el botón aceptar.
 - a. El sistema valida que todos los campos estén llenos
 - b. El sistema guarda el analista asignado a la muestra respectiva.

2. **Flujos Alternativos**

1. En el punto 3.a.- El sistema coloca una X en los datos requeridos faltantes.
3. Precondiciones
 1. El analista ha ingresado correctamente el login y password para ingresar al sistema.
 2. La Secretaria debe estar generando una Orden de Trabajo.
 4. Poscondiciones
 1. La secretaria debe guardar la asignación de personal junto a la Orden de Trabajo.

Generar Informe Resultados (Caso de uso CU21)

1. Descripción

Este caso de uso especifica la generación del Informe de Resultados, que tiene por objetivo el registro de los resultados de una muestra seleccionada de una Orden de Trabajo dada en un Informe de Resultados que será entregado al Cliente.

El analista es la persona encargada de registrar en el sistema, los Informes de Resultados generados por muestra de una Orden de Trabajo. El código de cada Informe de Resultados se autogenerará el momento de almacenar el mismo.
2. Flujo de Eventos
 1. **Flujo Básico**
 1. El analista selecciona una Orden de Trabajo de la cual se va a generar el Informe de Resultados.
 2. El sistema despliega la fecha con la cual se guardará el informe y los datos del Cliente al cual va dirigido.

3. El analista selecciona una muestra perteneciente a la Orden de Trabajo seleccionada, a la cual se le ingresarán los respectivos resultados.
 - a. El sistema devuelve los datos de la muestra seleccionada con sus respectivos servicios y métodos a ser realizados a dicha muestra.
4. El analista selecciona la opción ingresar resultados.
5. El analista selecciona la unidad de medida y la norma.
6. El analista ingresa los datos (requeridos: resultado, límite máximo permisible).
7. El analista selecciona la opción aceptar.
 - a. El sistema valida que todos los campos requeridos estén llenos
 - b. El sistema guarda los resultados ingresados al respectivo servicio.
8. El analista selecciona la opción guardar Informe de Resultados.
 - a. El sistema valida que exista resultados asignados al detalle del Informe de Resultados.
 - b. El Sistema valida que los campos requeridos hayan sido ingresados.
 - c. El Sistema guarda Informe de resultados y emite un mensaje de éxito "Datos almacenados con éxito."

2. **Flujos Alternativos**

1. En el punto 3.a.- El sistema no despliega los datos de la muestra y los servicios asociados cuando la muestra ya tiene generado un informe de resultados. Emite un mensaje de error "muestra ya tiene Informe de Resultados."
2. En el punto 8.a.- El sistema emite un mensaje de error "Ingrese detalle", si no se ha ingresado servicios en el detalle proforma.
3. En el punto 8.b.- El sistema muestra campos con signos rojos en los campos faltantes.

3. Precondiciones

1. El analista ha ingresado correctamente el login y password para ingresar al sistema.
2. El analista ha seleccionado en el menú correspondiente la opción "Informe de Resultados".
3. Deben existir Órdenes de Trabajo ingresados en la base de datos.

4. Poscondiciones

1. Se deben generar informes de resultados por cada muestra de la orden de trabajo seleccionada.

Caso de uso Consultar Proforma Aceptadas (Caso de uso CU22)

1. Descripción

Este caso especifica la consulta de las proformas que han sido aceptadas por el cliente, de modo que esto permitirá la generación de Facturas.

La Tesorera es la persona encargada de consultar las proformas q han sido Aceptadas.

2. Flujo de Eventos

1. **Flujo Básico**

0. La Tesorera consulta las proformas que han sido aceptadas.

2. **Flujos Alternativos**

1. Ninguno

3. Precondiciones

1. La Tesorera ha ingresado correctamente el login y password para ingresar al sistema.
 2. La Tesorera ha seleccionado la interfaz gráfica "Factura" de su menú correspondiente.

4. Poscondiciones

1. Las proformas consultas podrán ser usadas para la generación de Factura.

Caso de uso Generar Factura (Caso de uso CU23)

1. Descripción

Este caso especifica la generación de una factura, seleccionar una de las proformas que han sido consultadas y que se encuentra en estado "Aceptada".

La Tesorera es la persona encargada de generar una Factura a partir de una proforma aceptada.

2. Flujo de Eventos

1. **Flujo Básico**

1. La tesorera selecciona una de las proformas aceptadas para la generación de la Factura.
2. El sistema despliega un panel con los datos de la Proforma selecciona y la fecha actual para la generación de la Factura.
3. La Tesorera selecciona guardar Factura.
4. El Sistema guarda la Factura y emite un mensaje de éxito “Datos almacenados con éxito.”

2. Flujos Alternativos

1. *Ninguno*
3. Precondiciones
 1. La Tesorera ha ingresado correctamente el login y password para ingresar al sistema.
 2. La Tesorera ha seleccionado la interfaz gráfica “Factura” de su menú correspondiente.
 3. Deben existir proformas con estado Aceptada.
4. Poscondiciones
 1. Las Facturas generadas serán usadas para la generación de Muestra y Orden de Trabajo.

2.2. ANÁLISIS

La etapa de análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales para determina los objetivos y límites del sistema, caracterizar su estructura y funcionamiento, marcar las directrices que permitan alcanzar los objetivos propuestos y evaluar sus consecuencias, para lo cual se elaboran diagramas de actividad y secuencia.

2.2.1. DIAGRAMAS DE SECUENCIA

Los diagramas de secuencia muestran la iteración de un conjunto de objetos en una aplicación, los cuales se comunican entre sí al transcurrir el tiempo, y el orden secuencial de las iteraciones se modela para cada operación de la clase. El diagrama de secuencias proporciona un camino a partir de los escenarios para describir las operaciones en una forma más detallada. Adicionalmente muestra el intercambio de mensajes, es decir la forma en que se invocan en un momento

dato. Los diagramas de secuencia ponen énfasis en el orden y en el momento en el que se envían los mensajes a los objetos

Los diagramas de secuencias se preparan durante la fase de análisis del ciclo de desarrollo, muestra un determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema.

Los diagramas de secuencias se modelan a nivel de objetos y utilizan tres elementos fundamentales: objetos, mensajes intercambiados entre objetos en una secuencia ordenada y línea de vida de los objetos.

Para el diseño del diagrama de secuencia tenemos:

- Los objetos los cuales están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.
- Los mensajes pueden ser o bien síncronos, el tipo normal de llamada del mensaje donde se pasa el control a objeto llamado hasta que el método finalice, o asíncronos donde se devuelve el control directamente al objeto que realiza la llamada. Los mensajes síncronos tienen una caja vertical en un lateral del objeto invocante que muestra el flujo del control del programa.

A continuación se presentan los diferentes diagramas de secuencia a partir de cada caso de uso del sistema de Gestión de Laboratorios donde se muestra la iteración entre los diferentes objetos a través de mensajes. [23] [50]

[23] MARCOS FOUCES LAGO, Diagramas de Secuencia

[50] WIKIPEDIA, Diagrama de Secuencia

2.2.1.1. Diagrama de Secuencia Ingresar al Sistema

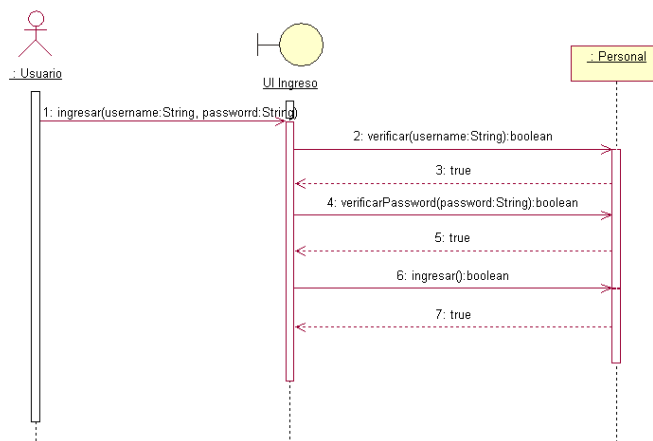


Figura 2. 9 Diagrama de Secuencia Ingresar al Sistema

2.2.1.2. Diagrama de Secuencia Registrar Unidad

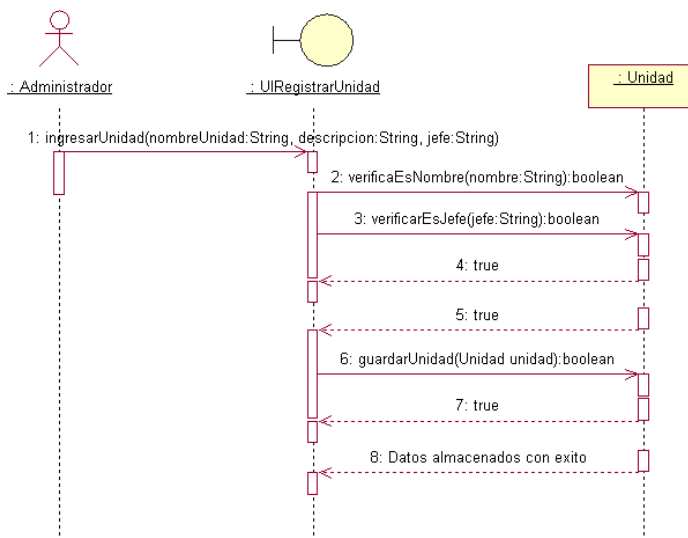


Figura 2. 10 Diagrama de Secuencia Registrar Unidad

2.2.1.3. Diagrama de Secuencia Registrar Tipo Cliente

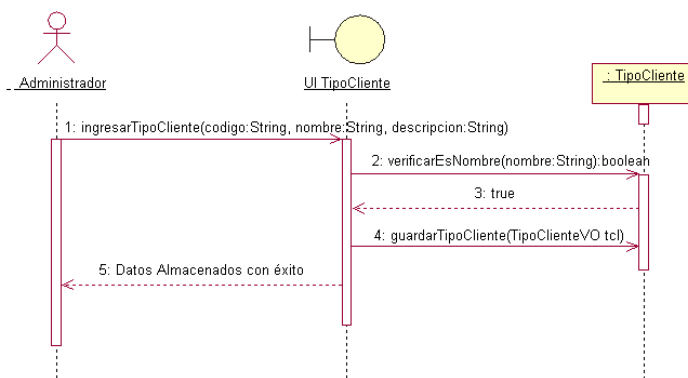


Figura 2. 11 Diagrama de Secuencia Registrar Tipo Cliente

2.2.1.4. Diagrama de Secuencia Registrar Tipo Servicio

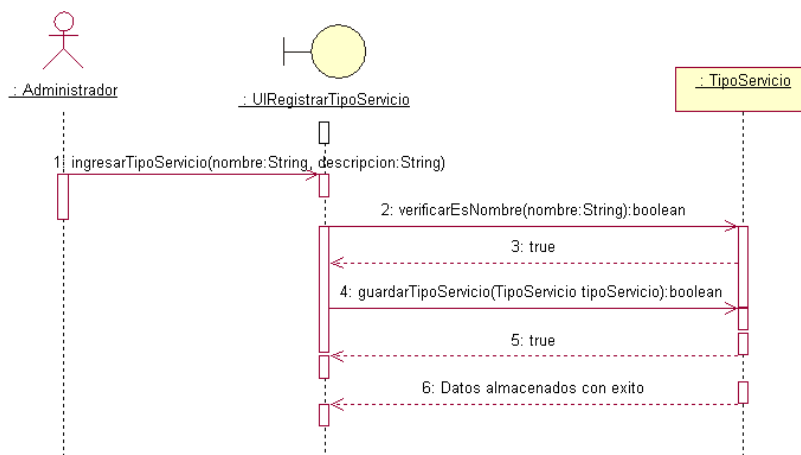


Figura 2. 12 Diagrama de Secuencia Registrar Tipo Servicio

2.2.1.5. Diagrama de Secuencia Registrar Tipo Producto

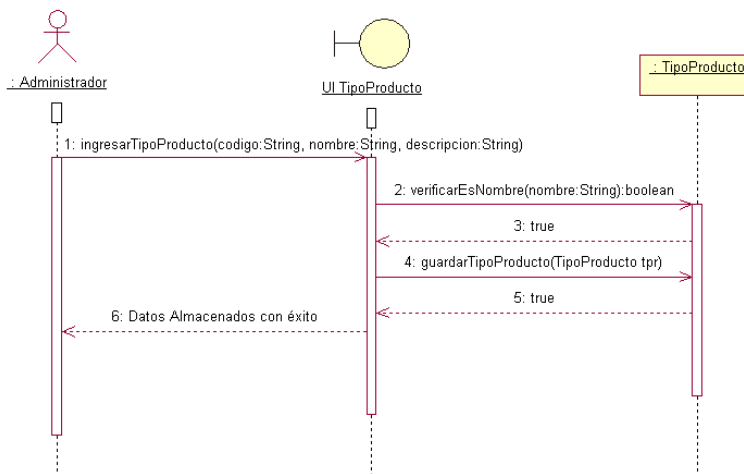


Figura 2. 13 Diagrama de Secuencia Registrar Tipo Producto

2.2.1.6. Diagrama de Secuencia Registrar Tipo Personal

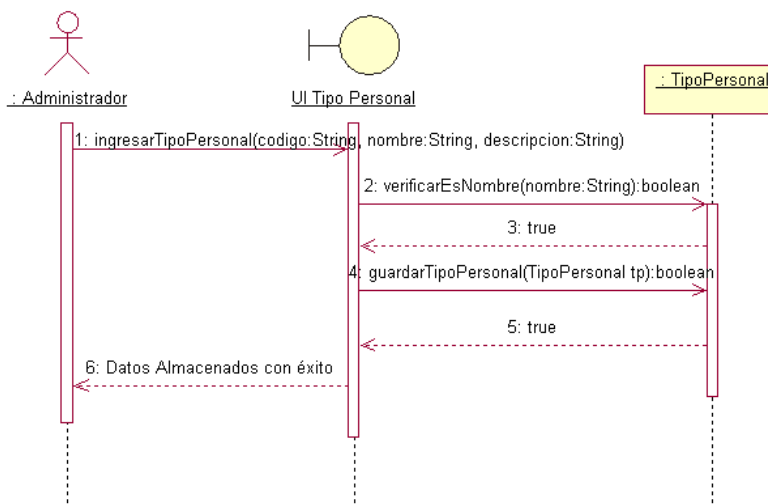


Figura 2. 14 Diagrama de Secuencia Registrar Tipo Personal

2.2.1.7. Diagrama de Secuencia Registrar Cargo Personal

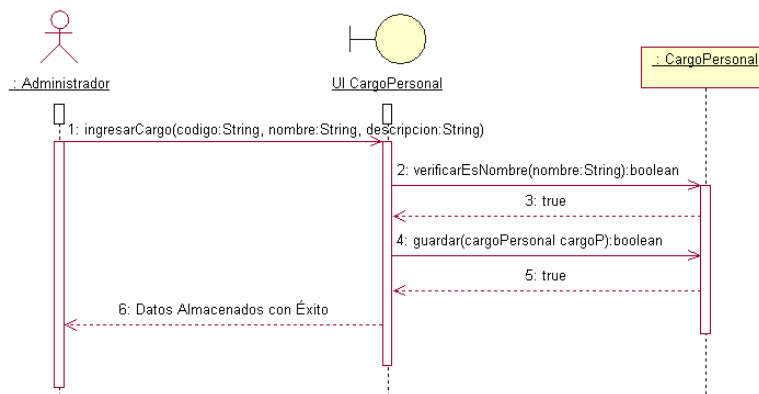


Figura 2. 15 Diagrama de Secuencia Registrar Cargo Personal

2.2.1.8. Diagrama de Secuencia Registrar Laboratorio

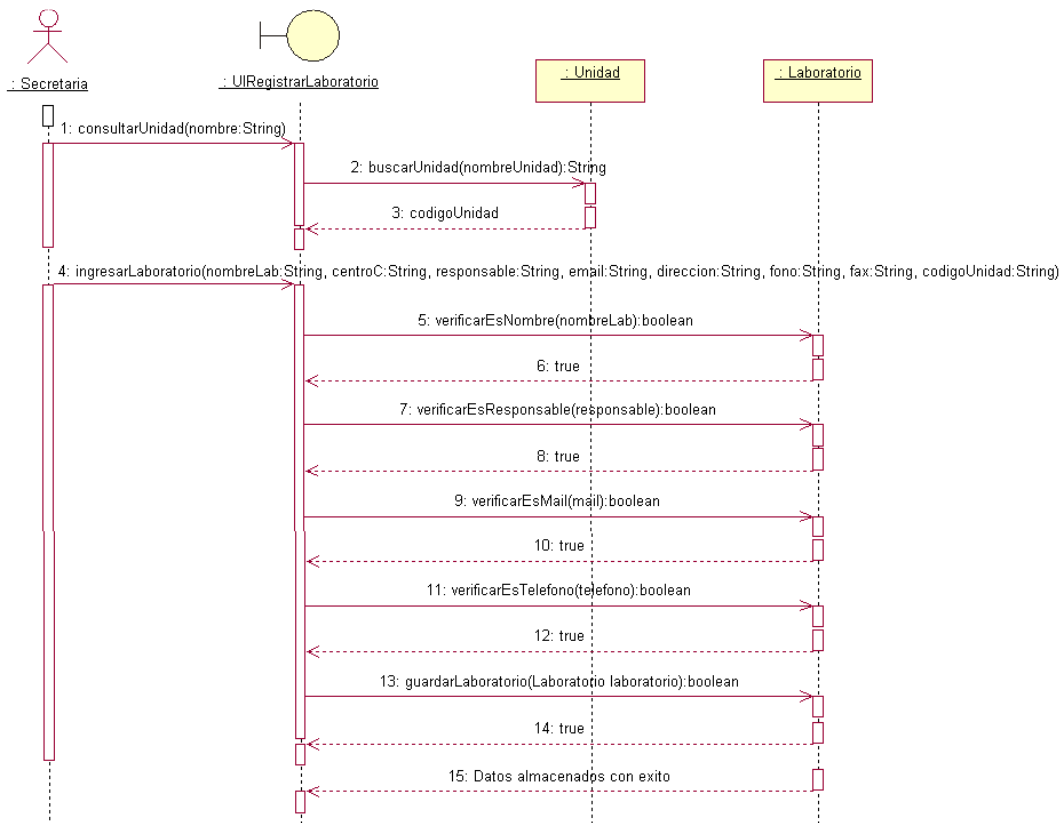


Figura 2. 16 Diagrama de Secuencia Registrar Laboratorio

2.2.1.9. Diagrama de Secuencia Registrar Personal

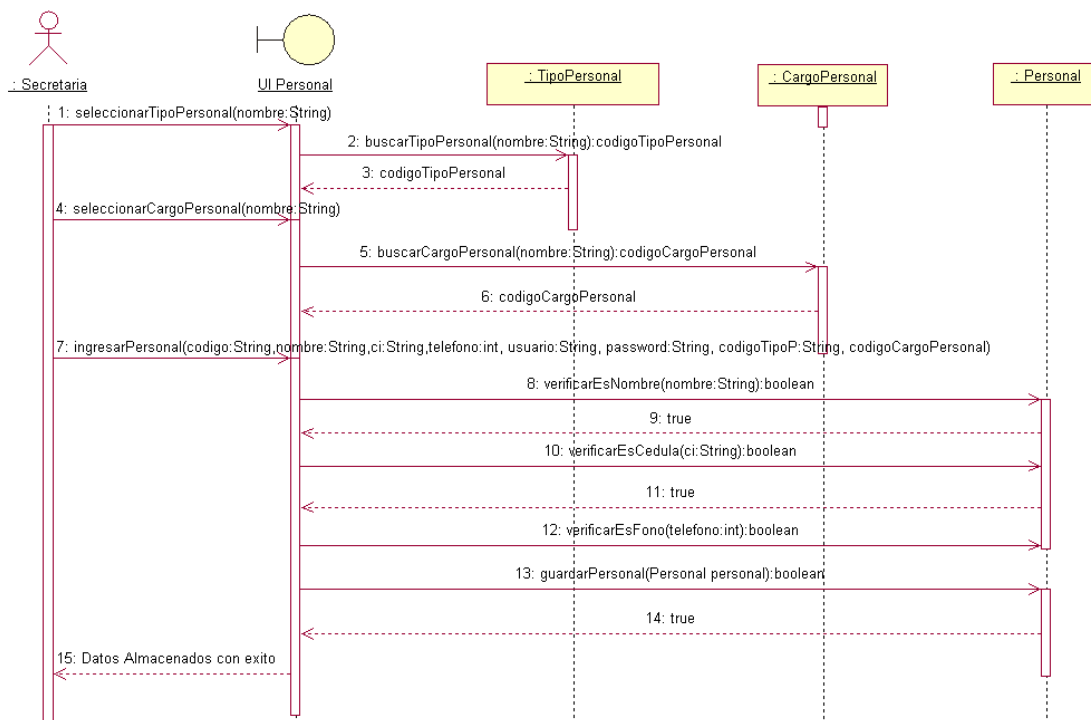


Figura 2. 17 Diagrama de Secuencia Registrar Personal

2.2.1.10. Diagrama de Secuencia Registrar Cliente

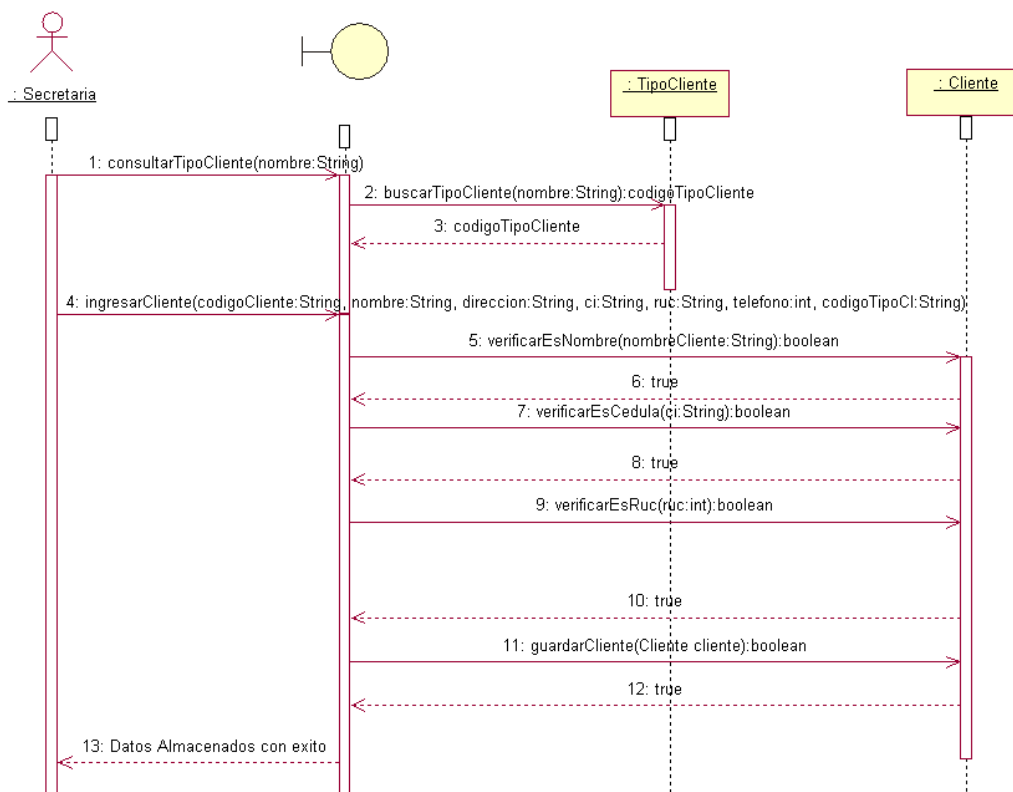


Figura 2. 18 Diagrama de Secuencia Registrar Cliente

2.2.1.11. Diagrama de Secuencia Registrar Unidad de Medida

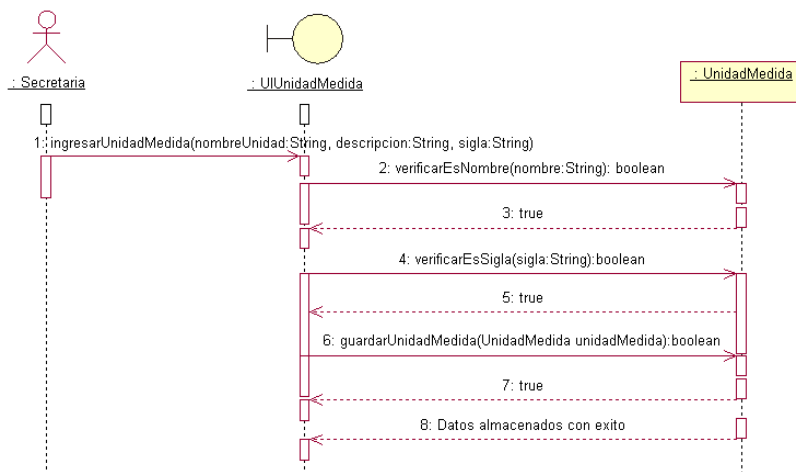


Figura 2. 19 Diagrama de Secuencia Registrar Unidad Medida

2.2.1.12. Diagrama de Secuencia Registrar Norma

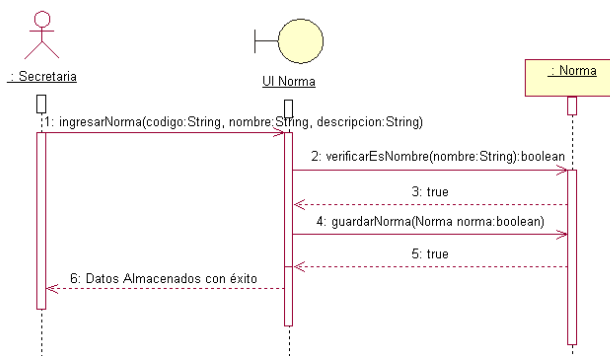


Figura 2. 20 Diagrama de Secuencia Registrar Norma

2.2.1.13. Diagrama de Secuencia Registrar Producto

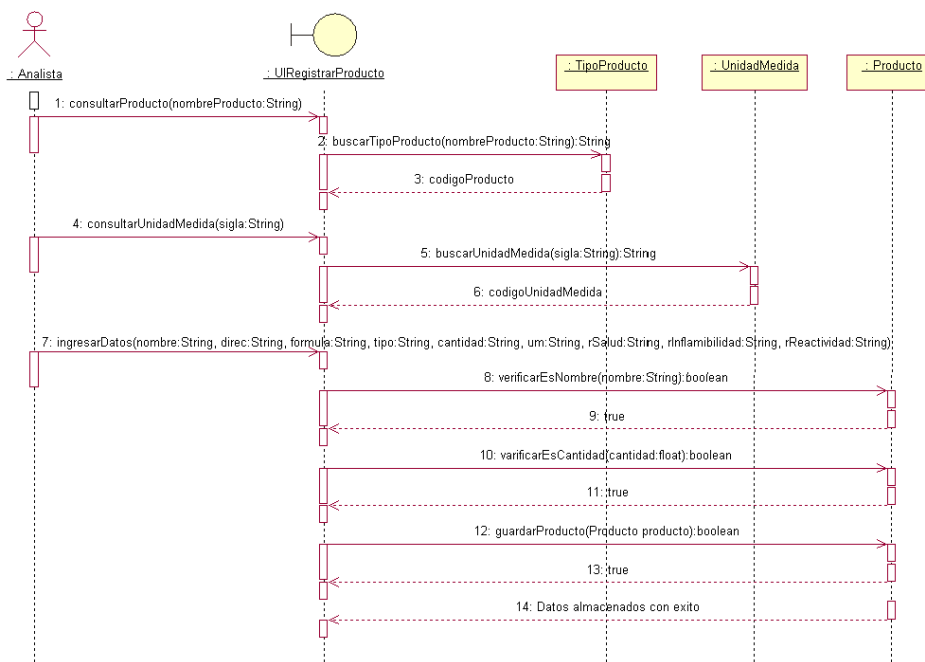


Figura 2. 21 Diagrama de Secuencia Registrar Producto

2.2.1.14. Diagrama de Secuencia Registrar Servicio

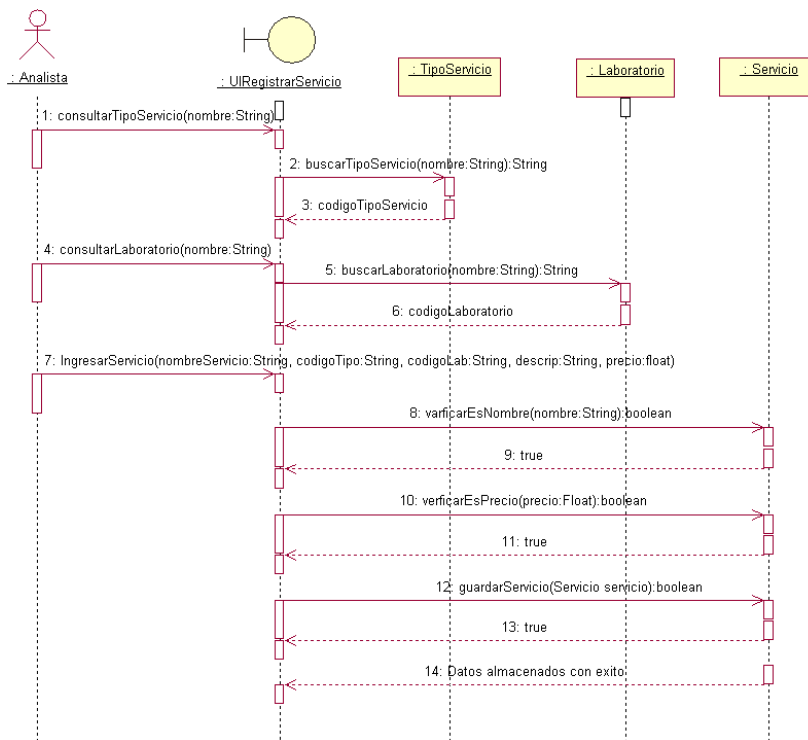


Figura 2. 22 Diagrama de Secuencia Registrar Servicio

2.2.1.15. Diagrama de Secuencia Registrar Método

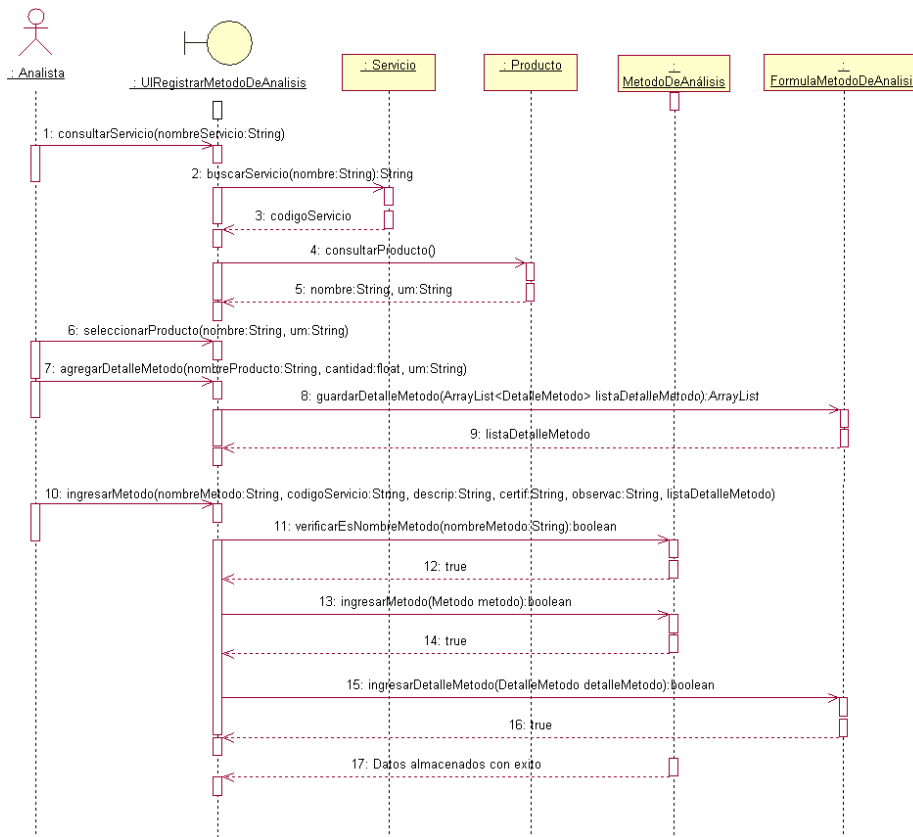


Figura 2. 23 Diagrama de Secuencia Registrar Método

2.2.1.16. Diagrama de Secuencia Generar Proforma

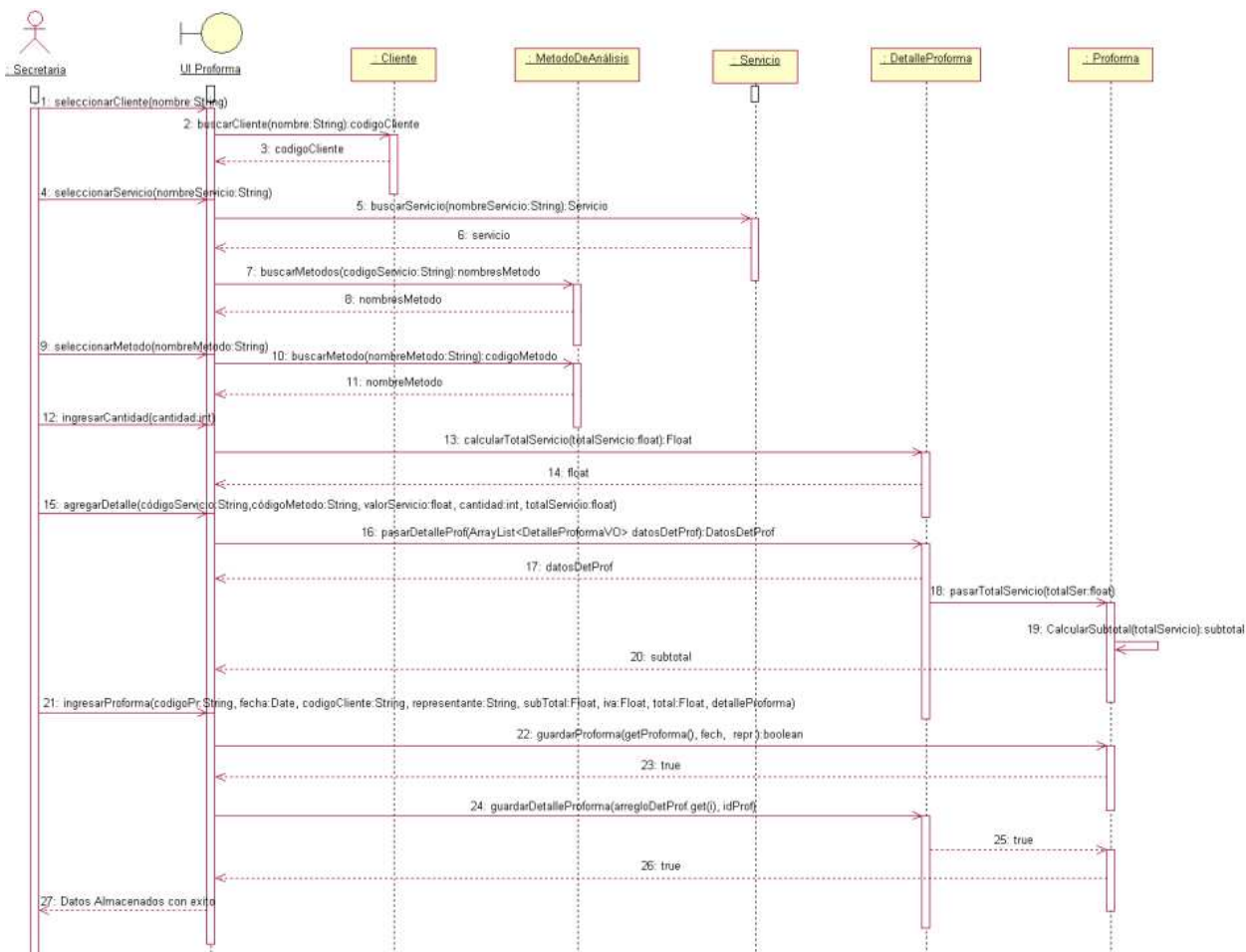


Figura 2. 24 Diagrama de Secuencia Generar Proforma

2.2.1.17. Diagrama de Secuencia Registrar Muestra

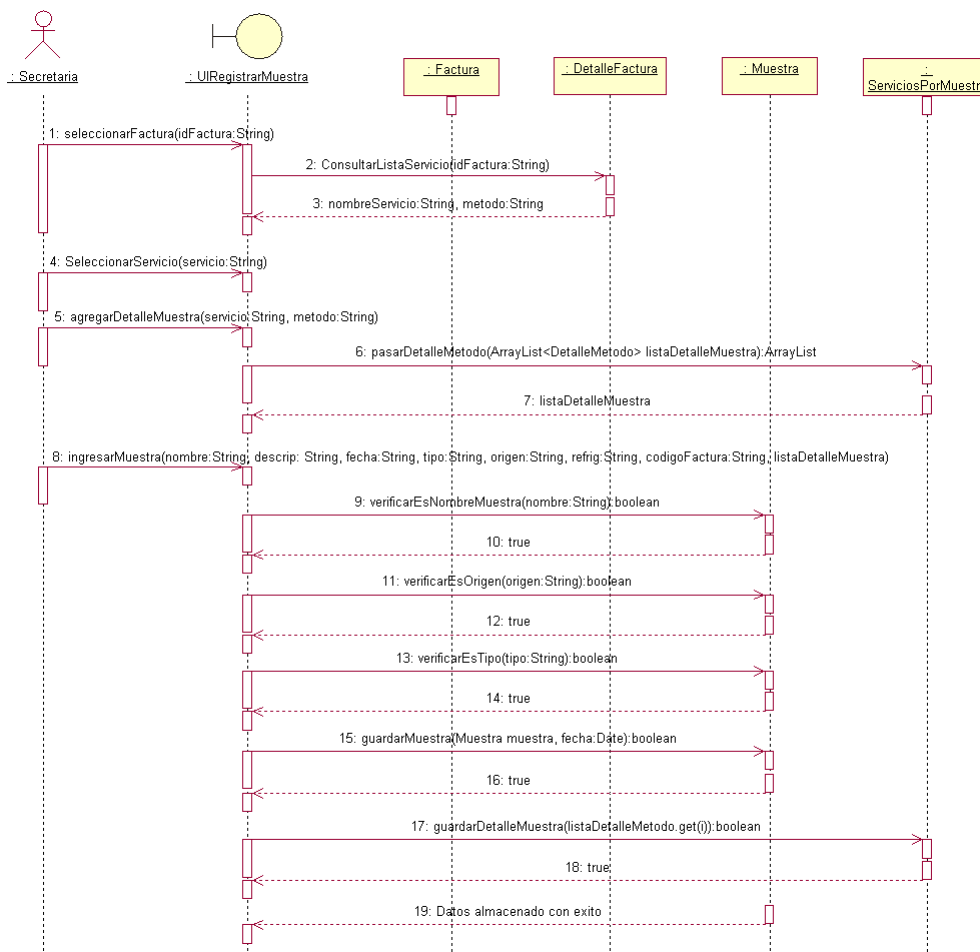


Figura 2. 25 Diagrama de Secuencia Registrar Muestra

2.2.1.18. Diagrama de Secuencia Consultar Facturas Canceladas

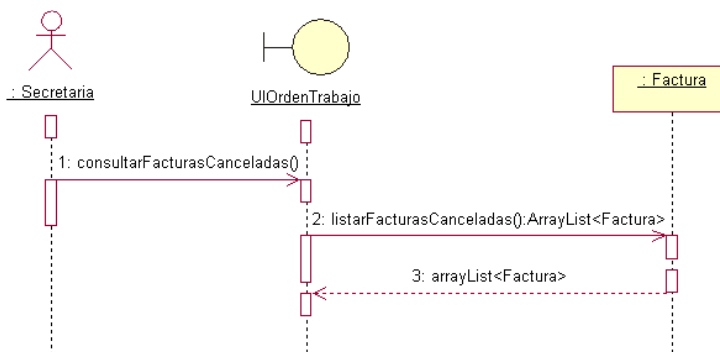


Figura 2. 26 Diagrama de Secuencia Consultar Facturas Canceladas

2.2.1.19. Diagrama de Secuencia Generar Orden de Trabajo

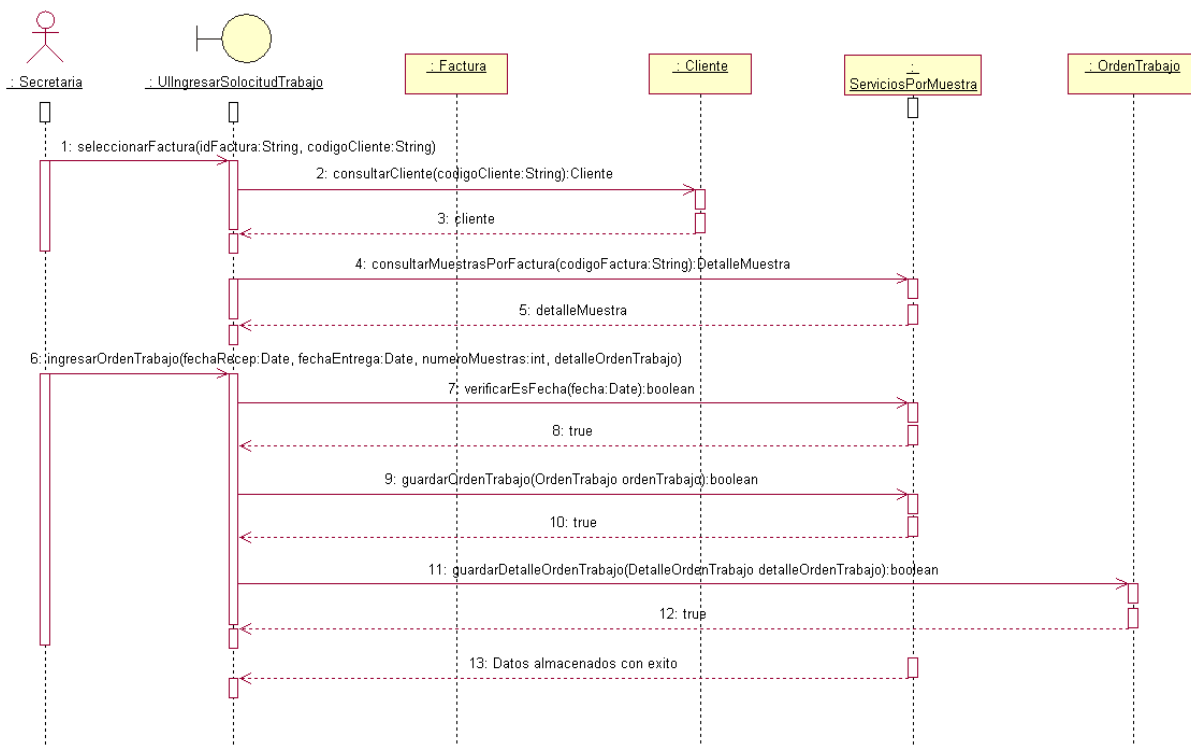


Figura 2. 27 Diagrama de Secuencia Generar Orden Trabajo

2.2.1.20. Diagrama de Secuencia Asignar Carga Trabajo

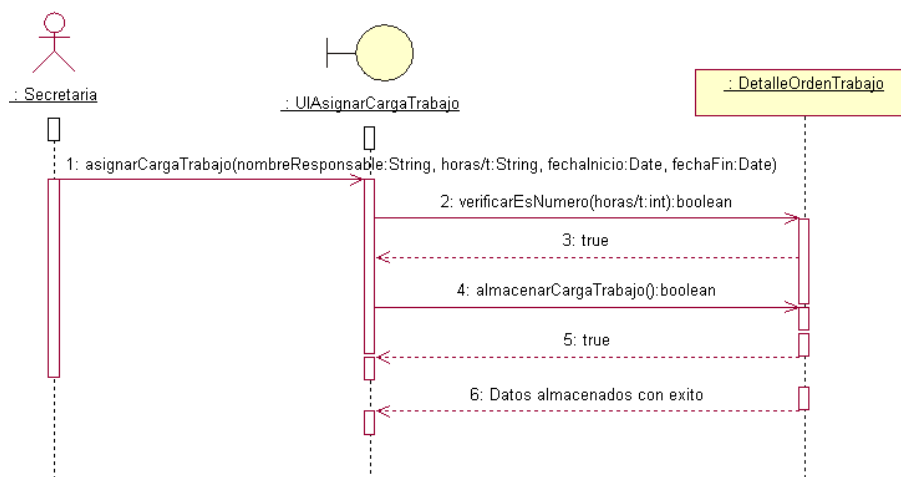


Figura 2. 28 Diagrama de Secuencia Asignar Carga Trabajo

2.2.1.21. Diagrama de Secuencia Generar Informe de Resultados

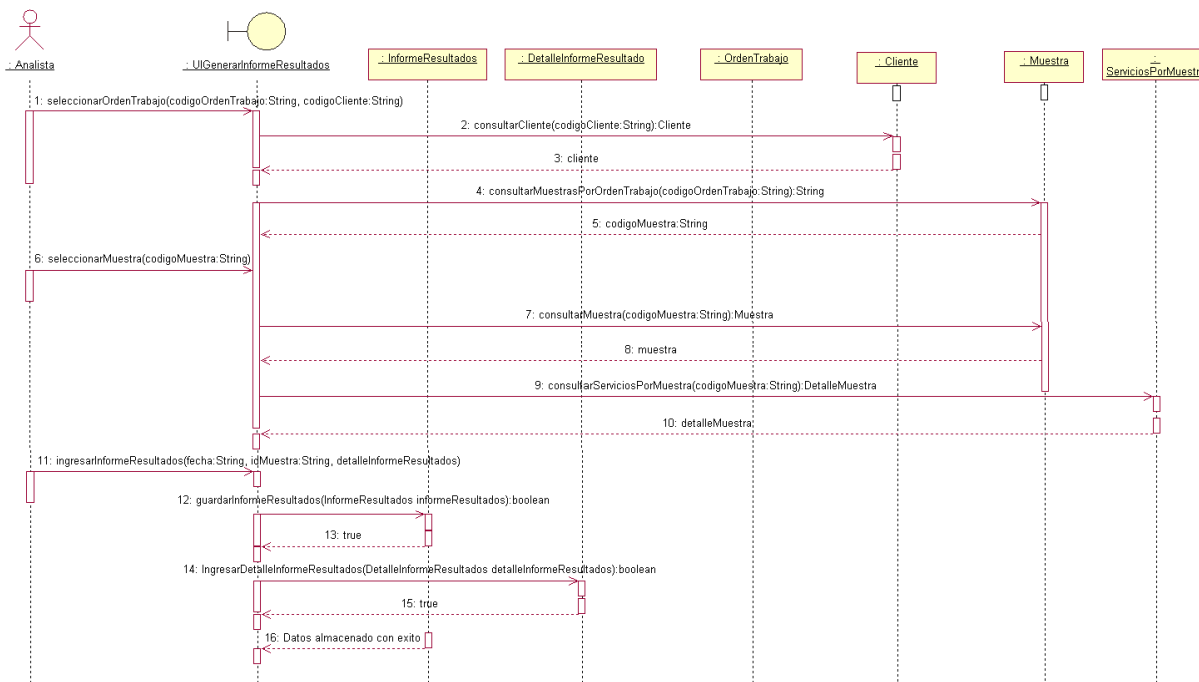


Figura 2. 29 Diagrama de Secuencia Generar Informe Resultados

2.2.1.22. Diagrama de Secuencia Consultar Proformas Aceptadas

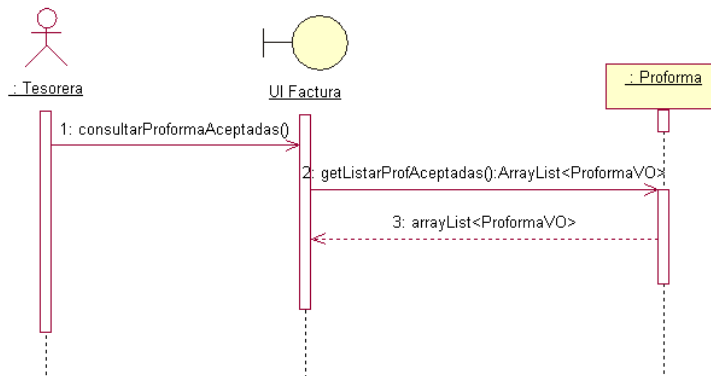


Figura 2. 30 Diagrama de Secuencia Consultar Proformas Aceptadas

2.2.1.23. Diagrama de Secuencia Generar Factura

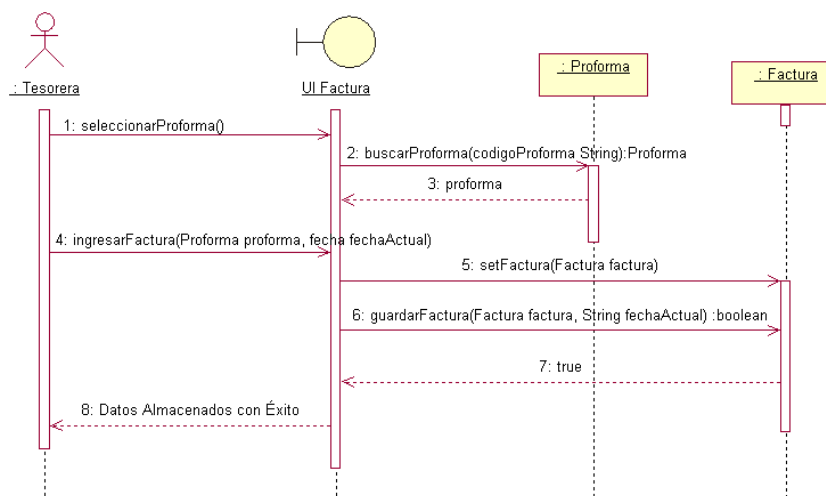


Figura 2. 31 Diagrama de Secuencia Generar Factura

2.2.2. DIAGRAMAS DE ACTIVIDAD

Un diagrama de Actividad muestra la serie de actividades que deben ser realizadas en un caso de uso, así como las distintas rutas que pueden irse desencadenando en el caso de uso, de esta forma se representan los aspectos dinámicos de un sistema, se puede describir procesos o casos de uso, permite elegir el orden en el que puede hacerse las cosas y establece las reglas de secuencia a seguir.

Los diagramas de actividad son similares a los diagramas de flujo procesales, con la diferencia de que todas las actividades están claramente unidas a objetos. Los diagramas de actividad siempre están asociados a una clase, a una operación o a un caso de uso.

Los diagramas de actividad soportan actividades tanto secuenciales como paralelas. La ejecución paralela se representa por medio de iconos de espera, y en el caso de las actividades paralelas, no importa en qué orden sean invocadas, pueden ser ejecutadas simultáneamente o una detrás de otra.

Una actividad es un único paso de un proceso y pueden formar jerarquías, lo que significa que una actividad puede estar formada de varias actividades de detalle,

en cuyo caso las transiciones entrantes y salientes deberían coincidir con las del diagrama de detalle. [33] [28]

2.2.2.1. Diagrama de Actividad Ingresar al Sistema

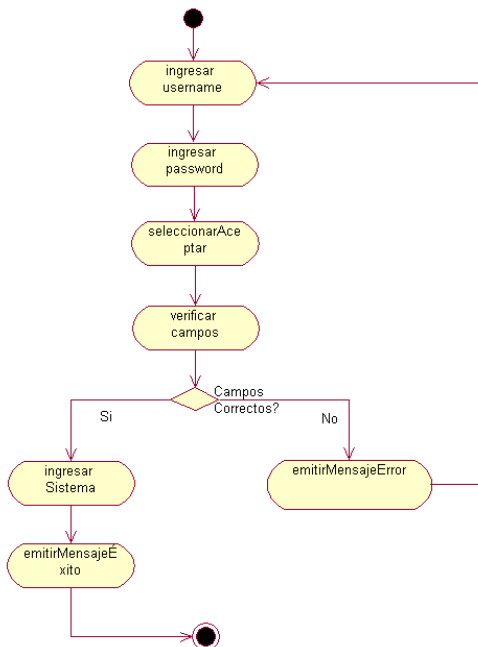


Figura 2. 32 Diagrama de Actividad Ingresar al Sistema

2.2.2.2. Diagrama de Actividad Registrar Unidad

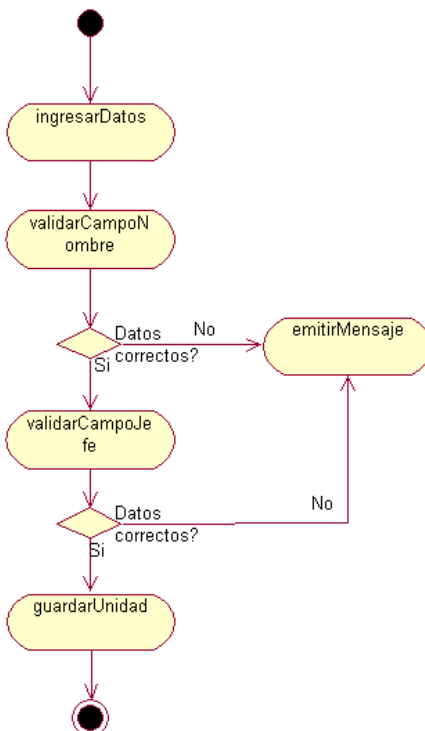


Figura 2. 33 Diagrama de Actividad Registrar Unidad

[33] OSMOSIS LATINA, Diagrama de Actividad
 [28] MARCOS FOUCES LAGO, Diagramas de Actividad

2.2.2.3. Diagrama de Actividad Registrar Tipo Cliente

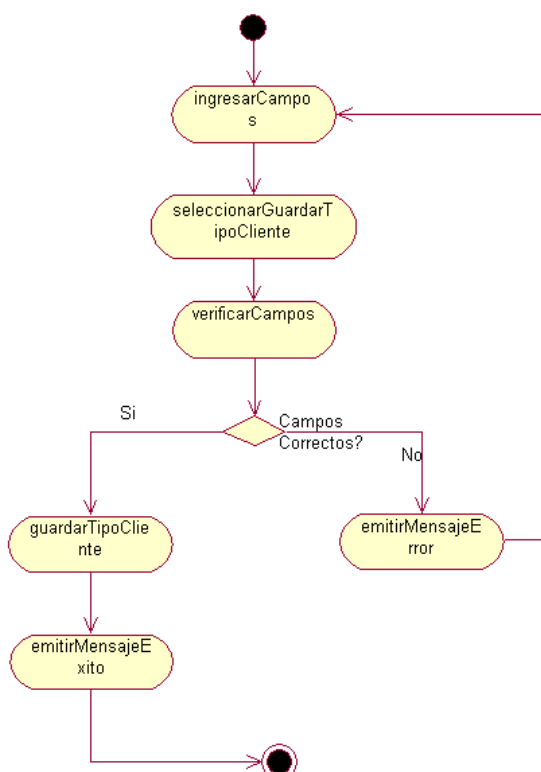


Figura 2. 34 Diagrama de Actividad Registrar Tipo Cliente

2.2.2.4. Diagrama de Actividad Registrar Tipo Servicio

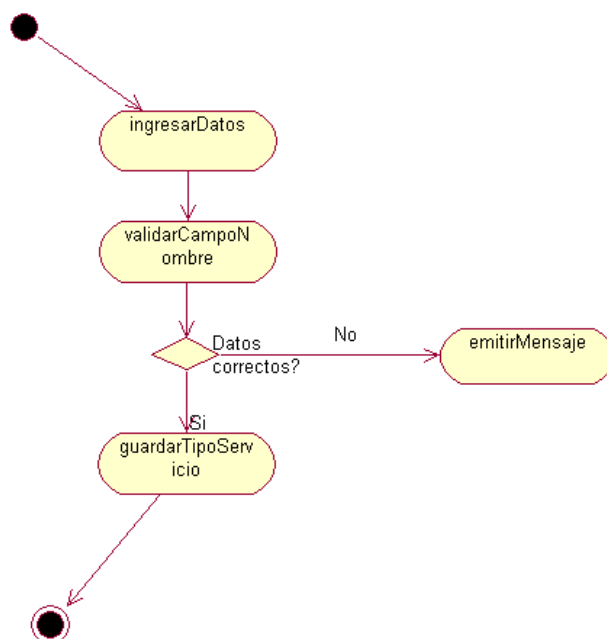


Figura 2. 35 Diagrama de Actividad Registrar Tipo Servicio

2.2.2.5. Diagrama de Actividad Registrar Tipo Producto

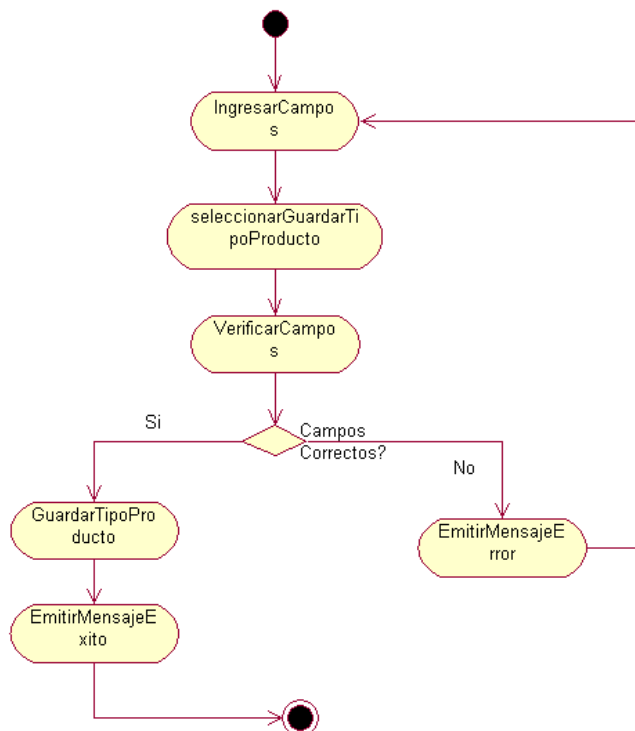


Figura 2. 36 Diagrama de Actividad Registrar Tipo Producto

2.2.2.6. Diagrama de Actividad Registrar Tipo Personal

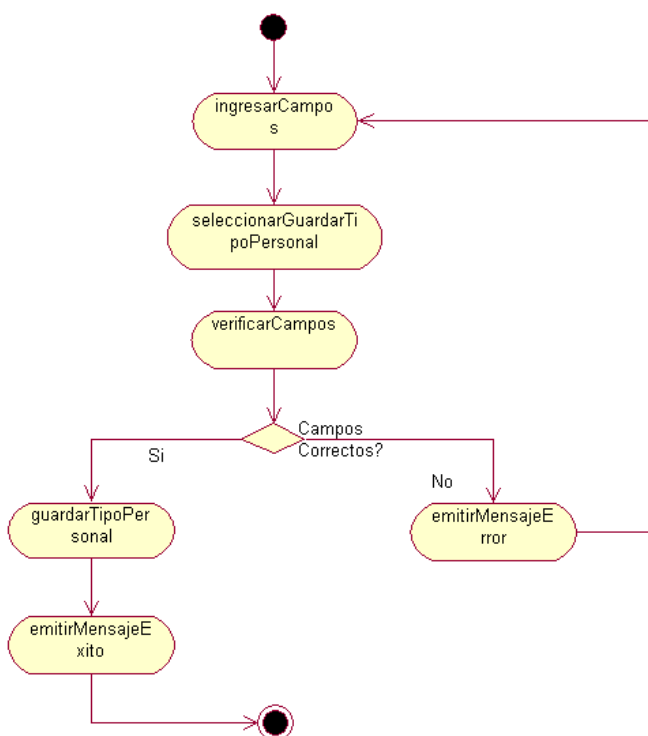


Figura 2. 37 Diagrama de Actividad Registrar Tipo Personal

2.2.2.7. Diagrama de Actividad Registrar Cargo Personal

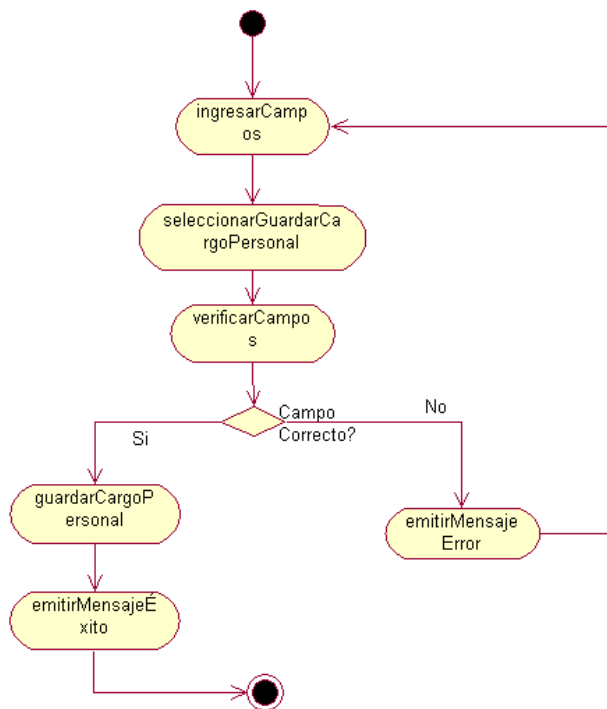


Figura 2. 38 Diagrama de Actividad Registrar Cargo Personal

2.2.2.8. Diagrama de Actividad Registrar Laboratorio

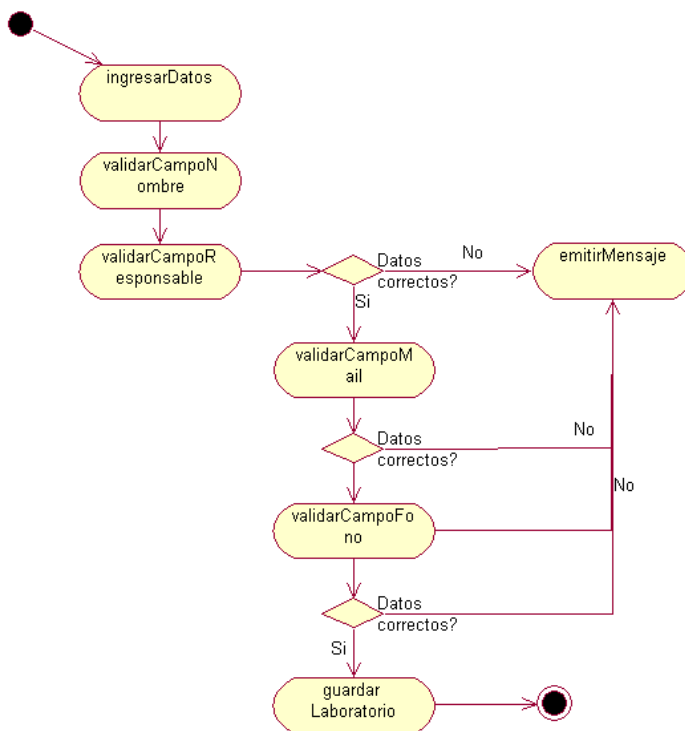


Figura 2. 39 Diagrama de Actividad Registrar Laboratorio

2.2.2.9. Diagrama de Actividad Registrar Personal

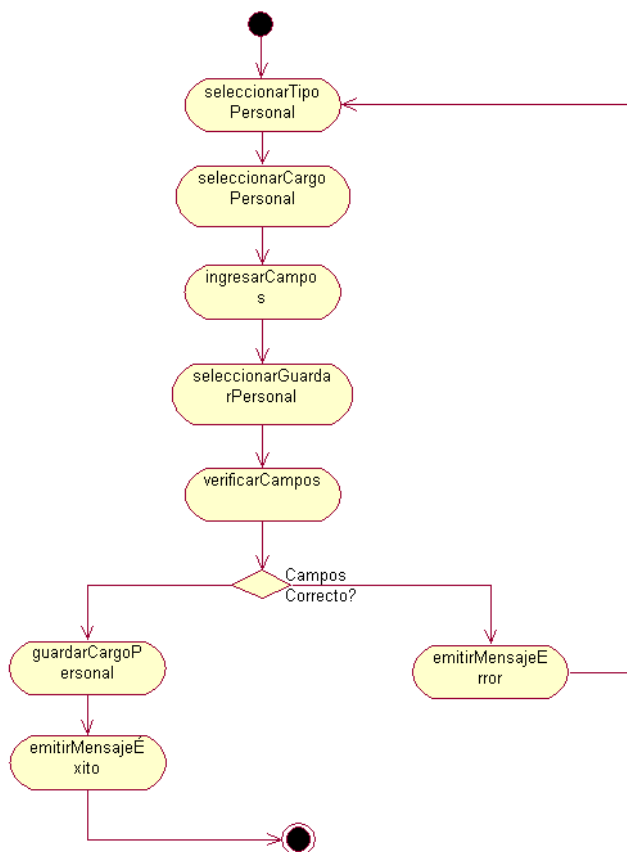


Figura 2. 40 Diagrama de Actividad Registrar Personal

2.2.2.10. Diagrama de Actividad Registrar Cliente

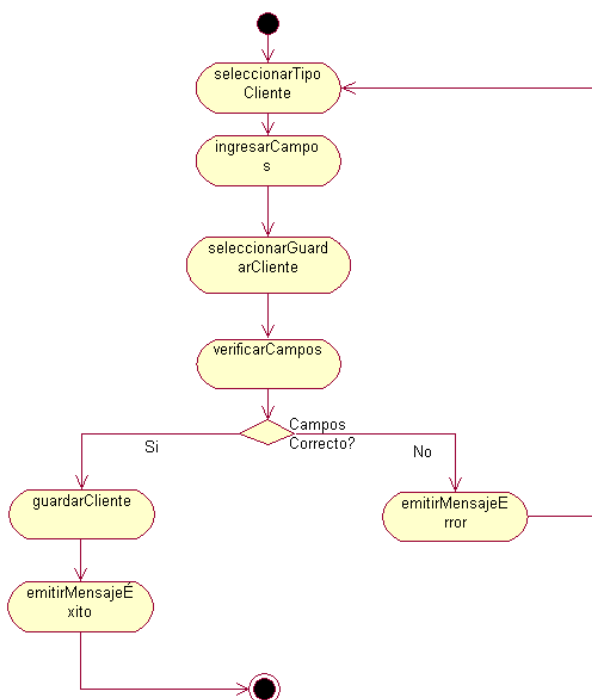


Figura 2. 41 Diagrama de Actividad Registrar Cliente

2.2.2.11. Diagrama de Actividad Registrar Unidad Medida

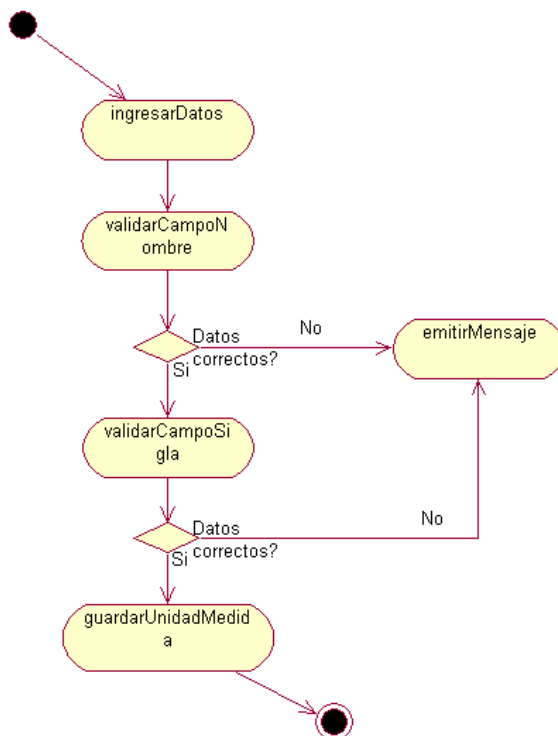


Figura 2. 42 Diagrama de Actividad Registrar Unidad Medida

2.2.2.12. Diagrama de Actividad Registrar Norma

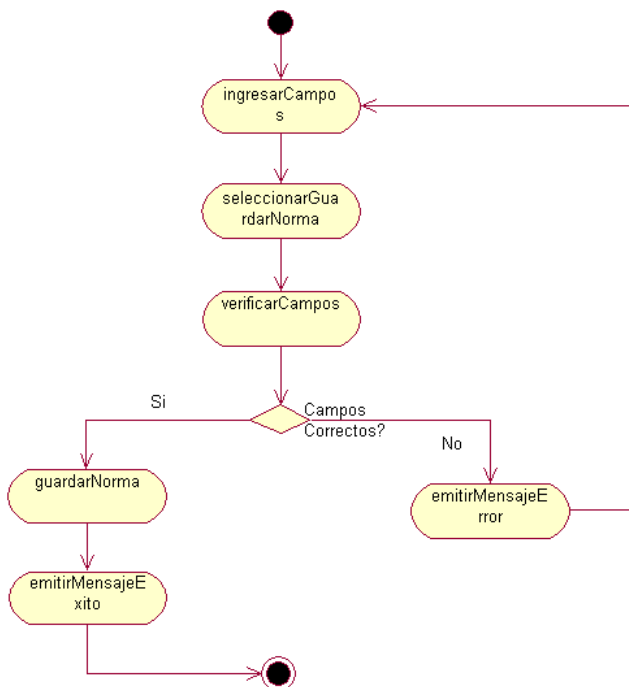


Figura 2. 43 Diagrama de Actividad Registrar Norma

2.2.2.13. Diagrama de Actividad Registrar Producto

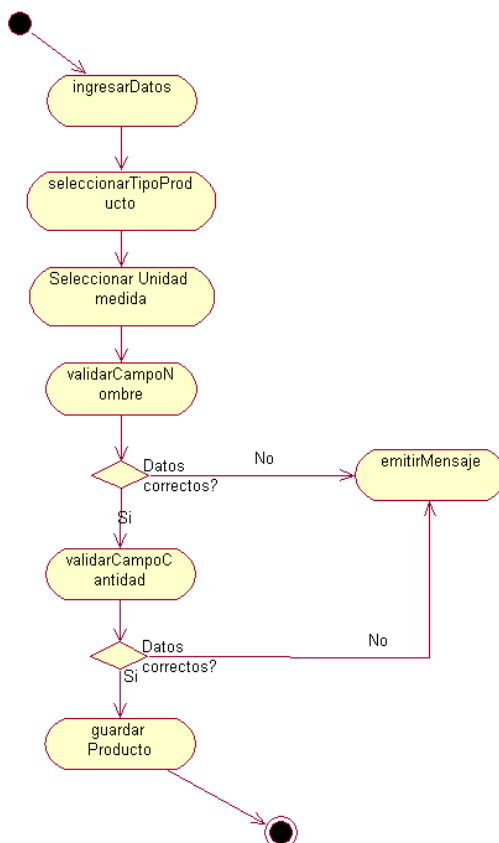


Figura 2. 44 Diagrama de Actividad Registrar Producto

2.2.2.14. Diagrama de Actividad Registrar Servicio

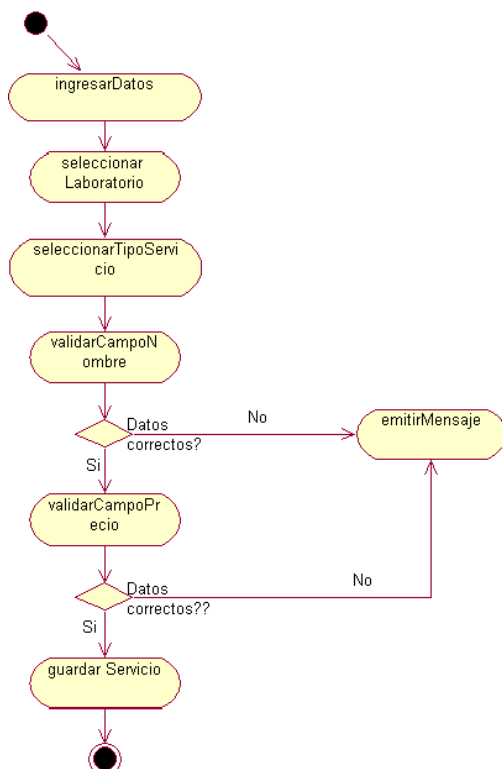


Figura 2. 45 Diagrama de Actividad Registrar Servicio

2.2.2.15. Diagrama de Actividad Registrar Método

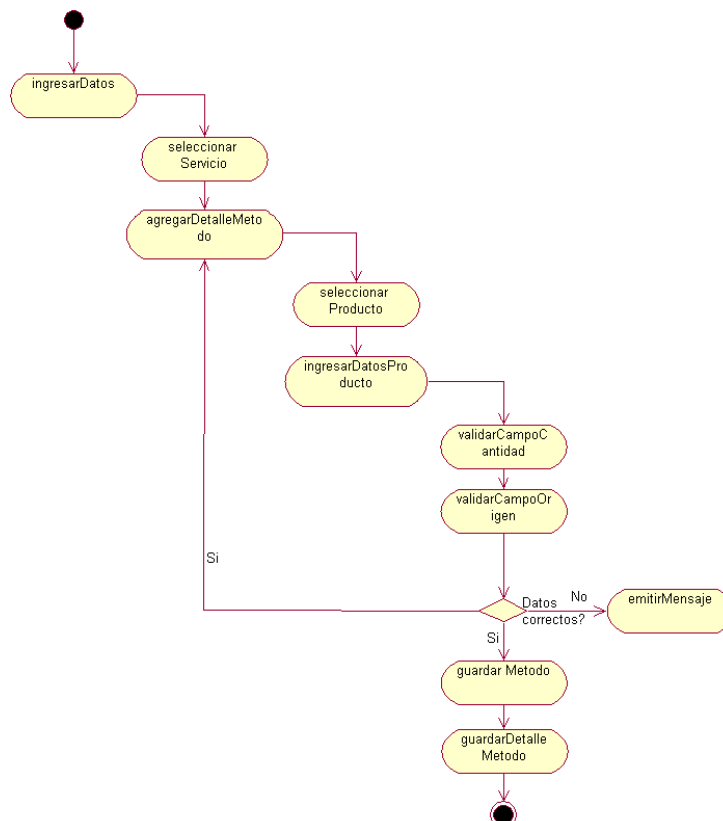


Figura 2. 46 Diagrama de Actividad Registrar Método

2.2.2.16. Diagrama de Actividad Registrar Muestra

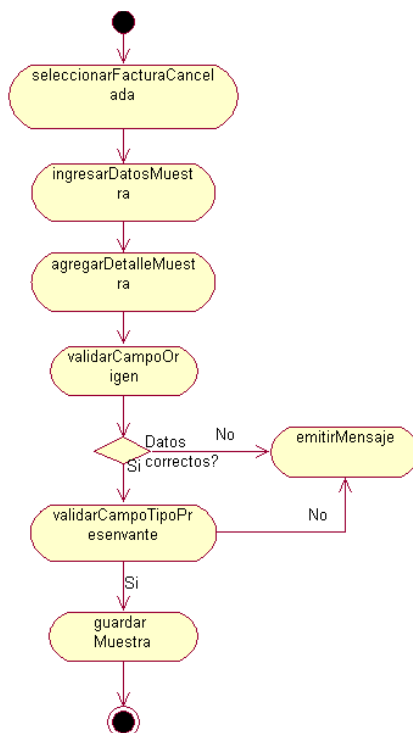


Figura 2. 47 Diagrama de Actividad Registrar Muestra

2.2.2.17. Diagrama de Actividad Generar Proforma

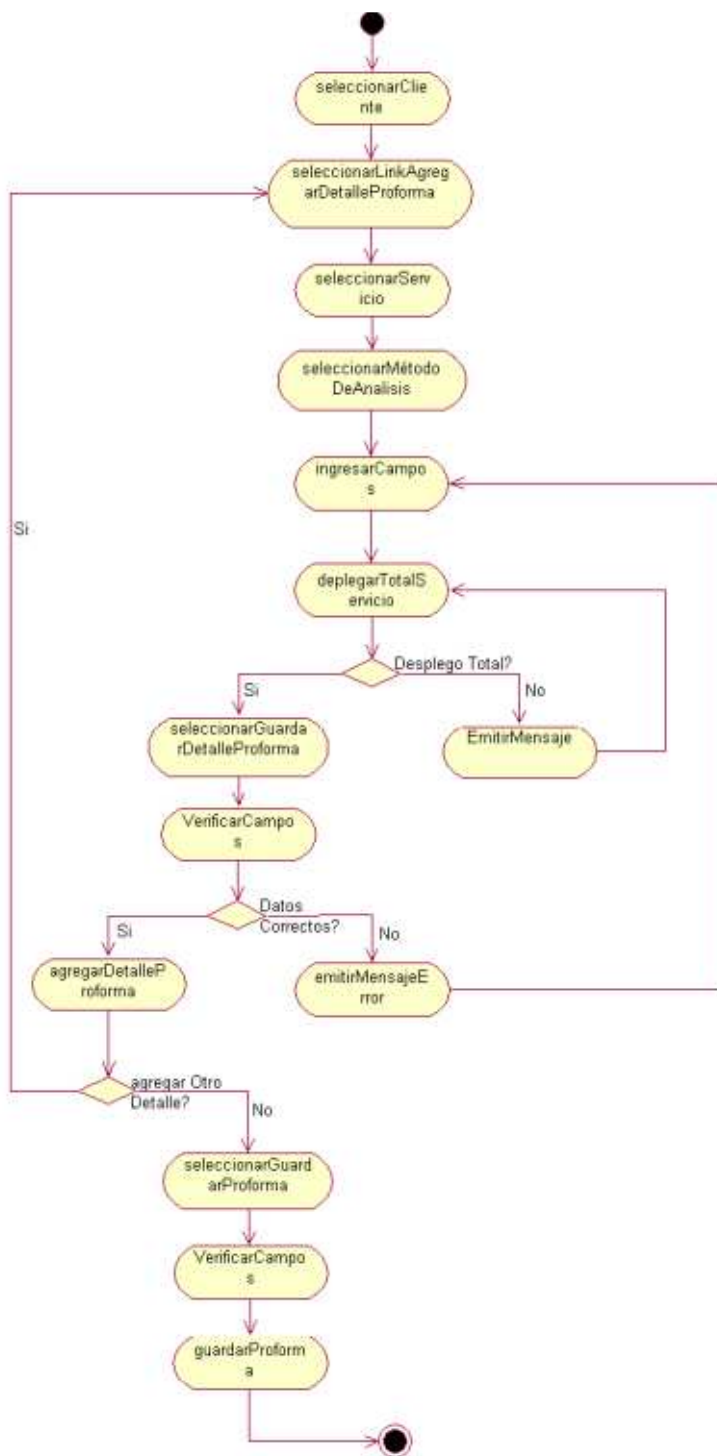


Figura 2. 48 Diagrama de Actividad Generar Proforma

2.2.2.18. Diagrama de Actividad Consultar Facturas Canceladas

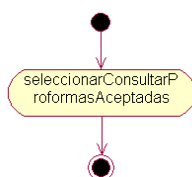


Figura 2. 49 Consultar Facturas Canceladas

2.2.2.19. Diagrama de Actividad Generara Orden Trabajo

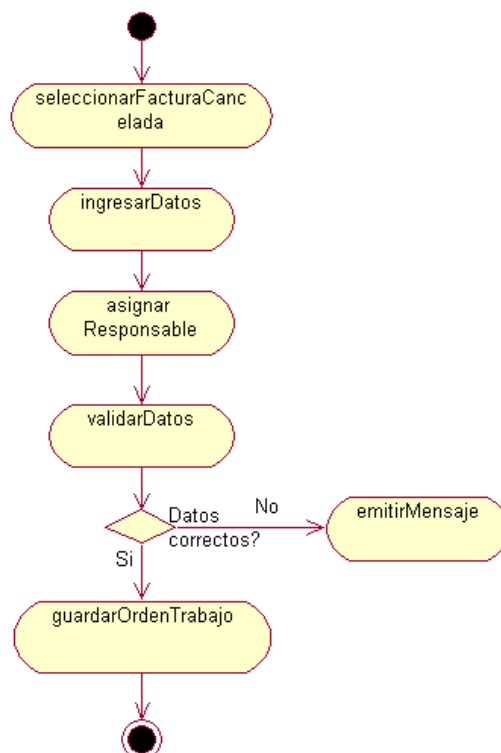


Figura 2. 50 Diagrama de Actividad Generar Orden Trabajo

2.2.2.20. Diagrama de Actividad Asignar Carga Trabajo

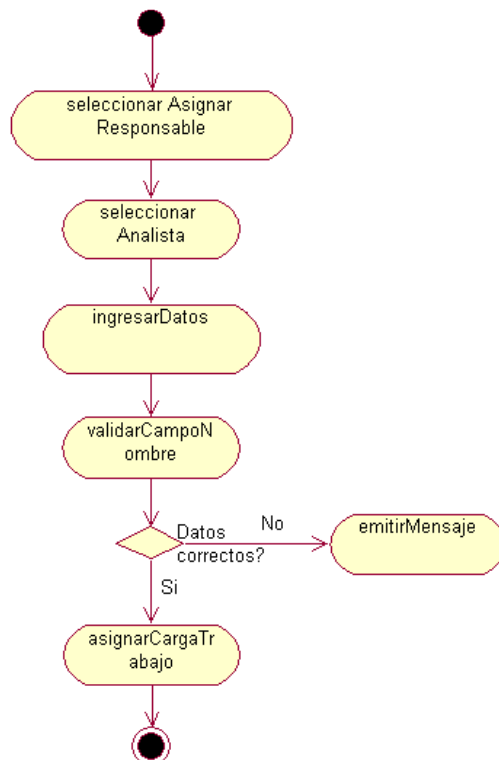


Figura 2. 51 Diagrama de Actividad Asignar Carga Trabajo

2.2.2.21. Diagrama de Actividad Generar Informe Resultados

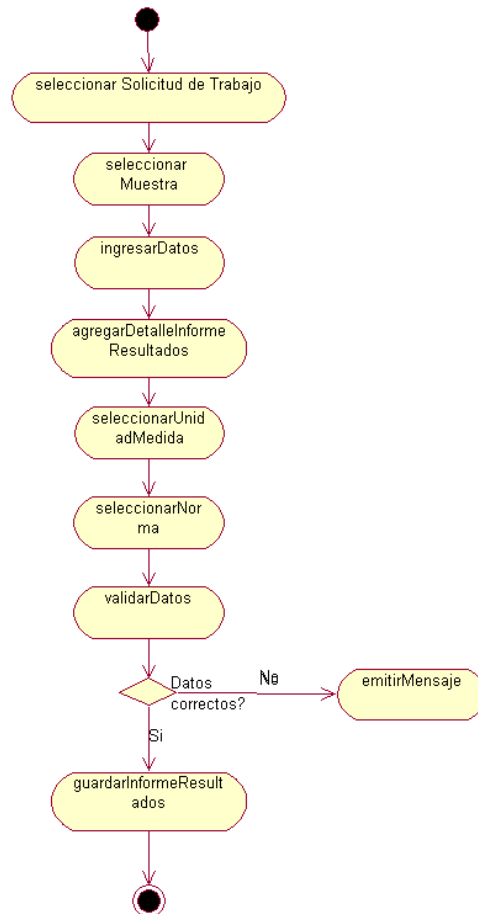


Figura 2. 52 Diagrama de Actividad Generar Informe Resultados

2.2.2.22. Diagrama de Actividad Consultar Proformas Aceptadas

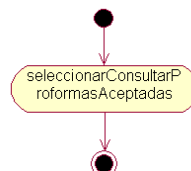


Figura 2. 53 Diagrama de Actividad Consultar Proformas Aceptadas

2.2.2.23. Diagrama de Actividad Generar Factura

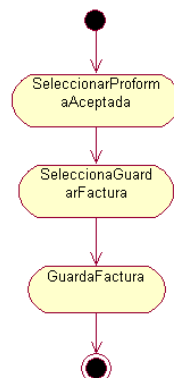


Figura 2. 54 Diagrama de Actividad Generar Factura

2.3. DISEÑO

La etapa de diseño consiste en el refinamiento del análisis considerando los requisitos no funcionales, para de esta manera definir cómo cumple el sistema sus objetivos, para ello se define la arquitectura del sistema, tomando en cuenta que se adapte al sistema general de información de la organización, que tenga flexibilidad para aceptar modificaciones futuras y que ofrezca seguridad contra el uso no autorizado. Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional se define el diagrama de clases y el modelo de datos.

2.3.1. ESQUEMA DE FUNCIONAMIENTO Y ARQUITECTURA DEL SISTEMA

2.3.1.1. Esquema de Funcionamiento del Sistema

A continuación se describe el esquema de funcionamiento general del sistema define que la aplicación será accesible vía web y se encontrará alojada en un servidor ubicado en la Unidad de Gestión Informática de la Escuela Politécnica Nacional.

Los usuarios podrán acceder a la aplicación sin la necesidad de tener una aplicación cliente instalada en su computador, debido a que solo necesita un navegador web como el Mozilla Firefox y su acceso será dentro de la Intranet de la EPN.

Cada usuario tiene la facultad de ingresar al sistema con su respectivo nombre de usuario y su contraseña, de modo que pueda acceder a la información que el sistema le presenta de acuerdo a su perfil y realizar las transacciones permitidas al usuario.

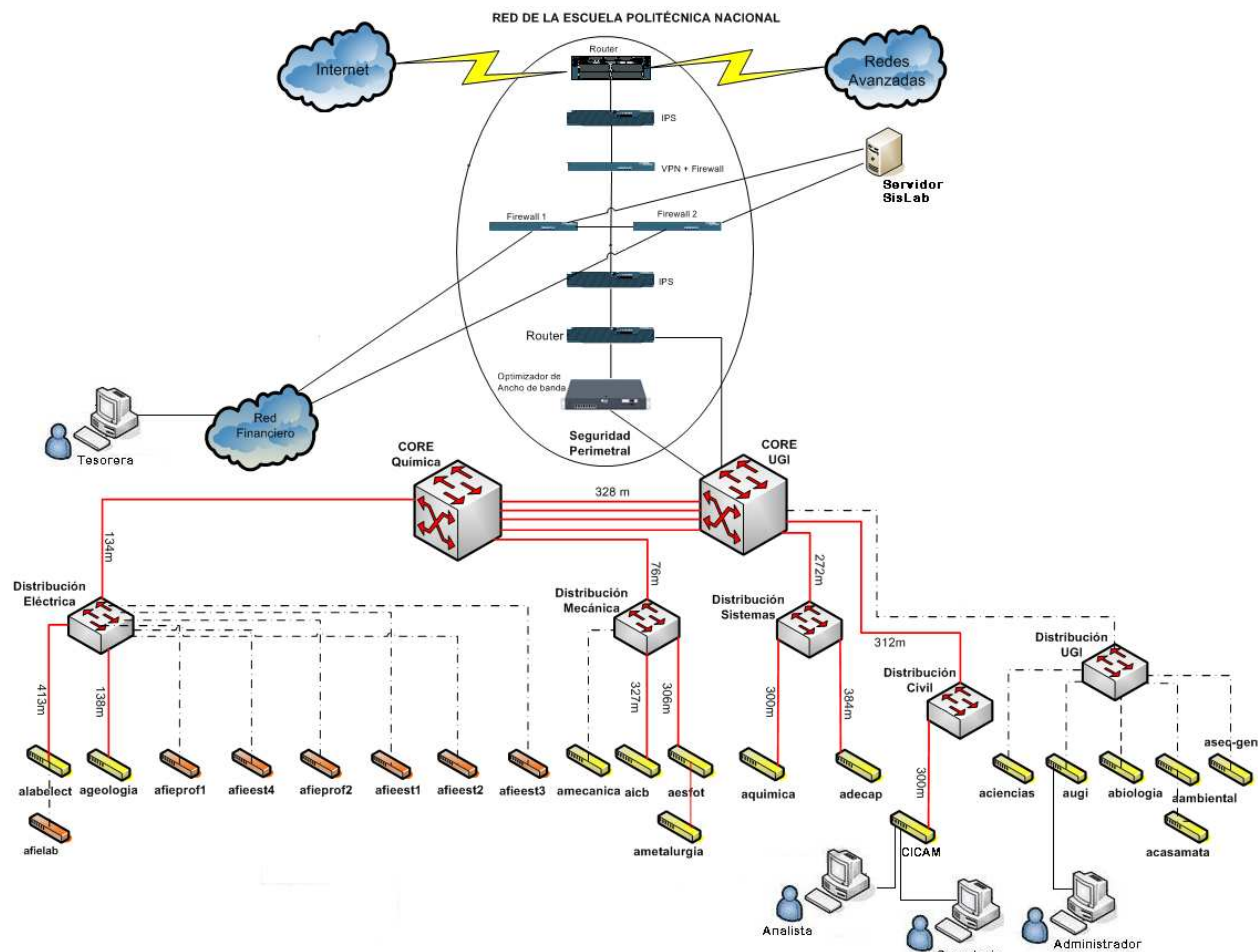


Figura 2. 55 Esquema de funcionamiento del sistema

2.3.1.2. Arquitectura del Sistema

El sistema fue desarrollado como una aplicación web que permite a los usuarios gestionar la información de los laboratorios de la EPN. Es por esta razón que el proyecto fue desarrollado en base al estándar empresarial J2EE, utilizando la arquitectura de tres capas: Capa del Cliente, Capa Intermedia la cual contiene la presentación y la lógica del negocio y la Capa de Datos.

La aplicación fue completamente codificada en lenguaje JAVA y se implemento las tres capas descritas anteriormente de la siguiente manera:

- La Capa de Datos fue implementada utilizando PostgreSQL 8.4.
- La Capa Intermedia fue implementada utilizando el Servidor JBoss y el framework JSF.
- El Cliente se comunica con la lógica del negocio a través de un navegador como el Mozilla Firefox.

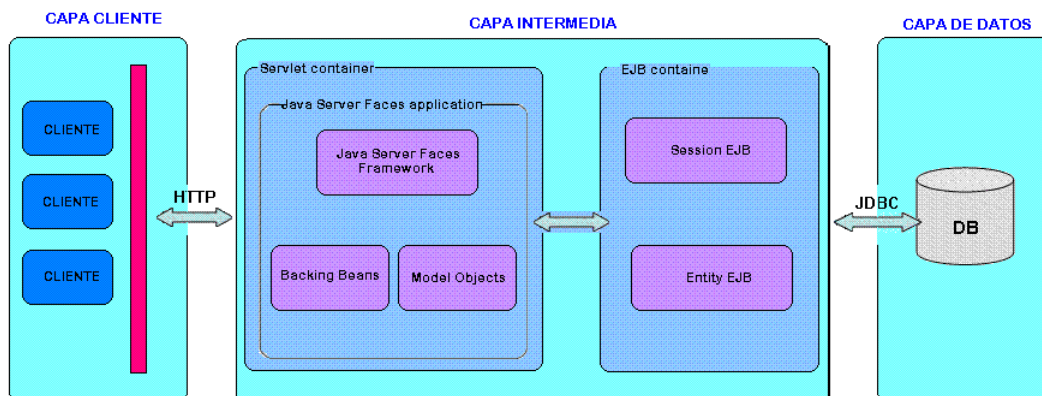


Figura 2. 56 Arquitectura del Sistema

2.3.2. DIAGRAMA DE CLASES

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

Los diagramas de Clases son los más útiles para ilustrar las relaciones entre las clases e interfaces, además nos sirve para realizar la transacción de un dominio, formalizar el análisis de conceptos, definir una solución del diseño y construir componentes de software. Las generalizaciones, las agregaciones y las asociaciones son todas valiosas para reflejar la herencia, la composición o el uso y las conexiones respectivamente.

Se dice que los diagramas de clases son diagramas estáticos porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases conocen a qué otras clases o qué clases son parte de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

Para el diseño del diagrama de clase tenemos:

- Clases las cuales definen atributos y métodos de una serie de objetos, todos los objetos de esta clase tienen el mismo comportamiento y el mismo conjunto de atributos.

- Los atributos se muestran con su nombre, su tipo, valor inicial, etc. Los métodos u operaciones se muestran con su nombre, sus parámetros o valor de entorno.
- Asociación de clases pueden ser las siguientes: generalización, asociaciones, acumulación, composición. Otros componentes del diagrama de clases son interfaces, tipo de datos, enumeraciones, paquetes.

A continuación se presenta el diagrama de clases elaborado para el desarrollo del sistema de Gestión de Laboratorios donde se muestra la iteración entre las clases caracterizadas por sus atributos y operaciones como se muestra en la figura46.

[13] [40] [48] [62]

[13] DOMINGO HERNÁNDEZ H, Introducción al UML

[40] RICARDO BORILLO DOMENECH, Introducción al modelado

[48] WIKIPEDIA, Diagrama de clases

[62] Y. RUMBAUGH, M. BLAHA, W. PREMERLANI, F. EDDY Y W. LORENSEN

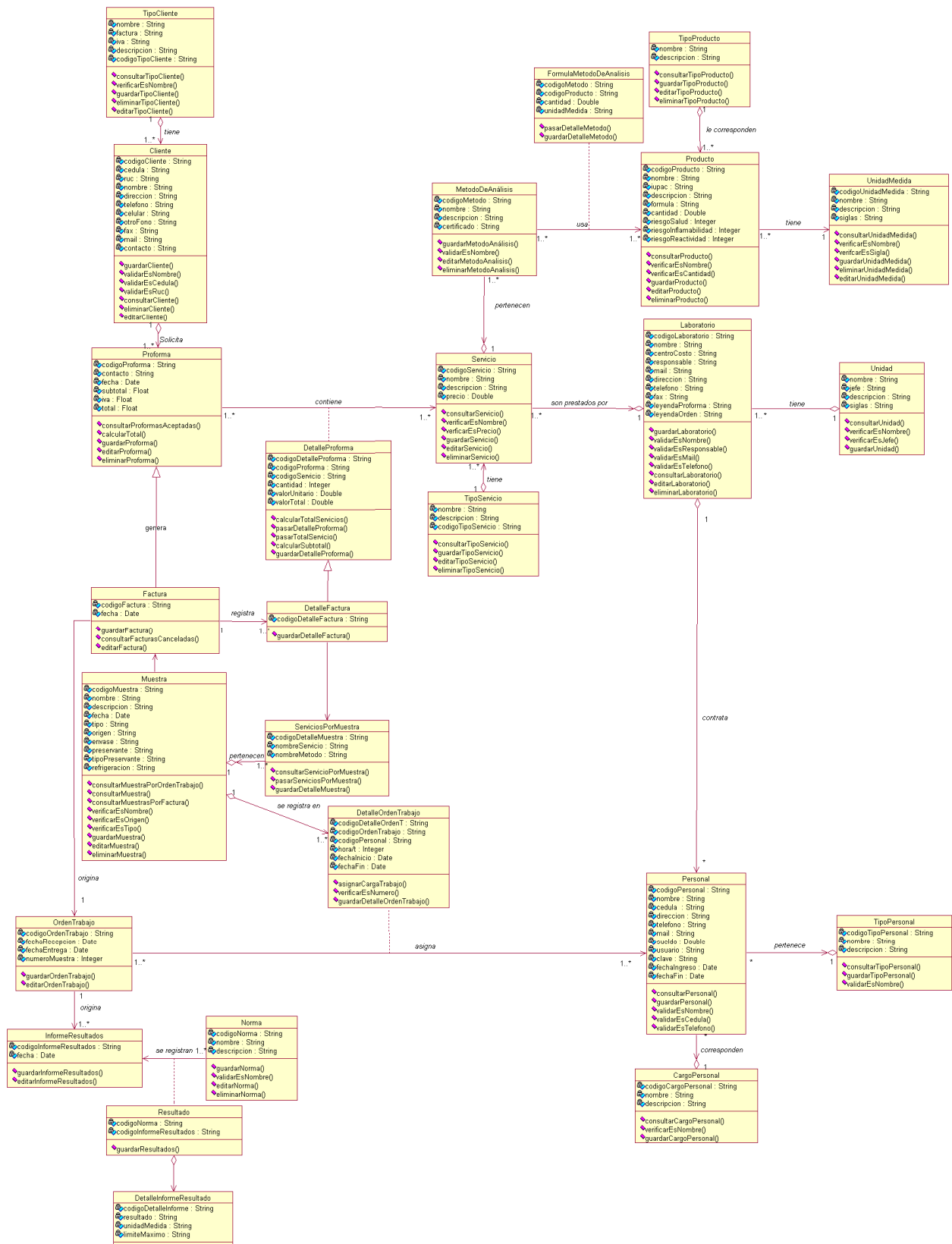


Figura 2. 57 Diagrama de Clases

2.3.3. MODELO DE DATOS RELACIONAL

Es un conjunto de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos, su significado y restricciones de consistencia.

Las características principales del Modelo de Datos es analizar los aspectos de interés para una organización y la relación que tienen unos con otros. El modelo hace la pregunta 'qué' orientada al procesamiento de datos.

Entre los principales beneficios del modelo de datos es registrar los requerimientos de datos de un proceso de negocio. Permite observar patrones de datos y usos potenciales de los datos.

El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos. [3] [11] [55]

[3] CARLOS PROAL AGUILAR, Modelo de Datos

[11] DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN DEL INSTITUTO TECNOLÓGICO DE LA PAZ

[55] WIKIPEDIA, Modelo Relacional

CAPITULO 3: IMPLEMENTACIÓN Y PRUEBAS

Durante la fase de implementación se lleva a cabo la construcción de producto mediante una serie de interacciones luego de verificar que los requerimientos hayan sido plasmados en los entregables anteriores, de no ser así, se rediseña el análisis y diseño hasta conseguir los modelos que se adapten a los requerimientos del usuario final.

En esta fase se realiza una pequeña cascada para cada ciclo, se realizan una serie de iteraciones hasta que se obtenga la implementación final del producto.

3.1. IMPLEMENTACIÓN

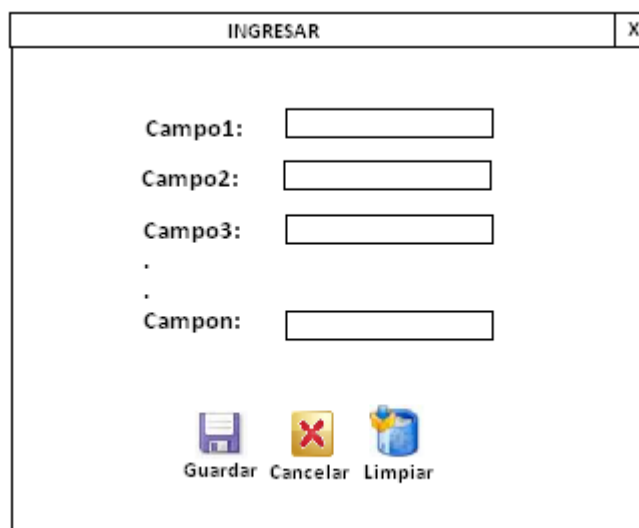
3.1.1. PROTOTIPO DE INTERFACES DE USUARIO

En el presente capítulo se presenta los prototipos de interfaz diseñados de forma general para las operaciones de ingreso, actualización, consulta y eliminación, las cuales serán la base para el diseño de la aplicación.

3.1.1.1. Pantalla Normal

Pantalla para Ingresar

A continuación se muestra el prototipo de interfaz que le servirá al usuario para ingresar los datos en las diferentes clases, como se puede ver en la figura.



El prototipo de interfaz para el ingreso de datos se muestra en una ventana con el título "INGRESAR" y un botón de cerrar "X". El formulario contiene los siguientes elementos:

- Campo1:
- Campo2:
- Campo3:
- .
- .
- Campon:

En la parte inferior del formulario se encuentran tres botones de acción con iconos:

- Guardar (icono de un disco)
- Cancelar (icono de una X roja)
- Limpiar (icono de un cubo de basura)

Figura 3. 1 Prototipo de Interfaz para ingreso de datos

Pantalla para Consultar

A continuación se muestra el prototipo de interfaz que le servirá al usuario para consultar los datos, donde encontramos las opciones de editar y eliminar, como se puede ver en la figura 3.2.

LISTA			
Campo1	Campo2	Campo3	Acción
			Editar Eliminar

Figura 3. 2 Prototipo de Interfaz para consultar datos

Pantalla para Confirmar la eliminación de un dato

A continuación se muestra el prototipo de interfaz que se le presenta al usuario para confirmar la eliminación de un dato seleccionado, como se puede ver en la figura 3.3.

ELIMINAR		X
Está seguro que desea eliminar: _____		
Aceptar		Cancelar

Figura 3. 3 Prototipo de Interfaz para eliminar datos

Pantalla para Editar

A continuación se muestra el prototipo de interfaz que se presenta al usuario una vez que ha seleccionado el dato que quiere modificar, como se puede ver en la figura 3.4.

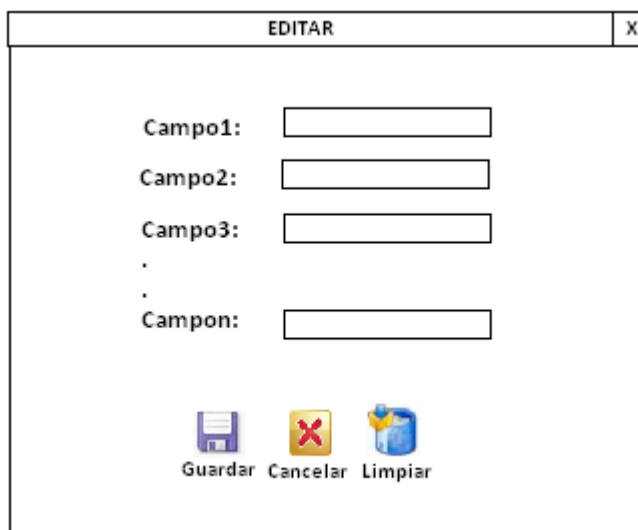


Figura 3. 4 Prototipo de Interfaz para editar datos

3.1.1.2.Pantalla con Encabezado y Detalle

A continuación se muestra el prototipo de interfaz que se presentará al usuario para ingresar datos que consten de un encabezado con su respectivo detalle, como se puede ver en la figura 3.5.

Pantalla para Ingresar

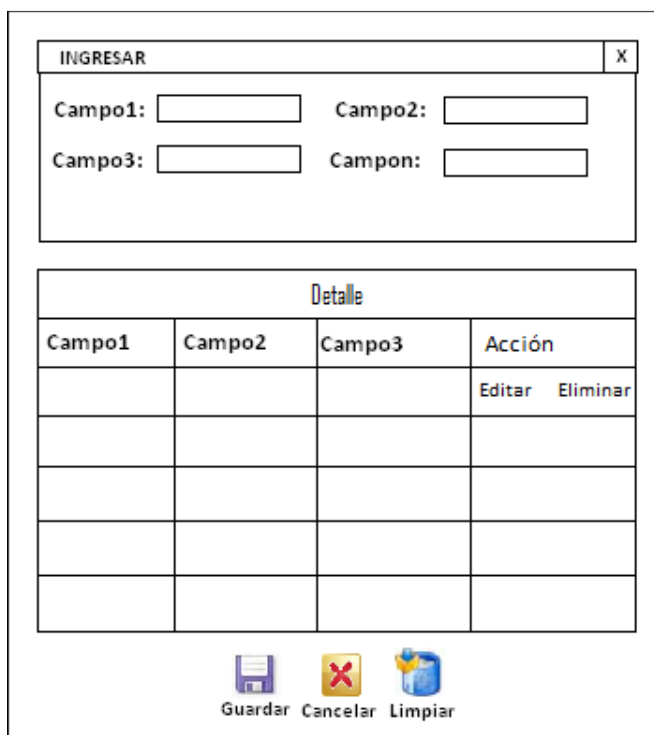


Figura 3. 5Prototipo de Interfaz para ingresar datos con encabezado y detalle

3.1.2. COMPONENTES Y DESPLIEGUE

3.1.2.1. Diagrama de Componentes

Un diagrama de componentes representa los fragmentos de software (componentes) que conforma un sistema y sus relaciones.

Un componente es una parte física y reemplazable de un sistema. Conformar un conjunto de interfaces y realiza esas interfaces. Usualmente se implementa por una o más clases de objetos en tiempo de ejecución.

Los componentes representan todos los tipos de elementos de software que se incorporan en la elaboración de aplicaciones informáticas. Ejemplo: archivos, paquetes, librerías, módulos, ejecutables.

Dado que un diagrama de paquetes muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones y que normalmente un paquete está pensado como un directorio, se dice que los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Dentro de las principales características de los componentes tenemos:

- Representa un elemento físico, existe en el mundo de los bits.
- Es posible reemplazar un componente por otro que sea compatible con las mismas interfaces.
- Como parte de un sistema representa un bloque de construcción fundamental sobre el cual se puede diseñar y construir sistemas. Así un sistema puede ser solo un componente en un nivel de abstracción mayor, compuesto por componentes. [45] [19]

A continuación se presentan los modelos definidos en RUP como diagrama de componentes.

3.1.2.2. Diagrama Global de Paquetes

El primer diagrama muestra la disposición de las partes integrantes de la aplicación y las dependencias entre los distintos módulos de la aplicación.

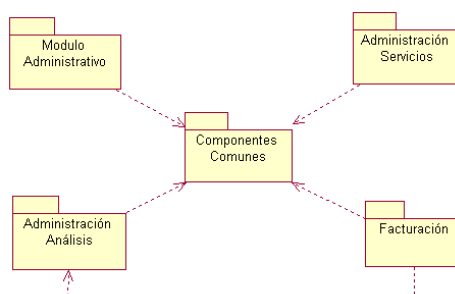


Figura 3. 6 Diagrama Global de Paquetes

Diagrama de Componentes Comunes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables.

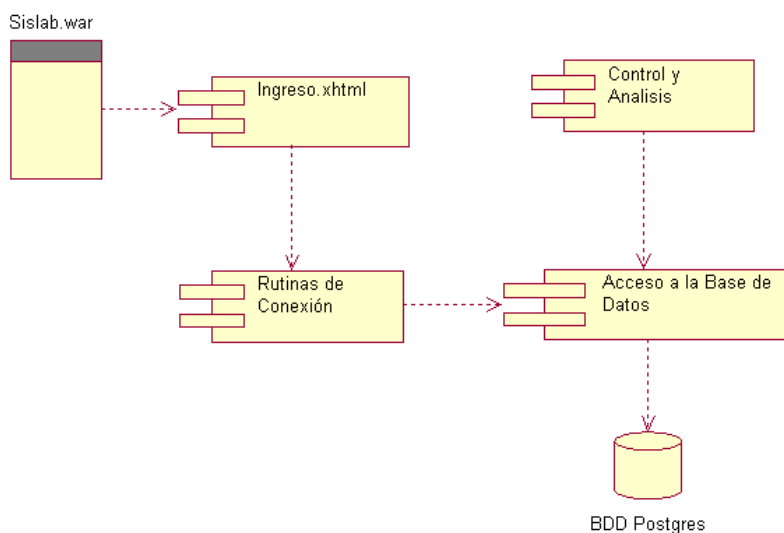


Figura 3. 7 Diagrama de Componentes Comunes

Diagrama de Componentes Modulo Administrativo

El siguiente diagrama de componentes muestra la organización del Módulo Administrativo.

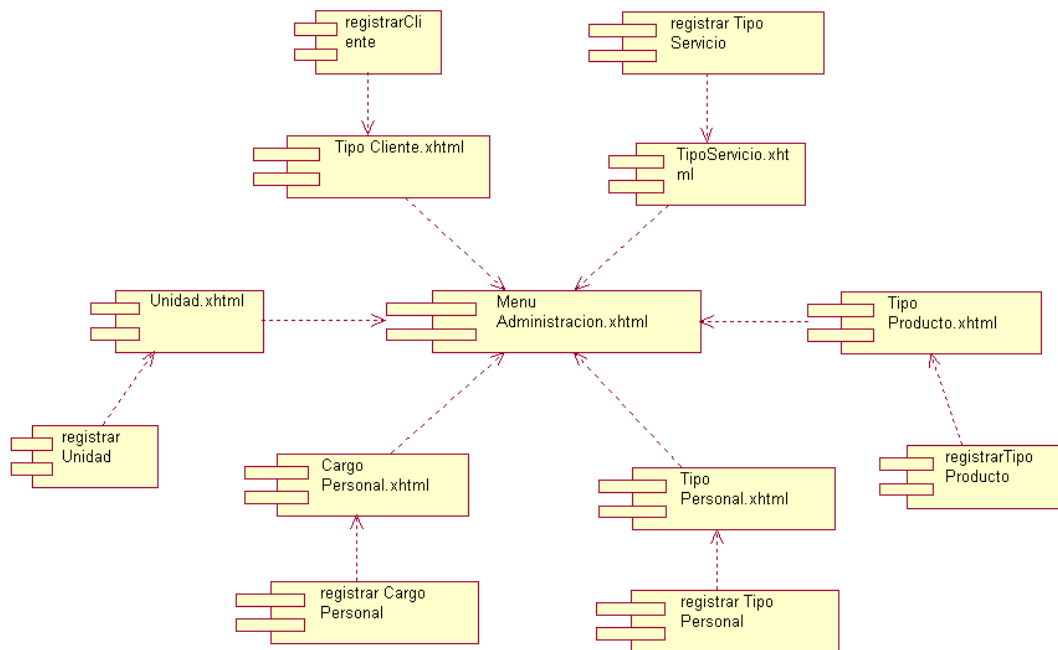


Figura 3. 8 Diagrama de Componentes Módulo Administrativo

Diagrama de Componentes Administración de Servicios

El siguiente diagrama de componentes muestra la organización del Módulo Administración de Servicios.

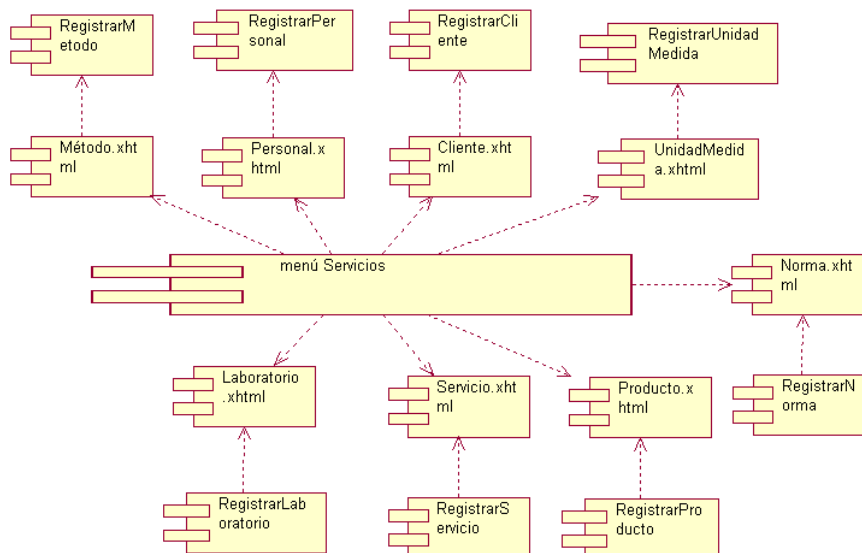


Figura 3. 9 Diagrama de Componentes Administración de Servicios

Diagrama de Componentes Administración de Análisis

El siguiente diagrama de componentes muestra la organización del Módulo de Administración de Análisis.

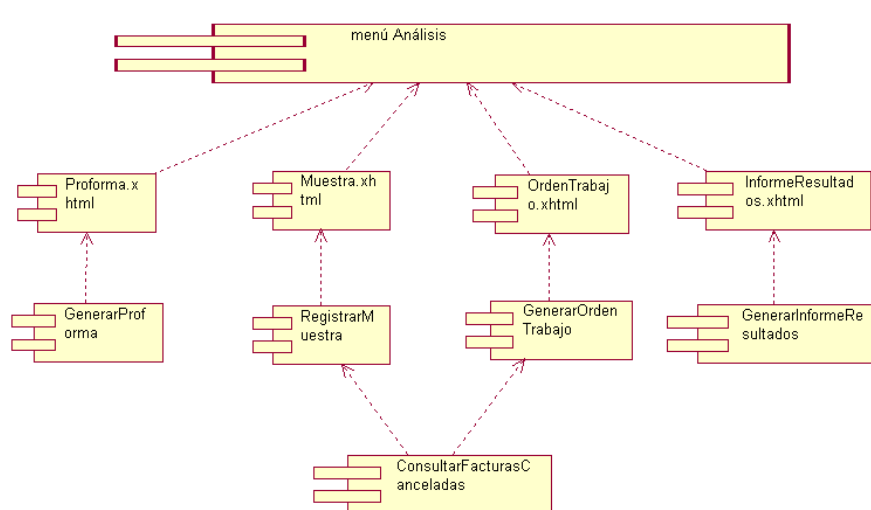


Figura 3. 10 Diagrama de Componentes Manejo de Flujos de Trabajo

Diagrama de Componentes Facturación

El siguiente diagrama de componentes muestra la organización y relación del Módulo de Facturación.

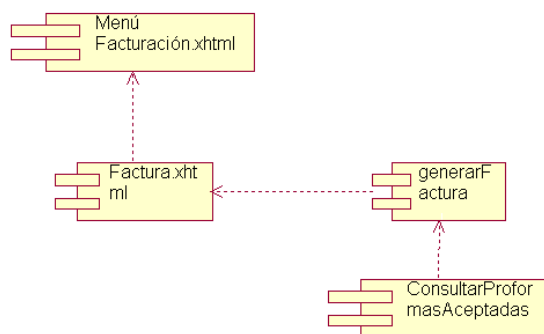


Figura 3. 11 Diagrama de Componentes Facturación

3.1.2.3. Diagrama de Despliegue

Los Diagramas de Distribución muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos.

Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene algo de memoria y, a menudo, capacidad de procesamiento.

Así, un diagrama de despliegue describen la arquitectura física del sistema durante la ejecución, en términos de:

- Dispositivos

- Procesadores
- Componentes de Software

Como los equipos se ubican en los nodos para modelar el despliegue del sistema, la ubicación es guiada por el uso de las especificaciones de despliegue, para lo cual se deben tomar decisiones sobre los parámetros de la configuración, funcionamiento, asignación de recursos, distribución y concurrencia. [32] [18] [49]

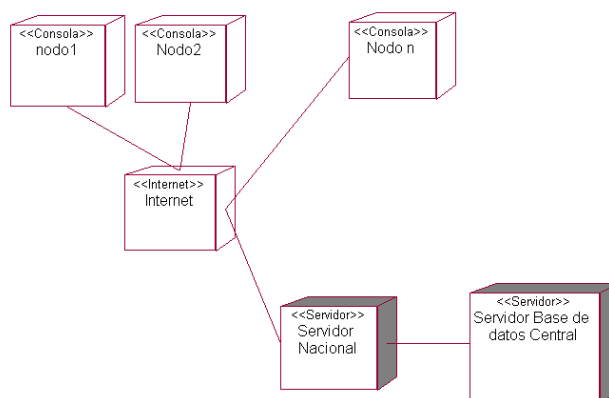


Figura 3. 12 Diagrama de Despliegue

3.1.3. CÓDIGO E INTERFACES

Con el objetivo de permitir que la aplicación a desarrollar mejore en cuanto a reusabilidad, se ha decidido utilizar la tecnología Java Server Faces la cual se basa en el patrón de diseño Modelo-Vista-Controlador (MVC), de modo que la lógica del negocio pueda ser separada de los almacenamientos de datos, permitiendo así que la aplicación con el tiempo pueda ser modificada a las condiciones que establezcan las nuevas reglas de negocio.

3.1.3.1. Modelo Vista Controlador en JSF

El patrón MVC (Modelo Vista Controlador), nos permite separar la lógica de control, la lógica de negocio y la lógica de presentación.

Utilizando este tipo de patrón es posible conseguir más calidad, un mantenimiento más fácil, al margen de todo esto, una de las cosas más importantes que permite el uso de este patrón consiste en normalizar y estandarizar el desarrollo de Software.

[32] OBJECTS AND DESIGN, Herramientas basadas en UML

[18] ING. MA. DE PILAR RAMÍREZ GIL, Ingeniería de Software

[49] 49. WIKIPEDIA, diagrama de despliegue

Para el diseño de la aplicación a desarrollar estructuramos los componentes de la misma siguiendo el patrón de diseño MVC con JSF como se muestra en la Figura 3.13, donde se observa el nombre del proyecto denominado SisLab, en el mismo los tres componentes del patrón de diseño separados en carpetas específicas. Se tiene la carpeta Java Source donde se encuentran las clases VO e Implem que trabajan como modelo, la carpeta vista donde se encuentran páginas creadas para la interfaz de la aplicación y la carpeta WEB-INF donde se encuentran el faces-config.xml y el web.xml los cuales nos sirven como controlador. [56]

Además, este modelo de arquitectura presenta otras importantes ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.
- Hay una API muy bien definida; cualquiera que use la API, podrá reemplazar el modelo, la vista o el controlador, sin demasiada dificultad.
- La conexión entre el modelo y sus vistas (ya que puede haber varias) es dinámica: se produce en tiempo de ejecución, no en tiempo de compilación.

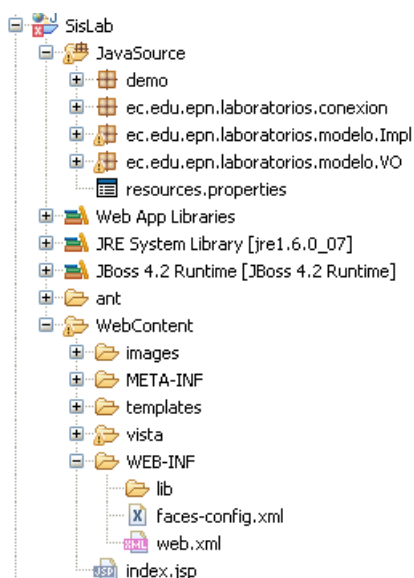


Figura 3. 13 Organización de componentes de la aplicación

3.1.3.2. Organización del Código de acuerdo al Patrón MVC en JSF

Para el proceso de codificación nos basamos en el patrón Modelo-Vista-Controlador mediante JSF el cual permite separar los distintos componentes que intervienen durante el proceso de codificación.

Modelo

El modelo es el objeto que representa y trabaja directamente con los datos del programa: gestiona los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los diferentes controladores y/o vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuándo deben reflejar un cambio en el modelo.

El modelo se ve reflejado mediante las clases impl y VO las cuales contienen los elementos necesarios para manejar los datos de la aplicación. A continuación se muestra en código de las clases mencionadas.

CLASE IMPLEMEN

```
/**
 * @author Proyecto Tesis
 * @version */

package ec.edu.epn.laboratorios.persistencia;

import java.io.Serializable;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import ec.edu.epn.laboratorios.VO.UnidadVO;
import ec.edu.epn.laboratorios.conexion.Conexion;

public class UnidadImpl extends Conexion implements Serializable{

    /** Atributos del objeto*/
    private int id_unidad;
    private String nombre_u;
    private String jefe_u;
    private String descr_u;
    private String sigla_u;
    private ArrayList<UnidadVO> unidad;

    private int scrollerPage;

    /** Constructores */
    public void UnidadImpl(){

    }

    public void UnidadImpl (int id_unidad, String nombre_u, String jefe_u, String descr_u,
        String sigla_u){
        this.id_unidad = id_unidad;
    }
}
```



```

        this.nombre_u = nombre_u;
        this.jefe_u = jefe_u;
        this.descr_u = descr_u;
        this.sigla_u = sigla_u;
    }

    /** Metodo Consultar todos los objetos Unidad*/
    public ArrayList<UnidadVO> getUnidad() {
        Connection con = null;
        PreparedStatement ps = null;
        try {
            con = super.getCon();
            if (con != null) {
                ps = con.prepareStatement("SELECT id_unidad, " +
                    "nombre_u, jefe_u, descr_u, sigla_u FROM unidad");

                ResultSet rs = ps.executeQuery();

                ArrayList<UnidadVO> listaUnidad = new ArrayList<UnidadVO>();

                while (rs.next()) {
                    id_unidad = rs.getInt(1);
                    nombre_u = rs.getString(2);
                    jefe_u = rs.getString(3);
                    descr_u = rs.getString(4);
                    sigla_u = rs.getString(5);

                    UnidadVO u = new UnidadVO();

                    u.setId_unidad(id_unidad);
                    u.setNombre_u(nombre_u);
                    u.setJefe_u(jefe_u);
                    u.setDescr_u(descr_u);
                    u.setSigla_u(sigla_u);

                    listaUnidad.add(u);
                }
                ps.close();
                con.close();

                return listaUnidad;
            } else {
                System.out.println("Error en la conexion");
                return null;
            }
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("ENTRAMOS AL catch 1");
            return null;
        } finally {
            super.cerrarConexion(con, ps);
        }
    }

    public void setUnidad(ArrayList<UnidadVO> unidad) {
        this.unidad = unidad;
    }
}

```

CLASE VO

```

/**
 * @author Proyecto Tesis
 * @version */

package ec.edu.epn.laboratorios.VO;

import java.io.Serializable;

import javax.faces.component.UIParameter;

```

```

import javax.faces.event.ActionEvent;

import ec.edu.epn.laboratorios.persistencia.UnidadImpl;

public class UnidadVO implements Serializable {

    /** Atributos del objeto */
    private int id_unidad;
    private String nombre_u;
    private String jefe_u;
    private String descr_u;
    private String sigla_u;
    private String sucesos = " ";
    private String mensajeExito="";

    /** Estilo Mensajes */
    private String style;
    private String styleSuccess = "COLOR: #2E2EFE;";
    private String styleError = "COLOR: #fc5868;";

    /** Constructores */
    public UnidadVO() {

    }

    public UnidadVO(int id_unidad, String nombre_u, String jefe_u,
                    String descr_u, String sigla_u) {
        this.id_unidad = id_unidad;
        this.nombre_u = nombre_u;
        this.jefe_u = jefe_u;
        this.descr_u = descr_u;
        this.sigla_u = sigla_u;
    }

    /** Set y get del objeto */
    public void setUnidad(UnidadVO u) {
        this.setId_unidad(u.getId_unidad());
        this.setNombre_u(u.getNombre_u());
        this.setJefe_u(u.getJefe_u());
        this.setDescr_u(u.getDescr_u());
        this.setSigla_u(u.getSigla_u());
    }

    public UnidadVO getUnidad() {
        return new UnidadVO(this.getId_unidad(), this.getNombre_u(), this
            .getJefe_u(), this.getDescr_u(), this.getSigla_u());
    }

    /** Metodo Guardar */
    public void guardarUnidad(ActionEvent event) {
        setSigla_u(getSigla_u().trim());

        setNombre_u(getNombre_u().toUpperCase());
        setJefe_u(getJefe_u().toUpperCase());
        setDescr_u(getDescr_u().toUpperCase());
        setSigla_u(getSigla_u().toUpperCase());

        UnidadImpl u = new UnidadImpl();

        if (u.validarNombreU(getNombre_u()) == "Existe") {

            this.setStyle(getStyleError());
            this.setSucesos("La unidad ya existe");
        }
    }
}

```

```

    } else if (u.validarNombreU(getNombre_u()) == "Error") {

    } else {
        if (u.ingresarUnidad(getUnidad()) == true) {
            this.setStyle(getStyleSuccess());
            setMensajeExito(getMensajeExito()+"Datos almacenados con exito!");

            String aux1 = "";
            String aux2 = "";
            String aux3 = "";
            String aux4 = "";

            setNombre_u(aux1);
            setJefe_u(aux2);
            setDescr_u(aux3);
            setSigla_u(aux4);
        }
    }
}

```

Vista

La vista es el objeto que maneja la presentación visual de los datos gestionados por el Modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interacciona con el modelo a través de una referencia al propio modelo.

La vista se ve reflejada mediante los template y las páginas como la index.jsp. A continuación el código de una página.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:a4j="http://richfaces.org/a4j"
  xmlns:rich="http://richfaces.org/rich"
  xmlns:s="http://jboss.com/products/seam/taglib">

<f:loadBundle basename="resources" var="msg" />
<!-- Replace path to template, title, header and body with actual data. -->

<ui:composition template="/templates/common.xhtml">
  <ui:define name="pageTitle">Unidad</ui:define>
  <ui:define name="pageHeader">Unidad</ui:define>
  <ui:define name="body">

    <!-- Panel para ingresar nueva Unidad -->
    <rich:modalPanel id="panelUnidad" width="470" height="300">
      <f:facet name="header">
        <h:panelGroup>
          <h:outputText value="Ingresar Unidad"></h:outputText>
        </h:panelGroup>
      </f:facet>

      <f:facet name="controls">
        <h:panelGroup >

```

```

        <a4j:form>
            <a4j:commandLink>
                <h:graphicImage id="clo1" value="/images/Close.png"
                    title="Cerrar" style="border:0px solid #CAD6E0,"
styleClass="hidelink" />
                <rich:componentControl for="panelUnidad" attachTo="clo1"
operation="hide" event="onclick"/>
            </a4j:commandLink>
        </a4j:form>
    </h:panelGroup>
</f:facet>

<a4j:form>
    <h:panelGrid style="border:1px solid #CAD6E0;
        height: 180px;
        left: 55px;
        top: 40px;
        position: absolute; "
        columns="3" title="Ingrese los datos" cellspacing="4" cellpadding="5"
        width="360" id="panelGridUnidad" >

        <h:panelGrid style="border:0px solid #CAD6E0;
            height: 20px;
            left: 55px;
            top: 55px;

            position: absolute; "
            columns="3" >

            <h:outputText value="Nombre Unidad: " />
            <h:inputText id="nombre_u" value="#{unidadVO.nombre_u}"
                required="true" tabindex="1" size="40">
            </h:inputText>
            <rich:message for="nombre_u">
                <f:facet name="passedMarker">
                    <h:graphicImage value="/images/passed.gif" />
                </f:facet>
                <f:facet name="errorMarker">
                    <h:graphicImage value="/images/error.gif" />
                </f:facet>
            </rich:message>

            <h:outputText value="Jefe Unidad: " />
            <h:inputText id="jefe_u" value="#{unidadVO.jefe_u}"
                required="true" tabindex="2" size="40">
            </h:inputText>
            <rich:message for="jefe_u">
                <f:facet name="passedMarker">
                    <h:graphicImage value="/images/passed.gif" />
                </f:facet>
                <f:facet name="errorMarker">
                    <h:graphicImage value="/images/error.gif" />
                </f:facet>
            </rich:message>
        </h:panelGrid>

        <h:panelGrid style="border:0px solid #CAD6E0;
            height: 20px;
            left: 55px;
            top: 105px;

            position: absolute; "

```

```

        columns="4" >

        <h:outputText value="Descripcion: " />
        <rich:spacer width="13"></rich:spacer>
        <h:inputTextarea id="descr_u" value="#{unidadVO.descr_u}" tabindex="3" />
        <rich:spacer width="0"></rich:spacer>

                <h:outputText value="Siglas:" />
                <rich:spacer width="5"></rich:spacer>
                <h:inputText id="sigla_u" value="#{unidadVO.sigla_u}"
                        required="true" tabindex="4" size="40">
        </h:inputText>
        <h:graphicImage id="ok_siglasU" value="/images/ok.gif" style="border:0px solid

#CAD6E0;" />

</h:panelGrid>

<h:panelGrid style="border:0px solid #CAD6E0;
        height: 20px;
left: 180px;
top: 190px;

position: absolute; "
        columns="2" >

        <h:graphicImage id="ok_Obser" value="/images/ok.gif" style="border:0px solid

#CAD6E0;" />

        <h:outputText value="Requerido" />
</h:panelGrid>

<f:facet name="footer" >
        <h:panelGroup style="border:1px solid #CAD6E0;
                height: 40px;
left: 170px;
top: 230px;

position: absolute; " >

                <a4j:commandLink actionListener="#{unidadVO.guardarUnidad}"
                        id="guardarUnidad"
reRender="estado_Unidad,DatosUnidad,panelGridUnidad" tabindex="6">
                        <h:graphicImage value="/images/guardar1.png"
title="Guardar" style="border:0px solid #CAD6E0;" />
                </a4j:commandLink>

                <a4j:commandLink tabindex="4" >
                        <h:graphicImage id="cancelar" value="/images/cancelar1.gif"
                                title="Cancelar" style="border:0px solid #CAD6E0;"
/>

                        <rich:componentControl for="panelUnidad"
                                operation="hide" event="onclick"/>
                </a4j:commandLink>

                <a4j:commandLink actionListener="#{unidadVO.limpiarUnidad}"
                        reRender="panelGridUnidad,estado_Unidad" id="limpiarUnidad"
                        tabindex="7">
                        <h:graphicImage value="/images/limpiar4.gif" title="Limpiar"
                                style="border:0px solid #CAD6E0;" />
                </a4j:commandLink>

```

```

        </h:panelGroup>
        </f:facet>
    </h:panelGrid>
</a4j:form>

<h:panelGrid style="border:0px solid #CAD6E0;
    height: 25px;
    left: 137px;
    top: 270px;
    position: absolute; " id="panelSucesos">
    <h:outputText value="#{unidadVO.sucesos}" id="estado_Unidad" style="#{unidadVO.style}" />
</h:panelGrid>
</rich:modalPanel>

```

Controlador

El controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Entra en acción cuando se realiza alguna operación, ya sea un cambio en la información del modelo o una interacción sobre la Vista. Se comunica con el modelo y la vista a través de una referencia al propio modelo.

Además, JSF opera como un gestor que reacciona ante los eventos provocados por el usuario, procesa sus acciones y los valores de estos eventos, y ejecuta código para actualizar el modelo o la vista.

El controlador se verá reflejado en las reglas de navegación contenidas en el fichero faces-config.xml y el servlet faces definido en el fichero web.xml.

- **Faces-config.xml**

```

<managed-bean>
  <managed-bean-name>unidadVO</managed-bean-name>
  <managed-bean-class>ec.edu.epn.laboratorios.modelo.Implem.UnidadVO</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>unidadImpl</managed-bean-name>
  <managed-bean-class>ec.edu.epn.laboratorios.modelo.VO.UnidadImpl</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>

```

- **Web.xml**

```

<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <description>Facelets StarterKit</description>
  <display-name>SisLab</display-name>

  <context-param>
  <param-name>org.richfaces.SKIN</param-name>
  <param-value>blueSky</param-value>
</context-param>

```

```

<!-- Making the RichFaces skin spread to standard HTML controls -->
<context-param>
    <param-name>org.richfaces.CONTROL_SKINNING</param-name>
    <param-value>enable</param-value>
</context-param>
<context-param>
    <param-name>org.jboss.jbossfaces.WAR_BUNDLES_JSF_IMPL</param-name>
    <param-value>>true</param-value>
</context-param>

<!-- Defining and mapping the RichFaces filter -->
<filter>
    <display-name>RichFaces Filter</display-name>
    <filter-name>richfaces</filter-name>
    <filter-class>org.ajax4jsf.Filter</filter-class>
</filter>
<filter-mapping>
    <filter-name>richfaces</filter-name>
    <servlet-name>Faces Servlet</servlet-name>
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>FORWARD</dispatcher>
    <dispatcher>INCLUDE</dispatcher>
</filter-mapping>
<listener>
    <listener-class>org.jboss.seam.servlet.SeamListener</listener-class>
</listener>
<servlet>
    <servlet-name>Seam Resource Servlet</servlet-name>
    <servlet-class>org.jboss.seam.servlet.SeamResourceServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>Seam Resource Servlet</servlet-name>
    <url-pattern>/seam/resource/*</url-pattern>
</servlet-mapping>
<servlet>
    <servlet-name>Document Store Servlet</servlet-name>
    <servlet-class>org.jboss.seam.document.DocumentStoreServlet</servlet-class>
</servlet>
<filter>
    <filter-name>Seam Filter</filter-name>
    <filter-class>org.jboss.seam.servlet.SeamFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>Seam Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
<context-param>
    <param-name>javax.faces.DEFAULT_SUFFIX</param-name>
    <param-value>.xhtml</param-value>
</context-param>
<context-param>
    <param-name>facelets.REFRESH_PERIOD</param-name>
    <param-value>2</param-value>
</context-param>
<context-param>
    <param-name>facelets.DEVELOPMENT</param-name>
    <param-value>>true</param-value>
</context-param>
<context-param>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>client</param-value>
</context-param>

```

```

<context-param>
  <param-name>com.sun.faces.validateXml</param-name>
  <param-value>>true</param-value>
</context-param>
<context-param>
  <param-name>com.sun.faces.verifyObjects</param-name>
  <param-value>>true</param-value>
</context-param>
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.jsf</url-pattern>
</servlet-mapping>
<login-config>
  <auth-method>BASIC</auth-method>
</login-config>
</web-app>

```

Interfaz

En esta sección se presentan las interfaces con las que el usuario interactuará, las mismas que son resultado del diseño realizado en el modelo vista, que se muestra a continuación.

- *Interfaz para Consultar*

La Figura 3.14 es la primera que se le presenta al usuario y es una consulta de todas las Unidades ingresadas, al final de cada unidad se muestra dos link, los cuales le ayudarán al usuario a editar si fuera necesario algún dato permitido o eliminar si se diera la necesidad.



Figura 3. 14 Pantalla Consultar Unidad

- *Interfaz para Ingresar*

La pantalla que se muestra en la Figura 3.15 le permite al usuario ingresar una Unidad, los campos que tienen un visto al lado derecho significan que son obligatorios y los que no tienen pueden o no ingresarlos.

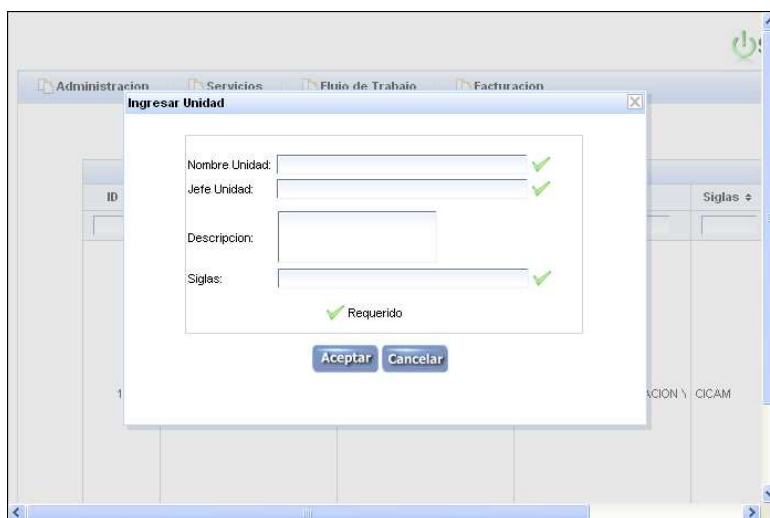


Figura 3. 15 Pantalla Ingresar Unidad

- *Interfaz para Editar*

La Figura 3.15 muestra la pantalla que le ayudará al usuario actualizar los datos ingresados, de igual manera los campos que tienen visto no pueden quedar vacíos porque son requeridos.



Figura 3. 16 Pantalla Editar Unidad

- *Interfaz para Confirmar*

La figura 3.17 es una pantalla de confirmación para eliminar un registro después de que el usuario a dado clic en el link eliminar que se encuentra al lado del registro, si el usuario acepta se elimina el registro seleccionado.

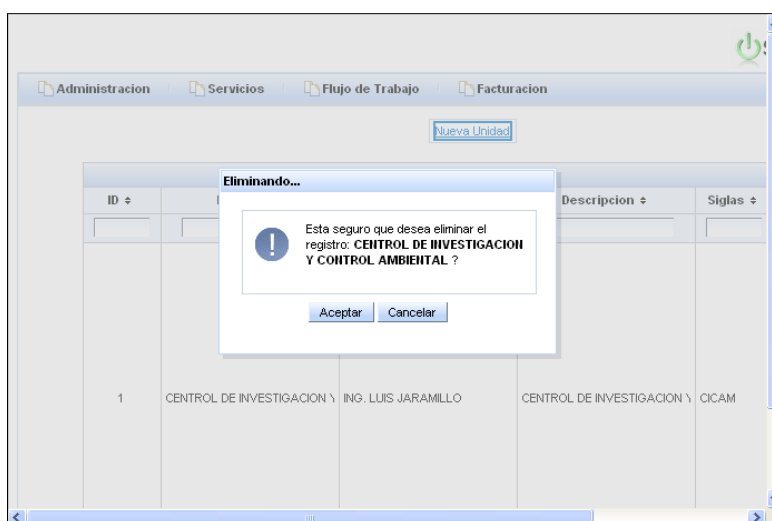


Figura 3. 17 Pantalla Eliminar Unidad

- *Interfaz para Ingreso con Encabezado y Detalle*

La Figura 3.18 muestra una pantalla que le permite al usuario ingresar un registro que conste de encabezado y detalle, para ingresar el detalle se debe dar clic en el link Agregar Detalle.



Figura 3. 18 Pantalla para Ingresar registros con Encabezado y Detalle

- *Interfaz para Ingreso de Detalle*

La figura 3.19 muestra la pantalla que le permite al usuario ingresar el Detalle de un registro.

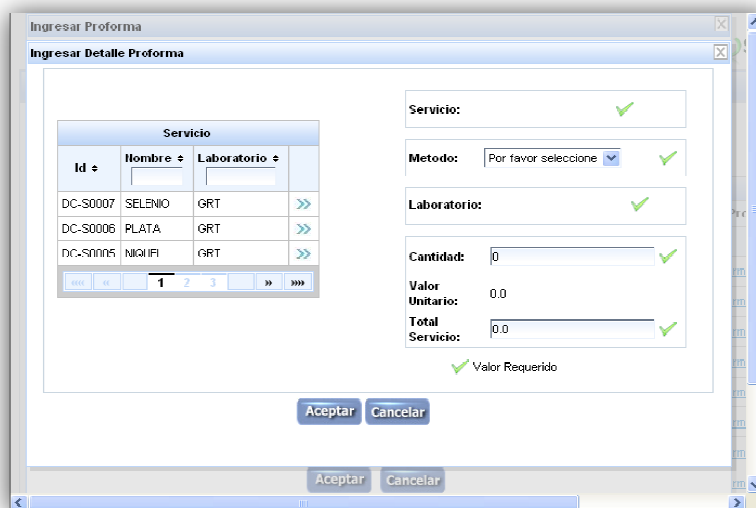


Figura 3. 19 Pantalla Ingresar Detalle

- *Descripción de Botones y Link's*

Botón	Condición	Al clic
Aceptar	Ingresar todos los campos	Después de ingresados los datos en los

	establecidos como obligatorios.	respectivos campos y una vez validados, el sistema guarda la información en la ubicación respectiva.
Cancelar	Ninguna	Regresa a la pantalla de anterior, sin guardar los datos.

Tabla 3. 1 Descripción Botones

Link	Condición	Al clic
Nuevo	Ninguna	Despliega el panel que permite Ingresar un nuevo registro.
Editar	Ubicar el registro a editar, en la Lista respectiva.	Despliega el panel que permite Editar un registro existente.
Eliminar	Ubicar el registro a editar, en la Lista respectiva.	Despliega el panel que permite Confirmar la Eliminación de un registro, siempre y cuando este no tenga datos asociados.
Agregar Detalle	Ninguna	Despliega el panel que permite seleccionar los detalles para un registro.
Generar	Los registros de los cuales dependen sean previamente ingresados.	Despliega el panel para ingreso de un registro (Factura, Orden de Trabajo, Informe de Resultados) cuyos datos están asociados y depende de otra información.
Editar Estado	Ninguna	Despliega el panel que permite Editar el estado de una proforma o una factura.
Ver	Ubicar el registro a visualizar	Despliega un panel con la información de encabezado y detalle para los registros de Proforma, Factura, Muestra, Método, Orden de Trabajo e Informe de Resultados.

Tabla 3. 2 Descripción de Links

3.2. PRUEBAS

3.2.1. CASOS DE PRUEBA

3.2.1.1.Caso de Prueba Ingresar al Sistema (CP01)

No. Caso de Prueba	CP01
Caso de Uso de Referencia	INGRESAR SISTEMA
Nombre	Ingresar Sistema
Descripción	Proceso de ingreso al sistema por parte de un usuario, previamente registrado en la nómina de personal.
Entrada	Perfil: Administrador Username: analista Password: analista1
Salida	N/A
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. En el browser colocar la dirección URL de la aplicación. 2. En la ventana de Ingreso, seleccionar el perfil Administrador e ingrese los datos de username y password indicados en la entrada. 3. Hacer clic en el botón Aceptar, para ingresar al sistema
Resultado Esperado	El sistema valida la información y presenta la ventana principal con el

	respectivo menú para el perfil Administrador.
--	---

Tabla 3. 3Caso de Prueba Ingresar al Sistema

3.2.1.2.Caso de Prueba Registrar unidad (CP02)

No. Caso de Prueba	CP02
Nombre	REGISTRAR UNIDAD
Descripción	Procedimiento para ingresar una nueva Unidad, que se almacena en la base de datos para que este pueda ser usado al momento de registrar un nuevo Laboratorio.
Entrada	Nombre: Centro de Investigación y Control Ambiental Descripción: Jefe: Ing. Luis Jaramillo Sigla: CICAM
Salidas	N/A
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ingrese al sistema con el perfil de Administrador. 2. Seleccione del menú la interfaz “Unidad”. 3. En la interfaz seleccione el link “Nueva Unidad”. 4. En el panel “Ingresar Unidad” que se despliega, ingrese los datos indicados en la entrada. 5. Seleccione el botón Aceptar, para registrar la nueva Unidad.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda la Unidad, y emite un mensaje de éxito.

Tabla 3. 4 Caso de Prueba Registrar Unidad

3.2.1.3.Caso de Prueba Registrar Tipo Cliente (CP03)

No. Caso de Prueba	CP03
Nombre	REGISTRAR TIPO CLIENTE
Descripción	Procedimiento para ingresar un nuevo Tipo de Cliente, que se almacena en la base de datos para que este pueda ser usado al momento de registrar un nuevo Cliente.
Entrada	Nombre: Externo Factura: Si IVA: Si Descripción: Tipo de cliente definido para cliente de administración privada.
Salidas	N/A
Procedimiento de Prueba	<ol style="list-style-type: none"> 6. Ingrese al sistema con el perfil de Administrador. 7. Seleccione del menú la interfaz “Tipo Cliente”. 8. En la interfaz seleccione el link “Nuevo Tipo Cliente”. 9. En el panel “Ingresar Tipo Cliente” que se despliega, ingrese los datos indicados en la entrada. 10. Seleccione el botón Aceptar, para registrar el nuevo cliente.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el Tipo Cliente, y emite un mensaje de éxito.

Tabla 3. 5 Caso de Prueba Registrar Tipo Cliente

3.2.1.4.Caso de Prueba Registrar Tipo Servicio (CP04)

No. Caso de Prueba	CP04
Nombre	REGISTRAR TIPO SERVICIO
Descripción	Procedimiento para el ingreso de un nuevo Tipo Servicio, que se almacena en la base de datos para que este pueda ser usado al momento de registrar un nuevo Servicio.
Entrada	Nombre: Análisis de Laboratorio Descripción:
Salidas	N/A
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ingrese al sistema con el perfil de Administrador. 2. Seleccione del menú la interfaz “Tipo Servicio”. 3. En la interfaz seleccione el link “Nuevo Tipo Servicio”. 4. En el panel “Ingresar Tipo Servicio” que se despliega, ingrese los datos indicados en la entrada. 5. Seleccione el botón Aceptar, para registrar el nuevo tipo servicio.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el Tipo Servicio, y emite un mensaje de éxito.

Tabla 3. 6 Caso de Prueba Registrar Tipo Servicio

3.2.1.5.Caso de Prueba Registrar Tipo Producto (CP05)

No. Caso de Prueba	CP05
Nombre	REGISTRAR TIPO PRODUCTO
Descripción	Procedimiento para el ingreso de un nuevo Tipo Producto, que se almacena en la base de datos para que este pueda ser usado al momento de registrar un nuevo Producto.
Entrada	Nombre: Inflamable Descripción: sustancias peligrosas reguladas, para las cuales se han fijado cantidades de reporte o de control.
Salidas	N/A
Procedimiento de Prueba	<ol style="list-style-type: none"> 6. Ingrese al sistema con el perfil de Administrador. 7. Seleccione del menú la interfaz “Tipo Producto”. 8. En la interfaz seleccione el link “Nuevo Tipo Producto”. 9. En el panel “Ingresar Tipo Producto” que se despliega, ingrese los datos indicados en la entrada. 10. Seleccione el botón Aceptar, para registrar el nuevo tipo producto.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el Tipo Producto, y emite un mensaje de éxito.

Tabla 3. 7 Caso de Prueba Registrar Tipo Producto

3.2.1.6.Caso de Prueba Registrar Tipo Personal (CP06)

No. Caso de Prueba	CP06
---------------------------	-------------

Nombre	REGISTRAR TIPO PERSONAL
Descripción	Procedimiento para ingresar un nuevo Tipo Personal, para que se almacene en la base de datos y esté disponible al momento de registrar un nuevo Personal.
Entrada	Nombre: Contrato Descripción: Personal que labora en la institución por un período de tiempo establecido en un contrato.
Salidas	N/A
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ingrese al sistema con el perfil de Administrador. 2. Seleccione del menú la interfaz “Tipo Personal”. 3. En la interfaz seleccione el link “Nuevo Tipo Personal”. 4. En el panel “Ingresar Tipo Personal” que se despliega, ingrese los datos indicados en la entrada. 5. Seleccione el botón Aceptar, para registrar el nuevo tipo personal.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el Tipo Personal, y emite un mensaje de éxito.

Tabla 3. 8 Caso de Prueba Registrar Tipo Personal

3.2.1.7.Caso de Prueba Registrar Cargo Personal (CP07)

No. Caso de Prueba	CP07
Nombre	REGISTRAR CARGO PERSONAL
Descripción	Procedimiento para ingresar un nuevo Cargo Personal, para que se almacene en la base de datos y esté disponible al momento de registrar un nuevo Personal.
Entrada	Nombre: Secretaria Descripción: Persona que ejerce un trabajo en oficina y que es muy indispensable su labor dentro de la Unidad en la que se desempeña.
Salidas	N/A
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ingrese al sistema con el perfil de Administrador. 2. Seleccione del menú la interfaz “Cargo Personal”. 3. En la interfaz seleccione el link “Nuevo Cargo Personal”. 4. En el panel “Ingresar Cargo Personal” que se despliega, ingrese los datos indicados en la entrada. 5. Seleccione el botón Aceptar, para registrar el nuevo Cargo personal.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el Cargo Personal, y emite un mensaje de éxito.

Tabla 3. 9 Caso de Prueba Registrar Cargo Personal

3.2.1.8.Caso de Prueba Registrar Laboratorio (CP08)

No. Caso de Prueba	CP08
Nombre	REGISTRAR LABORATORIO
Descripción	Procedimiento para ingresar un nuevo Laboratorio, para que se almacene en la base de datos y esté disponible al momento de registrar un nuevo Personal.

Entrada	Unidad: CENTRO DE INVESTIGACION Y CONTROL AMBIENTAL Nombre: Centro de Investigación y Control Ambiental. Centro de Costo: bs-003. Responsable: Ing. Fierro. Mail: “ Dirección: “. Teléfono: “. Fax: “. Leyenda Proforma: “. Leyenda Orden T: “.
Salidas	N/A
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ingrese al sistema con el perfil de Secretaria. 2. Seleccione del menú la interfaz “Laboratorio”. 3. En la interfaz seleccione el link “Nuevo Laboratorio”. 4. En el panel “Ingresar Laboratorio” que se despliega, ingrese los datos indicados en la entrada. 5. Seleccione el botón Aceptar, para registrar el nuevo laboratorio.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el Laboratorio, y emite un mensaje de éxito.

Tabla 3. 10 Caso de Prueba Registrar Laboratorio

3.2.1.9.Caso de Prueba Registrar Tipo Personal (CP09)

No. Caso de Prueba	CP09
Nombre	REGISTRAR PERSONAL
Descripción	Procedimiento para ingresar un nuevo Personal, para que se almacene en la base de datos y esté disponible al momento de registrar un nuevo Personal.
Entrada	Laboratorio: CICAM Tipo Personal: Contrato Cargo Personal: Secretaria Nombre : Sra. Geoconda Silva Dirección: Solanda Teléfono: Email: gsilva@epn.edu.ec Sueldo : 700 Fecha Inicio: Fecha Fin: Login: secre Password: secre
Salidas	N/A
Procedimiento de Prueba	<ol style="list-style-type: none"> 6. Ingrese al sistema con el perfil de Administrador. 7. Seleccione del menú la interfaz “Personal”. 8. En la interfaz seleccione el link “Nuevo Personal”. 9. En el panel “Ingresar Personal” que se despliega, ingrese los datos indicados en la entrada. 10. Seleccione el botón Aceptar, para registrar el nuevo personal.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el Personal, y emite un mensaje de éxito.

Tabla 3. 11 Caso de Prueba Registrar Personal

3.2.1.10.Caso de Prueba Registrar Cliente (CP10)

No. Caso de Prueba	CP10
Nombre	REGISTRAR CLIENTE
Descripción	Procedimiento para ingresar un nuevo Cliente, que se almacena en la base de datos.
Entrada	Tipo Cliente: Externo Nombre: Asesoría Técnica Cia Ltda Cédula: 1716598751 Ruc: 1716598751001 Dirección: A. LINCOLN Teléfono: 023876198 Celular: 089562315 Otro Fono: 023696122 Email: asesoratec@hotmail.com Contacto: Ing. Mauricio Pérez
Salidas	N/A
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ingrese al sistema con el perfil de Secretaria. 2. Seleccione del menú la interfaz “Cliente”. 3. En la interfaz seleccione el link “Nuevo Cliente”. 4. En el panel “Ingresar Cliente” que se despliega, ingrese los datos indicados en la entrada. 5. Seleccione el botón Aceptar, para registrar el nuevo cliente.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el Cliente, y emite un mensaje de éxito.

Tabla 3. 12 Caso de Prueba Registrar Cliente

3.2.1.11.Caso de Prueba Registrar Unidad Medida (CP11)

No. Caso de Prueba	CP11
Nombre	REGISTRAR UNIDAD MEDIDA
Descripción	Procedimiento para ingresar una nueva Unidad Medida, para que se almacene en la base de datos y esté disponible al momento de registrar un Producto.
Entrada	Nombre: Kilogramos Descripción: unidad medida de masa Sigla: kg
Salidas	N/A
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ingrese al sistema con el perfil de Secretaria. 2. Seleccione del menú la interfaz “Unidad Medida”. 3. En la interfaz seleccione el link “Nueva Unidad Medida”. 4. En el panel “Ingresar Unidad Medida” que se despliega, ingrese los datos indicados en la entrada. 5. Seleccione el botón Aceptar, para registrar la nueva Unidad Medida.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda la Unidad Medida, y emite un mensaje de éxito.

Tabla 3. 13 Caso de Prueba Registrar Unidad Medida

3.2.1.12.Caso de Prueba Registrar Norma (CP12)

No. Caso de Prueba	CP12
Nombre	REGISTRAR NORMA
Descripción	Procedimiento para ingresar una nueva Norma, para que se almacene en la base de datos y esté disponible al momento de generar un informe de resultados.
Entrada	Nombre: NMX-AA-019-1985, Protección Al Ambiente Descripción: Contaminación Del Suelo - Residuos Sólidos Municipales - Peso Volumétrico "In Situ".
Salidas	N/A
Procedimiento de Prueba	6. Ingrese al sistema con el perfil de Secretaria. 7. Seleccione del menú la interfaz "Norma". 8. En la interfaz seleccione el link "Nueva Norma". 9. En el panel "Ingresar Norma" que se despliega, ingrese los datos indicados en la entrada. 10. Seleccione el botón Aceptar, para registrar la nueva Norma.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda la Norma, y emite un mensaje de éxito.

Tabla 3. 14 Caso de Prueba Registrar Norma

3.2.1.13.Caso de Prueba Registrar Producto (CP13)

No. Caso de Prueba	CP13
Nombre	REGISTRAR PRODUCTO
Descripción	Procedimiento para ingresar un nuevo Producto, para que se almacene en la base de datos y esté disponible al momento de registrar Métodos.
Entrada	Nombre: ACETATO DE PLOMO' en Nombre del Producto Unidad Medida: Kg. Descripción del Producto: ". Fórmula del Producto: ". Tipo Producto: Inflamable Riesgo Específico: S3I1R0. Riesgo Salud del Producto: S3 Riesgo Inflamable del Producto: I1. Riesgo Reactividad del Producto: R0
Salidas	N/A
Procedimiento de Prueba	11. Ingrese al sistema con el perfil de Analista. 12. Seleccione del menú la interfaz "Producto". 13. En la interfaz seleccione el link "Nuevo Producto". 14. En el panel "Ingresar Producto" que se despliega, ingrese los datos indicados en la entrada. 15. Seleccione el botón Aceptar, para registrar el nuevo Producto.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el Producto, y emite un mensaje de éxito.

Tabla 3. 15 Caso de Prueba Registrar Producto

3.2.1.14.Caso de Prueba Registrar Servicio (CP14)

No. Caso de Prueba	CP14
Nombre	REGISTRAR SERVICIO

Descripción	Procedimiento para ingresar un nuevo Servicio, para que se almacene en la base de datos y esté disponible al momento de registrar Métodos.
Entrada	Nombre: Cloro libre residual Tipo de Servicio: Análisis de Laboratorio Laboratorio: CICAM Descripción del Servicio: ‘’, Precio del Servicio: 40.5
Salidas	N/A
Procedimiento de Prueba	16. Ingrese al sistema con el perfil de Analista. 17. Seleccione del menú la interfaz “Servicio”. 18. En la interfaz seleccione el link “Nuevo Servicio”. 19. En el panel “Ingresar Servicio” que se despliega, ingrese los datos indicados en la entrada. 20. Seleccione el botón Aceptar, para registrar el nuevo Servicio.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el Servicio, y emite un mensaje de éxito.

Tabla 3. 16 Caso de Prueba Registrar Servicio

3.2.1.15.Caso de Prueba Registrar Método (CP15)

No. Caso de Prueba	CP15
Nombre	REGISTRAR MÉTODO
Descripción	Procedimiento para ingresar un nuevo Método, para que se almacene en la base de datos y esté disponible al momento de generar una Proforma.
Entrada	Método: <ul style="list-style-type: none"> • Nombre: Cloro libre residual • Servicio: Cloro libre residual • Descripción del Método: ‘’ • Certificado del Método Detalle Método: <ul style="list-style-type: none"> • Producto: CM-00001 • Unidad Medida: 1 • Cantidad: 0.034
Salidas	N/A
Procedimiento de Prueba	1. Ingrese al sistema con el perfil de Analista. 2. Seleccione del menú la interfaz “Método”. 3. En la interfaz seleccione el link “Nuevo Método”. 4. En el panel “Ingresar Método” que se despliega, ingrese los datos indicados en la entrada. 5. Seleccione el botón Aceptar, para registrar el Nuevo Método.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda Método, y emite un mensaje de éxito.

Tabla 3. 17 Caso de Prueba Registrar Método

3.2.1.16.Caso de Prueba Generar Proforma (CP16)

No. Caso de Prueba	CP16
Nombre	GENERAR PROFORMA

Descripción	Procedimiento para ingresar una nueva Proforma, para que se almacene en la base de datos y esté disponible al momento de generar un informe de resultados.
Entrada	Proforma: <ul style="list-style-type: none"> • Cliente: Asesoría Técnica Cia Ltda • Representante: Ing. Mauricio Pérez Panel Detalle Proforma: <ul style="list-style-type: none"> • Servicio : CM-S0001 • Método : CM-M0001 • Cantidad: 2
Salidas	Proforma: <ul style="list-style-type: none"> • Fecha Actual: dd/MM/yyyy • Subtotal : 81.0 • IVA: 9.72 • Total: 90.72 Detalle Proforma: <ul style="list-style-type: none"> • Laboratorio: CICAM • Valor Unitario: 40.5 • Total Servicio: 81.0
Procedimiento de Prueba	6. Ingrese al sistema con el perfil de Secretaria. 7. Seleccione del menú la interfaz “Proforma”. 8. En la interfaz seleccione el link “Nueva Proforma”. 9. En el panel “Ingresar Proforma” que se despliega, ingrese los datos indicados en la entrada. 10. Seleccione el botón Aceptar, para registrar la nueva Proforma.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda la Proforma, y emite un mensaje de éxito.

Tabla 3. 18 Caso de Prueba Generar Proforma

3.2.1.17.Caso de Prueba Registrar Muestra (CP17)

No. Caso de Prueba	CP17
Nombre	REGISTRAR MUESTRA
Descripción	Procedimiento para ingresar una nueva Muestra, para que se almacene en la base de datos y esté disponible al momento de generar una Orden de Trabajo.
Entrada	Muestra: <ul style="list-style-type: none"> • Nombre del Muestra : Aguas saladas • Factura: CM-F0001 • Fecha de Toma: 2010-04-09 • Tipo de Muestra: ‘--’ • Origen: Agua • Tipo de Envase: Plástico • Preservante: No • Tipo de Preservante: ‘--’ • Refrigeración: No Detalle Muestra: <ul style="list-style-type: none"> • Servicio: CM-S0001 • Método: CM-M0001
Salidas	N/A

Procedimiento de Prueba	11. Ingrese al sistema con el perfil de Secretaria. 12. Seleccione del menú la interfaz “Muestra”. 13. En la interfaz seleccione el link “Nueva Muestra”. 14. En el panel “Ingresar Muestra” que se despliega, ingrese los datos indicados en la entrada. 15. Seleccione el botón Aceptar, para registrar la nueva Muestra.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda Muestra, y emite un mensaje de éxito.

Tabla 3. 19 Caso de Prueba Registrar Muestra

3.2.1.18.Caso de Prueba Generar Orden Trabajo (CP19)

No. Caso de Prueba	CP19
Nombre	GENERAR ORDEN TRABAJO
Descripción	Procedimiento para ingresar una nueva Orden Trabajo, para que se almacene en la base de datos y esté disponible al momento de generar un informe de resultados.
Entrada	Selecciona Factura: CM-F001 Orden Trabajo: <ul style="list-style-type: none"> • Fecha de recepción de análisis o trabajo: 2010-04-15 • Fecha de propuesta de entrega de informe: 2010-04-1 • Número de Muestras: 1 Panel Detalle Orden: <ul style="list-style-type: none"> • Servicio : CM-S0001 • Método : CM-M0001
Salidas	Orden Trabajo: <ul style="list-style-type: none"> • Fecha Actual: dd/MM/yyyy Detalle Proforma: <ul style="list-style-type: none"> • Laboratorio: CICAM
Procedimiento de Prueba	16. Ingrese al sistema con el perfil de Secretaria. 17. Seleccione del menú la interfaz “Orden Trabajo”. 18. En la interfaz seleccione el link “Nueva Orden Trabajo”. 19. En el panel “Ingresar Orden Trabajo” que se despliega, ingrese los datos indicados en la entrada. 20. Seleccione el botón Aceptar, para registrar la nueva Orden Trabajo.
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda la Orden Trabajo, y emite un mensaje de éxito.

Tabla 3. 20 Caso de Prueba Generar Orden Trabajo

3.2.1.19.Caso de Prueba Asignar Carga Trabajo (CP20)

No. Caso de Prueba	CP20
Nombre	ASIGNAR CARGA TRABAJO
Descripción	Procedimiento para asignar Carga Trabajo, para que se almacene en el Detalle de la Orden de Trabajo para generar una Orden de Trabajo.
Entrada	Responsable3 Fecha de inicio: 2010-04-16 Fecha de Finalización: 2010-04-17 Horas/Trabajo: 4
Procedimiento de Prueba	21. Ingrese al sistema con el perfil de Secretaria.

	<p>22. Seleccione el link “Asignar Responsable”.</p> <p>23. En el panel “Asignar Responsable” que se despliega, ingrese los datos indicados en la entrada.</p> <p>24. Seleccione el botón Aceptar, para Asignar el responsable.</p>
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el responsable en el detalle de la Orden de Trabajo, y emite un mensaje de éxito.

Tabla 3. 21 Caso de Prueba Asignar Carga Trabajo

3.2.1.20.Caso de Prueba Generar Informe Resultados (CP21)

No. Caso de Prueba	CP21
Nombre	GENERAR INFORME RESULTADOS
Descripción	Procedimiento para ingresar un nuevo Informe Resultados, para que se almacene en la base de datos.
Entrada	<p>Selecciona Orden Trabajo: CM-OT001</p> <p>Informe Resultados:</p> <ul style="list-style-type: none"> • Muestra: CM-MU0001 <p>Panel Detalle Informe Resultados:</p> <ul style="list-style-type: none"> • Servicio : CM-S0001 • Método : CM-M0001 • Norma: 1 • Límite máximo permisible: < 1.5 • Resultado: 1.2 • Fecha del análisis: 2010-04-16
Salidas	<p>Orden Trabajo:</p> <ul style="list-style-type: none"> • Fecha Actual: dd/MM/yyyy • Cliente: Asesoría Técnica Cia Ltda • Dirección: A. LINCOLN • Teléfono: 023876198 • Contacto: Ing. Mauricio Pérez • Nombre del Muestra : Aguas saladas • Fecha de Toma: 2010-04-09 • Origen: Agua • Tipo de Envase: Plástico • Preservante: No • Tipo de Preservante • Refrigeración: No
Procedimiento de Prueba	<p>25. Ingrese al sistema con el perfil de Analista.</p> <p>26. Seleccione del menú la interfaz “Informe Resultados”.</p> <p>27. En la interfaz seleccione el link “Nuevo Informe Resultados”.</p> <p>28. En el panel “Ingresar Informe Resultados” que se despliega, ingrese los datos indicados en la entrada.</p> <p>29. Seleccione el botón Aceptar, para registrar el nuevo Informe Resultados.</p>
Resultado Esperado	El sistema valida que los campos requeridos estén llenos y que sean correctos. El sistema guarda el Informe Resultados, y emite un mensaje de éxito.

Tabla 3. 22 Caso de Prueba Generar Informe Resultados

3.2.1.21.Caso de Prueba Consultar Proformas Aceptadas (CP22)

No. Caso de Prueba	CP22
Nombre	CONSULTAR PROFORMAS ACEPTADAS
Descripción	Procedimiento para consultar las proformas que se encuentran en estado “Aceptada”.
Entrada	N/A
Salidas	Lista de Proformas almacenadas en el sistema con estado “Aceptada”
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Ingrese al sistema con el perfil de Tesorera. 2. Seleccione del menú la interfaz “Factura”. 3. En la interfaz seleccione el link “Nueva Factura”. 4. Seleccione consultar Proformas Aceptadas.
Resultado Esperado	El sistema despliega una lista de las Proformas que se encuentran en estado Aceptada.

Tabla 3. 23 Caso de Prueba Consultar Proformas Aceptadas

3.2.1.22.Caso de Prueba Generar Factura (CP23)

No. Caso de Prueba	CP23
Nombre	GENERAR FACTURA
Descripción	Procedimiento para generar una Factura, a partir de una proforma almacenada en la base de datos en estado “Aceptada”.
Entrada	Id Factura: F0001
Salidas	Fecha Actual: dd/MM/yyyy Panel Factura con todos los datos de la Proforma seleccionada
Procedimiento de Prueba	<ol style="list-style-type: none"> 1. Seleccione una de las Proformas en estado aceptada de la lista que Consultar Proformas Aceptadas. 2. Seleccione la opción “Generar Factura”. 3. Ingrese el dato indicado en la entrada, luego de que el Panel Factura indicado en la salida sea desplegado. 4. Pulsamos el botón “Aceptar”, para generar la nueva Factura
Resultado Esperado	El sistema valida que los campos ingresados estén llenos y que sean correctos. El sistema guarda la Norma, y emite un mensaje de éxito.

Tabla 3. 24 Caso de Prueba Generar Factura

CAPITULO 4: IMPLANTACIÓN Y EVALUACIÓN DE RESULTADO

Al iniciar el proceso de implantación y evaluación, nos encontramos en la fase de transición de acuerdo a RUP; durante esta fase, buscamos garantizar que se tiene un producto preparado para ser entregado al usuario, para lo cual se desarrollan una serie de actividades tales como la definición de un ambiente de pruebas y la evaluación de resultados después de que el usuario interactúa como el sistema.

4.1. IMPLANTACIÓN

4.1.1. DEFINICIÓN DEL AMBIENTE DE PRUEBAS

Una vez que el sistema cuenta con la funcionalidad requerida, se procede a la evaluación real del sistema, para lo cual se debe definir el ambiente de pruebas en el laboratorio tomado como base (CICAM).

Para la definición del ambiente de pruebas se tiene la especificación de Perfiles de Usuario, Definición de Usuarios del Sistema y descripción de hardware y software recomendado, los mismos que se muestran en las tablas 4.1, 4.2, 4.3, 4.4, 4.5 y 4.6.

4.1.1.1. Perfil de Usuario

Perfil de Usuario	Descripción
Administrador	Conocimiento del funcionamiento de los laboratorios de la EPN Conocimiento de computación Habilidades de comunicación con la interfaz mediante dispositivos de entrada: teclado y mouse
Jefe	Jefe del laboratorio o delegado asignado Conocimientos de los procesos desarrollados en el Laboratorio de Análisis. Conocimiento básico de computación Habilidades de comunicación con la interfaz mediante dispositivos de entrada: teclado y mouse
Analista	Persona que realiza el análisis de las muestras entregadas por los clientes Conocimiento de los análisis que se realiza en el laboratorio Conocimiento básico de computación Habilidades de comunicación con la interfaz mediante dispositivos de entrada: teclado y mouse
Secretaria/ Recepcionista	Conocimiento de funciones de secretaria/recepcionista en un centro de análisis de laboratorio Conocimiento básico de computación

	Habilidades de comunicación con la interfaz mediante dispositivos de entrada: teclado y mouse
Tesorera	Conocimiento de funciones de finanzas. Conocimiento básico de computación Habilidades de comunicación con la interfaz mediante dispositivos de entrada: teclado y mouse

Tabla 4. 1 Perfiles de Usuario Sugeridos para el uso del sistema

4.1.1.2.Usuarios

Nombre de Usuario	Cargo en la Institución	Perfil del Sistema
Ing. Geovanna Saltos		Administrador
Jefe	Jefe de Unidad de Laboratorio	
Analista	Analista de Laboratorio	Analista
Secretaria/Recepcionista	Secretaria del Laboratorio	Secretaria
Tesorera	Tesorera de la EPN	Tesorera

Tabla 4. 2 Usuario del sistema

4.1.1.3.Hardware recomendado

Servidor

Dispositivo	Descripción
Procesador	Velocidad de 2Ghz o superior
Memoria RAM	4 GB
Capacidad en Disco	500 GB mínimo recomendado
Monitor	Resolución: 1300 x 768 pixeles, 32 bits
Mouse	102 Teclas
Teclado	Genérico

Tabla 4. 3 Hardware recomendado para el Servidor

Terminal

Dispositivo	Descripción
Procesador	Velocidad de 2Ghz o superior
Memoria RAM	1 GB
Capacidad en Disco	120
Monitor	Resolución: 1300 x 768 pixeles, 32 bits
Impresora	Tecnología matricial o superior
Mouse	102 Teclas
Teclado	Genérico
Tarjeta de Red	100MB (Recomendado)

Tabla 4. 4 Hardware recomendado para los Terminales

4.1.1.4.Software recomendado

Nombre del Software	Descripción
Windows XP Profesional con service pack 1 o superior	Sistema operativo
Mozilla Firefox 3.5	Navegador Web
JBOSS 4.2.3.GA	Servidor de Aplicaciones
Postgres	Servidor de Base de datos
Controladores de impresora y dispositivos de	Controladores

entrada	
---------	--

Tabla 4. 5 Software recomendado para los Terminales

4.2. EVALUACIÓN DE RESULTADOS

A continuación se muestra la Evaluación del Sistema en el ambiente descrito anteriormente, se ejecuta el plan de pruebas que se muestra en la sección 3.2.1 con usuarios reales. A continuación se muestra el formato de presentación de evaluación de resultados.

Resultados	Descripción del Comportamiento del usuario frente al Sistema
MUY SATISFECHO	El Usuario utiliza el sistema sin ningún inconveniente frente a un sistema que funciona adecuadamente.
SATISFECHO	El Usuario utiliza el sistema presentando algunos inconvenientes pero que pueden ser superados con aprendizaje.
INSATISFECHO	El Usuario utiliza el sistema presentando inconvenientes graves o el sistema no cumple con el funcionamiento requerido.

Tabla 4. 6 Formato de Presentación Evaluación

4.2.1. RESULTADOS: USUARIO ADMINISTRADOR

Usuario	Administrador
<i>Caso de Prueba</i>	<i>Ingreso al Sistema</i>
Resultado	<ul style="list-style-type: none"> • Usuario ingresa de forma satisfactoria al sistema
<i>Caso de Prueba</i>	<i>Registrar Unidad</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
<i>Caso de Prueba</i>	<i>Registrar Tipo Cliente</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
<i>Caso de Prueba</i>	<i>Registrar Tipo Servicio</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
<i>Caso de Prueba</i>	<i>Administrar Tipo Producto</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
<i>Caso de Prueba</i>	<i>Administrar Tipo Personal</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
<i>Caso de Prueba</i>	<i>Administrar Cargo Personal</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
Conclusión de la Evaluación	MUY SATISFECHO

Tabla 4. 7 Resultados Evaluación Usuario Administrador

4.2.2. RESULTADOS: USUARIO SECRETARIA

Usuario	Secretaria
<i>Caso de Prueba</i>	<i>Ingreso al Sistema</i>
Resultado	<ul style="list-style-type: none"> • Usuario ingresa de forma satisfactoria al sistema
<i>Caso de Prueba</i>	<i>Registrar Laboratorio</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
<i>Caso de Prueba</i>	<i>Registrar Cliente</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
<i>Caso de Prueba</i>	<i>Registrar Personal</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
<i>Caso de Prueba</i>	<i>Administrar Norma</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
<i>Caso de Prueba</i>	<i>Administrar Unidad Medida</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
<i>Caso de Prueba</i>	<i>Administrar Proforma</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario selecciona los detalles de la Proforma de forma correcta. • Usuario pregunta si pueden ser modificados los datos ingresados. • Usuario guarda la Proforma sin inconvenientes.
<i>Caso de Prueba</i>	<i>Administrar Muestra</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona. • Usuario pregunta como ingresa Factura • Usuario selecciona factura con un poco de inconvenientes. • Usuario ingresa los servicios de la muestra con un poco de inconvenientes. • Usuario guarda la Muestra sin problema.
<i>Caso de Prueba</i>	<i>Administrar Orden Trabajo</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario pregunta cuales son las Facturas que se muestran. • Usuario selecciona una Factura de forma correcta. • Usuario ingresa la fecha de recepción y de entrega de resultado sin inconvenientes. • Usuario pregunta cómo se asigna un responsable. • Usuario Asigna el Responsable de realizar el análisis con algunos inconvenientes. • Usuario guarda la Orden de Trabajo de forma satisfactoria.
Conclusión de la Evaluación	SATISFECHO

Tabla 4. 8 Resultados Evaluación Usuario Secretaria

4.2.3. RESULTADOS: USUARIO TESORERA

Usuario	Tesorera
<i>Caso de Prueba</i>	<i>Ingreso al Sistema</i>
Resultado	<ul style="list-style-type: none"> • Usuario ingresa de forma satisfactoria al sistema
<i>Caso de Prueba</i>	<i>Gestionar Factura</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario selecciona la Proforma dando clic en generar Factura. • Usuario ingresa el código de la factura sin inconveniente. • Usuario guardar la factura de manera correcta.
Conclusión de la Evaluación	MUY SATISFECHO

Tabla 4. 9 Resultados Evaluación Usuario Tesorera

4.2.4. RESULTADOS: USUARIO ANALISTA

Usuario	Analista
<i>Caso de Prueba</i>	<i>Ingreso al Sistema</i>
Resultado	<ul style="list-style-type: none"> • Usuario ingresa de forma satisfactoria al sistema
<i>Caso de Prueba</i>	<i>Registrar Producto</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario ejecuta el proceso satisfactoriamente
<i>Caso de Prueba</i>	<i>Registrar Servicio</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona • Usuario guarda el Servicio sin problema.
<i>Caso de Prueba</i>	<i>Registrar Método</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona. • Usuario pregunta como ingresa el detalle. • Usuario selecciona producto sin inconveniente. • Usuario ingresa los datos del producto sin inconveniente. • Usuario guarda el Método sin problema.
<i>Caso de Prueba</i>	<i>Registrar Informe Resultados</i>
Resultado	<ul style="list-style-type: none"> • Usuario pregunta cómo funciona. • Usuario pregunta cuantos informes se pueden generar por orden de trabajo. • Usuario selecciona la orden de trabajo sin inconveniente. • Usuario selecciona la muestra de la cual se va a generar el Informe de Resultados. • Usuario pregunta cómo debe ingresar el respectivo resultado. • Usuario ingresa resultados con un poco de inconvenientes. • Usuario guardar el Informe de Resultado de forma correcta.
Conclusión de la Evaluación	SATISFECHO

Tabla 4. 10 Resultados Evaluación Usuario Analista

4.2.5. RESUMEN DE LA EVALUACIÓN DE RESULTADOS

Perfil	Conclusión de Evaluación
Administrador	MUY SATISFECHO
Secretaria	SATISFECHO
Tesorera	MUY SATISFECHO
Analista	SATISFECHO
CONCLUSION FINAL DEL SISTEMA	El Usuario requiere de aprendizaje para utilizar de forma correcta el sistema, el cual cumple con la funcionalidad requerida.

Tabla 4. 11 Resultados Evaluación Usuario

CAPITULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

En la actualidad el desarrollo de las tecnologías y el auge de las comunicaciones hacen que día a día empresa, industrias e instituciones, se vean en la necesidad de automatizar los procesos, de modo que esto les permita trabajar de forma más efectiva. Es por ello que el sistema desarrollado ha sido concebido con la idea de mejorar los procesos desarrollados en las diferentes Unidades de Laboratorio de la Escuela Politécnica Nacional, tomando como base el Centro de Investigación de Control Ambiental (CICAM), lo que proveerá un mayor índice de rendimiento con tiempos de ejecución mínimos en los procesos de atención al cliente y, manejo y accesibilidad a la información de manera efectiva.

El Sistema desarrollado fue concebido con la idea de que sea implementado en los treinta laboratorios con los que cuenta la Escuela Politécnica Nacional, basado su desarrollo en los principales procesos del Centro de Investigación de Control Ambiental, logrando así que los demás laboratorios puedan adaptar sus procesos al sistema. El Desarrollo el sistema como una primera versión, deja abierta la posibilidad de lanzar nuevas versiones hacia la adaptación de los requerimientos del resto de laboratorios con los que cuenta la institución.

El uso y difusión de herramientas de software libre permitirá que día y día sean más las empresas que opten por el uso de estas, dado que esto significa reducción de costo en cuanto a licenciamiento, independencia de proveedor y libertad de uso del código fuente. En los próximos años el auge de estas herramientas contribuirá a que el soporte a usuarios e información de uso de las mismas, incremente de manera significativa.

El Sistema de administración centralizada ha sido desarrollo bajo el lenguaje de programación Java, soportando el acceso a una base de datos postgresql, es por ello que se tiene varios beneficios debido a que se obtienen aplicaciones simples, familiares, seguras, independientes a la arquitectura, accesibles desde internet, de esta manera el usuario final recibe un producto de calidad con un esquema fácil de aprender y utilizar.

Desarrollar un software óptimo depende de un sin número de actividades y etapas, donde el impacto de elegir la metodología para el desarrollo de un proyecto es trascendental para el éxito o fracaso del mismo.

El desarrollo de un proyecto de software usando la metodología RUP proporciona la posibilidad de realizar una serie de interacciones a lo largo del proyecto, lo que posibilita la detección de posibles errores a tiempo, para realizar los respectivos cambios si fuera necesario. Lo que contribuye a reafirmar y automatizar satisfactoriamente los requerimientos del usuario final.

El levantamiento de los requerimientos debe ser bien detallado y completo de modo que no se den cambios a la mitad o al finalizar el proyecto de software, ya que esto produce demoras que afectan al calendario de desarrollo, produciendo gastos innecesarios. Para evitar este tipo de inconvenientes se debe realizar una evaluación continua en cada fase, de modo que se detecten errores a tiempo, y por ende los cambios sean más fáciles de realizar y manejar.

La etapa de pruebas es una de las más importantes en el desarrollo de un sistema debido a que se verifica la validez de la aplicación, y se procede a la depuración de los errores. En esta etapa se debe tener en cuenta como el Usuario interactúa con la aplicación si se encuentra satisfecho con el producto, porque de esta manera se determinará si el producto cumplió con los requerimientos establecidos en etapa inicial.

5.2. RECOMENDACIONES

Considerar la actualización constante en cuanto a tecnología, pues esto permitirá que las empresas automaticen sus procesos de manera satisfactoria, logrando así incrementar el rendimiento de producción y sus niveles de ingreso.

Es importante que previo al desarrollo de un proyecto de software se considere su alcance y prioridades para determinar que la metodología a aplicar es la más adecuada, logrando así que el desarrollo del sistema arroje un producto acorde a las necesidades del usuario final, con tiempos mínimos en la ejecución de las etapas de desarrollo.

En el desarrollo de software y de acuerdo a la metodología utilizada se especifican las etapas de desarrollo y sus respectivos entregables, sin embargo, adicionalmente se puede incorporar enfoques, procedimientos y entregables complementarios que nos permitirán mejorar la probabilidad de éxito del proyecto.

Tener en cuenta que el levantamiento de los requerimientos es uno de los aspectos más importantes y trascendentales, es por ello que es necesario mantener una interacción directa y una colaboración estrecha con el cliente, de manera que se entiendan las necesidades del cliente y se obtengan los requisitos necesarios para definir los objetivos que debe cumplir la aplicación y no se produzcan cambios al final del desarrollo ocasionando demora en la entrega del producto final.

Es importante considerar que el presente proyecto es una versión en primera fase del Sistema de Administración de Laboratorios de la Escuela Politécnica Nacional, por lo que se han tomado en cuenta los requerimientos generales del Centro de Investigación de Control Ambiental (CICAM), para que posteriormente se puedan integrar a su uso los demás laboratorios de la EPN.

La realización de las pruebas de la aplicación debe ser realizada a todos los usuarios que utilizaran el proyecto porque cada uno tiene diferente perspectiva y una manera diferente de entender cómo funciona la aplicación y de reaccionar ante posibles inconvenientes. Adicionalmente se debe probar absolutamente todo el proyecto porque al momento de poner en producción se pueden presentar problemas de desarrollo que no se detectaron en la etapa de pruebas.

GLOSARIO

Analista de Sistema.- El perfil establecido es: Ingeniero en Informática con conocimientos de UML, uno de ellos al menos con experiencia en sistemas afines a la línea del proyecto.

Analistas – Programadores.- Con experiencia en el entorno de desarrollo Java Eclipse, conocimientos de UML, con el fin de que los prototipos puedan ser lo más cercanos posibles al producto final.

APIs.- Interfaz de Programación de Aplicaciones, provista por los creadores del lenguaje Java, y que da a los programadores los medios para desarrollar aplicaciones Java, a través de un conjunto de clases utilitarias para efectuar toda clase de tareas necesarias dentro de un programa.

Artefacto.- Es el término general para cualquier producto del trabajo: código, gráficos Web, esquema de base de datos, documentos de texto, diagramas o modelos.

Casos de Prueba.- Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

Casos de uso.- Escenario que comienza cuando algún usuario del sistema inicia alguna transacción o secuencia de eventos interrelacionados. Es un patrón o ejemplo concreto de utilización. A medida que se pasa por cada escenario, se deben identificar los objetos que participan en él, las responsabilidades de cada objeto y como esos objetos colaboran con otros, en términos de operaciones que invoca cada uno sobre el otro.

Contenedor EJB: gestiona la ejecución de los EJB. Esta ejecución requiere de un Server Enterprise Edition.

Diseño.- Enfatiza en la solución conceptual (en software y hardware) que cumpla con los requerimientos, más que en su implementación. Por ejemplo, una descripción de un esquema de base de datos y objetos de software.

Disponibilidad.- El grado con el que se mide la accesibilidad de un sistema o de un componente cuando es necesario su uso. Suele expresarse en términos de probabilidad. Ver también: tolerancia a errores, tolerancia a fallos, robustez.

EJB: Enterprise Java Bean, es una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE. Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, precisamente, los EJB.

EJB Engine: Contenedor EJB reservado para aplicaciones desarrolladas alrededor de los Enterprise Java Bean.

Fully J2EE Compliant: término que implica que se cumplen todas las especificaciones J2EE definidas por Sun.

Glosario.- Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología adecuada.

Implementación.- Proceso de transformación de un diseño en componentes de hardware, software o de ambos. Ver también: codificación.

Interfaz.- Componente de hardware o software que conecta dos o más componentes con el propósito de transmitir información entre ellos.

JSP (Java Server Pages): es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Puede considerarse como una manera alternativa, y simplificada, de construir servlets.

Lista de Riesgos.- Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.

Modelo de Análisis y Diseño.- Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis, es

decir sin incluir aspectos de implementación hacia una de diseño que se oriente hacia el entorno de implementación, de acuerdo al avance del proyecto.

Modelo de Casos de Uso del Negocio.- Es uno de los artefactos, que tiene por objetivo permitir el modelamiento de casos de uso del negocio y los actores externos, de modo que se describa cómo es usado el Negocio por sus socios y clientes. Este modelo describe al Negocio en términos de un Diagrama de Casos de Uso del Negocio, que corresponden con los generalmente denominados "Procesos", usando estereotipos específicos para este modelo.

Modelo de Casos de Uso.- El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

Modelo de Datos.- De acuerdo a la permanencia de la información del sistema, este será soportado por una base de datos relacional. Este modelo permite representar la información del mundo real de una manera intuitiva, introduciendo conceptos cotidianos y fáciles de entender por cualquier inexperto, puesto que se basa en la lógica de predicados y la teoría de conjuntos. Para expresar este modelo se utilizará un Diagrama de Clases (para lo cual se utilizará una herramienta UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.)

Modelo de Despliegue.- Este modelo muestra los diagramas de despliegue que representan los nodos del sistema y sus relaciones, en estos nodos se hará el despliegue de los componentes. Generalmente, los nodos son conectados por asociaciones de comunicación como enlaces de red, conexiones TCP/IP, microondas, entre otros.

Modelo de Implementación.- Este modelo es un compendio de componentes y subsistemas que los contiene. Los componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es sólo una versión preliminar al final de la fase de Elaboración, posteriormente se tiene bastante refinamiento).

Producto de software.- Conjunto de programas, procedimiento y opcionalmente documentación asociada que se entrega al usuario como resultado.

Producto.- Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva release al final de cada iteración.

Prototipos de Interfaces de Usuario.- Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados. Asimismo, este artefacto, será desechado en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

Rational Unified Process (RUP).- Proceso de Ingeniería del Software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades en las organizaciones de desarrollo de software. Se trata de un proceso integrado en un producto, desarrollado y mantenido por Rational Software, e integrado en su conjunto de herramientas de desarrollo. Se

Registrar resultados.- El objetivo es permitir que un usuario analista de laboratorio registre los resultados obtenidos después de realizado el análisis correspondiente.

Requisito.- Condición o facultad que necesita un usuario para resolver un problema.

Servlet: es una clase Java que se ajuste de acuerdo a las especificaciones del Servlet API de Java, mecanismo por el cual una clase Java puede responder a peticiones HTTP.

Servlet Engine: Servlets compatibles con casi todos los servidores web disponibles

Sistema.- Conjunto de procesos, hardware, software, instalaciones y personas necesarios para

Subsistema.- Sistema subordinado a otro mayor.

BIBLIOGRAFÍA

1. ADICTOS AL TRABAJO.COM, TUTORIAL INTERCEPTAR WJB, <http://www.adictosaltrabajo.com/tutoriales/interceptarEJB/interc2.jpg>, Fecha: 07-11-2009
2. ALEXANDER AGUILAR VARGAS, Administración de Proyectos de Software, <http://www.scribd.com/doc/3175911/Administracion-de-Proyectos-de-Software>, Fecha: 31-02-2010.
3. CARLOS PROAL AGUILAR, Modelo de Datos, <http://ict.udlap.mx/people/carlos/is341/bases02.html>, fecha: 12-05-2010
4. CENTRO DE INVESTIGACIÓN Y CONTROL AMBIENTAL, Manual de Calidad del CICAM, fecha: 28-08-2009
5. CIBERTEC, La ingeniería de Software y RUP, <http://www.slideshare.net/dersteppenwolf/la-ingeniera-de-software-y-rup>, fecha: 15-09-2009
6. CODE JAVA, J2EE 5.0 – Java 2 Enterprise Edition 5.0, http://www.codejava.org/detalle_nota.htm?idxnota=51844&destacada=1, fecha: 30-10-2009
7. COMPUTO ACADÉMICO UNAM, J2EE. Una plataforma para el cómputo empresarial, <http://www.enterate.unam.mx/Articulos/2003/junio/j2ee.htm>, fecha: 27-10-2009
8. COMUNIDAD DE PROFESIONALES DE IT, Aplicaciones Java con Eclipse, <http://www.conexionit.com/blog/java/aplicaciones-java-con-eclipse.html>, fecha: 05-11-2009
9. COMUNIDAD DE PROFESIONALES DE IT, Que es RUP, <http://www.conexionit.com/blog/metodologias/que-es-rup.html>, fecha: 17-09-2009
10. CULEBRO JUÁREZ, GÓMEZ HERRERA, TORRES SÁNCHEZ, Software Libre vs Software Propietario, <http://www.softwarelibre.cl/drupal//files/32693.pdf>, fecha: 30-09-2009
11. DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN DEL INSTITUTO TECNOLÓGICO DE LA PAZ, http://sistemas.itlp.edu.mx/tutoriales/basedat1/tema1_4.htm, fecha: 12-05-2010

12. DEPT. INFORMÁTICA, UNIVERSITAT DE VALÈNCIA, Eclipse y Java, <http://informatica.uv.es/it3guia/LP/laboratorio/P1/TutorialEclipse.pdf>, fecha: 02-11-2009
13. DOMINGO HERNÁNDEZ H, Introducción al UML, http://www.magma.com.ni/~jorge/Fupoli_uml/Frefs/FIntroduccion_UML.ppt, fecha: 05-05-2010
14. FAIRLEY, R. "Risk Management for Software Development" en Software Engineering editado por Dorffman, M y Thayer, R.H., IEEE Computer Society, 1997
15. GNU OPERATING SYSTEM, La Definición de Software Libre, <http://www.gnu.org/philosophy/free-sw.es.html>, fecha: 25-09-2009
16. HUMBERTO ESPINOZA, PostgreSQL, http://www.lgs.com.ve/pres/PresentacionES_PSQL.pdf, fecha: 05-1-2009
17. IBM, Rational Software, <http://www-01.ibm.com/software/rational/>, fecha: 17-11-2009
18. ING. MA. DE PILAR RAMÍREZ GIL, Ingeniería de Software, <http://www.monografias.com/trabajos34/ingenieria-software/ingenieria-software.shtml>, fecha: 10-05-2010
19. ING. MA. DE PILAR RAMÍREZ GIL, Ingeniería de Software, <http://www.monografias.com/trabajos34/ingenieria-software/ingenieria-software.shtml>, fecha: 10-05-2010
20. INGENIERÍA DEL SOFTWARE, Diagrama de Componentes y Despliegue
21. JACOBSON, I., BOOCH, G., RUMBAUGH, J. (2000). El proceso unificado de desarrollo de software. Addison-Wesley, Madrid.
22. JAVIER TUREGANO MOLINA, Administrando JBoss, <http://www.slideshare.net/setoide/administrando-jboss-3316944>, fecha: 15-11-2009
23. JESÚS GARCÍA MOLINA, M. JOSÉ ORTÍN, BEGOÑA MOROS, JOAQUÍN NICOLÁS, AMBROSIO TOVAL, De los Procesos del Negocio a los Casos de Uso, http://www.willydev.net/descargas/willydev_modeladodenegocio.pdf, fecha: 25-02-2010
24. JONES, C. Assesment and Control of Software Risks. Yourdon Press, Prentice Hall, 1994.

25. JOSÉ DE JESÚS RODRIGUEZ VELEZ, Administración de proyectos de desarrollo de sistemas de información, <http://www.gestiopolis.com/recursos2/documentos/fulldocs/ger/adproyisisinf.htm>, Fecha: 03-01-2010
26. MARC GIBERT GINESTÀ, OSCAR PÉREZ MORA, Base de datos en PostgreSQL, http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02152.pdf, fecha: 07-11-2009
27. MARCOS FOUCES LAGO, Diagramas de Secuencia, <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html#sequence-diagram>, fecha: 25-04-2010
28. MARCOS FOUCES LAGO, Diagramas de Actividad, <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html#activity-diagram>, fecha: 15-04-2010
29. MSC. MANUEL SANCHEZ CHERO, Desarrollo de un Sistema de Gestión, <http://www.slideshare.net/msch/utilizando-metodologia-rup-parte1>, fecha: 21-02-2010
30. MSC. MARTHA D. DELGADO DAPENA, Definición del modelo del negocio y del dominio utilizando Razonamiento Basado en Casos, <http://www.inf.udec.cl/~revista/ediciones/edicion8/Rbc.pdf>, fecha: 05-03-2010
31. MTBASE SYBASE DE COLOMBIA, Power Designer, <http://www.mtbase.com/productos/modelamientometadatos/powerdesigner>, fecha: 15-11-2009
32. OBJECTS AND DESIGN, Herramientas basadas en UML, www.objectsbydesign.com/tools/umltools_byPrice.html, fecha: 10-05-2010
33. OSMOSIS LATINA, diagrama de Actividad, <http://www.osmosislatina.com/lenguajes/uml/actividad.htm>, fecha: 22-04-2010
34. OSMOSIS LATINA, Guía de Jboss, URL: <http://www.osmosislatina.com/jboss/basico.htm>, fecha: 07-11-2009
35. PHILIPPE KRUCHTEN, THE RATIONAL UNIFIED PROCESSES AN INTRODUCTION, THIRD EDITION, and EDITORIAL: Addison-Wesley
36. PRESSMAN, R. (2001). Ingeniería del software. Un enfoque práctico”, 5a ed. McGraw-Hill.

37. RAMIRO LAGO, ARQUITECTURA EJ2EE, [http://www.proactiva-
calidad.com/java/arquitectura/index.html](http://www.proactiva-
calidad.com/java/arquitectura/index.html), Fecha: 30-10-2009
38. RATIONAL SOFTWARE CORPORATION, RATIONAL ROSE, http://www.slideshare.net/vivi_jocadi/rational-rose, fecha: 15-11-2009
39. RED HAT, JBOSS, Plataforma de Aplicaciones JBoss Enterprise, http://www.es.redhat.com/pdf/jboss/JBoss_Ent_app_platform_ES_web.pdf,
fecha: 15-11-2009
40. RICARDO BORILLO DOMENECH, Introducción al modelado, <http://www.scribd.com/doc/7090055/Clase-de-UML>, fecha: 05-05-2010
41. Rubén González Blanco, Sergio Pérez Tobalina, UNIVERSIDAD POLITECNICA DE CATALUNYA, Introducción a Rational Rose, <http://www.google.com.ec/url?sa=t&source=web&cd=3&ved=0CBsQFjAC&url=http%3A%2F%2Fwww.essi.upc.edu%2F~ese%2Fweb%2Fdocuments%2Ftab%2F0304Q2%2Flessons%2Flese-2%2FLESE-2%2520-%2520Introduccion%2520a%2520Rational%2520Rose.ppt&ei=uwl2TL6FOoP98AaV3LjMAw&usg=AFQjCNEz64ZXV5AtmvkH1QHjN5pJT-mhdA>, fecha: 17-11-2009
42. RUEDA CHACÓN JULIO CÉSAR, Aplicación de la Metodología RUP para el desarrollo rápido de aplicaciones basado en el estándar j2ee, http://biblioteca.usac.edu.gt/tesis/08/08_7691.pdf, fecha: 30-09-2009.
43. SOLUCIONES RACIONALES, RUP/EASY guía metodológica de desarrollo de sistemas, http://www.solucionesracionales.com/archivos/rupe_guia.pdf, fecha: 20-04-2010.
44. UNIDAD DE GESTIÓN DE INFORMACIÓN, Proyecto para la implementación de un sistema de administración centralizada de laboratorios en la Escuela Politécnica Nacional, fecha: 28-09-2009
45. UNIVERSIDAD SALESIANA DE BOLIVIA, INGENIERÍA DE SISTEMAS, Diagrama de Componentes, <http://virtual.usalesiana.edu.bo/web/practica/archiv/dcompone.doc>, fecha: 10-05-2010

46. VERÓNICA MACÍAS MENDOZA, Modelamiento Basado en el Dominio, <http://www.fiec.espol.edu.ec/resources/investigacion/articulo16.pdf>, fecha: 05-03-2010
47. WAPEDIA MOBI, Software Libre, http://wapedia.mobi/es/Software_libre, fecha: 25-09-2009
48. WIKIPEDIA, Diagrama de clases, http://es.wikipedia.org/wiki/Diagrama_de_clases, fecha: 07-05-2010
49. WIKIPEDIA, diagrama de despliegue, http://es.wikipedia.org/wiki/Diagrama_de_despliegue, fecha: 11-05-2010
50. WIKIPEDIA, Diagrama de Secuencia, http://es.wikipedia.org/wiki/Diagrama_de_secuencia, fecha: 12-04-2010
51. WIKIPEDIA, Eclipse (Software), [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software)), fecha: 02-11-2009
52. WIKIPEDIA, Java Community Process, http://es.wikipedia.org/wiki/Java_Community_Process, Fecha: 30-10-2009
53. WIKIPEDIA, Java EE, http://es.wikipedia.org/wiki/Java_EE, Fecha: 30-10-2009
54. WIKIPEDIA, Modelo del Negocio, http://es.wikipedia.org/wiki/Modelo_de_negocio, fecha: 20-02-2010
55. WIKIPEDIA, Modelo Relacional, http://es.wikipedia.org/wiki/Modelo_relacional, fecha: 10-05-2010
56. WIKIPEDIA, Modelo Vista Controlador, http://es.wikipedia.org/wiki/Modelo_Vista_Controlador, fecha: 20-05-2010
57. WIKIPEDIA, Power Designer, <http://en.wikipedia.org/wiki/PowerDesigner>, fecha: 15-11-2009
58. WIKIPEDIA, Proceso Unificado de Rational, http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational, fecha: 15-09-2009
59. WIKIPEDIA, Rational Rose, http://es.wikipedia.org/wiki/Rational_Software, fecha: 17-11-2009
60. WIKIPEDIA, Software Libre, http://es.wikipedia.org/wiki/Software_libre, fecha: 25-09-2009
61. YAMIL RAMOS, PILAR STRONGUILO, Modelo del Negocio con RUP y UML, <http://www.slideshare.net/david.motta/modelo-del-negocio-con-rup-y-uml-parte-3-1534304> , fecha: 20-02-2010

62.Y. RUMBAUGH, M. BLAHA, W. PREMERLANI, F. EDDY Y W. LORENSEN.
Object-oriented modeling and design. Prentice- Hall.

ANEXOS

A continuación se detallan los archivos que se encuentran en el CD como anexos:

- Casos de Uso y diagramas
- Modelo de Datos Relacional
- Base de Datos
- Código Fuente
- SisLab.war
- Plantillas de artefactos de acuerdo a la Metodología RUP