



UNIVERSIDAD NACIONAL DE LOJA

**ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES**

INGENIERÍA EN SISTEMAS

OPENLOJA 1.0

“Sistema Planificador de Recursos Empresariales (ERP) integrado con módulo de comunicación de clientes para la empresa de servicios informáticos Lojanet++ Cía. Ltda.”

Tesis previa a la obtención del
Título de Ingeniero en Sistemas

AUTORES:

Vladimir Yazber Romero Gonzaga.

Jorge Luis Tapia Guarnizo.

DIRECTOR

Ing. Wilman Patricio Chamba Zaragocín.

Loja-Ecuador

2012



CERTIFICACIÓN

Ing. Wilman Patricio Chamba Zaragocín.

**DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS DE LA
UNIVERSIDAD NACIONAL DE LOJA**

CERTIFICA:

Que el presente trabajo con el tema “**Sistema Planificador de Recursos Empresariales (ERP) integrado con módulo de comunicación de clientes para la empresa de servicios informáticos Lojanet++ Cía. Ltda.**”, previo a la obtención del título de ingeniero en sistemas, fue prolijamente dirigido, supervisado y revisado en cada uno de sus aspectos; razón por la que doy su validez y justificación, autorizando su presentación, sustentación y publicación.

.....
Ing. Wilman Patricio Chamba Zaragocín

DIRECTOR



AUTORÍA

Todos los conceptos, criterios y opiniones vertidos en este trabajo son de exclusiva responsabilidad de los autores.

Vladimir Yazber Romero Gonzaga

Jorge Luis Tapia Guarnizo



AGRADECIMIENTO

A todas las personas que hicieron posible la realización de este proyecto, de manera especial a nuestro director de tesis Ing. Wilman Chamba Zaragocín que con su paciencia, conocimientos y experiencia nos permitieron desarrollar y culminar este proyecto.

Así mismo extendemos un agradecimiento al Área de la Energía, las Industrias y los Recursos Naturales no Renovables, Carrera de Ingeniería en Sistemas quienes nos acogieron colaborando con nuestra formación profesional.

Vladimir Yazber Romero Gonzaga

Jorge Luis Tapia Guarnizo



DEDICATORIA

El presente proyecto va dedicado primeramente a Dios ya que gracias a su ayuda se ha logrado su culminación, a mi madre a quien apoyo e incentivo sirvieron para alcanzar los objetivos propuestos.

Jorge Tapia

El presente trabajo va dedicado a mis padres, hermanos y tíos, que con su apoyo incondicional supieron brindarme en todos los momentos de mi vida la fuerza necesaria para culminar mis objetivos, de manera muy especial a Dios y mi madre que siempre están en cada etapa de mi existencia.

Yazber Romero



CESIÓN DE DERECHOS

El presente trabajo queda a disposición de la Universidad Nacional de Loja, para que pueda determinar su utilización en los fines que considere pertinentes.

.....
Vladimir Yazber Romero Gonzaga

.....
Jorge Luis Tapia Guarnizo

RESUMEN

El sistema OpenLoja 1.0 es una aplicación que se ejecuta bajo entorno web, creado considerando la metodología ICONIX y desarrollado bajo el Marco de Trabajo(Framework) Seam , llevando consigo el paradigma dentro de las líneas del OpenSource, fue desarrollado con el objetivo de dar solución a manejo de la información de una forma ordenada y confiable, colaborando con cada uno de los procesos dentro de la Empresa de Servicios Informáticos Lojanet++, programado en módulos que permitan una mejor comprensión y manipulación de la aplicación, el presente sistema se encuentra cubriendo las áreas de empleados, clientes, proveedores, bodega, venta y comunicación; como punto de partida en el desarrollo de una solución robusta, confiable y sobre todo adaptable que permita el normal funcionamiento en la empresa, buscando la disponibilidad de acceso a usuarios desde cualquier parte del mundo.

El desarrollo de este sistema permite a sus autores cumplir uno de los requisitos propuestos por la Universidad Nacional de Loja, que consiste en el desarrollo de un proyecto investigativo una vez culminados todos los módulos de estudio, y así obtener el título de ingeniero en Sistemas, razones por las cuales se crea el Sistema Planificador de Recursos Empresariales OpenLoja 1.0.

La información contenida en el presente trabajo puede servir de apoyo para el futuro crecimiento de este proyecto mediante la integración de nuevos módulos, ya que su diseño le permite ser escalable o como base para la realización de aplicaciones similares.

SUMMARY

OpenLoja System 1.0 is an application that runs under web environment, created considering the ICONIX methodology and developed under the Framework Seam, carrying with it the paradigm within the lines of OpenSource, was developed with the aim of solving the management of information in an orderly and reliable, collaborating with each of the processes within the computer services company Lojanet++ , programed in modules that enable better understanding and manipulation of the application, this system is covering the areas of employees, customers, suppliers, warehouse, sales and communication, as a starting point in developing a robust solution, reliable and above all adaptable to allow the normal functioning in the company, looking for the availability of access to users from anywhere in the world.

The development of this system allows the authors fulfill one of the requirements proposed by the National University of Loja, that is to develop a research project upon completion of all modules of study, and get the title of systems engineers, for this reason is created Enterprise Resource Planning System OpenLoja 1.0.

The information contained in this work can support the future growth of this project through the integration of new modules, since its design allows it to be scalable or as the basis for the realization of similar applications.



ÍNDICE

ÍNDICE GENERAL

AUTORÍA.....	III
AGRADECIMIENTO	IV
DEDICATORIA	V
CESIÓN DE DERECHOS.....	VI
RESUMEN.....	VII
SUMMARY	VIII
ÍNDICE	IX
A. INTRODUCCIÓN.....	1
B. METODOLOGÍA	3
B.1. MATERIALES Y MÉTODOS.....	3
B.1.1. Métodos.....	3
B.1.2. Técnicas.....	5
B.1.3. Materiales.....	6
C. REVISIÓN DE LITERATURA.....	8
1. SISTEMA PLANIFICADOR DE RECURSOS EMPRESARIALES (ERP).....	8
1.1. INTRODUCCIÓN.....	8
1.2. DEFINICIÓN.....	8
1.3. TABLA DE COMPARACIÓN DE ERPs	9
1.4. ¿POR QUÉ UTILIZAR EL ERP OPENLOJA?.....	10
1.4.1. Aumentar la Competitividad.....	10
1.4.2. Agilizar Operaciones.....	10
1.4.3. Integración.....	10
1.5. OBJETIVOS PRINCIPALES DE OPENLOJA.....	11
1.6. CARACTERÍSTICAS PRINCIPALES DE OPENLOJA.....	11



1.7.	MÓDULOS DE OPENLOJA.....	12
1.8.	CONCLUSIONES DE OPENLOJA.....	13
2.	JBOSS SEAM FRAMEWORK.....	14
2.1.	INTRODUCCIÓN.....	14
2.2.	DEFINICIÓN.....	14
2.3.	CARACTERÍSTICAS.....	15
2.4.	SISTEMAS.....	16
2.4.1.	SISTEMA DE SEGURIDAD.....	16
2.4.2.	SISTEMA DE PLANTILLAS (TEMPLATES).....	16
2.4.3.	SISTEMA DE CACHÉ.....	16
3.	HIBERNATE.....	17
3.1.	DEFINICIÓN.....	17
3.2.	ARQUITECTURA DE OPENLOJA.....	18
3.2.1.	PATRÓN MODELO VISTA CONTROLADOR (MVC).....	18
3.2.2.	DESCRIPCIÓN DEL PATRÓN.....	18
3.2.2.1.	MODELO.....	18
3.2.2.2.	VISTA.....	20
3.2.2.3.	CONTROLADOR.....	21
3.3.	INTRODUCCIÓN ANOTACIONES.....	22
4.	RICHFACES.....	27
4.1.	INTRODUCCIÓN.....	27
4.2.	CARACTERÍSTICAS.....	27
4.3.	COMPONENTES.....	29
4.4.	INCONVENIENTES.....	31
4.5.	REQUISITOS E INCOMPATIBILIDADES.....	32
4.6.	INTERACCIÓN CON OTROS COMPONENTES.....	33
4.6.1.	Sun JSF RI.....	33
4.6.2.	Apache MyFaces.....	33



4.6.3. Facelets	33
4.6.4. JBoss Seam.....	33
4.7. RECOMENDACIONES DE USO.....	34
4.8. Enterprise JavaBeans.	34
4.8.1. Definición.	35
4.8.2. Tipos de Enterprise JavaBeans.....	35
4.8.2.1. EJB de Entidad(<i>EntityEJBs</i>):.....	36
4.8.2.2. EJB de Sesión (<i>SessionEJBs</i>):.....	36
4.8.2.3. EJB dirigidos por mensajes (<i>Message-drivenEJBs</i>):.....	37
4.8.3. Funcionamiento de un Enterprise JavaBean.....	37
4.8.4. Interfaz "Home"	38
4.8.5. Interfaz remota.....	38
4.8.6. Clase de implementación EJB.....	38
4.8.7. Correspondencia entre métodos de interfaz y métodos de implementación.	39
D. RESULTADOS	40
1. DESARROLLO DE LA PROPUESTA ALTERNATIVA	40
2. VALORACIÓN TÉCNICO ECONÓMICA AMBIENTAL.....	41
2.1. Académica	41
2.2. Científico – Técnica.....	42
2.3. Económica	42
E. DISCUSIÓN.....	43
1. REQUERIMIENTOS DEL SISTEMA.....	44
1.1. REQUERIMIENTOS FUNCIONALES.....	44
1.2. REQUERIMIENTOS NO FUNCIONALES	45
2. IDENTIFICACIÓN DE LOS ACTORES Y CASOS DE USO DE OPENLOJA	47
3. DISEÑO DE LOS MÓDULOS DEL SISTEMA.....	53
3.1. MÓDULO DE GESTIÓN DE CONFIGURACIÓN.....	53
3.1.1. MODELO DE DOMINIO	53



3.1.2. MODELO DE CASOS DE USO DE L SISTEMA.	54
3.1.2.1. DIAGRAMA DE CASOS DE USO.....	54
3.1.2.2. DESCRIPCIÓN DE CASOS DE USO	55
• CASO DE USO: Iniciar Sesión	55
• CASO DE USO: Cerrar Sesión.....	56
• CASO DE USO: Configurar Información General	57
• CASO DE USO: Crear Cuenta Usuario	59
• CASO DE USO: Modificar Cuenta Usuario	61
• CASO DE USO: Consultar Cuenta Usuario.....	63
3.1.3. MODELO DE INTERACCIÓN	64
3.1.3.1. DIAGRAMA DE SECUENCIA 001: Iniciar Sesión (DS001)	64
3.1.3.2. DIAGRAMA DE SECUENCIA 002: Cerrar Sesión (DS002)	65
3.1.3.3. DIAGRAMA DE SECUENCIA: Configurar Información General (DS003)	66
3.1.3.4. DIAGRAMA DE SECUENCIA 004: Crear Usuario (DS004)	67
3.1.3.5. DIAGRAMA DE SECUENCIA 005: Modificar Cuenta Usuario (DS005)..	68
3.1.3.6. DIAGRAMA DE SECUENCIA 006: Consultar Cuenta Usuario (DS006)....	69
3.1.4. DIAGRAMA DE CLASES FINAL	70
3.1.4.1. CLASES MODELO	70
3.1.4.2. CLASES DAO.....	71
3.1.4.3. CLASES CONTROLADOR	72
3.1.4.4. CLASES VISTAS.....	73
3.1.5. DISEÑO DE DATOS	74
3.1.5.1. MODELO ENTIDAD-RELACIÓN	74
3.1.5.2. MODELO CONCEPTUAL	74
3.1.5.3. MODELO RELACIONAL.....	77
3.2. MÓDULO DE GESTIÓN DE EMPLEADOS.....	79
3.2.1. MODELO DE DOMINIO	79



3.2.2. MODELO DE CASOS DE USO DEL SISTEMA.	80
3.2.2.1. DIAGRAMA DE CASOS DE USO.....	80
3.2.2.2. DESCRIPCIÓN DE CASOS DE USO	81
• CASO DE USO: Crear Empleado.....	81
• CASO DE USO: Modificar Empleado	83
• CASO DE USO: Consultar Empleado	85
• CASO DE USO: Crear Departamento	86
• CASO DE USO: Asignar Departamento	88
• CASO DE USO: Modificar Departamento.....	89
• CASO DE USO: Crear Horario	91
• CASO DE USO: Consultar Departamento	92
• CASO DE USO: Asignar Horario.....	93
• CASO DE USO: Registrar Entrada Salida.....	95
• CASO DE USO: Crear Permiso.....	99
• CASO DE USO: Modificar Permiso	101
• CASO DE USO: Consultar Permiso	103
• CASO DE USO: Crear Capacitación	105
• CASO DE USO: Modificar Capacitación.....	107
• CASO DE USO: Consultar Capacitación.....	109
• CASO DE USO: Crear Anticipo.....	111
• CASO DE USO: Modificar Anticipo	113
• CASO DE USO: Consultar Anticipo.....	115
• CASO DE USO: Generar Rol Individual	117
• CASO DE USO: Generar Rol General.....	119
3.2.3. MODELO DE INTERACCIÓN	121
3.2.3.1. DIAGRAMA DE SECUENCIA 007: Crear Empleado (DS007).....	121
3.2.3.2. DIAGRAMA DE SECUENCIA 008: Modificar Empleado (DS008).....	122



3.2.3.3.	DIAGRAMA DE SECUENCIA 009: Consultar Empleado (DS009)	123
3.2.3.4.	DIAGRAMA DE SECUENCIA 010: Crear Departamento (DS010).....	124
3.2.3.5.	DIAGRAMA DE SECUENCIA 011: Asignar Departamento (DS011)	125
3.2.3.6.	DIAGRAMA DE SECUENCIA 012: Modificar Departamento (DS012).....	126
3.2.3.7.	DIAGRAMA DE SECUENCIA 013: Consultar Departamento (DS013).....	127
3.2.3.8.	DIAGRAMA DE SECUENCIA 014: Crear Horario (DS014).....	128
3.2.3.9.	DIAGRAMA DE SECUENCIA 015: Asignar Horario (DS015).....	129
3.2.3.10.	DIAGRAMA DE SECUENCIA 016: Registro Entrada Salida (DS016)	130
3.2.3.11.	DIAGRAMA DE SECUENCIA 017: Consultar Asistencias (DS017)	131
3.2.3.12.	DIAGRAMA DE SECUENCIA 018: Crear Permiso (DS018)	132
3.2.3.13.	DIAGRAMA DE SECUENCIA 019: Modificar Permiso (DS019).....	133
3.2.3.14.	DIAGRAMA DE SECUENCIA 020: Consultar Permiso (DS020)	134
3.2.3.15.	DIAGRAMA DE SECUENCIA 021: Crear Capacitación (DS021)	135
3.2.3.16.	DIAGRAMA DE SECUENCIA 022: Modificar Capacitación (DS022).....	136
3.2.3.17.	DIAGRAMA DE SECUENCIA 023: Consultar Capacitación (DS023).....	137
3.2.3.18.	DIAGRAMA DE SECUENCIA 024: Crear Anticipo (DS024).....	138
3.2.3.19.	DIAGRAMA DE SECUENCIA 025: Modificar Anticipo (DS025)	139
3.2.3.20.	DIAGRAMA DE SECUENCIA 026: Consultar Anticipo (DS026).....	140
3.2.3.21.	DIAGRAMA DE SECUENCIA 027: Generar Rol Pago Individual (DS027)	141
3.2.3.22.	DIAGRAMA DE SECUENCIA 028: Generar Rol General (DS028)	142
3.2.4.	DIAGRAMA DE CLASES FINAL	143
3.2.4.1.	CLASES MODELO	143
3.2.4.2.	CLASES DAO.....	144
3.2.4.3.	CLASES CONTROLADOR	145
3.2.4.4.	CLASES VISTAS.....	146
3.2.5.	DISEÑO DE DATOS	147
3.2.5.1.	MODELO ENTIDAD-RELACIÓN	147
3.2.5.2.	MODELO CONCEPTUAL	147



3.2.5.3. MODELO RELACIONAL.....	152
3.3. MÓDULO DE GESTÓN DE CLIENTES.....	154
3.3.1. MODELO DE DOMINIO	154
3.3.2. MODELO DE CASOS USO DEL SISTEMA	155
3.3.2.1. DIAGRAMA DE CASOS DE USO.....	155
3.3.2.2. DESCRIPCIÓN DE CASOS DE USO	156
• CASO DE USO: Crear Cliente	156
• CASO DE USO: Modificar Cliente	157
• CASO DE USO: Consultar Cliente	159
3.3.3. MODELO DE INTERACCIÓN	161
3.3.3.1. DIAGRAMA DE SECUENCIA 029: Crear Cliente (DS029)	161
3.3.3.2. DIAGRAMA DE SECUENCIA 030: Modificar Cliente (DS030).....	162
3.3.3.3. DIAGRAMA DE SECUENCIA 031: Consultar Cliente (DS031)	163
3.3.4. DIAGRAMA DE CLASES FINAL	164
3.3.4.1. CLASES MODELO	164
3.3.4.2. CLASES DAO.....	165
3.3.4.3. CLASES CONTROLADOR	166
3.3.4.4. CLASES VISTA	167
3.3.5. DISEÑO DE DATOS	168
3.3.5.1. MODELO ENTIDAD-RELACIÓN	168
3.3.5.2. MODELO CONCEPTUAL.....	168
3.3.5.3. MODELO RELACIONAL.....	169
3.4. MÓDULO DE GESTIÓN DE PROVEEDORES.....	171
3.4.1. MODELO DE DOMINIO	171
3.4.2. MODELO DE CASOS DE USO DEL SISTEMA	172
3.4.2.1. DIAGRAMA DE CASOS DE USO.....	172
3.4.2.2. DESCRIPCIÓN DE CASOS DE USO	173
• CASO DE USO: Crear Proveedor	173



- CASO DE USO: Modificar Proveedor..... 175
- CASO DE USO: Consultar Proveedor 177
- 3.4.3. MODELO DE INTERACCIÓN 179
- 3.4.3.1. DIAGRAMA DE SECUENCIA 032: Crear Proveedor (DS032)..... 179
- 3.4.3.2. DIAGRAMA DE SECUENCIA 033: Modificar Proveedor (DS033)..... 180
- 3.4.3.3. DIAGRAMA DE SECUENCIA 034: Consultar Proveedor (DS034)..... 181
- 3.4.4. DIAGRAMA DE CLASES FINAL 182
- 3.4.4.1. CLASES MODELO 182
- 3.4.4.2. CLASES DAO..... 183
- 3.4.4.3. CLASES CONTROLADOR..... 184
- 3.4.4.4. CLASES VISTA 185
- 3.4.5. DISEÑO DE DATOS 186
- 3.4.5.1. MODELO ENTIDAD-RELACIÓN 186
- 3.4.5.2. MODELO CONCEPTUAL..... 186
- 3.4.5.3. MODELO RELACIONAL..... 187
- 3.5. MÓDULO DE GESTIÓN DE BODEGA..... 189
- 3.5.1. MODELO DE DOMINIO 189
- 3.5.2. MODELO DE CASOS DE USO DEL SISTEMA 190
- 3.5.2.1. DIAGRAMA DE CASOS DE USO..... 190
- 3.5.2.2. DESCRIPCIÓN DE CASOS DE USO 191
- CASO DE USO: Crear Producto 193
- CASO DE USO: Modificar Producto..... 195
- CASO DE USO: Consultar Producto 197
- CASO DE USO: Crear Movimiento Bodega 198
- CASO DE USO: Modificar Movimiento Bodega 202
- CASO DE USO: Consultar Movimiento Bodega 204
- CASO DE USO: Generar Inventario..... 206
- CASO DE USO: Generar Kárdex 207



3.5.3. MODELO DE INTERACCIÓN	209
3.5.3.1. DIAGRAMA DE SECUENCIA 035: Crear Familia de Producto (DS035) ..	209
3.5.3.2. DIAGRAMA DE SECUENCIA 036: Crear Producto (DS036)	210
3.5.3.3. DIAGRAMA DE SECUENCIA 037: Modificar Producto (DS037)	211
3.5.3.4. DIAGRAMA DE SECUENCIA 038: Consultar Producto (DS038)	212
3.5.3.5. DIAGRAMA DE SECUENCIA 039: Crear Movimiento Bodega(DS039)	213
3.5.3.6. DIAGRAMA DE SECUENCIA 040: Modificar Movimiento (DS040)	214
3.5.3.7. DIAGRAMA DE SECUENCIA 041: Consultar Movimientos Bodega (DS041)	215
3.5.3.8. DIAGRAMA DE SECUENCIA 042: Generar Inventario (DS042).....	216
3.5.3.9. DIAGRAMA DE SECUENCIA 043: Generar Kárdex (DS043)	217
3.5.4. DIAGRAMA DE CLASES FINAL	218
3.5.4.1. CLASES MODELO	218
3.5.4.2. CLASES DAO	219
3.5.4.3. CLASES CONTROLADOR	220
3.5.4.4. CLASES VISTA	221
3.5.5. DISEÑO DE DATOS	222
3.5.5.1. MODELO ENTIDAD-RELACIÓN	222
3.5.5.2. MODELO CONCEPTUAL	222
3.5.5.3. MODELO RELACIONAL.....	225
3.6. MÓDULO DE GESTIÓN DE VENTAS.....	227
3.6.1. MODELO DE DOMINIO	227
3.6.2. MODELO DE CASOS DE USO DEL SISTEMA	228
3.6.2.1. DIAGRAMA DE CASOS DE USO.....	228
3.6.2.2. DESCRIPCIÓN DE CASOS DE USO	229
• CASO DE USO: Crear Venta	229
• CASO DE USO: Modificar Venta.....	232
• CASO DE USO: Consultar Venta	234



3.6.3. MODELO DE INTERACCIÓN	236
3.6.3.1. DIAGRAMA DE SECUENCIA 044: Crear Venta (DS044)	236
3.6.3.2. DIAGRAMA DE SECUENCIA 045: Modificar Venta (DS045).....	237
3.6.3.3. DIAGRAMA DE SECUENCIA 046: Consultar Venta (DS046).....	238
3.6.4. DIAGRAMA DE CLASES FINAL	239
3.6.4.1. CLASES MODELO	239
3.6.4.2. CLASES DAO.....	240
3.6.4.3. CLASES CONTROLADOR	241
3.6.4.4. CLASES VISTA	242
3.6.5. DISEÑO DE DATOS	243
3.6.5.1. MODELO ENTIDAD-RELACIÓN	243
3.6.5.2. MODELO CONCEPTUAL.....	243
3.6.5.3. MODELO RELACIONAL.....	246
3.7. MÓDULO DE GESTIÓN DE COMUNICACIÓN.....	248
3.7.1. MODELO DE DOMINIO	248
3.7.2. MODELO DE CASOS DE USO DEL SISTEMA	249
3.7.2.1. DIAGRAMA DE CASOS DE USO.....	249
3.7.2.2. DESCRIPCIÓN DE CASOS DE USO	250
• CASO DE USO: Crear Notificación	250
• CASO DE USO: Crear Grupo Contacto.....	252
• CASO DE USO: Crear Contacto	254
• CASO DE USO: Modificar Contacto.....	255
3.7.3. MODELO DE INTERACCIÓN	258
3.7.3.1. DIAGRAMA DE SECUENCIA 047: Crear Notificación (DS047)	258
3.7.3.2. DIAGRAMA DE SECUENCIA 048: Crear Grupo contacto (DS048)	259
3.7.3.3. DIAGRAMA DE SECUENCIA 049: Crear Contacto (DS049)	260
3.7.3.4. DIAGRAMA DE SECUNCIA 050: Modificar Contacto (DS050)	261
3.7.4. DIAGRAMA DE CLASES FINAL	262



3.7.4.1.	CLASES DE MODELO	262
3.7.4.2.	CLASES DAO	263
3.7.4.3.	CLASES CONTROLADOR	264
3.7.4.4.	CLASES VISTA	265
3.7.5.	DISEÑO DE DATOS	266
3.7.5.1.	MODELO ENTIDAD-RELACIÓN	266
3.7.5.2.	MODELO CONCEPTUAL	266
3.7.5.3.	MODELO RELACIONAL	268
4.	DISEÑO DE LA ARQUITECTURA	269
4.1.	DIAGRAMA DE COMPONENTES	269
4.2.	DIAGRAMA DE DESPLIEGUE	270
4.3.	DIAGRAMA DE PAQUETES	270
4.4.	MODELO DE LA ARQUITECTURA	271
5.	DISEÑO NAVEGACIÓN	273
5.1.	INTRODUCCIÓN	273
5.2.	CABECERA	274
5.3.	MENÚ	275
5.4.	BLOQUE PRINCIPAL	276
5.5.	PIE DE PÁGINA	278
6.	PLAN DE VALIDACIÓN	280
6.1.	INTRODUCCIÓN	280
6.2.	PRUEBAS DE INTERFACES GRÁFICAS	280
6.3.	PRUEBAS DE FUNCIONALIDAD	284
6.3.1.	PRUEBAS DE SEGURIDAD	284
6.4.	TEST DE EVALUACIÓN DEL SISTEMA	287
G.	CONCLUSIONES	290
H.	RECOMENDACIONES	291
I.	BIBLIOGRAFÍA	292



J. ANEXOS.....	294
1. ANTEPROYECTO	294
1. TEMA	296
2. PROBLEMÁTICA	297
2.1. Antecedentes.....	297
2.2. Situación Problemática.....	297
2.3. Problema de Investigación.....	298
2.4. Delimitación.....	299
3. JUSTIFICACIÓN	300
3.1. Académica	300
3.2. Científico – Técnica.....	300
3.3. Económica	301
4. OBJETIVOS	302
4.1. Objetivo General	302
4.2. Objetivos Específicos	302
5. MARCO TEÓRICO.....	303
5.1. Esquema del Marco Teórico.....	303
CAPITULO I.....	306
1. SISTEMA DE PLANIFICACIÓN DE RECURSOS EMPRESARIALES (ERP). 306	
1.1. Introducción.....	306
1.2. Definición.....	306
1.3. ¿Por qué utilizar un ERP?	307
1.4. Objetivos principales de los sistemas ERP.....	308
1.5. Características principales de los sistemas ERP.....	308
1.6. Módulos de ERP.....	309
1.7. Ejemplos de ERP	310
1.8. Conclusiones de ERP.....	313
CAPITULO II.....	315



2.	JBoss Seam FRAMEWORK.....	315
2.1.	Introducción.....	315
2.2.	Definición	315
2.3.	Características.	316
2.4.	Sistemas	317
2.5.	Testeabilidad.....	318
2.6.	Configuración	318
	CAPITULO III.....	320
3.	HIBERNATE.....	320
3.1.	Definición	320
3.2.	Arquitectura De Hibernate	320
3.3.	Características	322
3.4.	Introducción Anotaciones	322
	CAPITULO IV	324
4.	RichFaces	324
4.1.	Introducción.....	324
4.2.	Características.....	324
4.3.	Inconvenientes	326
4.4.	Requisitos e Incompatibilidades	327
4.5.	Interacción con otros componentes	328
4.6.	Recomendaciones de Uso.....	329
4.7.	Enterprise JavaBeans	329
6.	METODOLOGÍA.....	335
1.1	Métodos	335
1.1.1	Método Científico	335
1.1.2	Método Inductivo	335
1.1.3	Método Deductivo	335
1.1.4	Método Analítico.....	336



1.1.5	Metodología ICONIX	336
1.2	Técnicas.....	337
1.2.1	Entrevista	337
1.2.2	Observación	337
7.	Cronograma	338
8.	PRESUPUESTOS Y FINANCIAMIENTO	339
8.1.	Recursos Humanos	339
8.2.	Recursos Técnicos y Tecnológicos	339
8.3	Recursos Materiales	339
8.4	Total de Recursos.....	340
9.	Bibliografía	341
10.	MATRICES.....	342
	MATRIZ DE CONSISTENCIA GENERAL	342
	MATRIZ DE CONSISTENCIA ESPECÍFICA (1).....	343
	MATRIZ DE OPERATIVIDAD DE OBJETIVOS (1)	344
	MATRIZ DE CONSISTENCIA ESPECÍFICA (2).....	345
	MATRIZ DE OPERATIVIDAD DE OBJETIVOS (2)	346
	MATRIZ DE CONSISTENCIA ESPECÍFICA (3).....	347
	MATRIZ DE OPERATIVIDAD DE OBJETIVOS (3)	348
	MATRIZ DE CONSISTENCIA ESPECÍFICA (4).....	349
	MATRIZ DE OPERATIVIDAD DE OBJETIVOS (4)	350
	MATRIZ DE CONSISTENCIA ESPECÍFICA (5).....	351
	MATRIZ DE OPERATIVIDAD DE OBJETIVOS (5)	352
	MATRIZ DE CONSISTENCIA ESPECÍFICA (6).....	353
	MATRIZ DE OPERATIVIDAD DE OBJETIVOS (6)	354
2.	ENCUESTAS	356
	ENCUESTA DIRIGIDA AL ADMINISTRADOR DEL SISTEMA DE LOJANET++	356
	ENCUESTA DIRIGIDA AL PERSONAL DE LA EMPRESA LOJANET++.....	359



CERTIFICACIÓN..... 361

ÍNDICE DE FIGURAS

Ilustración 1. Módulo de ERP.....	12
Ilustración 2. Logo Hibernate	17
Ilustración 3. Ejemplo Modelo OpenLoj.....	20
Ilustración 4. Ejemplo Vista de OpenLoja.....	21
Ilustración 5. Arquitectura de OpenLoja	22
Ilustración 6. Logo RichFaces.....	27
Ilustración 7. Componentes de RichFaces.....	29
Ilustración 8. Enterprise Java Beans	35
Ilustración 9. Zonas de Navegación	273
Ilustración 10. Descripción de Cabecera.....	274
Ilustración 11. Menú de la Aplicación.....	275
Ilustración 12. Descripción de Menú	275
Ilustración 13. Ejemplo Menú Empleado	276
Ilustración 14. Descripción de Bloque Principal.....	277
Ilustración 15. Ejemplo de Contenedor de Formularios.....	277
Ilustración 16. Pie de Página	278
Ilustración 17. Gráfica de Campos Obligatorios	281
Ilustración 18. Gráfica de Ejecución en Varios navegadores.....	282
Ilustración 19. Gráfica de Pantallas.....	283
Ilustración 20. Gráfica de Roles	285
Ilustración 21. Gráfica de Recuperar Contraseña	286
Ilustración 22. Gráfica de Sesión Inactiva	286
Ilustración 23. Logo OpenLoja	306
Ilustración 24. Módulos de un ERP	309
Ilustración 25. Logo Hibernate	320
Ilustración 26. Arquitectura de Hibernate	320
Ilustración 27. Logo RichFaces.....	324
Ilustración 28. Componentes de RichFaces.....	326
Ilustración 29. EJB.....	330

ÍNDICE DE DIAGRAMAS

Diagrama 1. Módulo de Configuración Versión Final	53
Diagrama 2. Gestión Configuración Versión Final.....	54
Diagrama 3. Secuencia Iniciar Sesión Versión Final.....	64
Diagrama 4. Secuencia Cerrar Sesión Versión Final	65
Diagrama 5. Secuencia Configurar Información General Versión Final.....	66
Diagrama 6. Secuencia Crear Usuario Versión Final	67
Diagrama 7. Modificar Cuenta Usuario Versión Final.....	68
Diagrama 8. Secuencia Consultar Cuenta Usuario Versión Final.....	69
Diagrama 9. Subsistema de Gestión de Configuración Versión Final	70
Diagrama 10. Clases Modelo: Gestión de Configuración Versión Final	71
Diagrama 11. Clases Controlador: Gestión de Configuración Versión Final.....	72
Diagrama 12. Vistas de Configuración Versión Final	73
Diagrama 13. Modelo Entidad-Relación: Gestión de Configuración Versión Final	74
Diagrama 14. Modelo Relacional: Gestión de Configuración Versión Final.....	77
Diagrama 15. Módulo Empleados Versión Final	79
Diagrama 16. Gestión Empleados Versión Final.....	80
Diagrama 17. Secuencia Crear Empleado Versión Final	121
Diagrama 18. Secuencia Modificar Empleado Versión Final.....	122
Diagrama 19. Secuencia Consultar Empleado Versión Final	123
Diagrama 20. Secuencia Crear Departamento Versión Final	124
Diagrama 21. Secuencia Asignar Departamento Versión Final.....	125
Diagrama 22. Secuencia Modificar Departamento Versión Final.....	126
Diagrama 23. Secuencia Consultar Departamento	127
Diagrama 24. Secuencia Crear Horario Versión Final.....	128
Diagrama 25. Secuencia Asignar Horario Versión Final.....	129
Diagrama 26. Secuencia Registro Entrada Salida Versión Final	130
Diagrama 27. Secuencia Consultar Asistencias Versión Final	131
Diagrama 28. Secuencia Crear Permiso Versión Final	132
Diagrama 29. Secuencia Modificar Permiso Versión Final.....	133
Diagrama 30. Secuencia Consultar Permiso Versión Final	134
Diagrama 31. Secuencia Crear Capacitación Versión Final.....	135
Diagrama 32. Secuencia Modificar Capacitación Versión Final	136
Diagrama 33. Secuencia Consultar Capacitación Versión Final.....	137



Diagrama 34. Secuencia Crear Anticipo Versión Final.....	138
Diagrama 35. Secuencia Modificar Anticipo Versión Final	139
Diagrama 36. Secuencia Consultar Anticipo Versión Final	140
Diagrama 37. Secuencia Generar Rol Pago Individual Versión Final.....	141
Diagrama 38. Secuencia Generar Rol Pago General Versión Final	142
Diagrama 39. Subsistema de Gestión de Empleados Versión Final.....	143
Diagrama 40. Clases DAO: Gestión de Empleados Versión Final.....	144
Diagrama 41. Clases Controlador: Gestión de Empleados Versión Final.....	145
Diagrama 42. Vistas de Empleados Versión Final	146
Diagrama 43. Modelo Entidad-Relación: Gestión de Empleado Versión Final	147
Diagrama 44. Modelo Relacional: Gestión de Empleados Versión Final.....	152
Diagrama 45. Módulo de Clientes Versión Final	154
Diagrama 46. Gestión Cliente Versión Final	155
Diagrama 47. Secuencia Crear Cliente Versión Final	161
Diagrama 48. Secuencia Modificar Cliente Versión Final.....	162
Diagrama 49. Secuencia Consultar Cliente Versión Final	163
Diagrama 50. Subsistema de Gestión de Clientes Versión Final.....	164
Diagrama 51. Clases Modelo: Gestión de Clientes Versión Final.....	165
Diagrama 52. Clases Controlador: Gestión de Clientes Versión Final.....	166
Diagrama 53. Vistas de Cliente Versión Final	167
Diagrama 54. Modelo Entidad-Relación: Gestión de Clientes y Ventas Versión Final	168
Diagrama 55. Modelo Relacional: Gestión de Clientes Versión Final.....	169
Diagrama 56. Módulo de Proveedores Versión Final	171
Diagrama 57. Gestión Proveedor Versión Final	172
Diagrama 58. Secuencia Crear Proveedor Versión Final	179
Diagrama 59. Secuencia Modificar Proveedor Versión Final.....	180
Diagrama 60. Secuencia Consultar Proveedor Versión Final.....	181
Diagrama 61. Subsistema de Gestión de Proveedores y de Bodega Versión Final ..	182
Diagrama 62. Clases Modelo: Gestión de Proveedores y Bodega Versión Final	183
Diagrama 63. Clases Controlador: Gestión de Proveedor Versión Final	184
Diagrama 64. Vistas de Proveedor Versión Final.....	185
Diagrama 65. Modelo Entidad-Relación: Gestión de Proveedores y Bodega Versión Final.....	186
Diagrama 66. Modelo Relacional: Gestión de Proveedores Versión Final.....	187



Diagrama 67. Módulo de Bodega Versión Final.....	189
Diagrama 68. Gestión Bodega Versión Final	190
Diagrama 69. Secuencia Crear Familia de Producto Versión Final	209
Diagrama 70. Secuencia Crear Producto Versión Final	210
Diagrama 71. Secuencia Modificar Producto Versión Final.....	211
Diagrama 72. Secuencia Consultar Producto Versión Final.....	212
Diagrama 73. Secuencia Crear Movimiento Bodega Versión Final	213
Diagrama 74. Secuencia Editar Movimiento Bodega Versión Final.....	214
Diagrama 75. Secuencia Consultar Movimientos Bodega Versión Final	215
Diagrama 76. Secuencia Generar Inventario Versión Final.....	216
Diagrama 77. Secuencia Generar Kárdex Versión Final	217
Diagrama 78. Subsistema de Gestión de Bodega Versión Final	218
Diagrama 79. Clases Modelo: Gestión de Proveedores Versión Final	219
Diagrama 80. Clases Controlador: Gestión de Bodega Versión Final	220
Diagrama 81. Vistas de Bodega Versión Final.....	221
Diagrama 82. Modelo Entidad-Relación: Gestión de Proveedores y Bodega Versión Final.....	222
Diagrama 83. Modelo Relacional: Gestión de Bodega Versión Final	225
Diagrama 84. Módulo de Ventas Versión Final	227
Diagrama 85. Gestión Ventas Versión Final	228
Diagrama 86. Secuencia Crear Venta Versión Final	236
Diagrama 87. Secuencia Modificar Venta Versión Final.....	237
Diagrama 88. Secuencia Consultar Venta Versión Final.....	238
Diagrama 89. Subsistema de Gestión de Venta Versión Final	239
Diagrama 90. Clases Modelo: Gestión de Proveedores Versión Final	240
Diagrama 91. Clases Controlador: Gestión de Venta Versión Final	241
Diagrama 92. Vistas de Ventas Versión Final.....	242
Diagrama 93. Modelo Entidad-Relación: Gestión de Clientes y Ventas Versión Final	243
Diagrama 94. Modelo Relacional: Gestión de Ventas Versión Final.....	246
Diagrama 95. Módulo de Comunicación Versión Final.....	248
Diagrama 96. Gestión Comunicación Versión Final	249
Diagrama 97. Secuencia Crear Notificación Versión Final	258
Diagrama 98. Secuencia Crear Grupo Contacto Versión Final	259
Diagrama 99. Secuencia Crear Contacto Versión Final	260



Diagrama 100. Secuencia Modificar Contacto Versión Final.....	261
Diagrama 101. Subsistema de Gestión de Comunicación Versión Final	262
Diagrama 102. Clases Modelo: Gestión de Comunicación Versión Final	263
Diagrama 103. Clases Controlador: Gestión de Comunicación Versión Final	264
Diagrama 104. Vistas de Comunicación Versión Final.....	265
Diagrama 105. Modelo Entidad-Relación: Gestión de Comunicación Versión Final..	266
Diagrama 106. Modelo Relacional: Gestión Comunicación Versión Final	268
Diagrama 107. Componentes Versión Final.....	269
Diagrama 108. Despliegue Versión Final	270
Diagrama 109. Paquetes Versión Final.....	270
Diagrama 110. Modelo de la Arquitectura Versión Final	271



ÍNDICE TABLAS

Tabla 1. Comparativa de ERPs.....	9
Tabla 2. Requerimientos Funcionales.....	45
Tabla 3. Requerimientos Funcionales.....	45
Tabla 4. Identificación de Casos de uso.....	50
Tabla 5. Pantalla Iniciar Sesión.....	55
Tabla 6. Descripción Caso de Uso Iniciar Sesión.....	56
Tabla 7. Pantalla Cerrar Sesión.....	56
Tabla 8. Descripción Caso de Uso Logout.....	57
Tabla 9. Pantalla Configuración General.....	57
Tabla 10. Descripción caso de Uso Configuración General.....	58
Tabla 11. Pantalla Crear Usuario.....	59
Tabla 12. Panel Modal Buscar Rol.....	59
Tabla 13. Descripción caso de uso Crear Usuario.....	60
Tabla 14. Pantalla Modificar Cuenta Usuario.....	61
Tabla 15. Panel Modal Buscar usuario.....	61
Tabla 16. Descripción Caso de Uso Modificar Cuenta Usuario.....	62
Tabla 17. Pantalla Consultar Cuenta Usuario.....	63
Tabla 18. Descripción Caso de Uso Consultar Cuenta Usuario.....	63
Tabla 19. Modelo Conceptual: Persona.....	74
Tabla 20. Modelo Conceptual: Dirección.....	75
Tabla 21. Modelo Conceptual: Usuario.....	75
Tabla 22. Modelo Conceptual: Rol.....	75
Tabla 23. Modelo Conceptual: Usuario_Rol.....	75
Tabla 24. Modelo Conceptual: Módulo.....	75
Tabla 25. Modelo Conceptual: Menú.....	76
Tabla 26. Modelo Conceptual: Submenú.....	76
Tabla 27. Modelo Conceptual: Módulo_Rol.....	76
Tabla 28. Modelo Conceptual: Configuración.....	77
Tabla 29. Pantalla Crear Empleado.....	81
Tabla 30. Descripción Caso de Uso Crear Empleado.....	82
Tabla 31. Pantalla Editar Empleado.....	83
Tabla 32. Panel Modal Buscar Empleado.....	83
Tabla 33. Descripción Caso de Uso Modificar Empleado.....	84
Tabla 34. Pantalla Buscar Empleado.....	85



Tabla 35. Descripción de Caso de Uso Consultar Empleado	85
Tabla 36. Pantalla Crear Departamento.....	86
Tabla 37. Descripción Caso de Uso Crear Departamento.....	87
Tabla 38. Pantalla Crear Empleado	88
Tabla 39. Panel Modal Departamento.....	88
Tabla 40. Descripción Caso de Uso Asignar Departamento.....	89
Tabla 41. Pantalla Editar Departamento	89
Tabla 42. Descripción Caso de Uso Modificar Departamento	90
Tabla 43. Pantalla Crear Horario	91
Tabla 44. Descripción Caso de Uso Crear Horario.....	92
Tabla 45. Pantalla Buscar Departamento.....	92
Tabla 46. Descripción Caso de Uso Consultar Departamento	93
Tabla 47. Pantalla Crear Empleado	93
Tabla 48. Panel Modal Horarios.....	94
Tabla 49. Descripción Caso de Uso Asignar Horario	94
Tabla 50. Pantalla Registro Asistencias	95
Tabla 51. Descripción Caso de Uso Registrar Entrada Salida	96
Tabla 52. Pantalla Asistencias	97
Tabla 53. Panel Modal Buscar Empleado	97
Tabla 54. Descripción Caso de Uso Consultar Empleado	98
Tabla 55. Pantalla Crear Permiso	99
Tabla 56. Panel Modal Empleados	99
Tabla 57. Descripción de Caso de Uso Crear Permiso	100
Tabla 58. Pantalla editar Permiso	101
Tabla 59. Panel Modal Permisos	101
Tabla 60. Descripción de Caso de Uso Modificar Permiso.....	102
Tabla 61. Pantalla Buscar Permiso	103
Tabla 62. Panel Modal Empleados	103
Tabla 63. Descripción Caso de Uso Consultar Permiso	104
Tabla 64. Pantalla Crear Capacitación.....	105
Tabla 65. Panel Modal Empleados	105
Tabla 66. Descripción Caso de Uso Crear Capacitación.....	106
Tabla 67. Pantalla Editar Capacitación	107
Tabla 68. Panel Modal Capacitación.....	107
Tabla 69. Descripción Caso de Uso Modificar Capacitación	108



Tabla 70. Pantalla Buscar Capacitación.....	109
Tabla 71. Panel Modal Empleado	109
Tabla 72. Descripción Caso de Uso Consultar Capacitación	110
Tabla 73. Pantalla Crear Anticipo.....	111
Tabla 74. Panel Modal Empleado	111
Tabla 75. Descripción Caso de Uso Crear Anticipo.....	112
Tabla 76. Pantalla Editar Anticipo	113
Tabla 77. Panel Modal Egresos	113
Tabla 78. Descripción Caso de Uso Modificar Anticipo	114
Tabla 79. Pantalla Buscar Anticipo	115
Tabla 80. Panel Modal Empleado	115
Tabla 81. Descripción Caso de Uso Consultar Anticipo	116
Tabla 82. Pantalla Generar Rol Individual.....	117
Tabla 83. Panel Modal Empleado	117
Tabla 84. Descripción Caso de Uso Generar Rol Individual.....	118
Tabla 85. Pantalla Generar Rol General	119
Tabla 86. Descripción Caso de Uso Generar Rol General	120
Tabla 87. Modelo Conceptual: Persona	148
Tabla 88. Modelo Conceptual: Dirección.....	148
Tabla 89. Modelo Conceptual: Empleado	148
Tabla 90. Modelo Conceptual: Capacitación.....	148
Tabla 91. Modelo Conceptual: Empleado_Capacitación	149
Tabla 92. Modelo Conceptual: Horario.....	149
Tabla 93. Modelo Conceptual: Empleado_Horario	149
Tabla 94. Modelo Conceptual: Departamento.....	149
Tabla 95. Modelo Conceptual: Empleado_Departamento	149
Tabla 96. Modelo Conceptual: Permiso	150
Tabla 97. Modelo Conceptual: Empleado_Permiso	150
Tabla 98. Modelo Conceptual: Asistencia	150
Tabla 99. Modelo Conceptual: Egresos_Rol	151
Tabla 100. Modelo Conceptual: Rol_pagos	151
Tabla 101. Pantalla Crear Cliente	156
Tabla 102. Descripción Caso de Uso Crear Cliente	157
Tabla 103. Pantalla Editar Cliente.....	157
Tabla 104. Panel Modal Cliente	158



Tabla 105. Descripción Caso de Uso Modificar Cliente.....	159
Tabla 106. Pantalla Buscar Cliente	159
Tabla 107. Descripción Consultar Cliente	160
Tabla 108. Modelo Conceptual: Persona	168
Tabla 109. Modelo Conceptual: Dirección.....	168
Tabla 110. Modelo Conceptual: Cliente	169
Tabla 111. Pantalla Crear Proveedor	173
Tabla 112. Descripción Caso de Uso Crear Proveedor	174
Tabla 113. Pantalla Modificar Proveedor	175
Tabla 114. Panel Modal Proveedor.....	175
Tabla 115. Descripción Caso de Uso Modificar Proveedor	176
Tabla 116. Pantalla Consultar Proveedor.....	177
Tabla 117. Descripción Caso de Uso Consultar Proveedor	178
Tabla 118. Modelo Conceptual: Persona	186
Tabla 119. Modelo Conceptual: Dirección.....	187
Tabla 120. Modelo Conceptual: Proveedor	187
Tabla 121. Crear Familia Producto	191
Tabla 122. Descripción Caso de Uso Crear Familia Producto.....	192
Tabla 123. Pantalla Crear Producto	193
Tabla 124. Descripción de Caso de Uso Crear Producto	194
Tabla 125. Pantalla Editar Producto.....	195
Tabla 126. Panel Modal Producto	195
Tabla 127. Descripción Caso de Uso Modificar Producto	196
Tabla 128. Pantalla Buscar Producto	197
Tabla 129. Descripción Caso de Uso Consultar Producto.....	197
Tabla 130. Pantalla Movimientos Bodega	198
Tabla 131. Panel Modal Proveedor.....	198
Tabla 132. Panel Modal Cliente	199
Tabla 133. Panel Modal Empleado	199
Tabla 134. Panel Modal Producto	200
Tabla 135. Descripción Caso de Uso Crear Movimiento Bodega	201
Tabla 136. Pantalla Editar Movimiento.....	202
Tabla 137. Panel Modal Movimiento	202
Tabla 138. Descripción Caso de Uso Editar Movimiento Bodega.....	203
Tabla 139. Pantalla Buscar Movimiento	204



Tabla 140. Descripción Caso de Uso Consultar Movimiento Bodega.....	205
Tabla 141. Pantalla Generar Inventario	206
Tabla 142. Descripción Caso de Uso Generar Inventario	206
Tabla 143. Pantalla Generar Kárdex.....	207
Tabla 144. Panel Modal Producto	207
Tabla 145. Descripción Caso de Uso Generar Kárdex.....	208
Tabla 146. Modelo Conceptual: Persona	222
Tabla 147. Modelo Conceptual: Dirección.....	223
Tabla 148. Modelo Conceptual: Proveedor.....	223
Tabla 149. Modelo Conceptual: Producto	223
Tabla 150. Modelo Conceptual: Familia_Producto	223
Tabla 151. Modelo Conceptual: Producto_Proveedor	223
Tabla 152. Modelo Conceptual: Cabecera_Comprobante.....	224
Tabla 153. Modelo Conceptual: Detalle_Comprobante	224
Tabla 154. Modelo Conceptual: Inventario.....	224
Tabla 155. Pantalla Crear Venta.....	229
Tabla 156. Panel Modal Cliente	229
Tabla 157. Panel Modal Producto	230
Tabla 158. Descripción Caso de Uso Crear Venta.....	231
Tabla 159. Pantalla Editar Venta	232
Tabla 160. Panel Modal Comprobante.....	232
Tabla 161. Descripción Caso de Uso Editar Venta	233
Tabla 162. Pantalla Buscar Venta.....	234
Tabla 163. Panel Modal Cliente	234
Tabla 164. Descripción Caso de Uso Consultar Ventas.....	235
Tabla 165. Modelo Conceptual: Persona	243
Tabla 166. Modelo Conceptual: Dirección.....	244
Tabla 167. Modelo Conceptual: Producto	244
Tabla 168. Modelo Conceptual: Familia_Producto	244
Tabla 169. Modelo Conceptual: Producto_Proveedor	244
Tabla 170. Modelo Conceptual: Inventario.....	244
Tabla 171. Modelo Conceptual: Cabecera_Venta.....	245
Tabla 172. Modelo Conceptual: Detalle_Venta	245
Tabla 173. Pantalla Crear Notificación.....	250
Tabla 174. Panel Modal Contacto	250

Tabla 175. Descripción Caso de Uso Crear Notificación	251
Tabla 176. Pantalla Crear Grupo	252
Tabla 177. Descripción Caso de Uso Crear Grupo Contactos	253
Tabla 178. Pantalla Crear Contacto	254
Tabla 179. Descripción Caso de Uso Crear Contacto	255
Tabla 180. Pantalla Editar Contacto.....	255
Tabla 181. Panel Modal Contacto	256
Tabla 182. Descripción Caso de Uso Modificar Contacto	257
Tabla 183. Modelo Conceptual: Persona	266
Tabla 184. Modelo Conceptual: Dirección.....	266
Tabla 185. Modelo Conceptual: Contacto	267
Tabla 186. Modelo Conceptual: Contacto_Grupo	267
Tabla 187. Modelo Conceptual: Contacto_En_Grupo	267
Tabla 188. Bandeja_Correo	267
Tabla 189. Pruebas Realizadas	284
Tabla 190. Pruebas de Seguridad.....	287
Tabla 191. Test de Instalación.....	288
Tabla 192. Test de Configuración	288
Tabla 193. Test de Manuales.....	289
Tabla 194. Test de Capacidad de Respuesta	289
Tabla 195. Objetivos principales de un ERP	308
Tabla 196. Comparativa de ERPs.....	313
Tabla 197. Recursos Humanos.....	339
Tabla 198. Recursos Técnicos y Tecnológicos	339
Tabla 199. Recursos Materiales	340
Tabla 200. Total Recursos	340
Tabla 201. Matriz de consistencia general	342
Tabla 202. Matriz de consistencia específica 1	343
Tabla 203. Matriz de operatividad de objetivos 1	344
Tabla 204. Matriz de consistencia específica 2	345
Tabla 205. Matriz de operatividad de objetivos 2	346
Tabla 206. Matriz de consistencia específica 3	347
Tabla 207. Matriz de operatividad 3	348
Tabla 208. Matriz de consistencia específica 4	349
Tabla 209. Matriz de operatividad de objetivos 4	350



Tabla 210. Matriz de consistencia específica 5.....	351
Tabla 211. Matriz de operatividad de objetivos 5.....	352
Tabla 212. Matriz de consistencia específica 6.....	353
Tabla 213. Matriz de operatividad de objetivos 6.....	354

A. INTRODUCCIÓN

El presente proyecto es el resultado de un trabajo sistemático, responsable y ordenado por parte de cada uno de los integrantes. El mismo que pretende aportar con el desarrollo de un Sistema Planificador de Recursos Empresariales (ERP), dirigidos a la creación de módulos en gestión de Empleados, Clientes, Proveedores, Bodega, Ventas, Usuarios y Comunicación.

Es indudable que el ambiente competitivo que se vive en el perímetro empresarial actualmente, requiere promover procesos y actividades de negocio que generan ventajas competitivas direccionadas a fortalecer la operatividad de compañías ante sus más fuertes competidores.

Por esto, desde hace varios años, se ha dado mayor importancia a las Tecnologías de Información y su alineación con las estrategias del negocio para mejorar sus procesos clave de operatividad. Prueba de ello, es el incremento sustancial en la adquisición de software empresarial, entre los que se destacan los ERP (Planificador de Recursos Empresariales), sistemas cuyos autores buscan seguir una traza cuyo paradigma es la unificación de procesos, ágiles, confiables, escalables, por mencionar alguna de las ventajas que brinda el sistema en mención, ventajas las cuales se han tomado en consideración para el desarrollo de este sistema denominado OpenLoja; y así permitir a la empresa en donde se ha implementado en este caso Lojanet++, automatizar la mayor parte de sus procesos y manejar adecuadamente la información que se maneja en las áreas de Recursos Humanos, Ventas, Bodega, Clientes, Proveedores y comunicación, con lo cual se pretende mejorar su productividad y la atención al cliente.

OpenLoja es una solución dirigida al manejo de la información de manera adecuada con disponibilidad. Desarrollada y construida modularmente cubriendo con todos los requerimientos en las áreas mencionadas anteriormente.

Por la actualidad e importancia del tema en mención determinamos conveniente el desarrollo del presente proyecto, y permitir que la empresa Lojanet++ cuente con un sistema el cual mejore sus procesos o actividades, administre de la manera más idónea toda la información que se maneja en las áreas que comprende éste sistema; y ayudar a nosotros los autores poner en práctica y mejorar los conocimientos que hemos adquirido en nuestro recorrido académico.



Recomendamos el desarrollo de éste tipo de aplicaciones ya que gracias a que se desarrollo mediante el uso de herramientas libres, permite incentivar a las personas que se interesen en desarrollar proyectos similares optar por el uso de software libre, así como también de optar por nuevas alternativas de solución.

B. METODOLOGÍA

B.1. MATERIALES Y MÉTODOS.

B.1.1. Métodos.

Entre los métodos que utilizamos dentro de nuestro proyecto está el Científico el cual sirve para organizar y sistematizar el proceso investigativo del presente trabajo para la demostración de resultados; en donde se partirá de la revisión de teorías sobre Planeación Financiera, recopilación de la información de la institución hasta llegar a obtener resultados reflejados en las conclusiones.

Otro de los métodos que utilizamos es el Inductivo o inductivismo¹, éste permite alcanzar una solución general para cada uno de los problemas que se pueden presentar en los clientes, empleados, proveedores formulando consigo una serie de suposiciones a problemas específicos que se pueden presentar dentro del entorno empresarial cumpliendo de esta manera con todos los requerimientos y expectativas de manera correcta y eficiente.

También se utilizó el método Deductivo el mismo que con su uso se puede optar por nuevas alternativas de solución a los inconvenientes presentados durante el desarrollo y ejecución de proyecto, mediante una solución general ayuda a solventar problemas de menor prioridad lo cual se puede apreciar en la interacción del usuario con el sistema permitiéndole realizar una actividad de distintas maneras de forma rápida y eficientemente.

El método analítico es de gran ayuda ya que facilita el análisis para llegar a soluciones adecuadas previa separación de cada problema en subproblemas, que contribuirá a profundizarnos en el objeto de estudio en cuestión.

Para el desarrollo de un proyecto de éste tipo es muy importante que éste se realice en base a una metodología. Para nuestro proyecto utilizamos ICONIX, metodología que nos permite llevar un orden adecuado de cada una de las actividades que se deben cumplir, alcanzando un esquema de desarrollo adecuado, en el que predomine el orden del modelado.

¹ obtiene conclusiones generales a partir de premisas particulares.

Iconix se divide en etapas o tareas las cuales exhiben una secuencia de pasos que se deben seguir, los cuales se describen a continuación:

Primeramente se realiza el análisis de requisitos: en ésta fase luego de interactuar con el personal de la empresa se prosiguió a determinar las necesidades y requerimientos de la empresa, con esta información se realizó un modelo de dominio general (simplificado) con los objetos de la vida real que consideramos debe almacenar nuestro sistema. Ya con todos estos datos se procedió a realizar un prototipado pequeño el cual con la ayuda e interacción del personal de la empresa se mejoró hasta obtener el prototipado final el cual se puede evidenciar claramente en la interfaz gráfica de la aplicación. Seguidamente se determinó los requerimientos funcionales y no funcionales de nuestro sistema, por consecuencia se pudo identificar los actores del sistema y los casos de uso que serán ejecutados por los mismos.

A continuación de esta etapa viene el análisis y diseño preliminar: en la cual realizamos la descripción de los casos de uso identificados en la etapa anterior, éstas descripciones se hacen en base al prototipado de pantallas realizado también en la fase anterior, esto se evidencia en nuestro proyecto en la sección de discusión, en el diseño de los módulos del sistema en el cual consta el modelo de casos de uso; donde cada uno de estos cuenta primeramente con un prototipado, seguido de la descripción del mismo. Con éstas descripciones de cada caso de uso se construye los diagramas de robustez, con los cuales se determina los distintos objetos que van a interactuar y como lo van a realizar en nuestra aplicación, como es de conocimiento estos objetos son de tres tipos; los objetos límite que en nuestro caso son las vistas es decir la interfaz gráfica de la aplicación, los objetos de control que en nuestro proyecto vienen a ser todas las clases Action que se han creado para cada clase del modelo de dominio de cada módulo del sistema, al igual que las interfaces Service implementadas correspondientemente por las clases Imp; finalmente todas las clases del modelo de dominio son el tercer tipo de objeto determinado en la aplicación denominados entidad. Con esto se culminó ésta etapa y se obtuvo un esqueleto aceptable de la arquitectura y el diseño para proseguir a la siguiente.

El diseño es la siguiente etapa en la cual se puede evidenciar ya en sí las tareas que realiza la aplicación y la manera o secuencia de cómo lo realiza, para lo cual se construyen los diagramas de secuencia los cuales están relacionados directamente a los casos de uso y su descripción, y se pueden apreciar en la parte correspondiente a

la discusión del proyecto en cada sección de diseño de los módulos denominados Diagramas de iteración.

Finalmente con todo lo realizado anteriormente procedimos a la última etapa de Iconix que es la implementación: donde codificamos, es decir programamos exactamente como se especificó en las etapas anteriores obteniendo como claro resultado de esta etapa el Sistema Planificador de Recursos Empresariales: OpenLoja 1.0 que luego de una serie de testeos y pruebas descritas en el plan de validación correspondiente a la parte de discusión del proyecto, garantiza que el sistema cumple con los requisitos iniciales obtenidos de la empresa donde lo entregamos e implementamos.

Cabe recalcar que debido a que una de las características principales de Iconix es que es iterativo e incremental, en cada fase se pudo mejorar y refinar el diagrama de clases de nuestro sistema obteniendo los datos adecuados y correctos que debe guardar o almacenar el sistema.

B.1.2. Técnicas.

La Entrevista, técnica que la utilizamos para obtener la mayor cantidad de información que nos provean todos las personas involucradas directa o indirectamente con la empresa.

La observación Contribuyo en la determinación de todos los inconvenientes que se presentan durante el desarrollo de las actividades dentro y fuera de la empresa.

La Encuesta, realizada al personal de la empresa Lojanet++ luego de su interacción con el sistema OpenLoja ya implementado, permite determinar si el sistema cumple con lo solicitado por la empresa, es decir permite realizar la validación de OpenLoja lo cual se puede corroborar en la parte de discusión en la sección del Plan de Validación así como también en la parte de anexos donde se encuentran adjuntadas las encuestas.

B.1.3. Materiales.

➤ **JBoss Seam FRAMEWORK.**

Herramienta de gran ayuda para la implementación de nuestro proyecto la cual gracias a que es adaptable con el entorno de desarrollo eclipse, puede ser integrado con bibliotecas de Richfaces y por lo tanto tiene soporte para Ajax. Creímos conveniente su uso para el desarrollo de esta aplicación web que en conjunto con la base de datos creada para nuestro sistema permite actualizar o guardar nuevos datos debido a que cuenta con una herramienta de línea de comandos llamada seam-gen y además podemos acceder fácilmente a cualquier componente EJB desde la capa de presentación utilizando su nombre de componente de seam lo cual nos facilitó enormemente al momento de realizar la codificación del diseño.

➤ **MYSQL**

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario, debido a que es muy utilizada en aplicaciones web, en distintas plataformas y tomando en cuenta sus características optamos por este tipo de base de datos para almacenar y mantener protegida toda la información que maneje el sistema.

➤ **HERRAMIENTA DE MODELADO UML DIA.**

Dia es una aplicación informática de propósito general para la creación de diagramas, la cual elegimos para facilitar la construcción de los diagramas de la aplicación en lo que respecta a análisis y diseño, los cuales podemos observar en la sección de discusión del proyecto. Se puede apreciar diagramas de: despliegue, componentes, secuencia, robustez, casos de uso entre otros.

ENTORNO DE DESARROLLO EMPRESARIAL ECLIPSE

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), 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.



Debido a su integración con Hibernate y Jboss seam optamos por el uso de éste entorno de desarrollo agilizando grandemente la implementación del proyecto ya que cuenta con una interfaz muy amigable.

C. REVISIÓN DE LITERATURA

1. SISTEMA PLANIFICADOR DE RECURSOS EMPRESARIALES (ERP)

1.1. INTRODUCCIÓN.

Es indudable que el ambiente competitivo en el que se vive en el ámbito empresarial actualmente, requiere de promover los procesos y actividades de negocio que generan las ventajas competitivas de las compañías ante sus más fuertes competidores.

Por esto, desde hace ya varios años, se ha dado mayor importancia a las Tecnologías de Información y su alineación con las estrategias del negocio para mejorar sus procesos clave de negocio. Prueba de ello, es el incremento tan sustancial de adquisiciones de paquetes de software empresariales tales como el ERP (*Enterprise Resource Planning*), con el cual los directivos de las compañías esperan tener integradas todas las áreas o departamentos de la compañía que apoyan para la generación de sus productos y servicios.

Hoy más que nunca las empresas requieren de herramientas que les proporcionen control y centralización de su información, esto con el fin tomar las mejores decisiones para sus procesos y estrategias de negocios. Los ERP son una solución robusta para aquellas empresas que buscan una solución universal a la centralización de su información, más específicamente OpenLoja que es un sistema que busca reducir los tiempos de ejecución en la realización de actividades, así como de llevar un control y manejo adecuado de la información de la empresa donde esta implementado.

1.2. DEFINICIÓN.

Ortuño propone la siguiente definición: “ERP son las siglas en inglés de Enterprise Resource Planning que en español significa planificación de recursos empresariales, es un sistema de gestión de la información estructurado para satisfacer la demanda de soluciones de gestión empresarial, basado en el ofrecimiento de una solución completa que permite a las empresas evaluar, implementar y gestionar más fácilmente su negocio. Las soluciones ERP se caracterizan por su modularidad,

integración de la información (dato único), universalidad, estandarización e interfaces con otras aplicaciones. Son sistemas abiertos y en la mayoría de los casos multiplataforma.”(2010).

Un ERP debe estar estructurado de forma modular e integrar a la vez estos módulos de tal forma que permite automatizar los procesos de la empresa en donde se está implementado, logrando así administrar y dirigir la empresa de manera correcta así como también de optimizar sus recursos, mejorando su competitividad, productividad y el servicio a sus clientes, y llevar un control adecuado de toda la información que se maneja dentro de la empresa.

1.3. TABLA DE COMPARACIÓN DE ERPs

Openbravo	OpenXpertya	OpenERP	OpenLoja
<ul style="list-style-type: none"> • Solución comercial en software libre y entorno web. • Para pequeñas y medianas empresas. • Utilizado alrededor de 50 países. • Su crecimiento se debe a la contribución de su comunidad internacional. • Modelo del negocio basado en el software libre comercial. • Elimina coste de licencias. • Ofrece servicios, soporte y mejoras mediante una suscripción anual. 	<ul style="list-style-type: none"> • Solución integral para empresas que engloba ERP y CRM. • Integra servicios en línea B2B o B2C e incluso B2E con soporte de exportación de datos al estándar EDI. • Posibilidad de trabajar con cubos multidimensionales OLAP. • Cubre necesidades de gestión de empresas de tamaño medio o grande. • Tiene capacidad multientidad, multiempresa, multicentro, multialmacén, entre otras. 	<ul style="list-style-type: none"> • Tiene libertad para elegir el proveedor que más le convenga según sus necesidades. • Permite contratar únicamente solo lo necesario. • Disponibilidad del código para realizar cualquier mejora sobre los módulos ya existentes, o crear uno nuevo. • Dispone de más de 400 módulos. 	<ul style="list-style-type: none"> • Diseñado para empresas pequeñas y medianas. • Diseñado y desarrollado por módulos e integra un módulo de comunicación entre empleados clientes y proveedores. • Permite adaptar nuevos módulos de acuerdo a las necesidades del usuario. • Diseño, construcción e implementación de bajo costo gracias a que se ha utilizado herramientas libres para su desarrollo. • Multiplataforma².

Tabla 1. Comparativa de ERPs

²software que puedan funcionar en diversas plataformas.

En la tabla anterior se puede apreciar las características de algunos ERPs entre los cuales se incluye OpenLoja, como podemos observar está diseñado para empresas de tamaño pequeño o mediano, en comparación con los otros que son desarrollados para empresas grandes, con mayores prestaciones; sin considerar mucho esto OpenLoja tiene características similares a los otros ERPs, así como también ventajas y desventajas las cuales se describen mejor a continuación.

1.4. ¿POR QUÉ UTILIZAR EL ERP OPENLOJA?

Ortuño (2010) propone algunas razones por las cuales una empresa debe implementar un ERP, así como también en que aspectos la beneficia.

OpenLoja siendo un ERP construido por módulos permite cumplir con los requerimientos de la empresa donde se encuentra implementado, permitiéndole a ésta utilizar los módulos que se determine convenientes o aumentarlos en caso que se requiera, es decir se puede personalizar de acuerdo a las necesidades que el cliente solicite.

Gracias a esto OpenLoja cumple con los aspectos por los cuales fue diseñado y construido permitiendo aumentar la competitividad, agilizar operaciones e integrar toda la información de las áreas que comprende éste sistema que son: Ventas, Bodega, Recursos Humanos, Clientes, Proveedores y Comunicación; de la siguiente manera:

1.4.1. Aumentar la Competitividad.

Optimizar los recursos de la empresa en las áreas mencionadas, lo que conlleva consigo a mejorar la administración de los mismos, disminuyendo costos y por lo tanto mejorando la productividad.

1.4.2. Agilizar Operaciones.

Mejora el desarrollo de las operaciones o actividades para las cuales fue diseñado el sistema, ya que agiliza los tiempos de respuesta a las diversas peticiones que realice el usuario cuando este lo requiera.

1.4.3. Integración.

Manejar la información de la empresa en cada sección correspondiente de manera aislada, para que posteriormente al integrar toda esta información contribuya en la

toma de decisiones generando así una solución global en base a todos los datos manejados dentro de la empresa.

1.5. OBJETIVOS PRINCIPALES DE OPENLOJA.

EL propósito principal de OpenLoja es brindar un manejo eficiente de la información de la empresa, lo cual permite disminuir costos en las operaciones y ayudar en la toma de decisiones, mejorando así la atención al cliente y el tiempo de respuesta para solucionar cualquier inconveniente o problema que se pueda presentar.

Este propósito abarca:

- ✓ Integración de toda la información manejada, así como de su correcto manejo, uso y respaldo.
- ✓ Excluir funciones inapropiadas e información relevante.
- ✓ Automatización de los procesos o actividades empresariales de las áreas que comprende el sistema.
- ✓ Disminución de costos y manejo adecuado de recursos.
- ✓ Rapidez y eficacia para resolver inconvenientes.
- ✓ Mejor atención para el cliente.

1.6. CARACTERÍSTICAS PRINCIPALES DE OPENLOJA.

El ERP no es simplemente un software que se compra, instala y usa como Windows o un juego de computadora.

Consiste en una revolución que involucra todos los procesos internos y debe ser precedido de una revaluación³ de todos los departamentos, sus funciones, mecanismos de decisión y formas de actuación.

En lo que respecta al sistema OpenLoja se puede mencionar las siguientes características.

- ✓ Adaptabilidad para utilizar solo los módulos que se requieran y disminuir la modificación de procesos al momento de implementar el sistema.

³ significa un aumento del precio de los bienes o productos.

- ✓ Escalabilidad si requiere adaptar nuevas funciones al sistema para un correcto funcionamiento del ERP.
- ✓ Base de datos centralizada
- ✓ Interacción entre todos los módulos del sistema lo cual le permite llevar un manejo y control adecuado de toda la información.

Además OpenLoja está integrado por un conjunto de módulos de tal forma que permite realizar operaciones en cada área a la cual corresponde cada módulo, así como de un manejo de datos aislado para su posterior integración y obtención de toda la información utilizada dentro de la organización con la que se pueda apoyar la toma de decisiones al momento de solucionar algún problema o inconveniente que se presente.

1.7. MÓDULOS DE OPENLOJA.

OpenLoja es un ERP conformado por un conjunto de módulos, en donde cada módulo corresponde a un departamento o área, que a su vez se encuentran relacionados entre sí mediante la información que comparten, la cual es generada por los procesos que se realizan. Estos módulos se encuentran diseñados para cumplir todos los requerimientos del usuario y así de esta manera cumplir con todas sus expectativas.

Los módulos con los que cuenta OpenLoja son Gestión de: Ventas, Bodega, Configuración, Comunicación, Empleados, Clientes y Proveedores.



Ilustración 1. Módulos de OpenLoja

1.8. CONCLUSIONES DE OPENLOJA

En la actualidad el uso de tecnologías es muy importante dentro del entorno empresarial ya que permite mejorar sus estrategias de negocios, permitiendo que estas agilicen sus procesos, optimizando tiempo y recursos; aumentando así la productividad y competitividad.

Asimismo le sirve a las empresas para soportar sus estrategias competitivas, ya sea para ir un paso delante de la competencia o reducir las ventajas que la misma pueda presentar.

OpenLoja es un ERP que permite automatizar los procesos, administrando todos los recursos de la manera más adecuada.

Como todo sistema, tiene sus ventajas y sus desventajas, de los beneficios más comunes e importantes podemos mencionar:

- ✓ Diseñado y construido de acuerdo a los requerimientos del cliente para funcionar correctamente en donde se halla implementado.
- ✓ Reduce costos y permite un manejo correcto de recursos de las áreas para las cuales está diseñado.
- ✓ Integra y relaciona las áreas que abarca el sistema, así como también toda la información que se maneja en ellas.
- ✓ La mayoría de las operaciones o procesos que se realizan son manejados a través de éste sistema.
- ✓ Es escalable por lo cual se puede adaptar nuevas funciones cuando se lo requiera.
- ✓ Interfaz muy amigable para el usuario.

De las desventajas podemos mencionar:

- ✓ A pesar que se utilizó herramientas para su elaboración su costo es un poco elevado.
- ✓ La implementación de éste sistema requiere de modificaciones en la organización o en algunos de sus procesos.
- ✓ No cubre el área de contabilidad, la cual al momento de implementarla en el sistema mejoraría aún más su funcionamiento y rendimiento.

2. JBOSS SEAM FRAMEWORK.

2.1. INTRODUCCIÓN.

Delgado (2007) compila información referente a SEAM para poder determinar sus características y los sistemas que lo componen.

JBoss Seam es un framework⁴ desarrollado por JBoss, una división de Red Hat. El líder del proyecto es Gavin King, también autor del framework para mapeo objeto relacional Hibernate. Combina a los 2 frameworks Enterprise JavaBeans EJB3 y JavaServerFaces JSF. Se puede acceder a cualquier componente EJB desde la capa de presentación refiriéndote a él mediante su nombre de componente seam.

Seam introduce el concepto de contextos. Cada componente de Seam existe dentro de un contexto. El contexto conversacional por ejemplo captura todas las acciones del usuario hasta que éste sale del sistema o cierra el navegador, inclusive puede llevar un control de múltiples pestañas y mantiene un comportamiento consistente cuando se usa el botón de regresar del navegador.

Permite generar automáticamente una aplicación web de altas, bajas; cambio y modificaciones a partir de una base de datos existente utilizando una herramienta de línea de comandos llamada seam-gen incluida con el framework. Seam puede ser integrado con las bibliotecas de componentes JSF JBossRichFaces o con ICEsoftICEFaces. Ambas bibliotecas poseen soporte para AJAX.

2.2. DEFINICIÓN

JBossSeam es un framework que integra y unifica los distintos standards de la plataforma Java EE 5.0, pudiendo trabajar con todos ellos siguiendo el mismo modelo de programación.

Ha sido diseñado intentando simplificar al máximo el desarrollo de aplicaciones, basando el diseño en Plain Old Java Objects (POJOs) con anotaciones. Estos componentes se usan desde la capa de persistencia hasta la de presentación, poniendo todas las capas en comunicación directa.

⁴ estructura conceptual y tecnológica de soporte definido

El núcleo principal de Seam está formado por las especificaciones Enterprise JavaBeans 3 (EJB3) y JavaServer Faces (JSF).

A grandes rasgos podemos definir EJB3 como una arquitectura para un sistema transaccional (como bases de datos) de objetos distribuidos basado en componentes que permite construir aplicaciones portables, reusables y escalables.

JSF es un framework de la capa de presentación que define componentes para el interfaz gráfico y "managedbeans" para la lógica de la aplicación que interactúan a través de un sistema de eventos.

Sin embargo, estos frameworks tienen algunas limitaciones y no han sido diseñados para trabajar juntos (esto pretende resolverse con la futura especificación web beans); tienen distinto tipo de configuraciones (JSF usa archivos XML mientras que EJB3 usa anotaciones), distinto ciclo de vida y no pueden comunicarse directamente a nivel de framework.

Para hacerlos cooperar necesitaríamos escribir "clases fachada" y multitud de código de relleno que se encargase de pasar las llamadas de una capa de la aplicación a otra. Ahí es donde entra en juego Seam.

Seam elimina la barrera existente entre estas tecnologías, permitiendo usar EJBs directamente como "backingbeans" de JSF y lanzar o escuchar eventos web.

2.3. CARACTERÍSTICAS.

- ✓ Tiene un diseño 100% basado en componentes.
- ✓ JBoss Seam ofrece una solución completa para el desarrollo web que abarca todas las capas, desde la de presentación hasta la de integración.
- ✓ Sigue un modelo Pull en el que desde una página puede referenciarse cualquier componente que esté asociado a un contexto accesible.
- ✓ Proporciona varios componentes para manejar los mensajes multidioma.
- ✓ Determina automáticamente el Locale de cada petición y lo usa si está disponible en la aplicación. También permite sobrescribir el Locale detectado y/o establecer uno por defecto. Se pueden configurar los mensajes en archivos de recursos tipo bundle.

- ✓ Dispone de una clase que se encarga de parsear⁵ automáticamente los formatos horarios al formato establecido.
- ✓ Para los mensajes de error o éxito se puede hacer uso de la clase de FacesMessages incluida con JSF.

2.4. SISTEMAS.

2.4.1. SISTEMA DE SEGURIDAD.

El Seam Security API es una parte de Seam que proporciona funcionalidades de autenticación y autorización basado en JAAS (Java Authentication and AuthorizationService). Puede usarse el modo sencillo por defecto que se basa en roles o usar el framework JBoss Rules, que ofrece un sistema más poderoso basado en reglas. El sistema se encarga del manejo de errores de autorización o autenticación permitiendo redirigir al usuario a una página determinada.

Las restricciones pueden establecerse mediante el uso de anotaciones. Permite restringir tanto acciones como entidades o páginas o componentes de la página.

2.4.2. SISTEMA DE PLANTILLAS (TEMPLATES).

Seam viene integrado por defecto con Facelets.

Facelets es un sistema de plantillas para JSF. Permite definir la vista mediante xml en forma de árbol de manera que se pueden definir componentes como composición de otros componentes. Entre sus características destacar que soporta el lenguaje EL, composición a partir de diferentes archivos, decorators y reporte de errores indicando la etiqueta/línea precisa.

2.4.3. SISTEMA DE CACHÉ.

En total podemos encontrar varios sistemas de caché.

En primer lugar podemos encontrar la importantísima caché de la base de datos.

⁵ transforma una entrada de texto en una estructura de datos.

Después existe otra caché en la solución ORM adoptada.

El contexto de persistencia manejado por Seam actúa como caché de los datos leídos en la conversación actual.

También pueden guardarse datos no transaccionales en el contexto de aplicación.

La aplicación puede almacenar datos en memoria utilizando el sistema JBoss Cache, que ofrece una cache estructurada en forma de árbol, con posibilidad de clúster y transaccional.

3. HIBERNATE.



Ilustración 2. Logo Hibernate

3.1. DEFINICIÓN.

Minter Y Linwood (2009) describen una definición breve y concisa acerca de la herramienta Hibernate.

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL.⁶

⁶ Association for Information Systems (CAIS)

3.2. ARQUITECTURA DE OPENLOJA

Fernández (2011) determina como está estructurado Hibernate, así como también cuáles son sus características y sus respectivos componentes, es decir describe la estructura de ésta herramienta.

Para el diseño de este sistema se tomo en consideración el patrón Modelo-Vista-Controlador mediante el cual se pudo obtener los diagramas de clases de diseño, y así determinar cómo es la arquitectura manejada en OpenLoja.

3.2.1. PATRÓN MODELO VISTA CONTROLADOR (MVC)

El patrón de diseño Modelo-Vista-Controlador plantea un método formal para separar los módulos de entrada, de procesamiento y de salida de datos en una aplicación y la forma de comunicación entre dichos módulos. Es decir, separa la lógica de negocio de la interfaz de usuario, facilitando así la evolución por separado de ambos aspectos e incrementando la reutilización y la flexibilidad. El Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno MCV, se ve frecuentemente en aplicaciones web donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

3.2.2. DESCRIPCIÓN DEL PATRÓN

3.2.2.1. MODELO

Ésta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

Entre otras cosas es responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.

- Define las reglas de negocio (la funcionalidad del sistema).
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

Dentro de OpenLoja el Modelo está conformado por la base de datos en donde se almacena la información la cual está constituida por tablas, elaboradas en correspondencia con las clases modelo del sistema, por ejemplo para la clase Empleado del sistema existe una tabla Empleados en la base de datos; estas tablas son construidas también tomando en consideración mucho lo que respecta a la multiplicidad y relación entre otras entidades o clases.

EMPLEADO

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	No		
Clave	varchar	No		
sueldoBasico	boolean	No		
fechaIngreso	varchar	Si		
contrato	boolean	No		
codRegistro	Varchar	No		
estado	Boolean	No		

Esta es la tabla Empleado que se crea en la base de datos, la cual se basa y relaciona directamente a la clase Empleado del modelo del sistema la cual se puede observar a continuación:

```

Empleado
- cabeceraComprobantes: List<CabeceraComprobante> = new ArrayList<C...
- cargo: String
- clave: String
- sueldoBasico: Double
- codRegistro: String
- fechaIngreso: Date
- contrato: Boolean
- empleadoCapacitaciones: List<EmpleadoCapacitacion> = new ArrayList<E...
- asistencias: List<Asistencia> = new ArrayList<A...
- estado: Boolean
- rolPagoses: List<RolPagos> = new ArrayList<R...
- empleadoPermisos: List<EmpleadoPermiso> = new ArrayList<E...
- empleadoHorarios: List<EmpleadoHorario> = new ArrayList<E...
- empleadoDepartamentos: List<EmpleadoDepartamento> = new ArrayList<E...
- empleadoEgresos: List<EmpleadoEgreso> = new ArrayList<E...

+ Empleado()
+ getClave(): String
+ setClave(String): void
+ getCargo(): String
+ setCargo(String): void
+ getSueldoBasico(): Double
+ setSueldoBasico(Double): void
+ getFechaIngreso(): Date
+ setFechaIngreso(Date): void
+ getContrato(): Boolean
+ setContrato(Boolean): void
+ getCodRegistro(): String
+ setCodRegistro(String): void
+ getEstado(): Boolean
+ setEstado(Boolean): void
+ getRolPagoses(): List<RolPagos>
+ setRolPagoses(List<RolPagos>): void
+ getEmpleadoDepartamentos(): List<EmpleadoDepartamento>
+ setEmpleadoDepartamentos(List<EmpleadoDepartamento>): void
+ getEmpleadoEgresos(): List<EmpleadoEgreso>
+ setEmpleadoEgresos(List<EmpleadoEgreso>): void
+ getEmpleadoPermisos(): List<EmpleadoPermiso>
+ setEmpleadoPermisos(List<EmpleadoPermiso>): void
+ getEmpleadoCapacitaciones(): List<EmpleadoCapacitacion>
+ setEmpleadoCapacitaciones(List<EmpleadoCapacitacion>): void
+ getEmpleadoHorarios(): List<EmpleadoHorario>
+ setEmpleadoHorarios(List<EmpleadoHorario>): void
+ getAsistencias(): List<Asistencia>
+ setAsistencias(List<Asistencia>): void
+ getCabeceraComprobantes(): List<CabeceraComprobante>
+ setCabeceraComprobantes(List<CabeceraComprobante>): void

```

Ilustración 3. Ejemplo Modelo OpenLoja

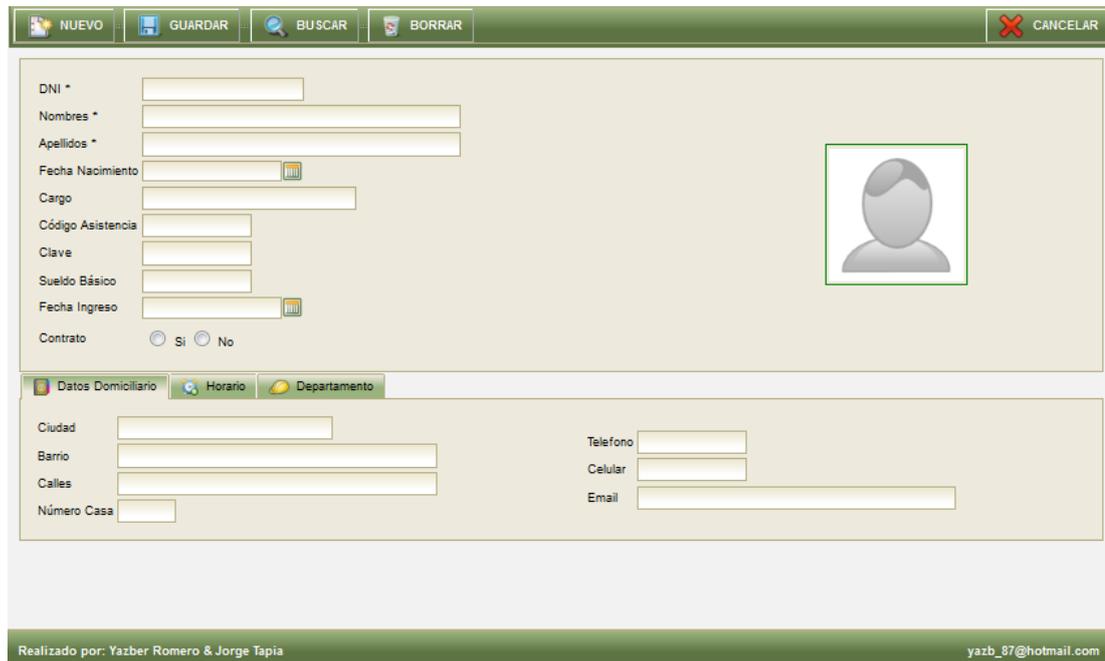
3.2.2.2. VISTA

Éste presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Las vistas son responsables de:

- Recibir datos del modelo y a su vez mostrarlos al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).

En el sistema OpenLoja la vista está constituida por todas la pantallas que van interactuar con el usuario, es decir es la interfaz grafica con la que cuenta el sistema que no se relaciona directamente con el modelo, pero está construida en conformidad con este, ya que el usuario es aquí en donde realizara todas sus funciones o actividades. Continuando con el ejemplo anterior para la clase Empleado se han diseñado varias pantallas en cómo: Crear Empleado, Editar Empleado y Consultar Empleado, que son las principales; pero hay otras como Crear Anticipos, Editar Anticipos Crear Capacitaciones, etc. Que también se relacionan con la entidad Empleado.



The screenshot shows a web-based form for user registration. At the top, there is a navigation bar with buttons for 'NUEVO', 'GUARDAR', 'BUSCAR', 'BORRAR', and 'CANCELAR'. The main form area is divided into two sections. The first section contains fields for 'DNI *', 'Nombres *', 'Apellidos *', 'Fecha Nacimiento', 'Cargo', 'Código Asistencia', 'Clave', 'Sueldo Básico', 'Fecha Ingreso', and 'Contrato' (with radio buttons for 'Si' and 'No'). A placeholder for a profile picture is shown on the right. The second section, titled 'Datos Domiciliario', 'Horario', and 'Departamento', includes fields for 'Ciudad', 'Barrio', 'Calle', 'Número Casa', 'Telefono', 'Celular', and 'Email'. At the bottom, the footer indicates 'Realizado por: Yazber Romero & Jorge Tapia' and 'yazb_87@hotmail.com'.

Ilustración 4. Ejemplo Vista de OpenLoja

3.2.2.3. CONTROLADOR

Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

Esta parte es la conexión entre la vista y el modelo, en el sistema OpenLoja contiene dos tipos de controladores para cada clase o entidad con la que cuenta el sistema en éste caso son las Interfaces terminadas en Action las cuales están implementadas a su vez por las clases terminadas en Imp que son las que nos permitirán realizar las funciones de actualizar y guardar información que el usuario ingresa en las vistas, en la base de datos es decir desde la vista llegar hasta el modelo; y el otro tipo que son las clases terminadas en List que se utilizan para realizar lo que son las consultas es decir nos permiten acceder a la información almacenada en el modelo.

Para explicarlo mejor y continuando con el mismo ejemplo tenemos que para la clase Empleado del sistema tenemos la interfaz EmpleadoAction implementada a su vez por la clase EmpleadoImp las cuales permitirán actualizar y guardar todos los datos que el usuario ingrese en la tabla Empleados de la base de datos, mientras que para las consultas como se dijo anteriormente se utiliza la clase EmpleadoList la cual nos permitirá acceder y obtener ésta información almacenada en el modelo.

Una vez descrito el patrón utilizado para realizar la arquitectura del sistema OpenLoja, ésta queda estructurada de la siguiente manera:

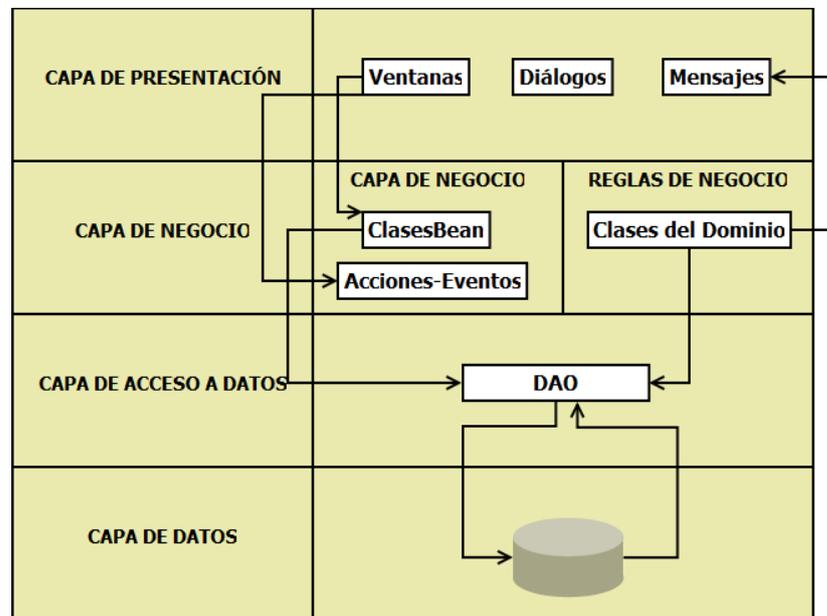


Ilustración 5. Arquitectura de OpenLoja

3.3. INTRODUCCIÓN ANOTACIONES

Gómez (2007) desarrolla e implementa una aplicación web basada en herramientas J2EE e Hibernate, donde explica y demuestra como es el manejo de anotaciones, describe una pequeña introducción para el manejo de éstas.

Tradicionalmente, la comunidad java ha estado en una confusión profunda acerca precisamente de qué tipo de meta información debe estar como configuración. J2EE y populares contenedores ligeros han proporcionado descriptores de despliegue basados en XML tanto para cosas que son verdaderamente configurables entre diferentes despliegues del sistema, y para cualquier otro tipo de declaración la cual no puede ser fácilmente expresada en java. Las anotaciones de Java 5 cambian todo esto.

Ejb3 adopta las anotaciones y la "configuración por excepción" como la manera más fácil para proporcionar información a el contenedor en una forma declarativa. Seam extiende las anotaciones proporcionadas por Ejb3 con un conjunto de anotaciones para gestión del estado en forma declarativa y demarcación del contexto también en

forma declarativa. Esto permite eliminar las declaraciones de beans gestionados en Jsf (managed beans) y reduce el XML requerido a justamente la información que verdaderamente pertenece a XML (las reglas de navegación Jsf).

A continuación se describen algunas de las anotaciones más utilizadas por Hibernate:

Primeramente se crea o elige una clase simple la cual se utilizará como Entidad.

```
public class Contacto implements Serializable{
    private long id;
    private String nombre;
    private String email;
    private String telefono;
    public Contacto(){ }
    public Contacto(String nombre, String email, String telefono){
        this.nombre = nombre;
        this.email = email;
        this.telefono = telefono;
    }
    public String getEmail(){
        return email;
    }
    public void setEmail(String email){
        this.email = email;
    }
    public long getId(){
        return id;
    }
    private void setId(long id){
        this.id = id;
    }
    public String getNombre(){
        return nombre;
    }
    public void setNombre(String nombre){
        this.nombre = nombre;
    }
    public String getTelefono(){
        return telefono;
    }
    public void setTelefono(String telefono){
        this.telefono = telefono;
    }
}
```

La anotación más importante es "javax.persistence.Entity", la cual indica que la clase es una "Entidad", por lo que la anotación se coloca a nivel de clase.

```
@Entity
public class Contacto implements Serializable{

}
```

La anotación "javax.persistence.Table" se utiliza para indicar cuál será el nombre de la tabla generada (o si ya tenemos una base de datos creada, cuál es el nombre de la tabla en la que se almacenarán estas entidades):

```
@Entity
@Table(name="contactos")
public class Contacto implements Serializable {
}
```

Después se debe indicar cuál atributo de la clase será el identificador. Esto lo hacemos con la anotación "javax.persistence.Id". Este atributo podemos colocarlo en dos lugares: en el getter del atributo identificador, o directamente en el atributo. Cuando colocamos la anotación en el atributo decimos que se realiza un acceso por "atributo" (field access), o sea que Hibernate lee y establece los valores directamente de los atributos. Cuando la colocamos en el getter decimos que se realiza un acceso por "propiedad" (property access), o sea que Hibernate accesa a los valores mediante los setters y getters. No pueden mezclarse los tipos de acceso, así que se debe elegir con cuidado.

En este caso la anotación está colocada en el atributo:

```
@Id
private long id;
```

Se puede personalizar la forma en la que se generará el valor del identificador usando la anotación "javax.persistence.GeneratedValue" e indicando la "estrategia" que se usará para generarlo. Existen solo 4 estrategias en JPA (si usamos las anotaciones de Hibernate podemos usar todos los generadores de Hibernate):

- AUTO
- IDENTITY
- SEQUENCE

- TABLE

Las más usadas son "AUTO" e "IDENTITY". La primera usa la estrategia por default de la base de datos, mientras que la segunda genera los identificadores en orden (1, 2, 3, etc.).

```
@Id  
@GeneratedValue(strategy=GenerationType.IDENTITY)  
private long id;
```

Esto es todo lo que es necesario hacerle a la clase, el resto de las propiedades que se persistirán serán obtenidas mediante reflexión. Todos los atributos que no estén marcados como "transient" o con la anotación "javax.persistence.Transient" serán persistidas.

Si queremos personalizar la columna de la tabla que se generará de un atributo, podemos hacerlo con la anotación "javax.persistence.Column". Por ejemplo, si no se desea que la columna generada del atributo "email" sea "email", sino que sea "e_mail", se agrega la anotación de la siguiente forma:

```
@Column(name="e_mail")  
private String email;
```

Al final la clase "Contacto" queda de la siguiente forma (omitiendo los imports)

```
@Entity
@Table(name="contactos")
public class Contacto implements Serializable{
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private long id;
    private String nombre;

    @Column(name="e_mail")
    private String email;
    private String telefono;

    public Contacto(){
    }
    public Contacto(String nombre, String email, String telefono){
        this.nombre = nombre;
        this.email = email;
        this.telefono = telefono;
    }
    public String getEmail(){
        return email;
    }
    public void setEmail(String email){
        this.email = email;
    }
    public long getId(){
        return id;
    }
    private void setId(long id){
        this.id = id;
    }
    public String getNombre(){
        return nombre;
    }
    public void setNombre(String nombre {
        this.nombre = nombre;
    }
    public String getTelefono(){
        return telefono;
    }
    public void setTelefono(String telefono {
        this.telefono = telefono;
    }
}
```

4. RICHFACES.



Ilustración 6. Logo RichFaces

4.1. INTRODUCCIÓN.

RichFaces es una librería de componentes visuales para JSF, escrita en su origen por Exadel y adquirida por Jboss. Además, RichFaces posee un framework avanzado para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF.

Se determinó conveniente el uso de Richfaces para el desarrollo de éste proyecto ya que es una librería con muchas ventajas entre las cuales se destacan la creación de Skins o uso de los que vienen creados por defecto, la creación de vistas complejas basándose en los requerimientos que el usuario requiere, también la utilización de etiquetas dentro de las vistas para la actualización de las mismas. Es decir se utiliza para la realización de la Interfaz Grafica del sistema, permitiendo desarrollarla de manera rápida, con un entorno agradable y entendible para que el usuario se adapte al sistema fácilmente, así como también controlar los datos que se van a manejar dentro de las vistas; Richfaces es una librería que cuenta con grandes ventajas en comparación a otras, razón principal por lo cual se la tomó en consideración para el desarrollo del presente sistema.

4.2. CARACTERÍSTICAS

Coral y Gordillo (2010) compilan información para describir y explicar cómo está estructurado RichFaces, cuales son su componentes, que características posee y que inconvenientes se pueden presentar al utilizarlo para desarrollar una aplicación; así como también como éste se relaciona con EJB y como EJB está estructurado.

En este apartado se pretende dar una visión de las características que RichFaces permite:

- ✓ Intensificar el conjunto de beneficios JSF al trabajar con Ajax. RichFaces está completamente integrado en el ciclo de vida de JSF. Mientras que otros marcos sólo dan acceso a los managedbean, RichFaces permite acceder al action y al valor del listener, así como invocar a validadores y convertidores durante el ciclo de petición-respuesta de Ajax.
- ✓ Añadir capacidad Ajax a aplicaciones JSF. El framework proporciona dos librerías de componentes (Core Ajax y la interfaz de usuario). La librería Core nos permite agregar la funcionalidad Ajax en las páginas que queramos sin necesidad de escribir nada de código JavaScript. RichFaces permite definir eventos en la propia página. Un evento invoca a una petición Ajax, sincronizándose así zonas de la página y componentes JSF después de recibir la respuesta del servidor por Ajax.
- ✓ Crear rápidamente vistas complejas basándose en la caja de componentes. La librería UI (Interfaz de usuario) que contiene componentes para agregar características de interfaz de usuario a aplicaciones JSF. Se amplía el framework de Rich Faces incluyendo un gran conjunto de componentes "habilitación de Ajax" que extiende el soporte de la página. Además, los componentes de RichFaces están diseñados para ser usados sin problemas con otras librerías de componentes en la misma página, de modo que existen más opciones para el desarrollo de aplicaciones
- ✓ Escribir componentes propios con función soportada por Ajax. El CDK o Kit de Desarrollo de Componentes basado en maven, incluye un generador de código para plantillas JSP utilizando una sintaxis similar. Estas capacidades ayudan a evitar un proceso de rutina de un componente de creación.
- ✓ Proporciona un paquete de recursos con clases de aplicación Java. Además de su núcleo, la funcionalidad de RichFaces para Ajax proporciona un avanzado soporte a la gestión de diferentes recursos: imágenes, código JavaScript y hojas de estilo CSS. El framework de recursos hace posible empaquetar fácilmente estos recursos en archivos jar junto con el código de los componentes personalizados.
- ✓ Generar fácilmente recursos binarios sobre la marcha. Los recursos del framework pueden generar imágenes, sonidos, hojas de cálculo de Excel, etc.
- ✓ Crear una moderna interfaz de usuario 'look-and-feel' basadas en tecnología de skins. RichFaces proporciona una función que permite definir y administrar

fácilmente diferentes esquemas de color y otros parámetros de la interfaz de usuario, con la ayuda de los parámetros del skin. Por lo tanto, es posible acceder a los parámetros del skin desde el código JSP y el código de Java (por ejemplo, para ajustar las imágenes generadas sobre la marcha basadas en la interfaz de usuario). RichFaces viene con una serie de skins predefinidas para empezar, pero también se pueden crear fácilmente skins propios.

- ✓ Permite gracias a una herramienta del framework generar casos de prueba para los componentes que estas creando (actions, listeners, etc.).
- ✓ Los componentes de la interfaz de usuario de RichFaces vienen preparados para su uso fuera del paquete, así los desarrolladores ahorrarán tiempo y podrán disponer de las ventajas mencionadas para la creación de aplicaciones Web. Como resultado, la experiencia puede ser más rápida y fácil de obtener.
- ✓ RichFaces permite definir (por medio de etiquetas de JSF) diferentes partes de una página JSF que se desee actualizar con una solicitud Ajax, proporcionando así varias opciones para enviar peticiones Ajax al servidor.

4.3. COMPONENTES.

Entre los elementos que componen RichFaces se encuentran:

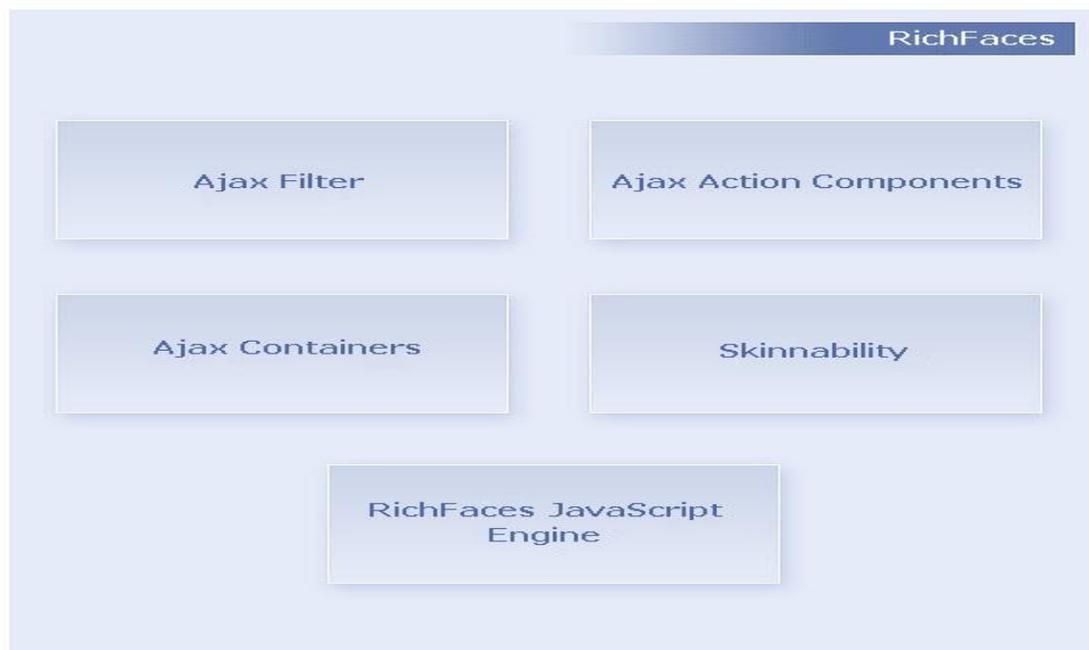


Ilustración 7. Componentes de RichFaces.

El marco Ajax RichFaces es una biblioteca de componentes JSF que añade la capacidad Ajax a sus páginas, a diferencia del apoyo tradicional y limitada de los componentes Ajax. Esto significa que usted puede utilizar componentes Ajax para invocar peticiones Ajax que se sincronizan automáticamente con el árbol de componentes JSF y actualizan las áreas individuales sin tener que recargar toda la página Web.

No es diferente de una página estándar de JSF, y ni siquiera necesita escribir código JavaScript al utilizar los componentes Ajax RichFaces. Dentro de la página usted puede definir las diferentes áreas que desea actualizar, después de la petición Ajax. La arquitectura del marco se compone de cinco partes:

- **Ajax Filter:** Esto es esencial para agregar capacidades Ajax para su aplicación JSF utilizando RichFaces. Administra todas las solicitudes (tanto Ajax y JSF estándar), corrige y valida el código de marcado enviado, administra la secuencia de comandos y la carga de estilo, la caché de los recursos y así sucesivamente. Usted tiene que colocar el filtro en el archivoweb.xml.
- **Componentes de Acción Ajax:** Son los componentes estándar de JSF que puede utilizar para enviar peticiones Ajax.
- **Contenedores Ajax:** El marco compatible con la interfaz AjaxContainer que describe un área ("región") de la página, que debe ser decodificada durante una petición Ajax. La mayor región es toda la página de JSF, pero usted puede definir cuántas regiones desea tener dentro de la página.
- **Skinnability:** Esta es una parte muy útil de la estructura y añade la capacidad de cambiar el aspecto a su aplicación.
- **Motor JavaScript RichFaces:** Se ejecuta en el navegador del cliente y administra las peticiones Ajax así como las respuestas. Son administrados automáticamente por el marco, por lo que no se tiene que utilizar directamente.

Puede decidir cuándo utilizar una petición JSF estándar (con una recarga completa de la página web) o cuándo utilizar una solicitud Ajax. En este último caso, sólo la región que participan Ajax es procesada, y el marcado Delta Ajax es enviado de vuelta al cliente después de que el filtro se ha analizado y verificado.

La verificación se hará porque la función JavaScript XMLHttpRequest envía la solicitud en formato XML, el marcado dentro de la petición de XML no se valida ni se corrigen. El filtro XML puede eliminar automáticamente los problemas de código HTML, pero es una buena práctica escribir en los estándares compatibles con XHTML y el código HTML.

Los componentes del marco RichFaces tienen una gran cantidad de atributos de Ajax, que son muy útiles para administrar las opciones que Ajax tiene.

Los componentes de atributos siguientes son muy importantes y se puede encontrar en todos los componentes Ajax permitidos en el marco RichFaces:

- reRender: Con el fin de decidir qué área debe ser actualizado después de una petición.
- ajaxRendered: Si es cierto, el área se actualiza después de cada petición Ajax (incluso si no es en el atributo reRender).
- limitToList: con el fin de forzar el motor de JavaScript para actualizar las únicas áreas en el atributo reRender.

4.4. INCONVENIENTES.

Como inconvenientes, podríamos decir que:

- ✓ Usando Ajax4JSF tenemos que indicar qué parte de la pantalla tiene que repintarse. No es tan simple como ICEfaces, pero implica tener más control sobre los eventos que se producen en la interfaz de usuario.
- ✓ En las últimas versiones tiene algunos inconvenientes en algunos componentes, que merma la funcionalidad y donde, por ejemplo, funcionaba la subida de ficheros mediante un componente JSF con barra de progreso en Internet Explorer, ahora solo funciona en Firefox. Aunque también es cierto que se detecta y soluciona en la siguiente versión.

4.5. REQUISITOS E INCOMPATIBILIDADES.

RichFaces fue desarrollado con una arquitectura de código abierto para ser compatible con la más amplia variedad de entornos.

Para poder trabajar con RichFaces es necesario disponer de una aplicación JSF. Además hay que tener en cuenta la versión de los diferentes componentes implicados:

Versiones de Java soportadas:

- ✓ JDK 1.5 y superiores.

Implementaciones de JavaServer Faces soportadas:

- ✓ Sun JSF 1.X RI
- ✓ MyFaces 1.1.1 - 1.X
- ✓ Facelets JSF 1.1.1 - 1.X
- ✓ Seam 1.2. - 2.X

Servidores soportados

- ✓ Apache Tomcat 4.1 - 6.X
- ✓ IBM WebSphere 5.1 - 6.X
- ✓ BEA WebLogic 8.1 - 9.X
- ✓ Oracle AS/OC4J 10.1.X
- ✓ Resin 3.X
- ✓ Jetty 5.1.X
- ✓ SunApplication Server 8 (J2EE 1.4)
- ✓ Glassfish (J2EE 5)
- ✓ JBoss 3.2 - 4.2.x
- ✓ SybaseEAServer 6.0.X

Navegadores admitidos

- ✓ Internet Explorer 6.0 - 7.X
- ✓ Firefox 1.5 - 2.X
- ✓ Opera 8.5 - 9.X

4.6. INTERACCIÓN CON OTROS COMPONENTES

RichFaces se integra con diferentes componentes para poder realizar aplicaciones más potentes:

4.6.1. Sun JSF RI

RichFaces trabaja con cualquier implementación de JSF (1.1 y 1.2) y con la mayoría de las bibliotecas de componentes JSF sin ninguna configuración adicional.

4.6.2. Apache MyFaces

RichFaces trabaja con todos Apache MyFaces versiones (1.1.1 - 1.1.6), incluidas las bibliotecas específicas como Tomahawk, Sandbox y Trinidad (el anterior ADF Faces). Sin embargo, hay algunas consideraciones a tener en cuenta para la configuración de las aplicaciones para trabajar con MyFaces y RichFaces. Hay algunos problemas con diferentes filtros definidos en el archivo web.xml. Para evitar estos problemas, el filtro de RichFaces debe ser el primero entre otros filtros dentro del archivo de configuración web.xml. Aún más, existen algunos problemas si se opta por utilizar MyFaces + Seam. Si se usa esta combinación, se debe usar el tag **a4j: page** dentro del f: view.

4.6.3. Facelets

RichFaces ofrece un alto nivel de soporte para Facelets. Cuando se trabaja con RichFaces, no hay diferencia entre las diferentes releases de Facelets que se utilizan.

4.6.4. JBoss Seam

Una de las novedades de RichFaces 3,1 es la integración con JBossSeam. Esto mejora aún más la experiencia del usuario mediante la simplificación de la configuración y el "plumbingcode", así como la prestación de estado y la concurrencia para la gestión de Ajax.

4.7. RECOMENDACIONES DE USO.

Con el fin de crear aplicaciones RichFaces correctamente, se debe tener en cuenta los siguientes puntos:

- ✓ Cualquier framework Ajax no debería añadir o suprimir, sólo sustituir elementos de la página. Para que las actualizaciones funcionen correctamente, en la página destino debe haber un elemento con el mismo ID que existe en la respuesta del servidor. Si se desea añadir cualquier código a la página, debería ponerse en un marcador de posición (cualquier lugar con algún elemento vacío). Por la misma razón, se recomienda colocar mensajes en el componente "AjaxOutput".
- ✓ No utilizar f:verbatim, ya que éste componente es transitorio y no se guarda en el árbol.
- ✓ Las peticiones Ajax se realizan gracias a funciones de XMLHttpRequest en formato XML, pero ese XML no pasa por la mayoría de validaciones y correcciones que pueden realizarse en el navegador. Por ello, sólo debe crearse código compatible con los estándares de HTML y XHTML, sin saltarse ninguno de los elementos o atributos requeridos. Cualquier corrección necesaria del XML se realizará automáticamente por el filtro XML en el servidor, pero muchos de los efectos inesperados pueden ser producidos por un código HTML incorrecto.

4.8. Enterprise JavaBeans.

Los Enterprise JavaBeans (también conocidos por sus siglas **EJB**) son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE (ahora JEE 5.0) de Oracle Corporation (inicialmente desarrollado por Sun Microsystems). Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, precisamente, los EJB:

- Comunicación remota utilizando CORBA
- Transacciones
- Control de la concurrencia
- Eventos utilizando JMS (Java messaging service)
- Servicios de nombres y de directorio
- Seguridad

- Ubicación de componentes en un servidor de aplicaciones.

La especificación de EJB define los papeles jugados por el contenedor de EJB y los EJB, además de disponer los EJB en un contenedor.

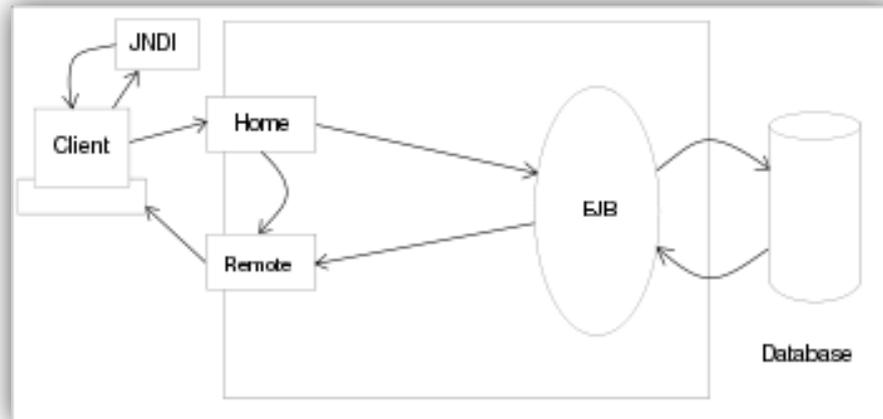


Ilustración 8. Enterprise Java Beans

4.8.1. Definición.

Los EJB proporcionan un modelo de componentes distribuido estándar del lado del servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad, etc.) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.

No hay que confundir los Enterprise JavaBeans con los JavaBeans. Los JavaBeans también son un modelo de componentes creado por Oracle - Sun Microsystems para la construcción de aplicaciones, pero no pueden utilizarse en entornos de objetos distribuidos al no soportar nativamente la invocación remota (RMI).

4.8.2. Tipos de Enterprise JavaBeans

Existen tres tipos de EJBs:

4.8.2.1. EJB de Entidad (*EntityEJBs*):

Su objetivo es encapsular los objetos del lado del servidor que almacena los datos. Los EJB de entidad presentan la característica fundamental de la persistencia:

- ✓ **Persistencia gestionada por el contenedor (CMP):** el contenedor se encarga de almacenar y recuperar los datos del objeto de entidad mediante el mapeo o vinculación de las columnas de una tabla de la base de datos con los atributos del objeto.
- ✓ **Persistencia gestionada por el bean (BMP):** el propio objeto entidad se encarga, mediante una base de datos u otro mecanismo, de almacenar y recuperar los datos a los que se refiere, por lo cual, la responsabilidad de implementar los mecanismos de persistencia es del programador.

4.8.2.2. EJB de Sesión (*SessionEJBs*):

Gestionan el flujo de la información en el servidor. Generalmente sirven a los clientes como una fachada de los servicios proporcionados por otros componentes disponibles en el servidor. Puede haber dos tipos:

- ✓ **Con estado (*stateful*).** En un bean de sesión con estado, las variables de instancia del bean almacenan datos específicos obtenidos durante la conexión con el cliente. Cada bean de sesión con estado, por tanto, almacena el estado conversacional de un cliente que interactúa con el bean. Este estado conversacional se modifica conforme el cliente va realizando llamadas a los métodos de negocio del bean. El estado conversacional no se guarda cuando el cliente termina la sesión.
- ✓ **Sin estado (*stateless*).** Los beans de sesión sin estado son objetos distribuidos que carecen de estado asociado permitiendo por tanto que se los acceda concurrentemente. No se garantiza que los contenidos de las variables de instancia se conserven entre llamadas al método.

4.8.2.3. EJB dirigidos por mensajes (*Message-drivenEJBs*):

Son los únicos beans con funcionamiento asíncrono. Usando el *Java MessagingSystem (JMS)*, se suscriben a un tema (*topic*) o a una cola (*queue*) y se activan al recibir un mensaje dirigido a dicho tema o cola. No requieren de su instanciación por parte del cliente.

4.8.3. Funcionamiento de un Enterprise JavaBean.

Los EJB se disponen en un contenedor EJB dentro del servidor de aplicaciones. La especificación describe cómo el EJB interactúa con su contenedor y cómo el código cliente interactúa con la combinación del EJB y el contenedor.

Cada EJB debe facilitar una clase de implementación Java y dos interfaces Java. El contenedor EJB creará instancias de la clase de implementación Java para facilitar la implementación EJB. Las interfaces Java son utilizadas por el código cliente del EJB. Las dos interfaces, conocidas como interfaz "home" e interfaz remota, especifican las firmas de los métodos remotos del EJB. Los métodos remotos se dividen en dos grupos:

- Métodos que no están ligados a una instancia específica, por ejemplo aquellos utilizados para crear una instancia EJB o para encontrar una entidad EJB existente. Estos métodos se declaran en la interfaz "home".
- Métodos ligados a una instancia específica. Se ubican en la interfaz remota.

Dado que se trata simplemente de interfaces Java y no de clases concretas, el contenedor EJB genera clases para esas interfaces que actuarán como un proxy⁷ en el cliente. El cliente invoca un método en los proxies generados que a su vez sitúa los argumentos método en un mensaje y envía dicho mensaje al servidor EJB. Los proxies usan RMI-IIOP para comunicarse con el servidor EJB.

El servidor llamará a un método correspondiente a una instancia de la clase de implementación Java para manejar la llamada del método remoto.

⁷ programa o dispositivo que realiza una acción en representación de otro.

4.8.4. Interfaz "Home"

La interfaz "Home" permite al código cliente manipular métodos de clase del EJB que no están asociados a ninguna instancia particular. La Interface "Home" permite crear las instancias de EJB de entidad o sesión a través del método create que puede ser sobrecargado.

La especificación EJB 1.1 establece el tipo de métodos de clase que se pueden definir como métodos que crean un EJB o para encontrar un EJB existente si es un "bean" de entidad.

La especificación EJB 2.0 permite a los desarrolladores de aplicaciones definir nuevos métodos de clase sin limitarse a su sola creación o borrado.

4.8.5. Interfaz remota.

La interfaz remota especifica los métodos de instancia públicos encargados de realizar las operaciones.

Una sesión bean puede implementar múltiples interfaces, con cada interfaz apuntada por un tipo de cliente diferente. La interfaz local es para aquellos clientes que corren en la misma máquina virtual que el contenedor EJB. La interfaz remota es para clientes remotos. Frente a una consulta del cliente, el contenedor retorna un stub serializado del objeto que implementa la interfaz remota. El stub conoce cómo pasar llamadas a procedimientos remotos (RPCs) al servidor. Este tipo de interfaz es también un POJO.

4.8.6. Clase de implementación EJB

Las clases de implementación EJB las suministran los desarrolladores de aplicaciones, que facilitan la lógica de negocio ("businesslogic") o mantienen los datos ("business data") de la interfaz de objeto, esto es, implementan todos los métodos especificados por la interfaz remota y, posiblemente, algunos de los especificados por la interfaz "home".

4.8.7. Correspondencia entre métodos de interfaz y métodos de implementación.

Las llamadas al método en la interfaz "home" se remiten al método correspondiente de la clase de implementación del bean con el prefijo 'ejb' añadido y con la primera letra de la interfaz "home" convertida en mayúscula y manteniendo exactamente el mismo tipo de argumentos⁸. Por ejemplo:

create --->ejbCreate.

Las llamadas a métodos en la interfaz remota se remiten al método de implementación correspondiente del mismo nombre y argumentos en la clase del bean.

La complejidad ciclomática de las unidades semánticas de navegación (NavigationSemanticUnit (NSU)) no cumple el estándar UML 2.0 que recomienda el uso de screenshots sobre componentes programáticos.

⁸Developing EJB Applications

D. RESULTADOS

1. DESARROLLO DE LA PROPUESTA ALTERNATIVA

- Analizar los Requerimientos del sistema OPENLOJA.

Para obtener los requerimientos del sistema que no son más que los requisitos iniciales de la empresa, se debe interactuar directamente con el personal de la misma para obtener la mayor cantidad de información y determinar dichos requerimientos los cuales se pueden visualizar en las tablas de requerimientos funcionales y no funcionales del sistema correspondiente a la sección de discusión de nuestro proyecto.

- Diseñar los módulos de bodega, ventas, configuración, RR HH y comunicación del sistema OPENLOJA.

El diseño del sistema se puede observar en la sección de discusión del proyecto, el mismo que se realizó por módulos entre los que constan bodega, ventas, configuración, RR HH, y comunicación, los cuales cuentan con sus respectivos casos de uso, descripción y diagramas de los mismos; modelos de interacción (secuencia), de datos y de clases; así como también de la arquitectura del sistema descrita en los diagramas de despliegue, paquetes y componentes; éste se realizó de manera eficiente aplicando todos los conocimientos adquiridos y a través del uso de la metodología ICONIX para su desarrollo cumpliendo con todos los requerimiento obtenidos anteriormente; logrando así obtener el diseño más de adecuado y proceder a realizar la construcción del sistema.

- Construir los módulos de ventas, bodega, RR HH, comunicación, configuración del sistema OPENLOJA.

La implementación del diseño anteriormente realizado se lo hizo bajo el marco de Seam integrado en el entorno de desarrollo eclipse considerando el desarrollo orientado a objetos, razón por la cual se utilizó el patrón Modelo Vista Controlador. Esto se puede observar fácilmente en la aplicación la cual está diseñada y construida modularmente, para lo cual se encuentran disponibles los siguientes menús de gestión de: Empleados, Clientes, Proveedores, Bodega, Ventas, Configuración y Comunicación, cubriendo así todos los requerimientos del sistema.

- Integrar los módulos del sistema OPENLOJA previamente construidos.
El sistema OpenLoja 1.0 es la prueba clara de esto donde se puede observar todos los módulos construidos e integrados, interactuando correctamente con el usuario y cumpliendo con todas las solicitudes para lo cual está diseñado.
- Implementar el sistema OPENLOJA en la empresa Lojanet++.
La implementación del sistema se ejecuto eficientemente ya que se utilizan herramientas libres, se configuro un servidor para su normal funcionamiento y de esta manera se cumple con los requerimientos de la empresa lo cual se puede constatar en el certificado otorgado por la empresa Lojanet++ ubicado dentro de los anexos.
- Ejecutar el plan de validación al sistema OPENLOJA.
Mediante el uso de testeos considerados convenientemente para aplicarlos al personal de la empresa interactuando con el sistema, se garantizó su calidad y cumplimiento de todos los requerimientos. Este plan se encuentra ubicado en la sección de discusión, el mismo que contiene pruebas de la interfaz gráfica, funcionalidad y seguridad; lo cual corrobora que el plan de validación se ejecutó en la empresa Lojanet++ a través de la encuestas realizadas al personal las cuales se pueden observar en el apartado de Anexos correspondiente a éste proyecto.

2. VALORACIÓN TÉCNICO ECONÓMICA AMBIENTAL

Durante el desarrollo de éste proyecto, se propuso éste tema, tomando en consideración el auge de las tecnologías de código abierto (**OpenSource**), y la gran necesidad de contar con un sistema que permita una adecuada organización de los recursos empresariales y que le permita a la entidad una correcta comunicación con los entes involucrados en los procesos cotidianos de la empresa, por lo cual hemos creído conveniente justificar el siguiente proyecto en los siguientes ámbitos:

2.1. Académica

La Universidad Nacional de Loja dentro de sus objetivos es la formación de profesionales capaces de desenvolverse en el accionar profesional y consientes de la

gran importancia que tiene poner en práctica los conocimientos adquiridos durante el transcurso de la carrera, detallado en el Sistema Académico Modular por Objetos de Transformación (SAMOT); lo cual nos permitirá actuar eficientemente ante cualquier situación que se nos presente durante nuestra futura vida profesional, hemos considerado conveniente justificar el proyecto en lo académico, aportando a la realización profesional, reafirmar sustentos teóricos y ponerlos en práctica, previo a la obtención del título de Ingeniero en Sistemas mediante la ejecución de éste proyecto, logrando profesionales con presencia a nivel local y nacional.

2.2. Científico – Técnica

Tomando en consideración el sustento técnico y tecnológico para el normal progreso del proyecto, el tema que ponemos a consideración brinda las facilidades necesarias tanto en hardware como software.

En lo que respecta al hardware el proyecto presta las facilidades para cubrir cada uno de los requerimientos que se presenta durante el desarrollo del proyecto en mención, de igual forma el software brinda la versatilidad necesaria dando que el tema en mención se encuentra enmarcado en los ámbitos del código abierto (OpenSource), brindando el sustento teórico necesario que facilita a cada uno de los integrantes del grupo de trabajo como foros, salas de aprendizaje entre otros.

2.3. Económica

Durante el progreso de este proyecto, nuestro grupo contará con los recursos necesarios, que serán proporcionados por la empresa involucrada, la misma que es consciente de la inversión que realiza y los resultados a corto plazo a obtener, por otra parte para la elaboración del mismo utilizaremos software de licencia libre, para lo cual no necesitaremos un aporte económico adicional, facilitándonos el cumplimiento de todos los requerimientos que se presenten durante el transcurso del proyecto.



E. DISCUSIÓN

REQUERIMIENTOS DEL SISTEMA

1. REQUERIMIENTOS DEL SISTEMA

1.1. REQUERIMIENTOS FUNCIONALES

El sistema permitirá:

CÓDIGO	DESCRIPCIÓN	CATEGORIA
RF001	Al usuario ingresar mediante nombre de usuario y contraseña.	Visible
RF002	Al usuario salir del sistema.	Visible
RF003	Al usuario configurar de forma general el sistema tomando en cuenta los datos de la organización, entidad o empresa.	Visible
RF004	Al usuario crear cuentas de usuario con privilegios de acceso.	Visible
RF005	Al usuario modificar cuentas de usuario.	Visible
RF006	Al usuario consultar cuentas de usuario.	
RF007	Al usuario crear y modificar clientes, empleados, proveedores, productos.	Visible
RF008	Al usuario consultar generando listado de clientes, empleados, proveedores, productos.	Visible
RF009	Al usuario crear horarios de trabajo y días festivos.	Visible
RF010	Al usuario crear, editar y consultar: departamentos, permisos de trabajo, capacitaciones de empleados y anticipos de pago.	Visible
RF011	Al usuario registrar su Asistencia eficientemente (hora entrada, hora salida, número de horas) considerando vacaciones, permisos, faltas.	Visible
RF012	Al usuario consultar las asistencias de un empleado en específico.	Visible
RF013	Al usuario asignar departamentos y horarios de trabajo a los empleados correspondientes.	Visible
RF014	Al usuario generar Roles de Pago de Empleados generales (de todos los empleados) tomando en consideración beneficios, multas, asistencias.	Visible
RF015	Al usuario generar Roles de pago individual (de un solo empleado) tomando en consideración beneficios, multas, descuentos, asistencias.	Visible
RF016	Al usuario crear familias de producto.	Visible
RF017	Al empleado registrar ingreso de productos a bodega tomando en consideración compra de producto, stock bajo, devoluciones.	Visible
RF018	Al empleado registrar salida de productos de bodega tomando en consideración ventas, devoluciones.	Visible
RF019	Al usuario editar los registros de ingresos o egresos de bodega anulándolos.	Visible
RF020	Al usuario realizar consultas de ingresos y egresos de productos a bodega.	Visible
RF021	Al usuario generar inventario general de productos actualizado.	Visible
RF022	Al usuario generar Kárdex de cada producto existente en bodega.	Visible

RF023	Al usuario registrar comprobantes de compra entregados por el proveedor.	Visible
RF024	Al usuario emitir comprobante (nota de venta, factura) de venta tomando en consideración (stock bodega, pedidos, ofertas, descuentos, forma de pago).	Visible
RF025	Al usuario anular comprobante de venta cuando se realice una venta o devolución.	Visible
RF026	Al usuario visualizar historial de ventas de productos.	Visible
RF027	Al usuario enviar notificaciones a Empleados, Clientes y Proveedores vía email.	Visible
RF028	Al usuario crear grupos de contactos.	Visible
RF029	Al usuario crear y editar contactos de agenda para el manejo de notificaciones.	Visible

Tabla 2. Requerimientos Funcionales

1.2. REQUERIMIENTOS NO FUNCIONALES

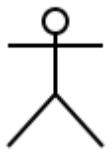
CÓDIGO	DESCRIPCIÓN	CATEGORIA
RNF001	El sistema será desarrollado bajo la plataforma de programación JAVA.	Oculto
RNF002	El sistema se desenvolverá bajo entorno web utilizando para ello framework JBoss Seam que integra Hibernate, richfaces, jpa, entre otros.	Oculto
RNF003	El sistema deberá ser escalable es decir que nos permita agregar módulos adicionales.	Oculto
RNF004	El sistema será multiusuario.	Oculto
RNF005	El sistema será multiplataforma o funcionará en varias plataformas.	Oculto
RNF006	El sistema utilizará como base de datos MySQL.	Oculto

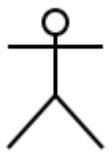
Tabla 3. Requerimientos Funcionales.

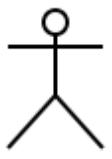


IDENTIFICACIÓN DE LOS ACTORES Y DE CASOS DE USO

2. IDENTIFICACIÓN DE LOS ACTORES Y CASOS DE USO DE OPENLOJA

	REQUERIMIENTO	CASOS DE USO	PAQUETES
 Usuario	Ingresar al sistema a través de su cuenta	Login	Action
	Cerrar Sesión	Logout	Action
	Configuración general del sistema	Configuración General	Action
	Crear cuenta de Usuario	Crear Cuenta Usuario	Action
	Editar Cuenta de Usuario	Modificar Cuenta Usuario	Action
	Consultar Cuenta de Usuario	Consultar Cuenta Usuario	Dao
	Crear Empleado	Crear Empleado	Action
	Editar Empleado	Modificar Empleado	Action
	Buscar Empleado	Consultar Empleado	Dao
	Crear Departamento	Crear Departamento	Action
	Asignar Departamento	Asignar Departamento	Action
	Editar Departamento	Modificar Departamento	Action
	Consultar Departamento	Consultar Departamento	Dao
	Crear Horario	Crear Horario	Action

 Usuario	Asignar Horario	Asignar Horario	Action
	Registrar Asistencias	Registro Entrada/Salida	Action
	Consultar Asistencias de Empleado	Consultar Asistencia	Dao
	Crear Permiso de trabajo	Crear Permiso	Action
	Editar Permiso de trabajo	Editar Permiso	Action
	Consultar Permisos de trabajo	Consultar Permiso	Dao
	Crear Capacitación	Crear Capacitación	Action
	Editar Capacitación	Modificar Capacitación	Action
	Consultar Capacitación	Consultar Capacitación	Dao
	Crear Anticipo de Pago de Sueldo	Crear Anticipo	Action
	Editar Anticipo de Pago	Modificar Anticipo	Action
	Generar Rol de Pago General	Generar Rol General	Action
	Generar Rol de Pago Individual	Generar Rol Individual	Action
	Consultar Anticipos de Empleado	Consultar Anticipo	Dao
	Crear Cliente	Crear Cliente	Action
	Editar Cliente	Modificar Cliente	Action
Consultar Cliente	Consultar Cliente	Dao	

 Usuario	Crear Proveedor	Crear Proveedor	Action
	Modificar Proveedor	Modificar Proveedor	Action
	Consultar Proveedor	Consultar Proveedor	Dao
	Crear Familia de Producto	Crear Familia Producto	Action
	Crear Producto	Crear Producto	Action
	Editar Producto	Modificar Producto	Action
	Consultar Producto	Consultar Producto	Dao
	Registrar Ingresos y Egresos de Bodega	Crear Movimiento Bodega	Action
	Editar Registros de Ingreso y Egreso	Modificar Movimiento Bodega	Action
	Consultar registros de ingreso y egreso	Consultar Movimiento Bodega	Dao
	Generar Inventario	Generar Inventario	Action
	Generar Kárdex de Producto	Generar Kárdex	Action
	Emitir comprobante de Venta	Crear Venta	Action
	Anular comprobante de Venta	Modificar Venta	Action
	Visualizar Historial de Ventas	Consultar Venta	Dao
	Enviar notificaciones vía e-mail	Crear Notificación	Action
Crear Grupo de Contactos	Crear Grupo Contactos	Action	



	Crear Contacto de Agenda	Crear Contacto	Action
	Editar Contacto de Agenda	Modificar Contacto	Action

Tabla 4. Identificación de Casos de uso



DISEÑO DE LOS MÓDULOS DEL SISTEMA



GESTIÓN DE CONFIGURACIÓN

3. DISEÑO DE LOS MÓDULOS DEL SISTEMA

3.1. MÓDULO DE GESTIÓN DE CONFIGURACIÓN

3.1.1. MODELO DE DOMINIO

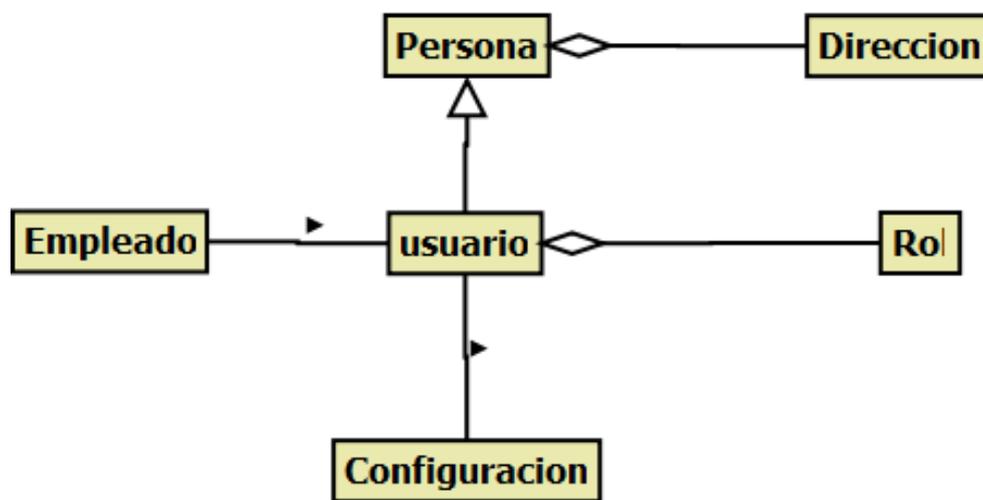


Diagrama 1. Módulo de Configuración Versión Final

3.1.2. MODELO DE CASOS DE USO DE L SISTEMA.

3.1.2.1. DIAGRAMA DE CASOS DE USO

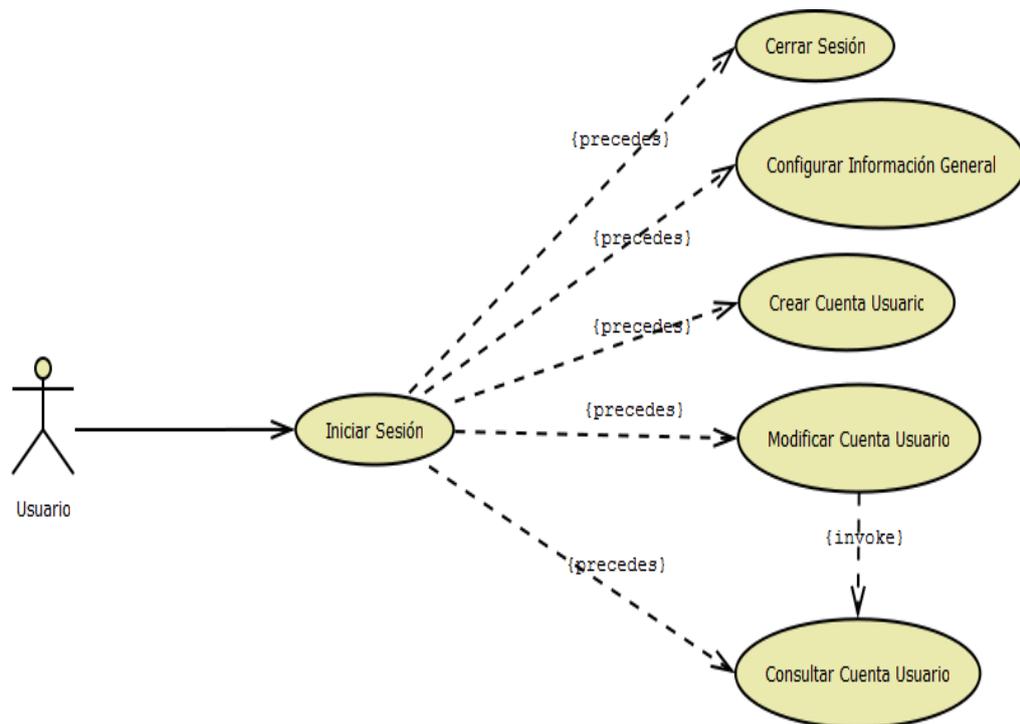


Diagrama 2. Gestión Configuración Versión Final

3.1.2.2. DESCRIPCIÓN DE CASOS DE USO

➤ CASO DE USO: Iniciar Sesión

NOMBRE: Login	CÓDIGO	OP001
NOMBRE PROGRAMADO	login.xhtml	

Tabla 5. Pantalla Iniciar Sesión

Nombre:	Iniciar Sesión	Código: 001
Requerimiento Funcional:	RF001	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir a un usuario registrado hacer uso de sistema.	
Descripción:	Que el usuario una vez ingresado a la pantalla "Login" ingrese su nombre de usuario y contraseña, eligiendo Login para acceder al sistema y poder realizar las funciones correspondientes al rol asignado.	
Precondiciones:	El usuario debe estar ubicado en la Pantalla Login.	
Poscondiciones:	Sesión de Usuario iniciada.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario ingresa nombre de usuario y contraseña en la pantalla web "Login". 2. El usuario elige el botón Login de la pantalla web "Login". 3. El sistema valida que los campos obligatorios no estén vacíos. 4. El sistema verifica que los datos ingresados correspondan a una Cuenta de Usuario existente. 5. El sistema recupera la cuenta de usuario en base al nombre de usuario. 6. El sistema muestra la pantalla web "OpenLoja" con una sesión iniciada, activando las 		

funciones que puede realizar en relación a su rol asignado.
7. El caso de uso finaliza.
FLUJO ALTERNO:
<p>A. CAMPOS OBLIGATORIOS VACIOS.</p> <p>A.3 El sistema muestra un mensaje de campos obligatorios vacíos.</p> <p>A.4 El caso de uso continúa en el paso 1 del flujo normal de eventos.</p> <p>B. CUENTA DE USUARIO NO EXISTE.</p> <p>B.4 El sistema muestra un mensaje de Cuenta de Usuario no existe.</p> <p>B.5 El caso de uso continúa en el paso 1 del flujo normal de eventos.</p>

Tabla 6. Descripción Caso de Uso Iniciar Sesión

➤ **CASO DE USO: Cerrar Sesión**

NOMBRE: Logout		CÓDIGO	-----
CASO DE USO	Cerrar Sesión		
NOMBRE PROGRAMADO	-----		
			

Tabla 7. Pantalla Cerrar Sesión

Nombre:	Logout		Código: 002
Requerimiento Funcional:	RF002		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Al usuario permitir cerrar sesión de trabajo.		
Descripción:	Qué el usuario previamente autenticado pueda cerrar su sesión de trabajo.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario selecciona la opción [Logout] de la pantalla web principal "OPENLOJA 1.0" 2. El sistema cierra la sesión de la Cuenta de Usuario. 3. El sistema muestra la pantalla web "Login". 4. El caso de uso finaliza. 			

Tabla 8. Descripción Caso de Uso Logout

➤ CASO DE USO: Configurar Información General

NOMBRE: Configuración General	CÓDIGO	OP002
NOMBRE PROGRAMADO	Xgeneral.xhtml	

GUARDAR
BORRAR
CANCELAR

Nombre Empresa:

Responsable:

ruc

Autorización SRI

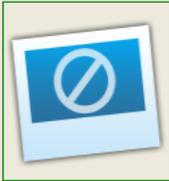
Inicio Facturación

Fecha Creación Jun 20, 2012

aportePersonal %

aportePatronal %

Estado Empresa: Activo Inactivo



Datos Correo Horario

Ciudad

Telefono

Celular

Tabla 9. Pantalla Configuración General

Nombre:	Configurar Información General	Código: 004
Requerimiento Funcional:	RF002	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol pueda configurar la información general del sistema.	
Descripción:	Qué el usuario al cual se ha asignado este rol pueda configurar la información general del sistema tomando en cuenta los datos de la organización o empresa.	
Precondiciones:	El usuario tiene que estar logueado. El usuario tiene que seguir la siguiente ruta Menú Aplicación, Submenú Gestión de Configuración y selecciona el Ítem Entidad.	
Poscondiciones:	Configuración General del Sistema.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción [General] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Configuración General". 3. El usuario ingresa la información general tomando de la empresa, entidad u organización donde se vaya a utilizar el sistema.. 4. El usuario selecciona el botón [Guardar] de la pantalla web "Configuración General". 5. El sistema valida que los campos obligatorios no estén vacíos. 6. El sistema guarda la información ingresada por el usuario. 7. El sistema muestra un mensaje de Configuración exitosa. 8. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.5 El sistema muestra un mensaje de campos obligatorios vacíos.</p> <p>A.6 El caso de uso continúa en el paso 3 del flujo normal de eventos.</p>		

Tabla 10.Descripción caso de Uso Configuración General

➤ CASO DE USO: Crear Cuenta Usuario

NOMBRE: Crear Usuario	CÓDIGO	OP003
NOMBRE PROGRAMADO	XcrearUsuario.xhtml	

GUARDAR
BORRAR

DNI *

Nombres *

Apellidos *

Fecha Nacimiento 

Usuario

Contraseña

Descripcion

Estado



roles
datos

asignar

Rol	Estad
No data	

Tabla 11. Pantalla Crear Usuario

NOMBRE: Panel Modal Buscar Rol	CÓDIGO	PM01
NOMBRE PROGRAMADO		

Rol


Nombre*	Estado	
ADMINISTRADOR	●	<input type="checkbox"/>
EMPLEADO	●	<input type="checkbox"/>
BODEGUERO	●	<input type="checkbox"/>
TEST	●	<input type="checkbox"/>
VENTAS	●	<input type="checkbox"/>
BODEGUERO	●	<input type="checkbox"/>

ACEPTAR
CANCELAR

Tabla 12. Panel Modal Buscar Rol

Nombre:	Crear Cuenta Usuario	Código: 004
Requerimiento Funcional:	RF004	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con el rol asignado pueda crear cuentas de usuarios.	
Descripción:	Qué el usuario previamente autenticado con privilegios pertinentes pueda crear cuenta usuarios.	
Precondiciones:	El usuario tiene que estar logueado. El usuario tiene que seguir la siguiente ruta Menú Aplicación, Submenú Gestión de Configuración y selecciona el Ítem Usuario.	
Poscondiciones:	Cuenta de Usuario creada.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción [Crear] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea la Cuenta de Usuario y muestra la pantalla web "Crear Usuario". 3. El usuario ingresa los datos del Usuario. 4. El usuario elige la pestaña Rol y elige la opción [Asignar]. 5. El sistema muestra el Panel Modal "Rol". 6. El usuario selecciona el Rol que va asignar al Usuario. 7. El sistema carga los datos del Rol seleccionado a la pantalla web "Crear Usuario" y automáticamente se cierra el panel modal "Rol". 8. El usuario selecciona el botón [Guardar] de la pantalla web "Crear Usuario". 9. El sistema valida que los campos obligatorios no estén vacíos. 10. El sistema verifica que se haya asignado un Rol al Usuario. 11. El sistema verifica a través del DNI que la Cuenta de Usuario no coincida con otra ya existente. 12. El sistema guarda la nueva Cuenta de Usuario. 13. El sistema muestra un mensaje de Usuario se ha creado con éxito. 14. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.9 El sistema muestra un mensaje de campos obligatorios vacíos.</p> <p>A.10 El caso de uso continúa en el paso 3 del flujo normal de eventos.</p> <p>B. ROL NO ASIGNADO</p> <p>B.10 El sistema muestra un mensaje de no se ha asignado un Rol al Usuario.</p> <p>B.11 El caso de uso continúa en paso 4 del flujo normal de eventos.</p> <p>C. CUENTA USUARIO EXISTENTE</p> <p>C.11 El sistema muestra un mensaje de Cuenta de Usuario ya existe.</p> <p>C.12 El caso de uso continúa en el paso 3 del flujo normal de eventos.</p>		

Tabla 13. Descripción caso de uso Crear Usuario

➤ CASO DE USO: Modificar Cuenta Usuario

NOMBRE: Modificar Cuenta Usuario	CÓDIGO	OP004
NOMBRE PROGRAMADO	XeditarUsuario.xhtml	

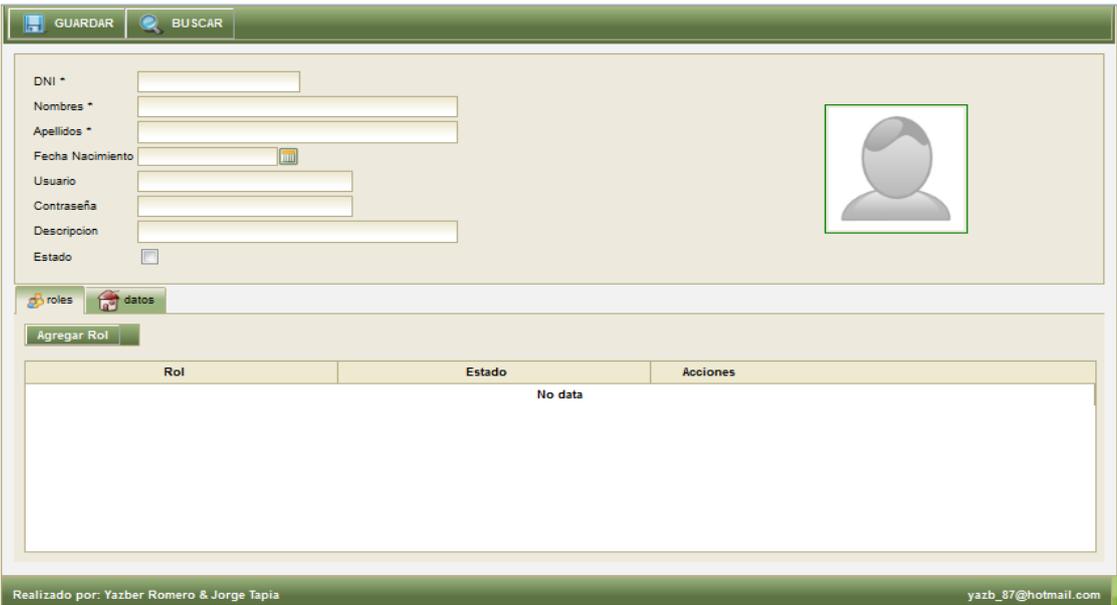


Tabla 14. Pantalla Modificar Cuenta Usuario

NOMBRE: Panel Modal Buscar Usuario	CÓDIGO	PM02
NOMBRE PROGRAMADO		

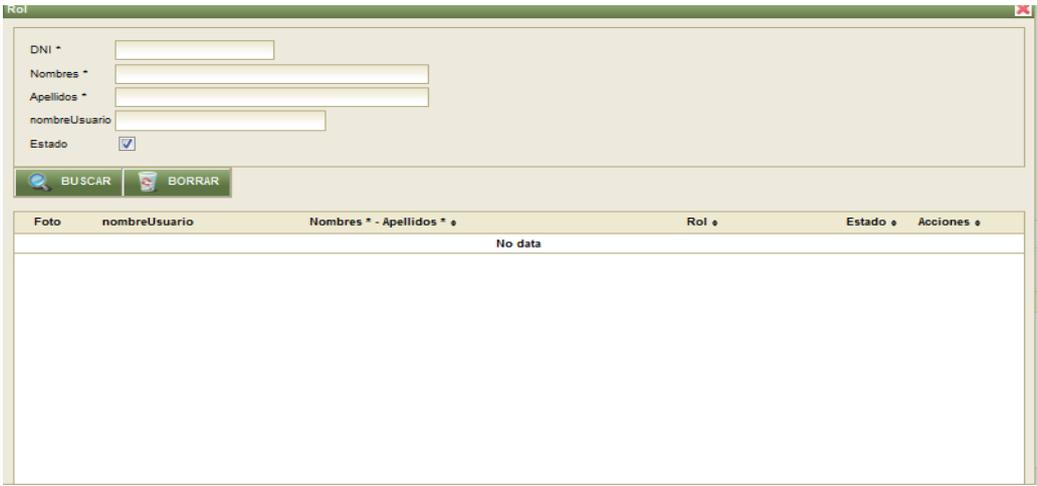


Tabla 15. Panel Modal Buscar usuario

Nombre:	Modificar Cuenta Usuario	Código: 005
Requerimiento Funcional:	RF005	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con el rol asignado pueda modificar una cuenta usuario previamente creado.	
Descripción:	Al usuario realizar modificaciones de cuenta usuarios previamente creados.	
Precondiciones:	El usuario tiene que estar logueado. El usuario tiene que seguir la siguiente ruta Menú Aplicación, Submenú Gestión de Configuración y selecciona el Ítem Usuario.	
Poscondiciones:	Cuenta de Usuario seleccionada modificada.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Editar Usuario". 3. El usuario selecciona la opción [Buscar Usuario]. 4. El sistema muestra el panel modal "Buscar Usuario" 5. Se invoca (se utiliza) el Caso de Uso <u>Consultar Cuenta Usuario</u>. 6. El selecciona el Usuario que va a modificar del Panel Modal "Buscar Usuario". 7. El sistema carga los datos del Usuario seleccionado en la pantalla "Editar Usuario" y automáticamente se cierra el panel modal "Buscar Usuario". 8. El usuario modifica los datos que desea cambiar en la Cuenta del Usuario. 9. El usuario selecciona la opción [Guardar]. 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica que se haya asignado un Rol al Usuario. 12. El sistema verifica a través del DNI que la Cuenta de Usuario no coincida con otra ya existente. 13. El sistema guarda los nuevos datos de la Cuenta de Usuario editada. 14. El sistema muestra un mensaje de Cuenta de Usuario se ha modificado con éxito. 15. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.10. El sistema muestra un mensaje de campos obligatorios vacíos.</p> <p>A.11. El caso de uso continúa en el paso 8 del flujo normal de eventos.</p> <p>B. ROL NO ASIGNADO</p> <p>B.11. El sistema muestra un mensaje de no se ha asignado un Rol al Usuario.</p> <p>B.12. El caso de uso continúa en paso 8 del flujo normal de eventos.</p> <p>C. CUENTA USUARIO EXISTENTE</p> <p>C.12. El sistema muestra un mensaje de Cuenta de Usuario ya existe.</p> <p>C.13. El caso de uso continúa en el paso 8 del flujo normal de eventos.</p>		

Tabla 16. Descripción Caso de Uso Modificar Cuenta Usuario

➤ CASO DE USO: Consultar Cuenta Usuario

NOMBRE: Buscar Usuario	CÓDIGO	OP005
NOMBRE PROGRAMADO	XlistarUsuario.xhtml	

NUEVO
IMPRIMIR
✖ CANCELAR

DNI *

Nombres *

Apellidos *

nombreUsuario

Estado

🔍 BUSCAR
🗑 BORRAR

Foto	nombreUsuario	Nombres * - Apellidos * *	Rol *	Estado *
No data				

Tabla 17. Pantalla Consultar Cuenta Usuario

Nombre:	Consultar Cuenta Usuario	Código: 006
Requerimiento Funcional:	RF006	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario poder consultar una cuenta previamente creado.	
Descripción:	Al usuario realizar consultas de cuentas de usuarios previamente creados.	
Precondiciones:	El usuario tiene que estar logueado. El usuario tiene que seguir la siguiente ruta Menú Aplicación, Submenú Gestión de Configuración y selecciona el Ítem Usuario.	
Poscondiciones:	Cuenta de Usuario visualizada.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción [Consultar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla "Buscar Usuario". 3. El usuario ingresa los datos de la Cuenta que desea buscar. 4. El usuario elige la opción [Buscar]. 5. El sistema muestra las Cuentas de Usuario que coinciden con los datos de búsqueda en la pantalla web "Buscar Usuario". 6. El caso de uso finaliza. 		

Tabla 18. Descripción Caso de Uso Consultar Cuenta Usuario

3.1.3. MODELO DE INTERACCIÓN

3.1.3.1. DIAGRAMA DE SECUENCIA 001: Iniciar Sesión (DS001)

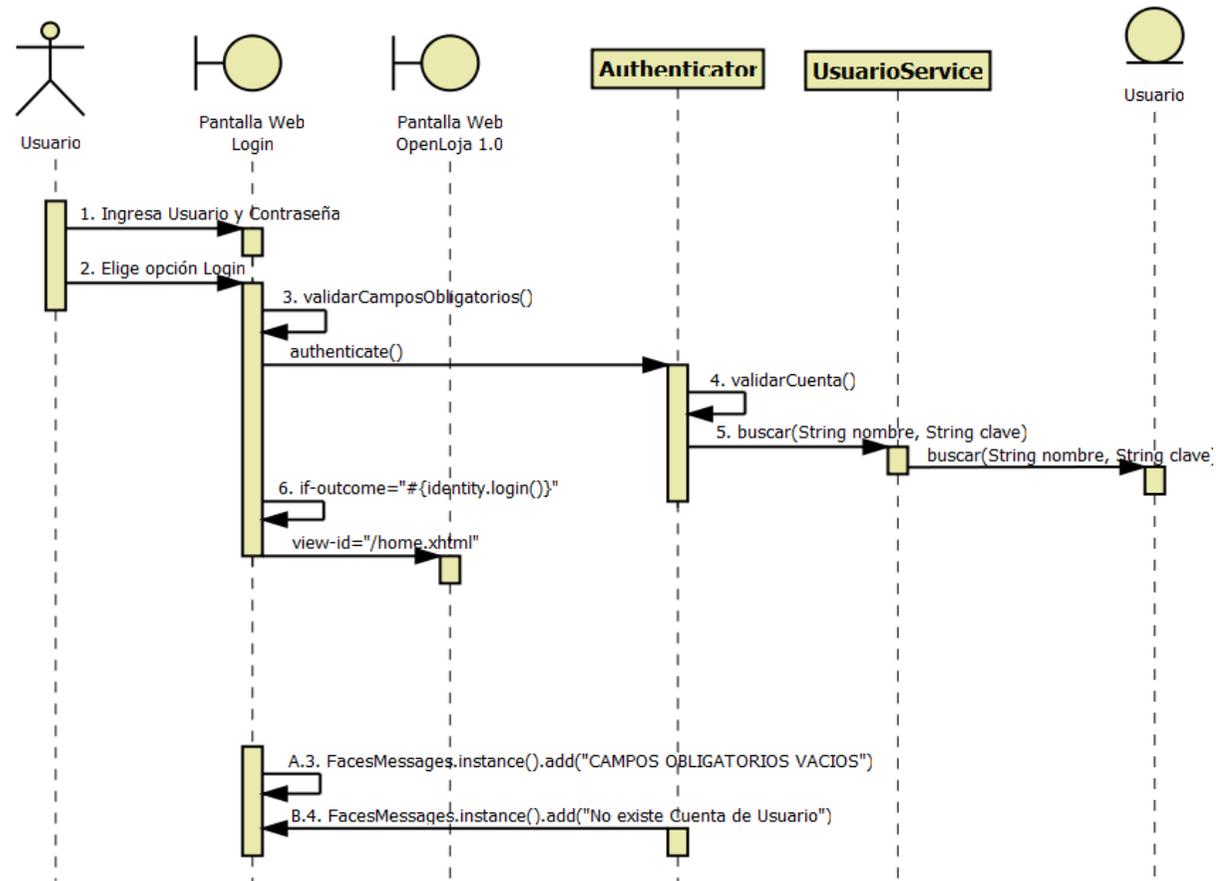


Diagrama 3. Secuencia Iniciar Sesión Versión Final

3.1.3.2. DIAGRAMA DE SECUENCIA 002: Cerrar Sesión (DS002)

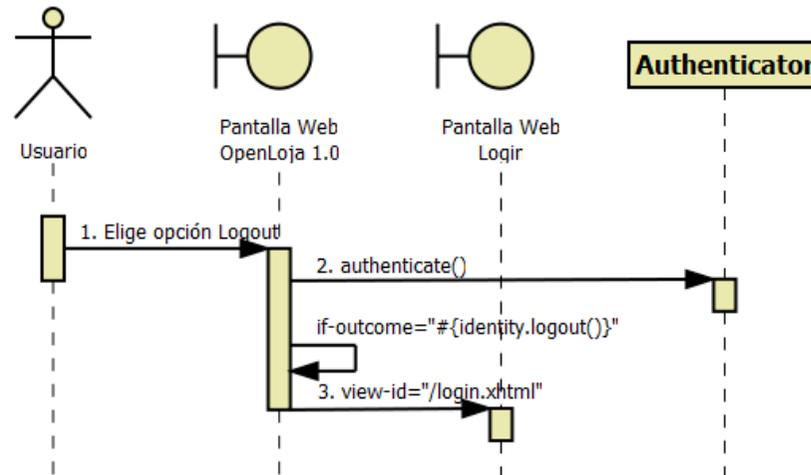


Diagrama 4. Secuencia Cerrar Sesión Versión Final

3.1.3.3. DIAGRAMA DE SECUENCIA: Configurar Información General (DS003)

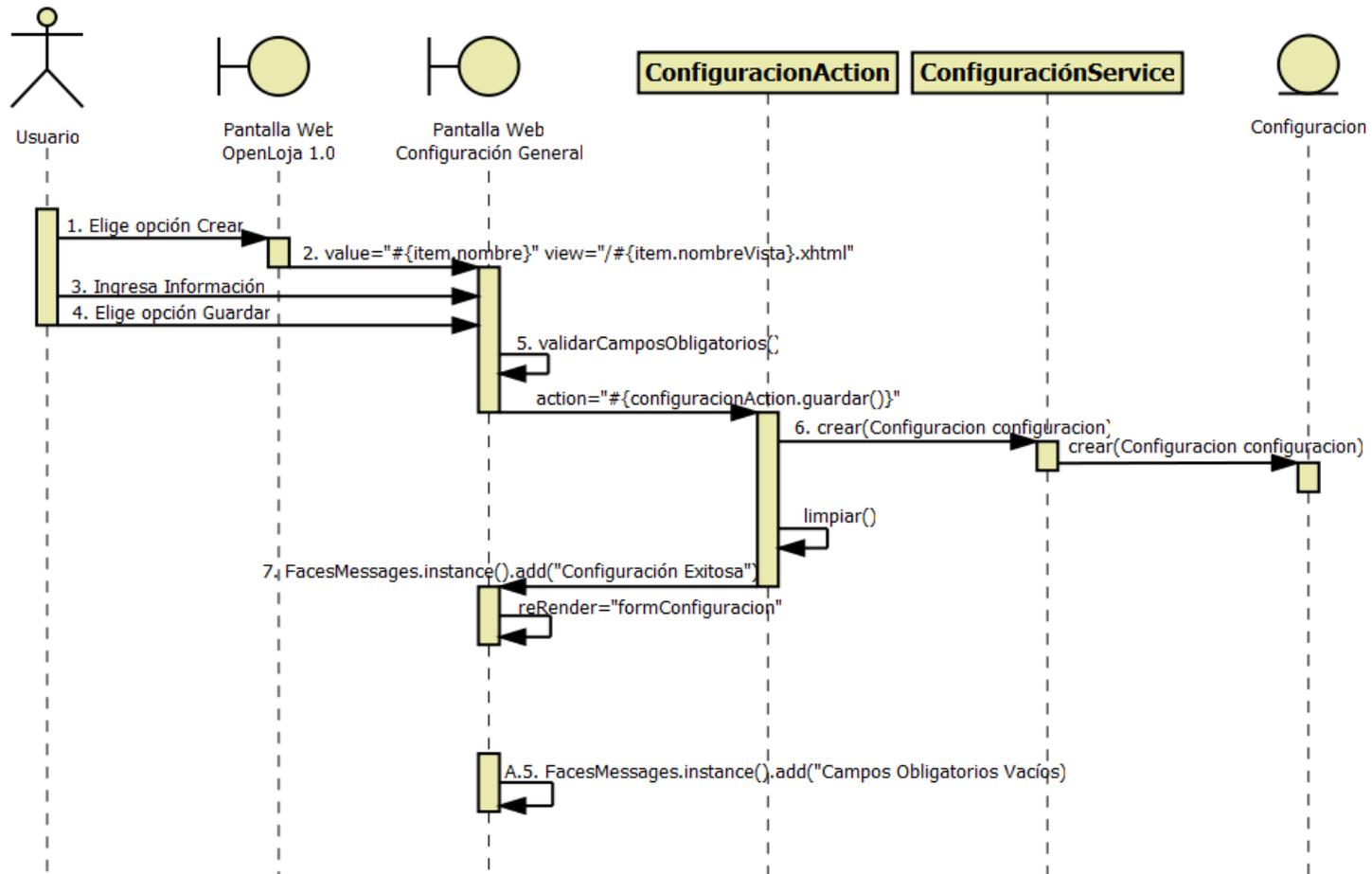


Diagrama 5. Secuencia Configurar Información General Versión Final

3.1.3.4. DIAGRAMA DE SECUENCIA 004: Crear Usuario (DS004)

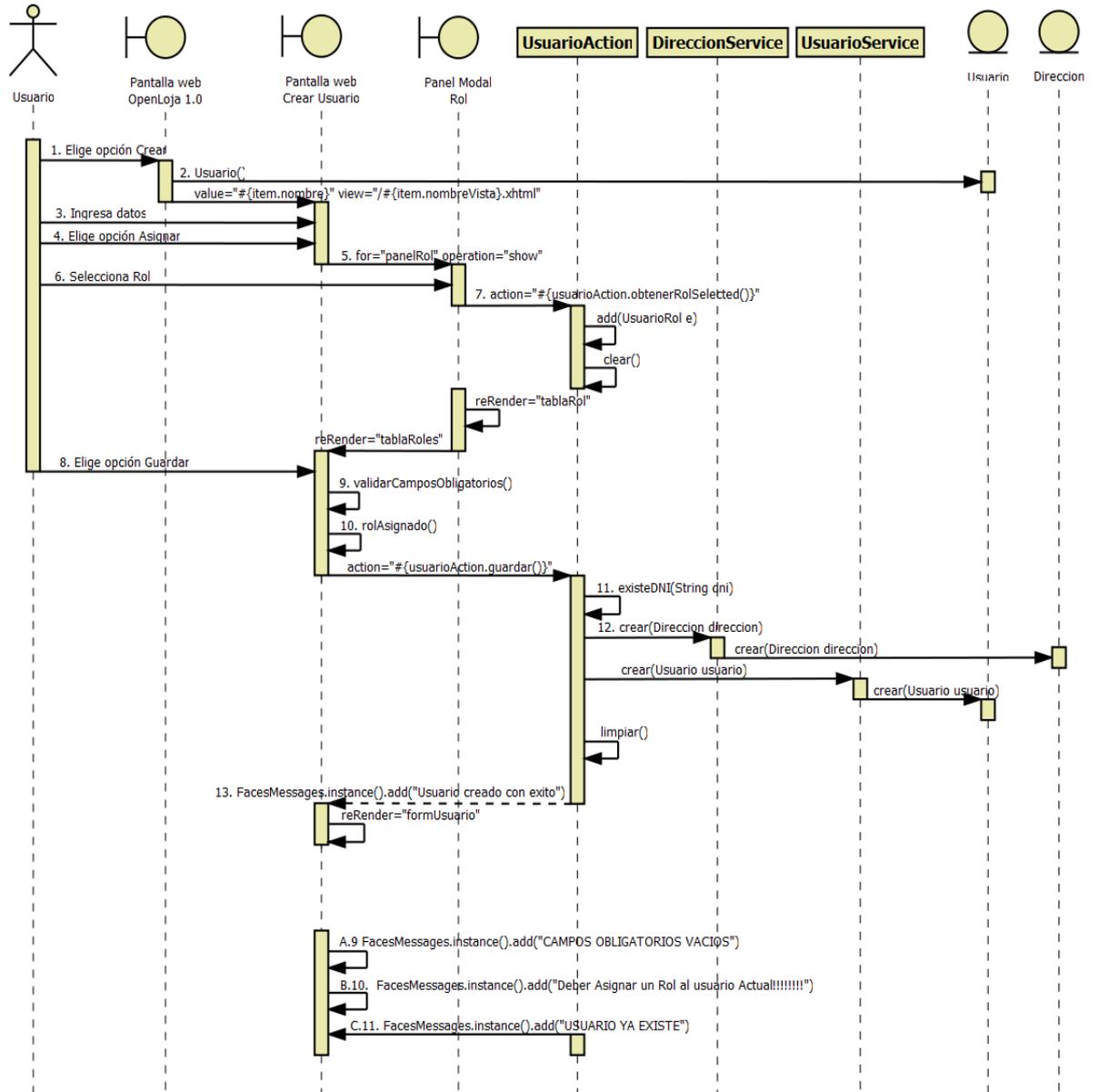


Diagrama 6. Secuencia Crear Usuario Versión Final

3.1.3.5. DIAGRAMA DE SECUENCIA 005: Modificar Cuenta Usuario (DS005)

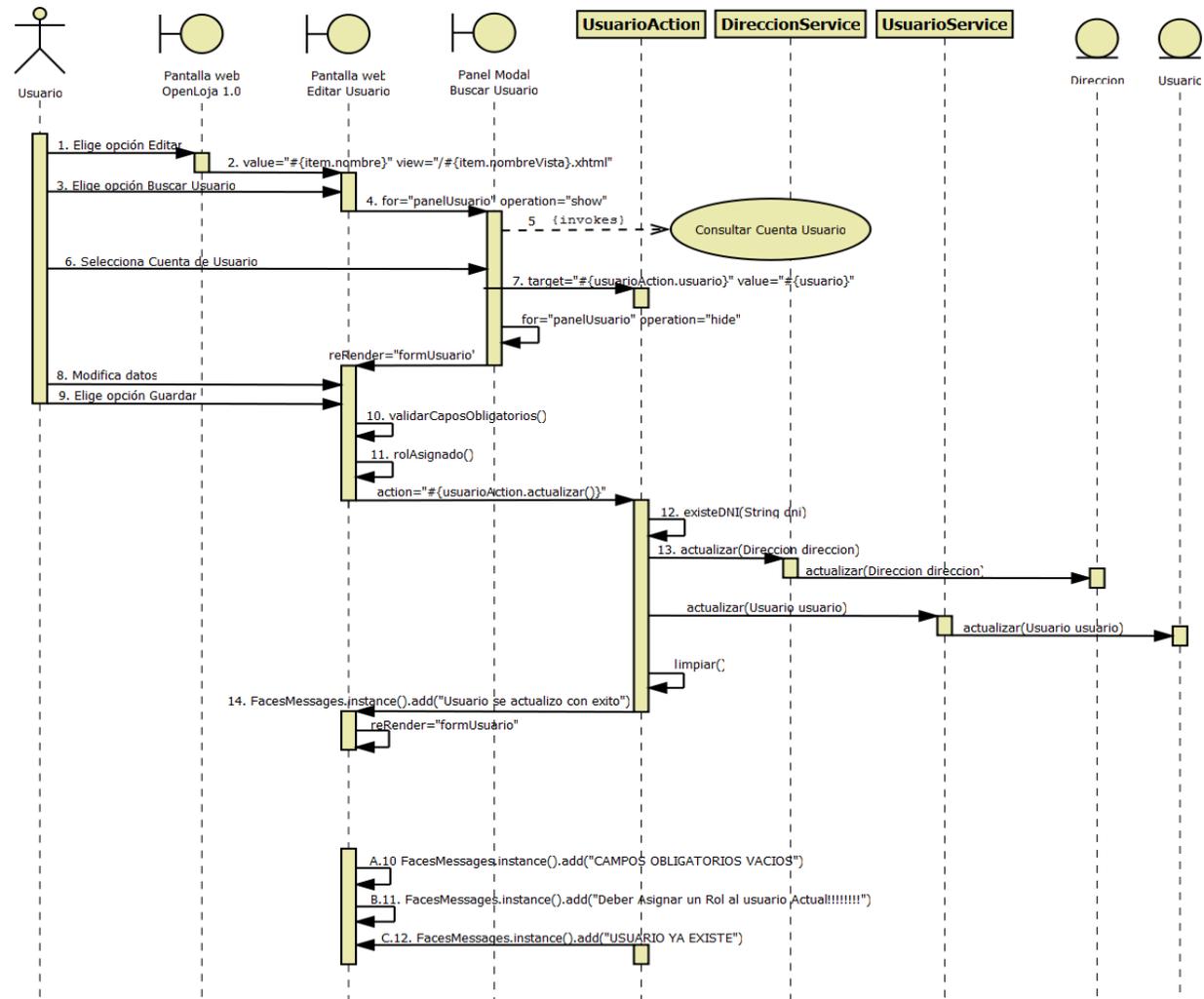


Diagrama 7. Modificar Cuenta Usuario Versión Final

3.1.3.6. DIAGRAMA DE SECUENCIA 006: Consultar Cuenta Usuario (DS006)

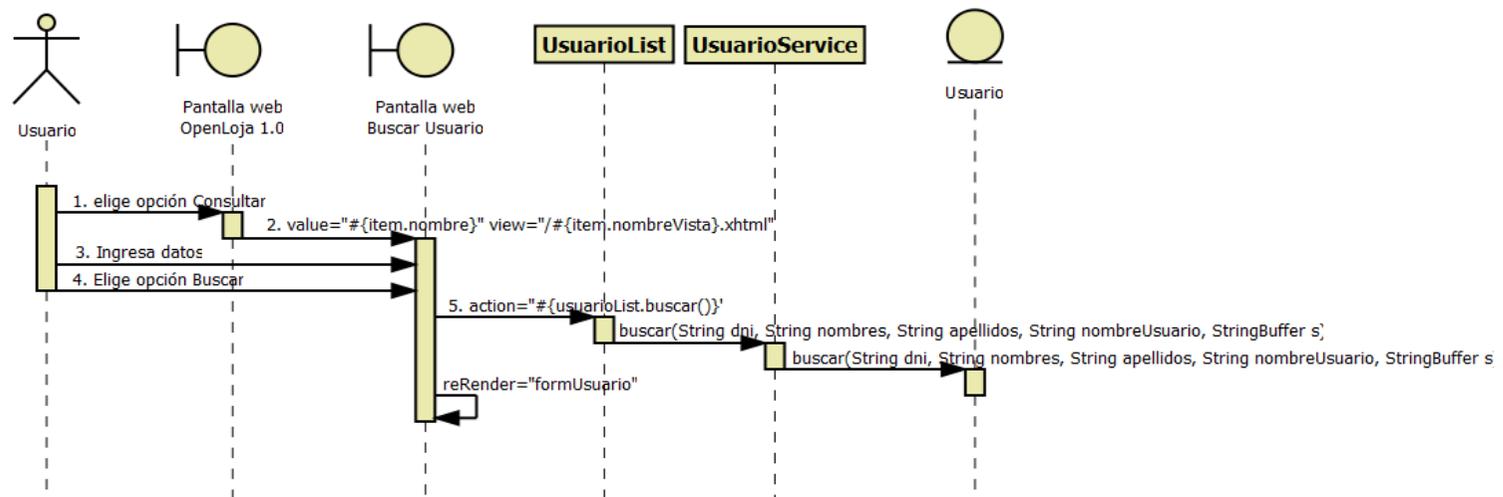


Diagrama 8. Secuencia Consultar Cuenta Usuario Versión Final

3.1.4. DIAGRAMA DE CLASES FINAL

3.1.4.1. CLASES MODELO

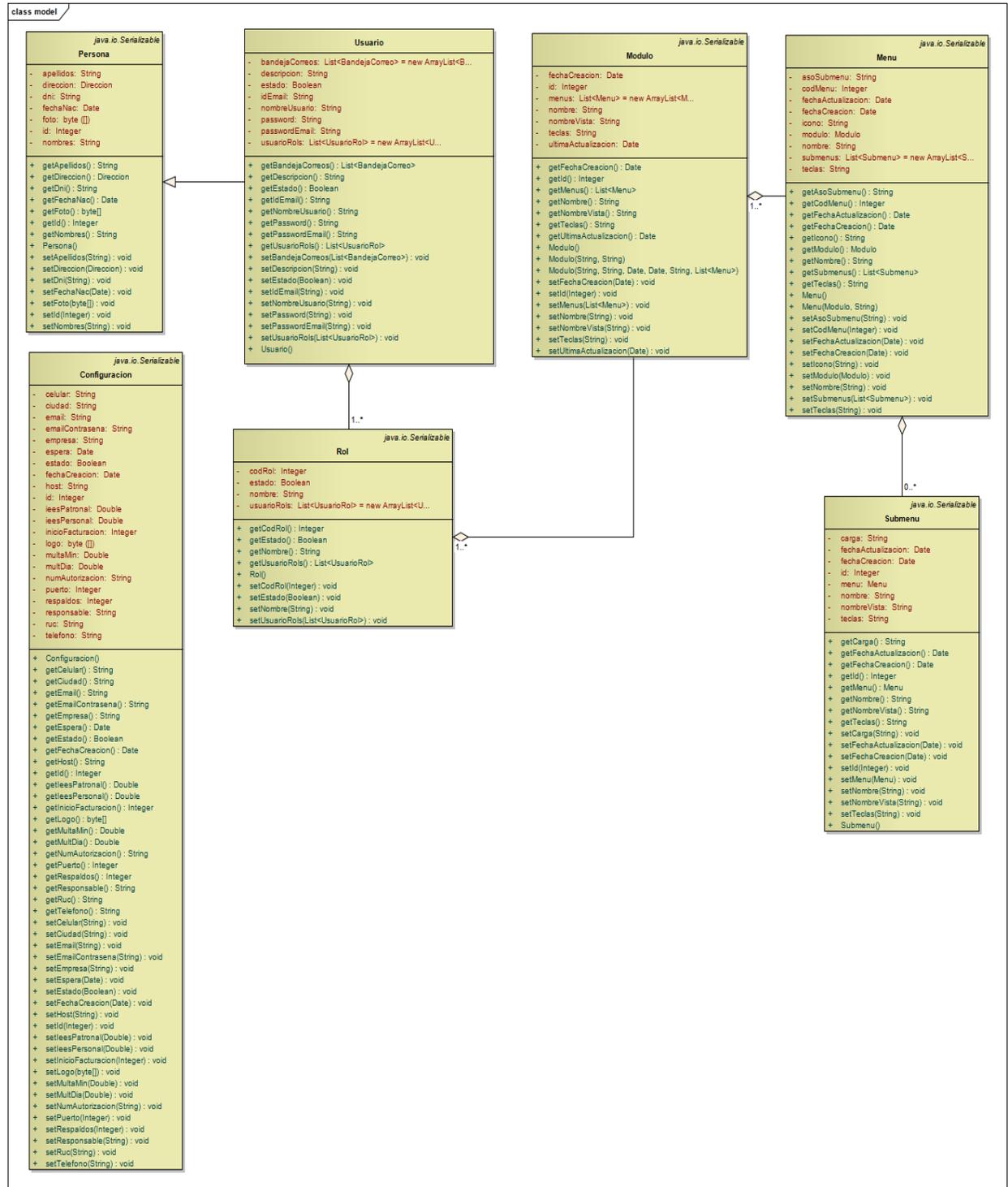


Diagrama 9. Clases Modelo: Gestión de Configuración Versión Final

3.1.4.2. CLASES DAO

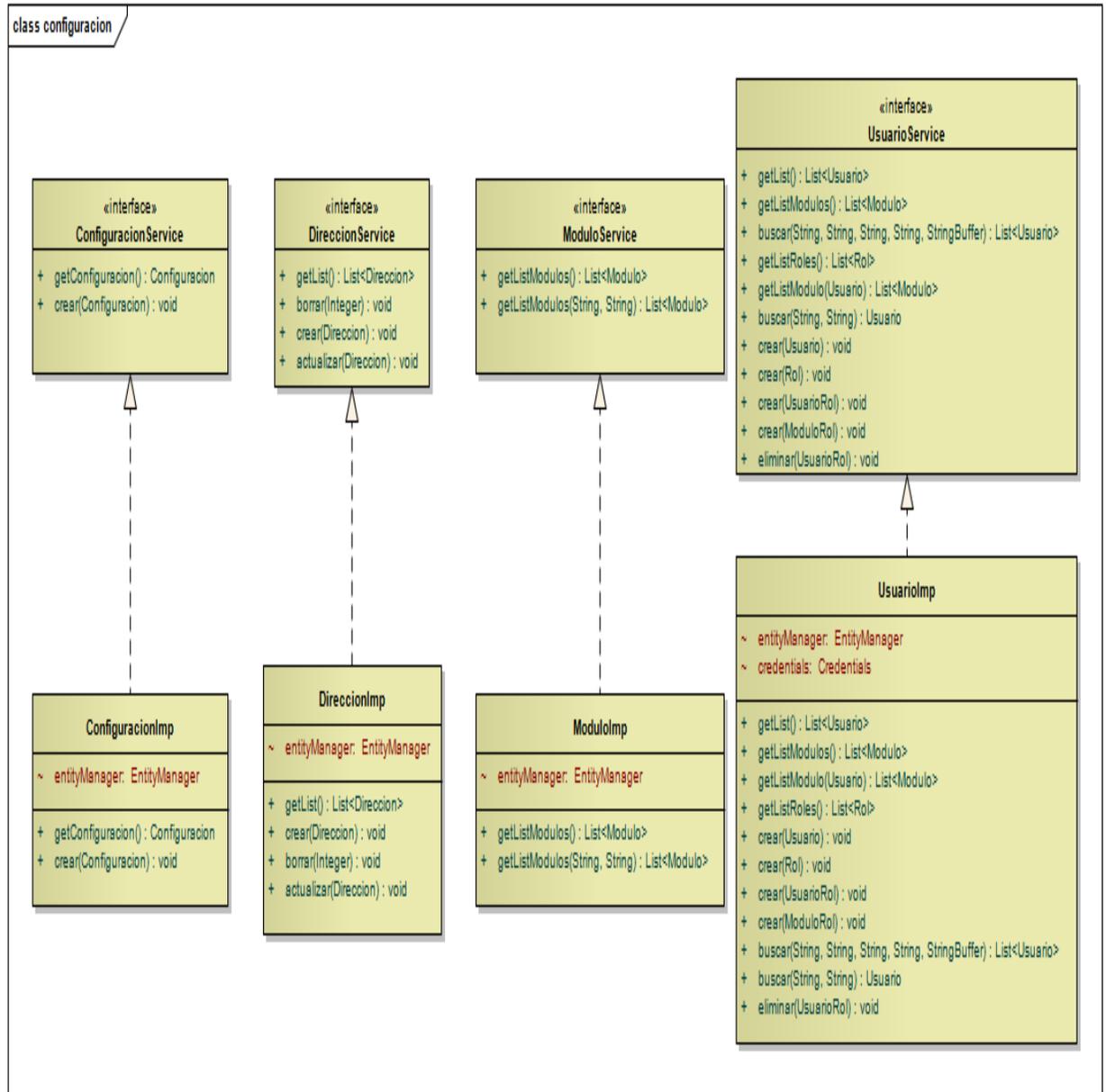


Diagrama 10. Clases DAO: Gestión de Configuración Versión Final

3.1.4.3. CLASES CONTROLADOR

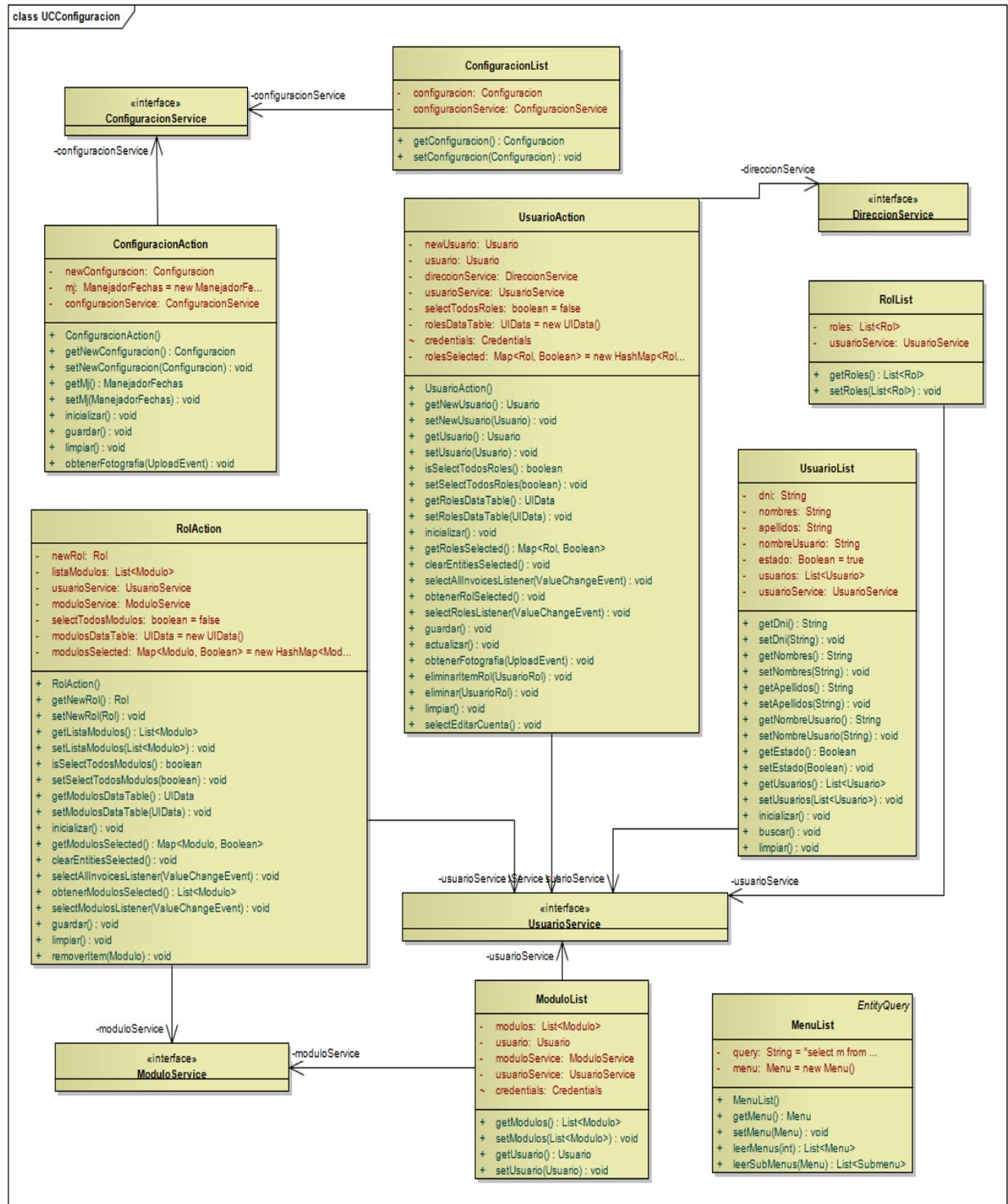


Diagrama 11. Clases Controlador: Gestión de Configuración Versión Final

3.1.4.4. CLASES VISTAS

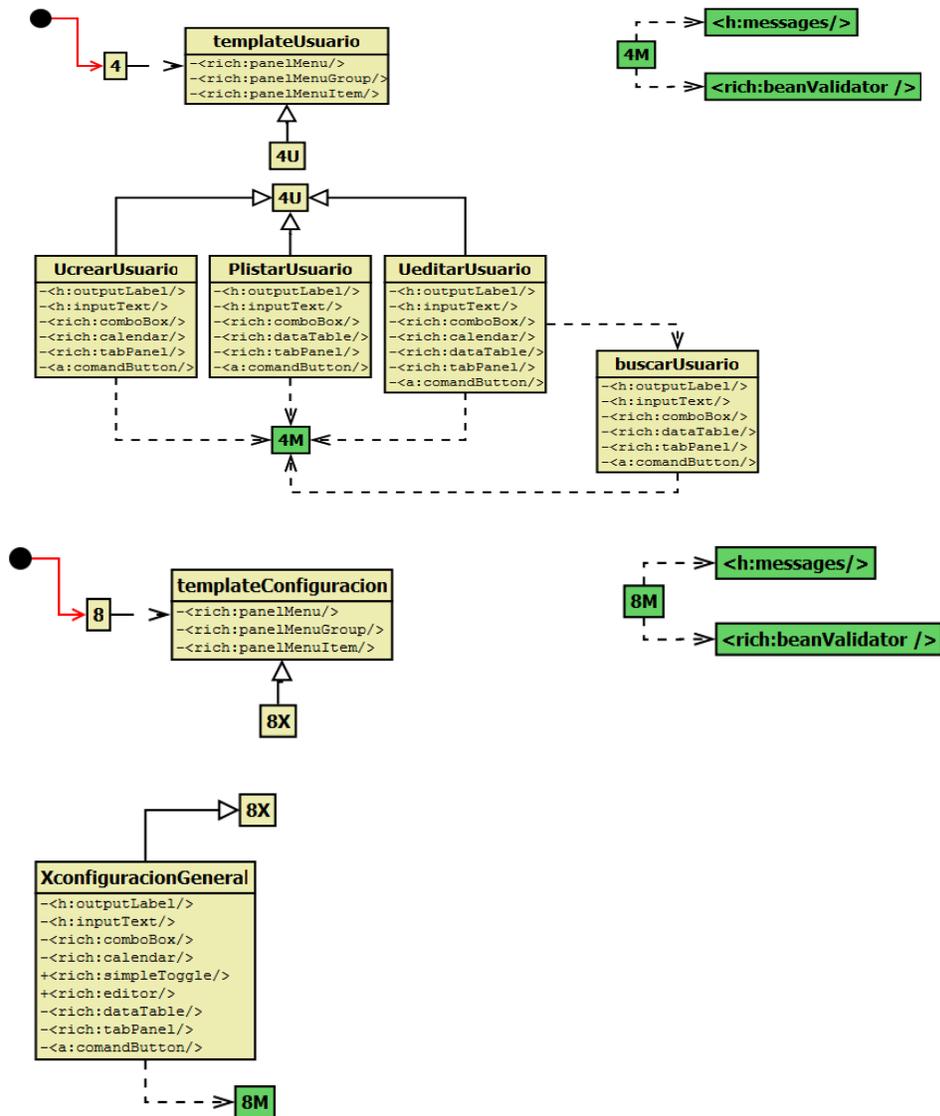


Diagrama 12. Vistas de Configuración Versión Final

3.1.5. DISEÑO DE DATOS

3.1.5.1. MODELO ENTIDAD-RELACIÓN

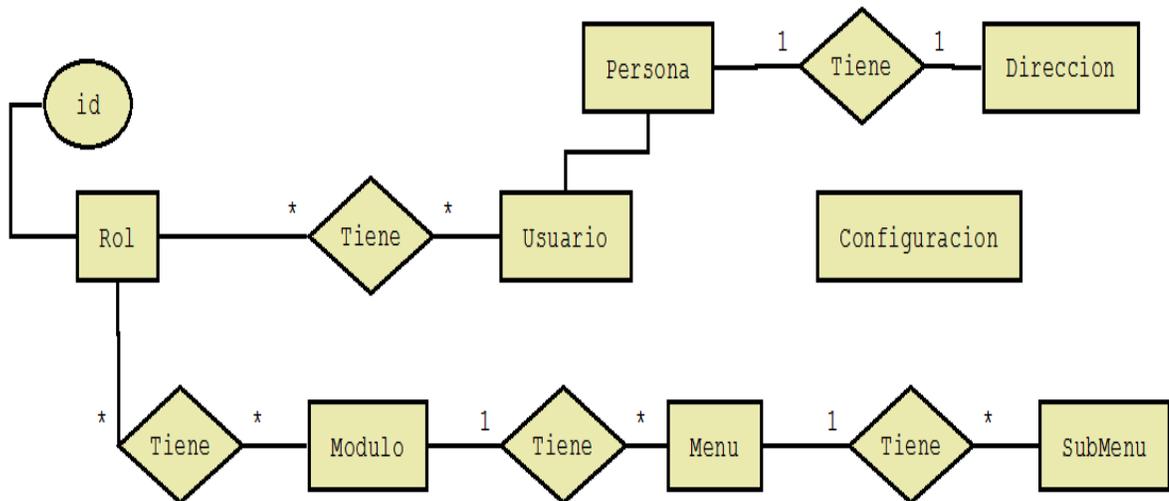


Diagrama 13. Modelo Entidad-Relación: Gestión de Configuración Versión Final

3.1.5.2. MODELO CONCEPTUAL

PERSONA

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no	🔑	
dni	Varchar	no		
foto	Blob	si		
nombres	Varchar	no		
apellidos	Varchar	no		
fechaNac	Date	si		
idDireccion	Integer	no		🔑

Tabla 19. Modelo Conceptual: Persona

DIRECCIÓN

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no	🔑	
ciudad	Varchar	no		
barrio	Varchar	si		
calles	Varchar	si		
teléfono	Integer	no		
celular	Integer	si		
email	Varchar	si		

numCasa	Integer	si		
---------	---------	----	--	--

Tabla 20. Modelo Conceptual: Dirección

USUARIO

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
nombreUsuario	Varchar	no		
password	Varchar	no		
estado	Boolean	no		
descripción	Varchar	si		
idEmail	Varchar	si		
passwordEmail	Varchar	si		

Tabla 21. Modelo Conceptual: Usuario

ROL

ATRIBUTO	TIPO	NULO	PK	FK
codRol	Integer	no		
nombre	Varchar	no		
estado	Boolean	no		
descripción	Varchar	si		

Tabla 22. Modelo Conceptual: Rol

USUARIO_ROL

ATRIBUTO	TIPO	NULO	PK	FK
cod	Integer	no		
codUsuario	Integer	no		
codRol	Integer	no		

Tabla 23. Modelo Conceptual: Usuario_Rol

MÓDULO

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
nombre	Varchar	no		
teclas	Varchar	no		
fechaCreacion	Date	si		
ultimaActualizacion	Date	no		
nombreVista	Varchar	si		
codUsuario	Integer	no		
archivo	Varchar	si		
estado	Boolean	no		

Tabla 24. Modelo Conceptual: Módulo

MENÚ

ATRIBUTO	TIPO	NULO	PK	FK
codMenu	Integer	no		
nombre	Varchar	no		
teclas	Varchar	no		
fechaCreacion	Date	si		
ultimaActualizacion	Date	no		
asoSubMenu	Varchar	no		
asoModulo	Integer	no		

Tabla 25. Modelo Conceptual: Menú

SUBMENÚ

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
nombre	Varchar	no		
teclas	Varchar	no		
fechaCreacion	Date	si		
ultimaActualizacion	Date	no		
nombreVista	Varchar	no		
codMenu	Integer	no		
carga	Varchar	si		

Tabla 26. Modelo Conceptual: Submenú

MÓDULO_ROL

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
idRol	Integer	no		
idModulo	Integer	no		

Tabla 27. Modelo Conceptual: Módulo_Rol

CONFIGURACIÓN

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
empresa	Varchar	no		
responsable	Varchar	no		
ruc	Varchar	no		
estado	Boolean	no		
fechaCreacion	Date	no		
email	Varchar	no		
ciudad	Varchar	si		
telefono	Integer	si		
celular	Integer	si		

emailContrasena	Varchar	no		
host	Varchar	no		
puerto	Integer	no		
numfac1	Integer	no		
numfac2	Integer	no		
numfac3	Integer	no		
numAutorizacion	Varchar	no		

Tabla 28. Modelo Conceptual: Configuración

3.1.5.3. MODELO RELACIONAL

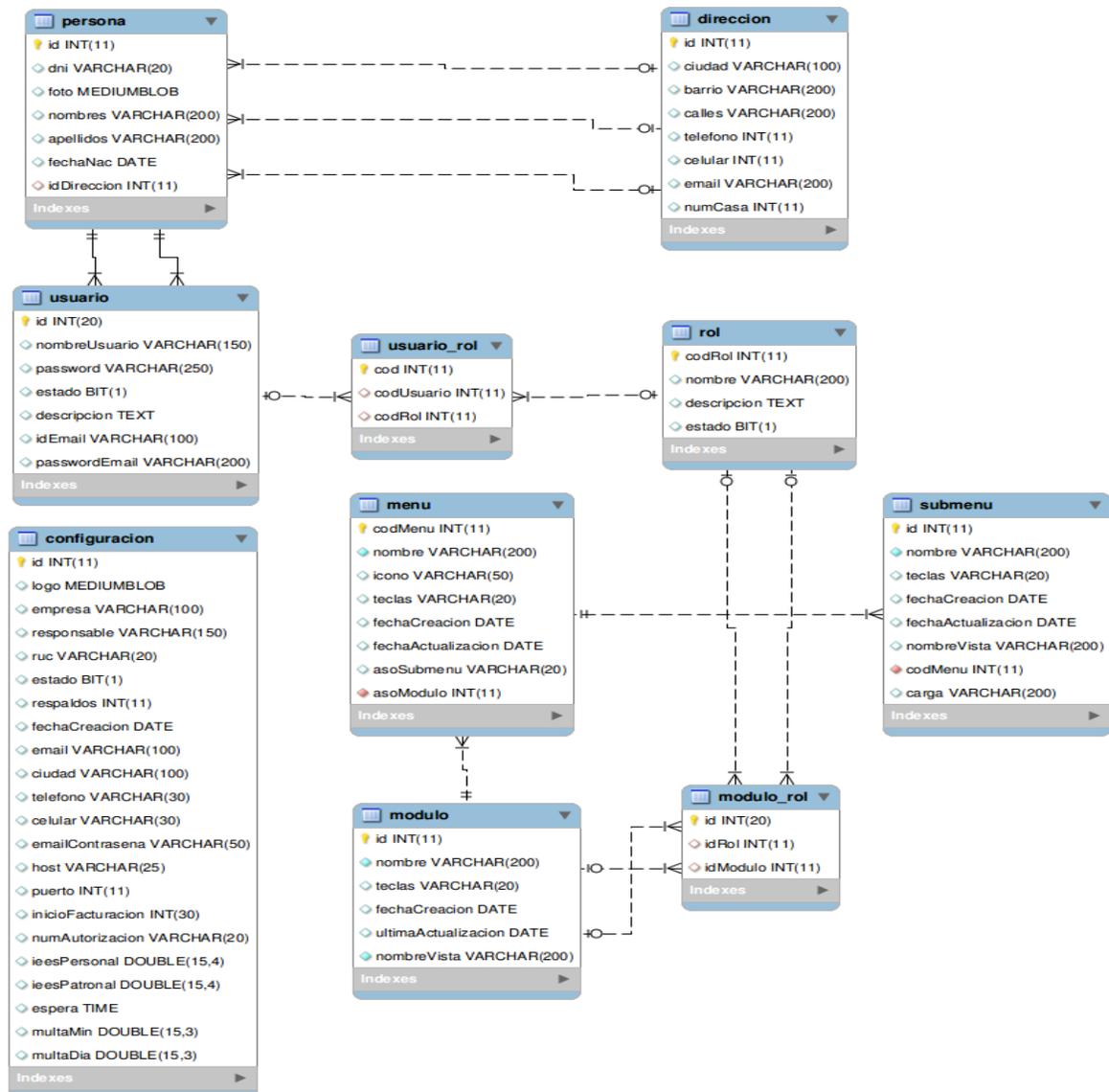


Diagrama 14. Modelo Relacional: Gestión de Configuración Versión Final



GESTIÓN DE EMPLEADOS

3.2. MÓDULO DE GESTIÓN DE EMPLEADOS

3.2.1. MODELO DE DOMINIO

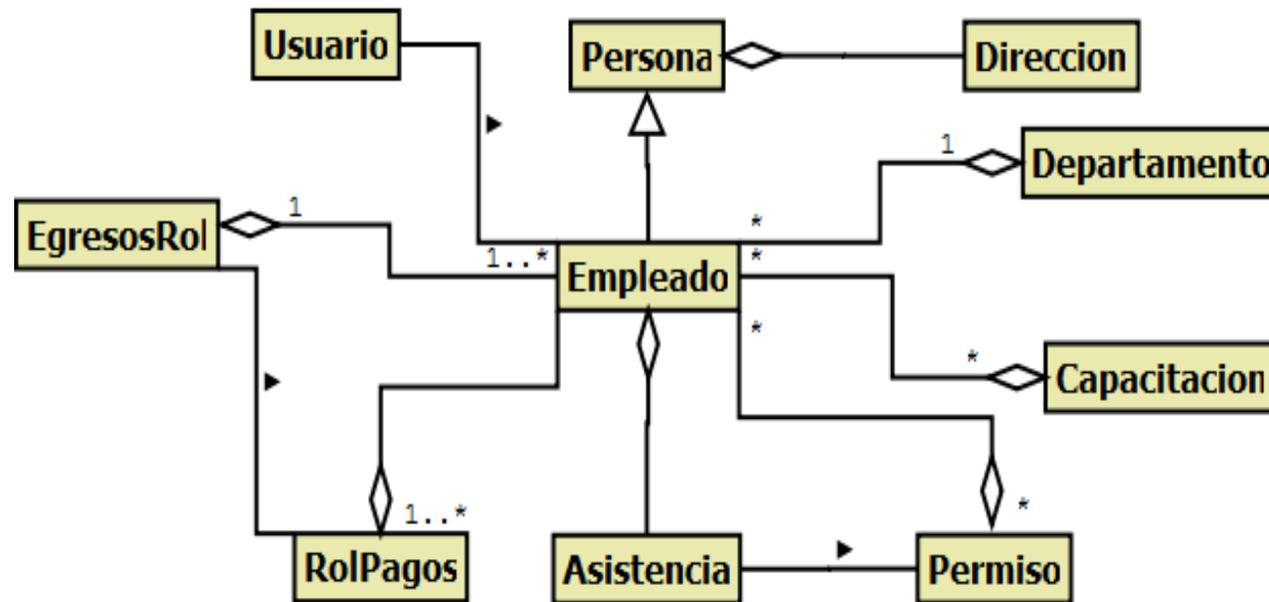


Diagrama 15. Módulo Empleados Versión Final

3.2.2. MODELO DE CASOS DE USO DEL SISTEMA.

3.2.2.1. DIAGRAMA DE CASOS DE USO

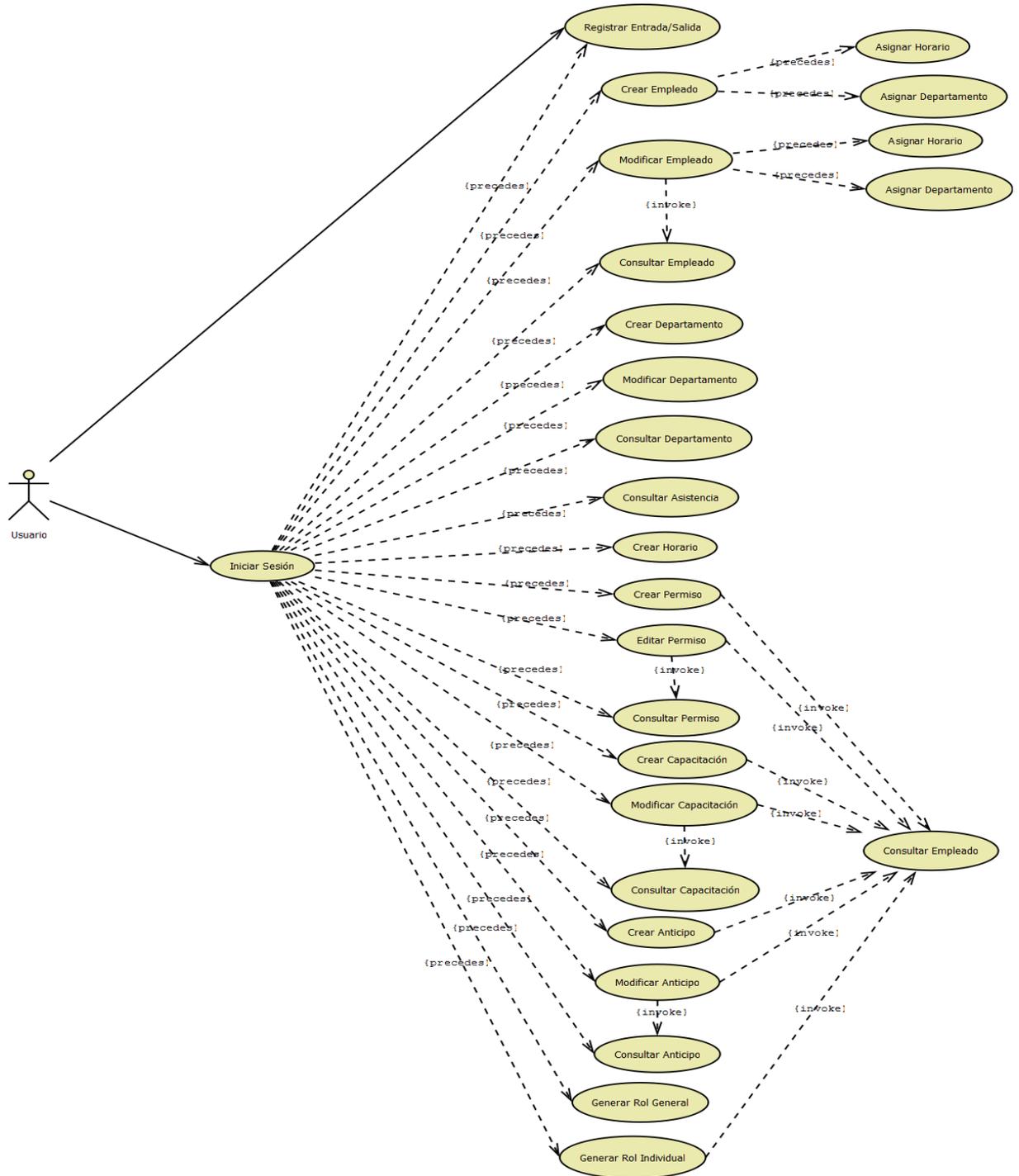


Diagrama 16. Gestión Empleados Versión Final

3.2.2.2. DESCRIPCIÓN DE CASOS DE USO

➤ CASO DE USO: Crear Empleado

NOMBRE: Crear Empleado		CÓDIGO	OP006
NOMBRE PROGRAMADO	EcrearEmpleado.xhtml		

NUEVO
GUARDAR
BUSCAR
BORRAR
X CANCELAR

DNI *

Nombres *

Apellidos *

Fecha Nacimiento

Cargo

Código Asistencia

Clave

Sueldo Básico

Fecha Ingreso

Contrato Si No



Datos Domiciliario | **Horario** | **Departamento**

Ciudad

Barrio

Calles

Número Casa

Telefono

Celular

Email

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com

Tabla 29. Pantalla Crear Empleado

Nombre:	Crear Empleado	Código: 007
Requerimiento Funcional:	RF007	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con privilegios especiales pueda crear empleados.	
Descripción:	Qué el usuario previamente autenticado con privilegios pertinentes pueda crear empleados.	
Precondiciones:	El usuario debe estar logueado. El usuario debe seguir la siguiente ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Empleado.	
Poscondiciones:	Empleado creado	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Crear] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea el Empleado y muestra la pantalla web "Crear Empleado". 3. El usuario ingresa los datos del Empleado que va a guardar en el sistema. 4. El usuario asigna horario(s) al Empleado mediante el caso de uso <u>Asignar Horario</u>. 		

5. El usuario asigna un Departamento al Empleado a través del caso de uso Asignar departamento.
6. El usuario selecciona el botón [Guardar].
7. El sistema valida que los campos obligatorios no estén vacíos.
8. El sistema verifica que se haya asignado por lo menos un Departamento al Empleado.
9. El sistema verifica que el Empleado tenga por lo menos un Horario asignado.
10. El sistema verifica mediante el DNI que el Empleado no coincida con otro ya existente.
11. El sistema guarda el nuevo Empleado.
12. El sistema muestra un mensaje de Empleado se ha creado con éxito.
13. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACIOS

A.7. El sistema muestra mensajes en los campos obligatorios vacíos.

A.8. El caso de uso continúa en el paso 3 del flujo normal de evento.

B. DEPARTAMENTO NO ASIGNADO

B.8. El sistema muestra un mensaje de que no se asignado ningún departamento.

B.9. El caso de uso continúa en el paso 4 del flujo normal de evento.

C. HORARIO NO ASIGNADO

C.9. El sistema muestra un mensaje no se asignado ningún horario de trabajo.

C.10. El caso de uso continúa en el paso 5 del flujo normal de evento.

D. EMPLEADO EXISTENTE

D.10. El sistema muestra un mensaje de Empleado ya existe.

D.11. El caso de uso continúa en el paso 3 del flujo normal de evento.

Tabla 30. Descripción Caso de Uso Crear Empleado

➤ CASO DE USO: Modificar Empleado

NOMBRE: Editar Empleado	CÓDIGO	OP007
NOMBRE PROGRAMADO	EeditaEmpleado.xhtml	

CREAR
GUARDAR
BUSCAR
BORRAR
X CANCELAR

DNI *

Nombres *

Apellidos *

Fecha Nacimiento

Código Asistencia

Clave

Sueldo Básico

Fecha Ingreso

Contrato



Datos Domiciliario
Horario
Departamento

Ciudad

Barrio

Número Casa

Telefono*:

Celular

Email

Tabla 31. Pantalla Editar Empleado

NOMBRE: Panel Modal Buscar Empleado	CÓDIGO	PM03
NOMBRE PROGRAMADO	EmodalBuscarEmpleado.xhtml	

DNI *

Nombres *

Apellidos *

Estado

BUSCAR
RESET

Foto	DNI * *	Nombres * Apellidos * *	Email	Telefono	Celular	Acciones
No data						

Tabla 32. Panel Modal Buscar Empleado

Nombre:	Modificar Empleado	Código: 008
Requerimiento Funcional:	RF007	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con el rol asignado poder modificar un Empleado.	
Descripción:	Al usuario que se le asignado realizar modificaciones de empleados creados con anterioridad.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Empleado.	
Poscondiciones:	Datos de Empleado seleccionado modificados.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Editar Empleado". 3. El usuario selecciona el botón [Buscar Empleado]. 4. El sistema muestra el panel modal "Buscar Empleado". 5. El usuario busca el Empleado que va a modificar para lo cual utiliza (invoca) el caso de uso <u>Consultar Empleado</u>. 6. El usuario selecciona el Empleado de la pantalla web "Buscar Empleado". 7. El sistema carga los datos del empleado seleccionado en la pantalla web "Editar Empleado" y automáticamente se cierra el panel modal "Buscar Empleado". 8. El usuario modifica los datos que desea cambiar del Empleado. 9. El usuario selecciona la opción [Guardar] de la pantalla web "Editar Empleado". 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica si hay asignado por lo menos un Departamento al Empleado. 12. El sistema verifica si hay asignado por lo menos un Horario al Empleado. 13. El sistema verifica mediante el DNI que el Empleado no coincida con otro ya existente. 14. El sistema guarda los nuevos datos del Empleado editado. 15. El sistema muestra un mensaje de Empleado se ha modificado con éxito. 16. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.10. El sistema muestra mensajes en los campos obligatorios vacíos.</p> <p>A.11. El caso de uso continúa en el paso 5 del flujo normal de eventos.</p> <p>B. DEPARTAMENTO NO ASIGNADO</p> <p>B.11. El sistema muestra un mensaje de que no se asignado ningún departamento.</p> <p>B.12. El caso de uso continúa en el paso 5 del flujo normal de evento.</p> <p>C. HORARIO NO ASIGNADO</p> <p>C.12. El sistema muestra un mensaje no se asignado ningún horario de trabajo.</p> <p>C.13. El caso de uso continúa en el paso 5 del flujo normal de evento.</p> <p>D. EMPLEADO EXISTENTE</p> <p>D.13. El sistema muestra un mensaje de Empleado ya existe.</p> <p>D.14. El caso de uso continúa en el paso 5 del flujo normal de evento.</p>		

Tabla 33. Descripción Caso de Uso Modificar Empleado

➤ CASO DE USO: Consultar Empleado

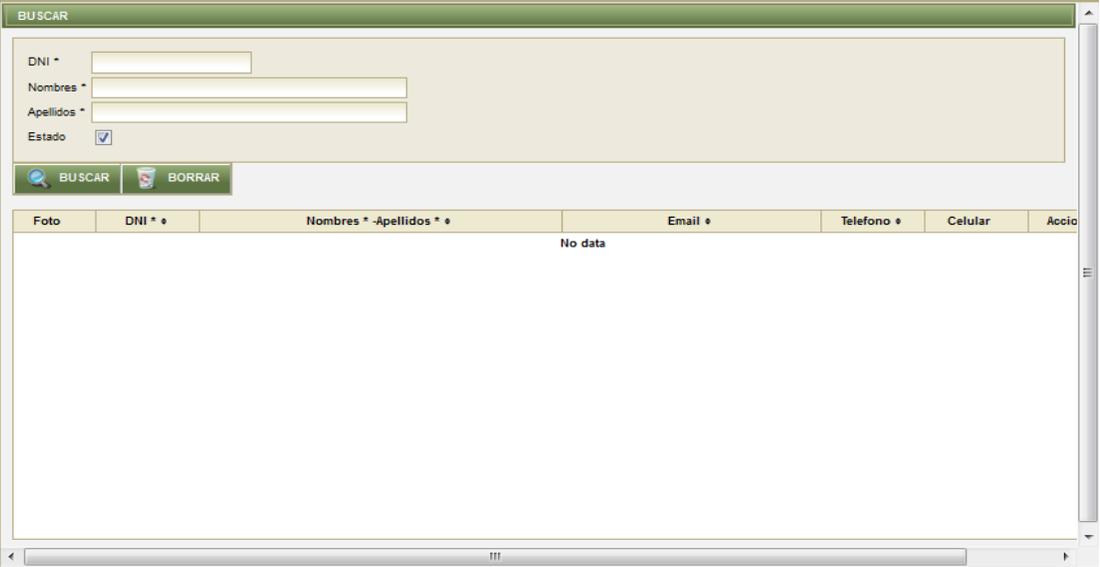
NOMBRE: Buscar Empleado		CÓDIGO	OP008
NOMBRE PROGRAMADO	ElistaEmpleado.xhtml		
			

Tabla 34. Pantalla Buscar Empleado

Nombre:	Consultar Empleado	Código: 009
Requerimiento Funcional:	RF008	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario asignado poder consultar un Empleado previamente creado.	
Descripción:	Al usuario realizar consultas de Empleados creados con anterioridad cuando lo requiera.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Empleado.	
Poscondiciones:	Empleado visualizado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Consultar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Buscar Empleado". 3. El usuario ingresa los datos del empleado que busca. 4. El usuario elige la opción [Buscar] de la pantalla web "Buscar Empleado". 5. El sistema muestra los empleados que coinciden con la búsqueda en la pantalla web "Buscar Empleado". 6. El caso de uso finaliza. 		

Tabla 35. Descripción de Caso de Uso Consultar Empleado

➤ CASO DE USO: Crear Departamento

NOMBRE: Crear Departamento		CÓDIGO	OP009
NOMBRE PROGRAMADO	EcrearDepartamento.xhtml		

GUARDAR BORRAR CANCELAR

Referencia:

Color:

Nombre*:

Color	Referencia	Nombre* *	Acciones
001		DEPARTAMENTO TECNICO	✎
002		CONTADURÍA	✎
003		GERENCIA	✎
004		BODEGA	✎
005		VENTAS	✎
005		COMPRAS	✎

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com

Tabla 36. Pantalla Crear Departamento

Nombre:	Crear Departamento	Código: 010
Requerimiento Funcional:	RF010	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario asignado poder crear departamentos.	
Descripción:	Que el usuario con este Rol asignado pueda crear departamentos que requiera.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Departamento.	
Poscondiciones:	Departamento creado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Crear] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea el Departamento y muestra la pantalla web "Crear Departamento". 3. El usuario ingresa los datos correspondientes al Departamento. 4. El usuario selecciona el botón [Guardar] de la pantalla web "Crear Departamento". 5. El sistema valida que los campos obligatorios no estén vacíos 		

6. El sistema verifica por medio de la Referencia que el Departamento no coincida con otro ya existente.
7. El sistema guarda el nuevo Departamento.
8. El sistema muestra un mensaje de Departamento creado con éxito.
9. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACIOS

- A.5. El sistema muestra un mensaje de campos obligatorios vacíos.
- A.6. El caso de uso continúa en el paso 3 del flujo normal de evento.

B. DEPARTAMENTO EXISTENTE

- B.6. El sistema muestra mensaje de Departamento ya existente.
- B.7. El caso de uso continúa en el paso 3 del flujo normal de evento.

Tabla 37.Descripción Caso de Uso Crear Departamento

➤ CASO DE USO: Asignar Departamento

NOMBRE: Crear Empleado o Editar Empleado	CÓDIGO	---
NOMBRE PROGRAMADO	EcrearEmpleado.xhtml o EeditaEmpleado.xhtml	

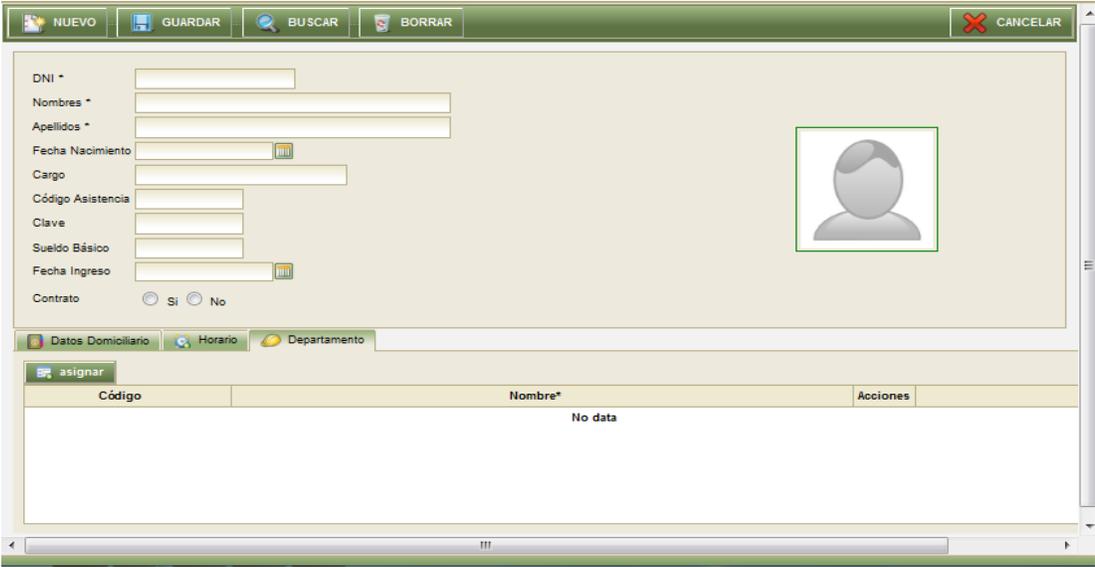


Tabla 38. Pantalla Crear Empleado

NOMBRE: Panel Modal Departamento	CÓDIGO	PM04
NOMBRE PROGRAMADO	EmodalDepartamento.xhtml	

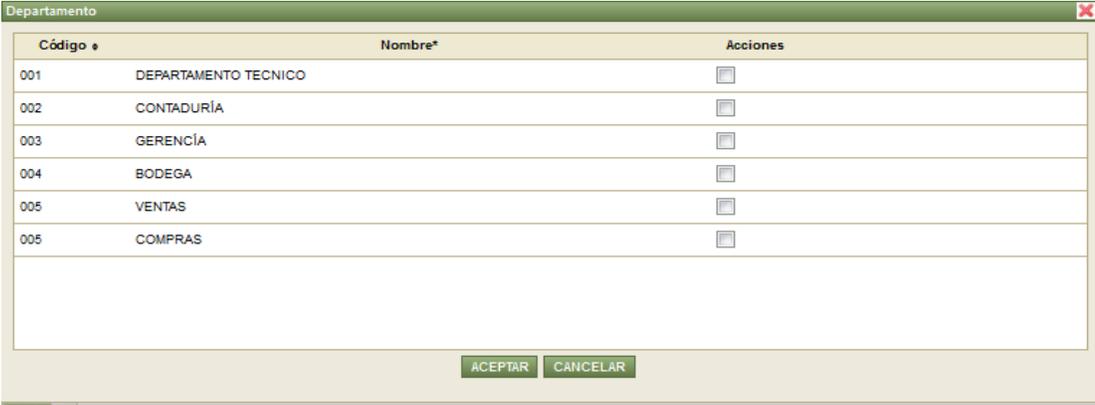


Tabla 39. Panel Modal Departamento

Nombre:	Asignar Departamento		Código: 011
Requerimiento Funcional:	RF014		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario con este rol poder asignar departamentos.		
Descripción:	Al usuario que se ha designado esta función asignar departamentos a los Empleados previa creación.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la pantalla web "Crear Empleado" ó "Editar Empleado".		
Poscondiciones:	Empleado con Departamento asignado.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario selecciona la pestaña (Tab Panel) <u>Departamento</u> de la pantalla web "Crear Empleado" ó "Modificar Empleado". 2. El usuario elige la opción [Asignar]. 3. El sistema muestra el panel modal "Departamentos". 4. El usuario selecciona el Departamento que va asignar al Empleado. 5. El usuario elige la opción [Aceptar] del panel modal "Departamentos". 6. El sistema carga los datos del Departamento seleccionado en la pantalla web "Crear Empleado" ó "Modificar empleado" según sea el caso. 7. El sistema cierra inmediatamente el panel modal "Departamentos". 8. El caso de uso finaliza. 			

Tabla 40. Descripción Caso de Uso Asignar Departamento

➤ **CASO DE USO: Modificar Departamento**

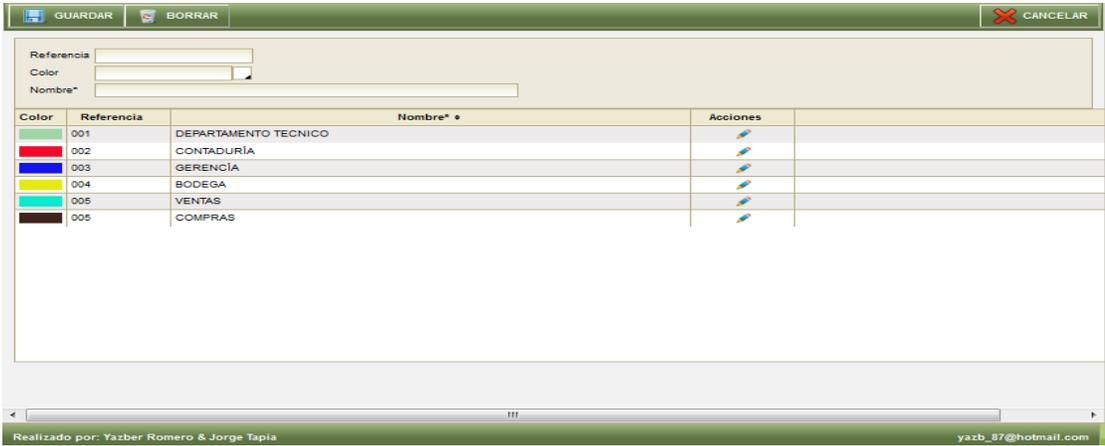
NOMBRE: Editar Departamento	CÓDIGO	OP010
NOMBRE PROGRAMADO	EeditarDepartamento.xhtml	
		

Tabla 41. Pantalla Editar Departamento

Nombre:	Modificar Departamento	Código: 012
Requerimiento Funcional:	RF010	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario asignado modificar un Departamento.	
Descripción:	Al usuario con este rol asignado realizar modificaciones de departamentos creados con anterioridad.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Departamento.	
Poscondiciones:	Departamento modificado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Editar Departamento". 3. El usuario selecciona el Departamento que va a modificar. 4. El sistema carga los datos del departamento seleccionado en los campos correspondientes de la pantalla web "Editar Departamento". 5. El usuario modifica los datos del departamento seleccionado. 6. El usuario elige la opción [Guardar] de la pantalla web "Editar Departamento". 7. El sistema valida que los campos obligatorios no estén vacíos. 8. El sistema verifica por medio de la Referencia que el Departamento no coincida con otro ya existente. 9. El sistema guarda los nuevos datos del Departamento modificado. 10. El sistema muestra un mensaje de Departamento se ha modificado con éxito. 11. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.6. El sistema muestra un mensaje de campos obligatorios vacíos.</p> <p>A.7. El caso de uso continúa en el paso 5 del flujo normal de evento.</p> <p>B. DEPARTAMENTO EXISTENTE</p> <p>B.6. El sistema muestra un mensaje de Departamento ya existe.</p> <p>B.7. El caso de uso continúa en el paso 5 del flujo normal de evento.</p>		

Tabla 42. Descripción Caso de Uso Modificar Departamento

➤ CASO DE USO: Crear Horario

NOMBRE: Crear Horario	CÓDIGO	OP012
NOMBRE PROGRAMADO	EcrearHorario.xhtml	

GUARDAR
BORRAR
CANCELAR

Nombre*

Hora Entrada

Hora Salida

Espera

Domingo Lunes Martes Miércoles Jueves Viernes Sábado

Nombre* *	hEntrada	hSalida	Espera	Dias	Acción
<input type="text"/>					
Matutino	9:0:0	13:0:0	0:10:0	Lunes, Martes, Miércoles, Jueves, Viernes,	
Vespertino	15:0:0	19:0:0	0:10:0	Lunes, Martes, Miércoles, Jueves, Viernes,	
Sabatino	10:0:0	14:0:0	0:10:0	Sábado	

Tabla 43. Pantalla Crear Horario

Nombre:	Crear Horario	Código: 014
Requerimiento Funcional:	RF009	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario poder crear horarios de trabajo.	
Descripción:	Permitir al usuario crear horarios de trabajo para posterior asignación a Empleados.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Horario.	
Poscondiciones:	Horario de Trabajo creado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Crear] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea el Horario y muestra la pantalla web "Crear Horario". 3. El usuario ingresa los datos correspondientes al Horario. 4. El usuario selecciona el botón [Guardar] de la pantalla web "Crear Horario". 5. El sistema valida que los campos obligatorios no estén vacíos 6. El sistema verifica por medio del nombre que el Horario no coincida con otro ya existente. 7. El sistema guarda el nuevo Horario. 8. El sistema muestra un mensaje de Horario creado con éxito. 		

9. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACIOS

- A.5. El sistema muestra un mensaje de campos obligatorios vacíos.
- A.6. El caso de uso continúa en el paso 3 del flujo normal de evento.

B. DEPARTAMENTO EXISTENTE

- B.6. El sistema muestra mensaje de Horario ya existente.
- B.7. El caso de uso continúa en el paso 3 del flujo normal de evento.

Tabla 44. Descripción Caso de Uso Crear Horario

➤ **CASO DE USO: Consultar Departamento**

NOMBRE: Buscar Departamento	CÓDIGO	OP011
NOMBRE PROGRAMADO	ElistarDepartamento.xhtml	

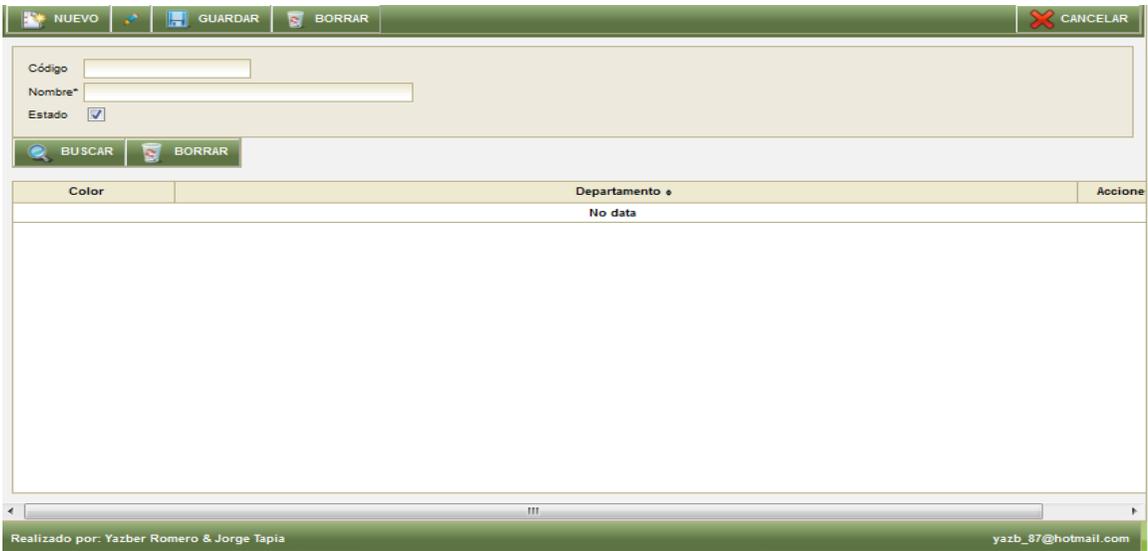


Tabla 45. Pantalla Buscar Departamento

Nombre:	Consultar Departamento		Código: 013
Requerimiento Funcional:	RF010		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario realizar consultas de Departamentos.		
Descripción:	Al usuario realizar consultas de los departamentos existentes en el sistema.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Departamento.		
Poscondiciones:	Departamento consultado y visualizado.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Consultar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Buscar Departamento". 3. El usuario ingresa los datos del Departamento que busca. 4. El usuario elige la opción [Buscar] de la pantalla web "Buscar Departamento". 5. El sistema muestra los Departamentos que coinciden con la búsqueda en la pantalla web "Buscar Departamento". 6. El caso de uso finaliza. 			

Tabla 46. Descripción Caso de Uso Consultar Departamento

➤ **CASO DE USO: Asignar Horario**

NOMBRE: Crear Empleado o Editar empleado	CÓDIGO	---
NOMBRE PROGRAMADO	EcrearEmpleado.xhtml o EeditaEmpleado.xhtml	

NUEVO
GUARDAR
BUSCAR
BORRAR
✖ CANCELAR

DNI *

Nombres *

Apellidos *

Fecha Nacimiento

Cargo

Código Asistencia

Clave

Sueldo Básico

Fecha Ingreso

Contrato Si No

Datos Domiciliario
Horario
Departamento

asignar

Nombre*	HORA ENTRADA	HORA SALIDA	Espera	Dias	Acciones
No data					

Tabla 47. Pantalla Crear Empleado

NOMBRE: Panel Modal Horarios		CÓDIGO	PM05
NOMBRE PROGRAMADO	EmodalHorario.xhtml		
			

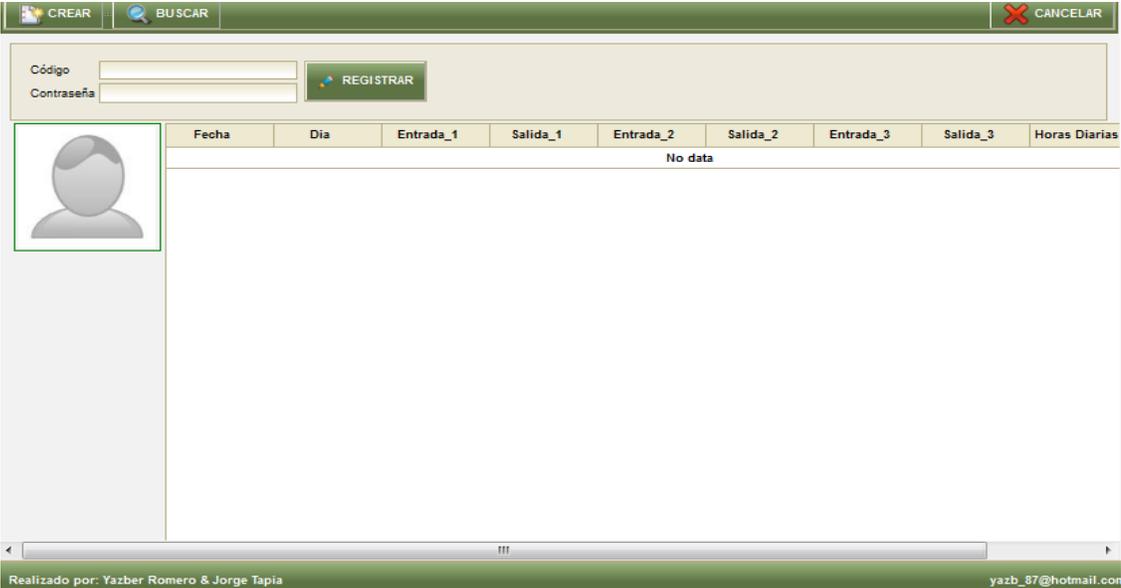
Tabla 48. Panel Modal Horarios

Nombre:	Asignar Horario	Código: 015
Requerimiento Funcional:	RF00	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol poder asignar horarios de trabajo.	
Descripción:	Al usuario que se ha designado esta función asignar horarios de trabajo a los Empleados previa creación.	
Precondiciones:	El usuario debe estar logueado. El usuario debe estar ubicado en la pantalla web "Crear Empleado" ó "Editar Empleado".	
Poscondiciones:	Empleado con Horario de trabajo asignado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario selecciona la pestaña (Tab Panel) <u>Horario</u> de la pantalla web "Crear Empleado" ó "Modificar Empleado". 2. El usuario elige la opción [Asignar]. 3. El sistema muestra el panel modal "Horarios". 4. El usuario selecciona los Horarios que va asignar al Empleado. 5. El usuario elige la opción [Aceptar] del panel modal "Horarios". 6. El sistema carga los datos del Departamento seleccionado en la pantalla web "Crear Empleado" ó "Modificar empleado" según sea el caso. 7. El sistema cierra inmediatamente el panel modal "Horarios". 8. El caso de uso finaliza. 		

Tabla 49. Descripción Caso de Uso Asignar Horario

➤ CASO DE USO: Registrar Entrada Salida

NOMBRE: Registro Asistencias		CÓDIGO	OP013
NOMBRE PROGRAMADO		EregistrarEntradaSalida.xhtml	



Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com



Tabla 50. Pantalla Registro Asistencias

Nombre:	Registrar Entrada Salida	Código: 016
Requerimiento Funcional:	RF011	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario poder registrar asistencias de entrada y salida.	
Descripción:	Al usuario crear registrar entradas y salidas de empleados en jornadas de trabajo.	
Precondiciones:	Si el usuario esta logueado se debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Control Asistencia. Si no está logueado se puede registrar directamente desde la pantalla web "Login"	
Poscondiciones:	Asistencia de Empleado registrada.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Registro Entrada/salida] de la pantalla web principal "OpenLoja 1.0" si esta logueado caso contrario va directamente al paso 3 de flujo normal de eventos. 2. El sistema muestra la pantalla web "Registro Asistencias". 3. El usuario (en este caso el empleado) ingresa sus datos para proceder a registrar su asistencia en el sistema. 4. El usuario selecciona la opción [Registrar]. 5. El sistema valida la información ingresada corresponda a un empleado existente. 6. El sistema verifica si el registro ingresado es de entrada o salida. 7. El sistema guara la información del registro al empleado correspondiente. 8. El sistema muestra el nuevo registro en la tabla de Asistencias. 9. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>A. INFORMACIÓN ERRÓNEA</p> <p>A.5. El sistema muestra mensaje la información no es correcta.</p> <p>A.6. El caso de uso continúa en el paso 3 del flujo normal de eventos.</p>		

Tabla 51. Descripción Caso de Uso Registrar Entrada Salida

➤ CASO DE USO: Consultar Asistencias

NOMBRE: Asistencias	CÓDIGO	OP014
NOMBRE PROGRAMADO	EconsultarAsistencias.xhtml	

IMPRIMIR EMAIL

Empleado:

DNI *

Cargo

Fecha Inicio: Jun 1, 2012

Fecha Fin: Jun 30, 2012



BU SCAR BORRAR

Fecha	Dia	Entrada_1	Salida_1	Entrada_2	Salida_2	Entrada_3	Salida_3	Horas Diarias
No data								

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com

Tabla 52. Pantalla Asistencias

NOMBRE: Panel Modal Buscar Empleado	CÓDIGO	PM06
NOMBRE PROGRAMADO	EmodalEmpleadoAsistencia.xhtml	

BU SCAR BORRAR

DNI *

Nombres *

Apellidos *

Estado



Foto	DNI *	Nombres * -Apellidos *	Email *	Telefono *	Celular	Acciones
No data						

Tabla 53. Panel Modal Buscar Empleado

Nombre:	Consultar Asistencias		Código: 017
Requerimiento Funcional:	RF012		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario con este rol poder consultar asistencias.		
Descripción:	Al usuario asignado realizar consultas de asistencias correspondientes a un empleado.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Control Asistencias.		
Poscondiciones:	Asistencias de Empleado consultadas y visualizadas.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Consultar Asistencia] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Asistencias". 3. El usuario selecciona la opción [Buscar Empleado]. 4. El sistema muestra el panel modal "Buscar Empleado". 5. El usuario busca el Empleado del cual va a consultar las asistencias para lo cual utiliza (invoca) el caso de uso <u>Consultar Empleado</u>. 6. El usuario selecciona el Empleado del panel modal "Buscar Empleado". 7. El sistema carga los datos del empleado seleccionado en la pantalla web "Asistencias" y automáticamente se cierra el panel modal "Buscar Empleado". 8. El usuario ingresa la fecha inicio y fecha fin para realizar la consulta. 9. El usuario selecciona la opción [Generar] 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica que las fechas ingresadas estén correctas. 12. El sistema muestra las asistencias del Empleado en la tabla de resultados de la pantalla web "Asistencias". 13. El caso de uso finaliza. 			
FLUJO ALTERNO:			
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.10. El sistema muestra mensaje en los campos obligatorios vacíos.</p> <p>A.11. El caso de uso continúa en el paso 3 del flujo normal de evento.</p> <p>B. FECHAS INCORRECTAS</p> <p>B.11. El sistema muestra mensaje fecha de consulta no son correctas.</p> <p>B.12. El caso de uso continúa en el paso 8 del flujo normal de evento.</p>			

Tabla 54. Descripción Caso de Uso Consultar Empleado

➤ CASO DE USO: Crear Permiso

NOMBRE: Crear Permiso		CÓDIGO	OP015
NOMBRE PROGRAMADO		EcrearPermiso.xhtml	

GUARDAR
BORRAR
CANCELAR

Fecha Creación: Jul 4, 2012

Monto: 40.0

Fecha Aplicación: Jul 4, 2012

Descripción: adelanto del mes de sueldo

Empleados

asignar

DNI *	Nombres *- Apellidos *	Email	Telefono	Celular	Acciones
1102253067	Nei Alicia Gonzaga Marquez				X
1104499874	Vladimir Yazber Romero Gonzaga	yazb_87@hotmail.com		85681893	X

Tabla 55. Pantalla Crear Permiso

NOMBRE: Panel Modal Empleados		CÓDIGO	PM07
NOMBRE PROGRAMADO		EcrearPermiso.xhtml	

BUSCAR
BORRAR

DNI *

Nombres *

Apellidos *

Estado

Foto	DNI * *	Nombres * Apellidos * *	Email	Telefono	Celular	<input type="checkbox"/>
	1104499874	Vladimir Yazber Romero Gonzaga				<input checked="" type="checkbox"/>
	1102253067	Nei Alicia Gonzaga Marquez				<input checked="" type="checkbox"/>

ACEPTAR
CANCELAR

Tabla 56. Panel Modal Empleados

Nombre:	Crear Permiso	Código: 018
Requerimiento Funcional:	RF013	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario al que se asignado este rol poder crear permisos de trabajo.	
Descripción:	Al usuario al cual se le asignado esta función pueda crear permisos de trabajo de los empleados que los requieran.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Permiso.	
Poscondiciones:	Permiso de Trabajo creado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Crear] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea el Permiso y muestra la pantalla web "Crear Permiso". 3. El usuario ingresa los datos correspondientes al Permiso creado. 4. El usuario elige la opción [Empleados]. 5. El sistema muestra el panel modal "Empleado". 6. El usuario busca los Empleados a los que se va asignar el permiso utilizando (invocando) el caso de uso <u>Consultar Empleado</u>. 7. El usuario selecciona los Empleados del panel modal "Empleado" y elige la opción [Aceptar]. 8. El sistema carga los datos de los Empleados seleccionados en la pantalla web "Crear Permiso" y automáticamente se cierra el panel modal "Empleado". 9. El usuario elige la opción [Guardar] de la pantalla web "Crear Permiso". 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica que haya por lo menos un Empleado asignado al Permiso. 12. El sistema verifica que las fechas ingresadas estén correctas. 13. El sistema guarda los datos del nuevo Permiso. 14. El sistema muestra un mensaje de Permiso creado con éxito. 15. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.10. El sistema muestra un mensaje de campos obligatorios vacíos.</p> <p>A.11. El caso de uso continúa en el paso 3 del flujo normal de evento.</p> <p>B. EMPLEADO NO ASIGNADO</p> <p>B.11. El sistema muestra un mensaje de Empleados no asignados.</p> <p>B.12. El caso de uso continúa en el paso 4 del flujo normal de eventos.</p> <p>C. FECHAS INCORRECTAS</p> <p>C.12. El sistema muestra mensaje fechas incorrectas.</p> <p>C.13. El caso de uso continúa en el paso 3 del flujo normal de eventos.</p>		

Tabla 57. Descripción de Caso de Uso Crear Permiso

➤ CASO DE USO: Modificar Permiso

NOMBRE: Editar Permiso		CÓDIGO	OP016
NOMBRE PROGRAMADO		EeditarPermiso.xhtml	

GUARDAR BORRAR BUSCAR CANCELAR

Fecha Creación:

Monto:

Fecha Aplicacion:

Descripcion:

Empleados

asignar

DNI *	Nombres * Apellidos *	Email	Telefono	Celular	Acciones
No data					

Tabla 58. Pantalla editar Permiso

NOMBRE: Panel Modal Permisos		CÓDIGO	PM08
NOMBRE PROGRAMADO		EmodalPermiso.xhtml	

Empleado

DNI *

Nombres *

Apellidos *

Estado

BUSCAR BORRAR

Foto	DNI * *	Nombres * Apellidos * *	Email	Telefono	Celular	
	1104499874	Vladimir Yazber Romero Gonzaga				<input checked="" type="checkbox"/>
	1102253067	Neli Alicia Gonzaga Marquez				<input checked="" type="checkbox"/>

ACEPTAR CANCELAR

Tabla 59. Panel Modal Permisos

Nombre:	Modificar Permiso		Código: 019
Requerimiento Funcional:	RF010		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir que el usuario con este rol pueda editar permisos de trabajo.		
Descripción:	Al usuario que se le ha otorgado esta función pueda editar permisos de trabajo cuando lo requiera.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Permiso.		
Poscondiciones:	Permiso de Trabajo editado.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Editar Permiso". 3. El usuario elige la opción [Buscar Permiso]. 4. El sistema muestra el panel Modal "Permisos". 5. El usuario busca el Permiso que va a modificar para lo cual utiliza (invoca) el caso de uso <u>Consultar Permiso</u>. 6. El usuario selecciona el Permiso que busca del panel modal "Permisos". 7. El sistema carga los datos del Permiso seleccionado en la pantalla web "Editar Permiso" y automáticamente se cierra el panel modal "Permisos". 8. El usuario modifica los datos del Permiso seleccionado. 9. El usuario elige la opción [Guardar] de la pantalla web "Editar Permiso". 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica que haya por lo menos un Empleado asignado al Permiso. 12. El sistema verifica que las fechas ingresadas estén correctas. 13. El sistema guarda los nuevos datos del Permiso modificado. 14. El sistema muestra un mensaje de Permiso modificado con éxito. 15. El caso de uso finaliza. 			
FLUJO ALTERNO:			
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.10. El sistema muestra un mensaje de campos obligatorios vacíos.</p> <p>A.11. El caso de uso continúa en el paso 3 del flujo normal de evento.</p> <p>B. EMPLEADO NO ASIGNADO</p> <p>B.11. El sistema muestra un mensaje de Empleados no asignados.</p> <p>B.12. El caso de uso continúa en el paso 4 del flujo normal de eventos.</p> <p>C. FECHAS INCORRECTAS</p> <p>C.12. El sistema muestra mensaje fechas incorrectas.</p> <p>C.13. El caso de uso continúa en el paso 3 del flujo normal de eventos.</p>			

Tabla 60. Descripción de Caso de Uso Modificar Permiso

➤ CASO DE USO: Consultar Permiso

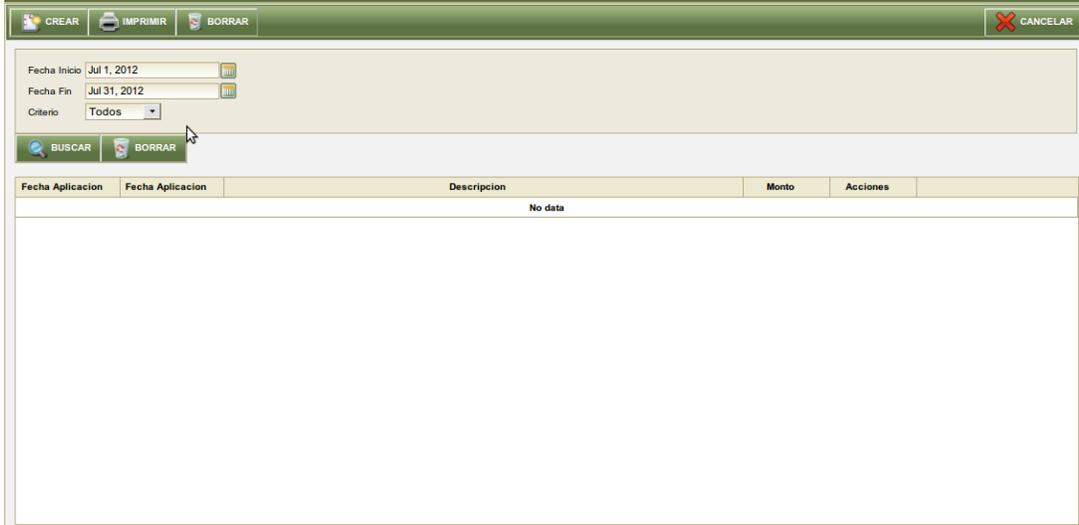
NOMBRE: Buscar Permiso	CÓDIGO	OP017
NOMBRE PROGRAMADO	EconsultaPermiso.xhtml	
		

Tabla 61. Pantalla Buscar Permiso

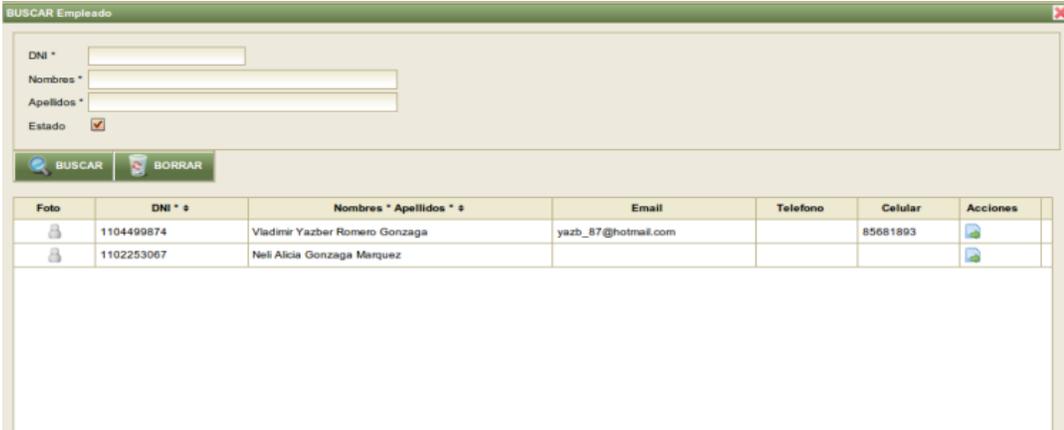
NOMBRE: Panel Modal Empleados	CÓDIGO	PM09
NOMBRE PROGRAMADO	EmodalEmpleadoPermiso.xhtml	
		

Tabla 62. Panel Modal Empleados

Nombre:	Consultar Permiso		Código: 020
Requerimiento Funcional:	RF010		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario con este rol asignado consultar Permisos.		
Descripción:	Que el usuario con esta función asignada realice consultas de Permisos de trabajo cuando lo necesite.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Permiso.		
Poscondiciones:	Lista de Permisos de Trabajo buscados.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Consultar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Buscar Permiso". 3. El usuario ingresa las fechas desde y hasta donde desea realizar la consulta. 4. El usuario elige el tipo de consulta que va a realizar: las capacitaciones de todos los empleados o las capacitaciones de un solo empleado, si elige la primera opción va directamente al paso 11. 5. Si el usuario elige la segunda opción el sistema actualiza la pantalla "Buscar Permiso" de acuerdo a este tipo de consulta. 6. El usuario selecciona la opción [Buscar Empleado]. 7. El sistema muestra el panel modal "Empleados". 8. El usuario busca el Empleado para consultar sus Permisos de trabajo utilizando (invocando) el caso de uso <u>Consultar Empleado</u>. 9. El usuario selecciona el Empleado que busca del panel modal "Empleados". 10. El sistema carga los datos del Empleado seleccionado en la pantalla web "Buscar Permiso" y automáticamente cierra el panel modal "Empleados". 11. El usuario elige la opción [Buscar] de la pantalla web "Buscar Permiso" 12. El sistema verifica que las fechas ingresadas estén correctas. 13. El sistema muestra los Permisos en la tabla de resultados de la pantalla web "Buscar Permiso". 14. El caso de uso finaliza. 			
FLUJO ALTERNO:			
A. FECHAS INCORRECTAS			
<ol style="list-style-type: none"> A.12. El sistema muestra un mensaje de fechas ingresadas incorrectas. A.13. El caso de uso continúa en el paso 3 del flujo normal de eventos. 			

Tabla 63. Descripción Caso de Uso Consultar Permiso

➤ CASO DE USO: Crear Capacitación

NOMBRE: Crear Capacitación	CÓDIGO	OP018
NOMBRE PROGRAMADO	EcrearCapacitacion.xhtmll	

GUARDAR
CANCELAR

Nombre*

Fecha Inicio: Jun 20, 2012

Fecha Termino: Jun 30, 2012

Duracion: Dias: Horas:

Valor:

Jornada: MATUTINA

Descripcion:

Empleados

asignar

DNI *	Nombres * Apellidos *	Cargo	Sueldo Básico	Fecha Ingreso	Accio
No data					

Tabla 64. Pantalla Crear Capacitación

NOMBRE: Panel Modal Empleados	CÓDIGO	PM10
NOMBRE PROGRAMADO	EmodalEmpleadoCapacitacion.xhtmll	

Empleado

DNI *

Nombres *

Apellidos *

Estado

BUSCAR
BORRAR

Foto	DNI * *	Nombres * Apellidos * *	Email	Telefono	Celular
No data					

ACEPTAR
CANCELAR

Tabla 65. Panel Modal Empleados

Nombre:	Crear Capacitación	Código: 021
Requerimiento Funcional:	RF010	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado crear capacitaciones para Empleados.	
Descripción:	Que el usuario con esta función asignada pueda crear capacitaciones de trabajo y asignar los Empleados que las deben cumplir.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Capacitación.	
Poscondiciones:	Capacitación creada.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Crear] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea la Capacitación y muestra la pantalla web "Crear Capacitación". 3. El usuario ingresa los datos correspondientes a la Capacitación creada. 4. El usuario elige la opción [Empleados]. 5. El sistema muestra el panel modal "Empleados". 6. El usuario busca los Empleados a los que se va asignar la Capacitación utilizando (invocando) el caso de uso <u>Consultar Empleado</u>. 7. El usuario selecciona los Empleados del panel modal "Empleados" y elige la opción [Aceptar]. 8. El sistema carga los datos de los Empleados seleccionados en la pantalla web "Crear Capacitación" y automáticamente se cierra el panel modal "Empleados". 9. El usuario elige la opción [Guardar] de la pantalla web "Crear Capacitación". 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica que haya por lo menos un Empleado asignado a la Capacitación. 12. El sistema verifica que las fechas ingresadas estén correctas. 13. El sistema guarda los datos de la nueva Capacitación. 14. El sistema muestra un mensaje de Capacitación creada con éxito. 15. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>B. CAMPOS OBLIGATORIOS VACIOS</p> <p>B.10. El sistema muestra un mensaje de campos obligatorios vacíos.</p> <p>B.11. El caso de uso continúa en el paso 3 del flujo normal de evento.</p> <p>C. EMPLEADO NO ASIGNADO</p> <p>C.11. El sistema muestra un mensaje de Empleados no asignados.</p> <p>C.12. El caso de uso continúa en el paso 4 del flujo normal de eventos.</p> <p>D. FECHAS INCORRECTAS</p> <p>C.14. El sistema muestra mensaje fechas incorrectas.</p> <p>C.15. El caso de uso continúa en el paso 3 del flujo normal de eventos.</p>		

Tabla 66. Descripción Caso de Uso Crear Capacitación

➤ CASO DE USO: Modificar Capacitación

NOMBRE: Editar Capacitación	CÓDIGO	OP019
NOMBRE PROGRAMADO	EeditarCapacitacion.xhtmll	

CREAR
GUARDAR
CAPACITACIONES
CANCELAR

Nombre*

Fecha Inicio

Fecha Fin

duracion Dia Horas

valor

Descripcion

Estado

Empleados

asignar

DNI *	Nombres * Apellidos *	Cargo	Acciones
No data			

Realizado por: Yazber Romero & Jorge Tapia
yazb_87@hotmail.com

Tabla 67. Pantalla Editar Capacitación

NOMBRE: Panel Modal Capacitación	CÓDIGO	PM11
NOMBRE PROGRAMADO	EmodalCapacitacion.xhtmll	

capacitacion
CANCELAR

Fecha Inicio

Fecha Fin

BU SCAR

Nombre*	Fecha Inicio	Fecha Fin	valor	Descripcion	Acciones
Seguridad y Redes	Apr 20, 2012	Apr 30, 2012	0.0	encriptacion y ...	Cargar
DESARROLLO WEB 2	May 28, 2012	May 31, 2012	0.0		Cargar
test del mes de junio 6 2045	Jun 6, 2012	Jun 30, 2012	1000.0	pruebas finale...	Cargar
test 898989	Jun 6, 2012	Jun 30, 2012		pruebas	Cargar

Tabla 68. Panel Modal Capacitación

Nombre:	Modificar Capacitación		Código: 022
Requerimiento Funcional:	RF010		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario con este rol asignado modificar Capacitación.		
Descripción:	Que el usuario con esta función designada modifique Capacitaciones que se han creado con anterioridad cuando lo requiera.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Capacitación.		
Poscondiciones:	Capacitación editada.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Editar Capacitación". 3. El usuario elige la opción [Buscar Capacitación]. 4. El sistema muestra el panel Modal "Capacitación". 5. El usuario busca la Capacitación que va a modificar para lo cual utiliza (invoca) el caso de uso <u>Consultar Capacitación</u>. 6. El usuario selecciona la Capacitación que busca del panel modal "Capacitación". 7. El sistema carga los datos de la Capacitación seleccionada en la pantalla web "Editar Capacitación" y automáticamente cierra el panel modal "Capacitación". 8. El usuario modifica los datos de la Capacitación seleccionada. 9. El usuario elige la opción [Guardar] de la pantalla web "Editar Capacitación". 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica que haya por lo menos un Empleado asignado a la Capacitación. 12. El sistema verifica que las fechas ingresadas estén correctas. 13. El sistema guarda los nuevos datos de la Capacitación modificada. 14. El sistema muestra un mensaje de Capacitación modificada con éxito. 15. El caso de uso finaliza. 			
FLUJO ALTERNO:			
A. CAMPOS OBLIGATORIOS VACIOS			
A.10. El sistema muestra un mensaje de campos obligatorios vacíos.			
A.11. El caso de uso continúa en el paso 3 del flujo normal de evento.			
B. EMPLEADO NO ASIGNADO			
B.11. El sistema muestra un mensaje de Empleados no asignados.			
B.12. El caso de uso continúa en el paso 8 del flujo normal de eventos.			
C. FECHAS INCORRECTAS			
C.12. El sistema muestra mensaje fechas incorrectas.			
C.13. El caso de uso continúa en el paso 8 del flujo normal de eventos.			

Tabla 69. Descripción Caso de Uso Modificar Capacitación

➤ CASO DE USO: Consultar Capacitación

NOMBRE: Buscar Capacitación	CÓDIGO	OP020
NOMBRE PROGRAMADO	ElistarCapacitacion.xhtmll	

CREAR IMPRIMIR

Fecha Desde: Hasta:

BUSCAR BORRAR

Nombre*	Fecha Inicio	Fecha Fin	jornada	precio	Acciones
MONITOREO DE REDES	Mar 20, 2012	Mar 31, 2012	Mañana	200.0	
PROGRAMACION JAVA	Mar 12, 2012	Mar 16, 2012	Matutina	650.0	
GESTIÓN DE REDES	Mar 1, 2012	Mar 5, 2012	Matutino	150.0	
GESTIÓN DE REDES	Mar 1, 2012	Mar 5, 2012	Matutino	135.0	
Seguridad y Redes	Apr 20, 2012	Apr 30, 2012	Matutino	0.0	

Tabla 70. Pantalla Buscar Capacitación

NOMBRE: Panel Modal Empleado	CÓDIGO	PM12
NOMBRE PROGRAMADO		

Empleado

DNI *
 Nombres *
 Apellidos *
 Estado

BUSCAR BORRAR

Foto	DNI * *	Nombres * Apellidos * *	Email	Telefono	Celular	Acciones
No data						

Tabla 71. Panel Modal Empleado

Nombre:	Consultar Capacitación		Código: 023
Requerimiento Funcional:	RF010		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario con este rol asignado consultar capacitaciones.		
Descripción:	Que el usuario con esta función asignada realice consultas de capacitaciones cuando lo necesite.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Capacitación.		
Poscondiciones:	Lista de Capacitaciones consultada.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Consultar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Buscar Capacitación". 3. El usuario ingresa las fechas desde y hasta donde desea realizar la consulta. 4. El usuario elige el tipo de consulta que va a realizar: las capacitaciones de todos los empleados o las capacitaciones de un solo empleado, si elige la primera opción va directamente al paso 11. 5. Si el usuario elige la segunda opción el sistema actualiza la pantalla "Buscar Capacitación" de acuerdo a este tipo de consulta. 6. El usuario selecciona la opción [Buscar Empleado] de la pantalla web "Buscar Capacitación". 7. El sistema muestra el panel modal "Empleado". 8. El usuario busca el Empleado para consultar sus Permisos de trabajo utilizando (invocando) el caso de uso <u>Consultar Empleado</u>. 9. El usuario selecciona el Empleado que busca del panel modal "Empleado". 10. El sistema carga los datos del Empleado seleccionado en la pantalla web "Buscar Capacitación" y automáticamente cierra el panel modal "Empleado". 11. El usuario elige la opción [Buscar] de la pantalla web "Buscar Capacitación". 12. El sistema verifica que las fechas ingresadas estén correctas. 13. El sistema muestra los Permisos en la tabla de resultados de la pantalla web "Buscar Capacitación". 14. El caso de uso finaliza. 			
FLUJO ALTERNO:			
A. FECHAS INCORRECTAS			
<ol style="list-style-type: none"> A.12. El sistema muestra un mensaje de fechas ingresadas incorrectas. A.13. El caso de uso continúa en el paso 3 del flujo normal de eventos. 			

Tabla 72. Descripción Caso de Uso Consultar Capacitación

➤ CASO DE USO: Crear Anticipo

NOMBRE: Crear Anticipo	CÓDIGO	OP021
NOMBRE PROGRAMADO	EcrearAnticipo.xhtml	

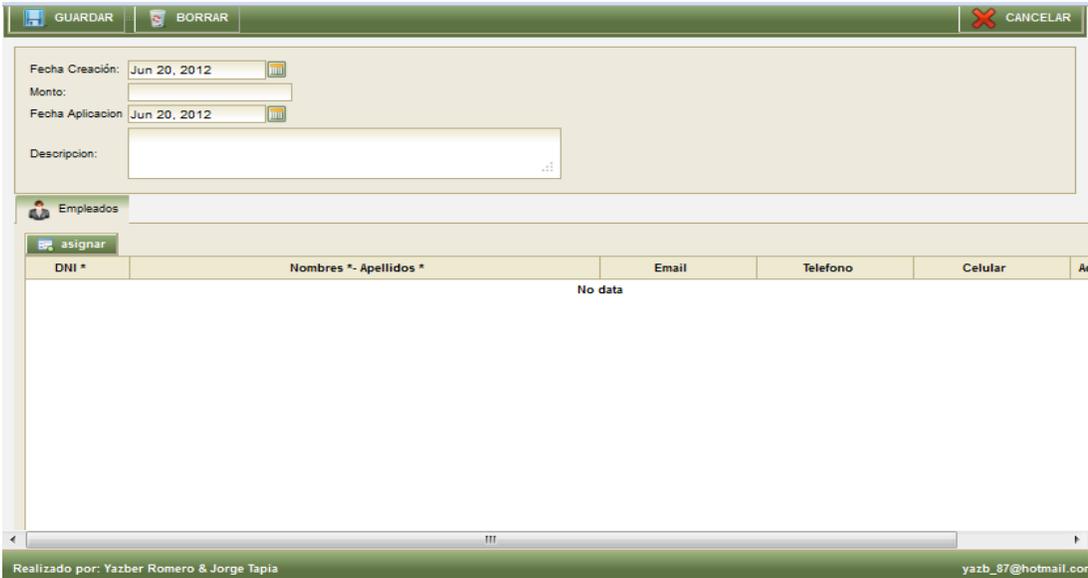


Tabla 73. Pantalla Crear Anticipo

NOMBRE: Panel Modal Empleado	CÓDIGO	PM13
NOMBRE PROGRAMADO	EmodalBuscarEmpleadoAnticipo.xhtml	

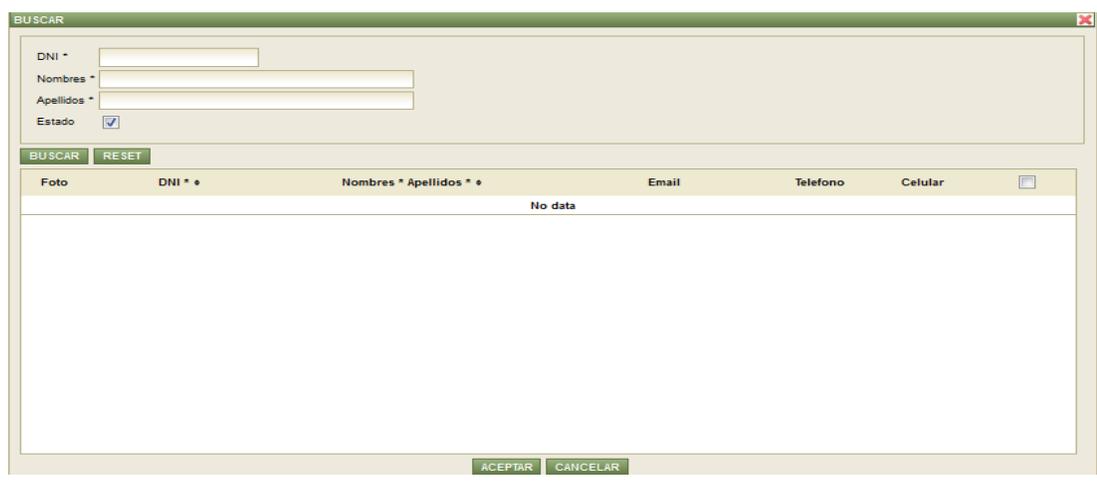


Tabla 74. Panel Modal Empleado

Nombre:	Crear Anticipo	Código: 024
Requerimiento Funcional:	RF010	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado crear Anticipos de Sueldo.	
Descripción:	Que el usuario con esta función asignada pueda crear anticipos de sueldo de los empleados que lo soliciten.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Anticipo.	
Poscondiciones:	Anticipo de Sueldo creado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Crear] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea el Anticipo y muestra la pantalla web "Crear Anticipo". 3. El usuario ingresa los datos correspondientes al Anticipo creado. 4. El usuario elige la opción [Empleados] de la pantalla web "Crear Anticipo". 5. El sistema muestra el panel modal "Empleado". 6. El usuario busca los Empleados a los que se va asignar el Anticipo utilizando (invocando) el caso de uso <u>Consultar Empleado</u>. 7. El usuario selecciona los Empleados del panel modal "Empleado" y elige la opción [Aceptar]. 8. El sistema carga los datos de los Empleados seleccionados en la pantalla web "Crear Anticipo" y automáticamente se cierra el panel modal "Empleado". 9. El usuario elige la opción [Guardar] de la pantalla web "Crear Anticipo". 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica que haya por lo menos un Empleado asignado para el Anticipo. 12. El sistema verifica que las fechas ingresadas estén correctas. 13. El sistema verifica que el monto del Anticipo sea mayor que cero. 14. El sistema guarda los datos del nuevo Anticipo. 15. El sistema muestra un mensaje de Anticipo creado con éxito. 16. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.10. El sistema muestra un mensaje de campos obligatorios vacíos.</p> <p>A.11. El caso de uso continúa en el paso 3 del flujo normal de eventos.</p> <p>B. EMPLEADO NO ASIGNADO</p> <p>B.11. El sistema muestra un mensaje de Empleados no asignados.</p> <p>B.12. El caso de uso continúa en el paso 4 del flujo normal de eventos.</p> <p>C. FECHAS INCORRECTAS</p> <p>C.12. El sistema muestra mensaje fechas incorrectas.</p> <p>C.13. El caso de uso continúa en el paso 3 del flujo normal de eventos.</p> <p>D. MONTO INCORRECTO</p> <p>D.13. El sistema muestra un mensaje de Monto debe ser mayor a cero.</p> <p>D.14. El caso de uso continúa en el paso 3 del flujo normal de eventos.</p>		

Tabla 75. Descripción Caso de Uso Crear Anticipo

➤ CASO DE USO: Modificar Anticipo

NOMBRE: Editar Anticipo	CÓDIGO	OP022
NOMBRE PROGRAMADO	EeditarAnticipo.xhtml	

GUARDAR BORRAR BUSCAR CANCELAR

Fecha Creación:

Monto:

Fecha Aplicacion:

Descripcion:

Empleados

asignar

DNI *	Nombres *- Apellidos *	Email	Telefono	Celular	Acciones
No data					

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com

Tabla 76. Pantalla Editar Anticipo

NOMBRE: Panel Modal Egresos	CÓDIGO	PM14
NOMBRE PROGRAMADO	EmodalAnticipos.xhtml	

egresos X

Fecha Inicio:

Fecha Fin:

Criterio:

BUSCAR BORRAR

Fecha Creación	Fecha Aplicacion	Descripcion	valor	Acciones
No data				

Tabla 77. Panel Modal Egresos

Nombre:	Modificar Anticipo		Código: 025
Requerimiento Funcional:	RF010		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario con este rol asignado modificar Anticipos de Sueldo.		
Descripción:	El usuario con esta función designada pueda modificar Anticipos de sueldo que se han creado con anterioridad.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Anticipo.		
Poscondiciones:	Anticipo de Sueldo editado.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Editar Anticipo". 3. El usuario elige la opción [Buscar Anticipo]. 4. El sistema muestra el panel Modal "Egresos". 5. El usuario busca el Anticipo que va a modificar para lo cual utiliza (invoca) el caso de uso <u>Consultar Anticipo</u>. 6. El usuario selecciona el Anticipo que busca del panel modal "Egresos". 7. El sistema carga los datos del Anticipo seleccionado en la pantalla web "Editar Anticipo" y automáticamente cierra el panel modal "Egresos". 8. El usuario modifica los datos del Anticipo seleccionado. 9. El usuario elige la opción [Guardar] de la pantalla web "Editar Anticipo". 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica que haya por lo menos un Empleado asignado para el Anticipo. 12. El sistema verifica que las fechas ingresadas estén correctas. 13. El sistema verifica que el monto ingresado sea mayor a cero. 14. El sistema guarda los nuevos datos del Anticipo modificado. 15. El sistema muestra un mensaje de Anticipo modificado con éxito. 16. El caso de uso finaliza. 			
FLUJO ALTERNO:			
A. CAMPOS OBLIGATORIOS VACIOS			
A.10. El sistema muestra un mensaje de campos obligatorios vacíos.			
A.11. El caso de uso continúa en el paso 3 del flujo normal de eventos.			
B. EMPLEADO NO ASIGNADO			
B.11. El sistema muestra un mensaje de Empleados no asignados.			
B.12. El caso de uso continúa en el paso 4 del flujo normal de eventos.			
C. FECHAS INCORRECTAS			
C.12. El sistema muestra mensaje fechas incorrectas.			
C.13. El caso de uso continúa en el paso 3 del flujo normal de eventos.			
D. MONTO INCORRECTO			
D.13. El sistema muestra un mensaje de Monto debe ser mayor a cero.			
D.14. El caso de uso continúa en el paso 3 del flujo normal de eventos.			

Tabla 78. Descripción Caso de Uso Modificar Anticipo

➤ CASO DE USO: Consultar Anticipo

NOMBRE: Buscar Anticipo	CÓDIGO	OP023
NOMBRE PROGRAMADO	EconsultarAnticipo.xhtml	

CREAR
IMPRIMIR
BORRAR
CANCELAR

Fecha Inicio:

Fecha Fin:

Criterio:

BUSCAR
BORRAR

Fecha Aplicacion	Fecha Aplicacion	Descripcion	Monto	Acciones
No data				

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com

Tabla 79. Pantalla Buscar Anticipo

NOMBRE: Panel Modal Empleado	CÓDIGO	PM15
NOMBRE PROGRAMADO	EmodalConsultaAnticipo.xhtml	

Empleado
CANCELAR

DNI *

Nombres *

Apellidos *

Estado

BUSCAR
BORRAR

Foto	DNI * *	Nombres * Apellidos * *	Email	Telefono	Celular	Acciones
No data						

Tabla 80. Panel Modal Empleado

Nombre:	Consultar Anticipo		Código: 026
Requerimiento Funcional:	RF010		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario con este rol asignado consultar anticipos de sueldo.		
Descripción:	Que el usuario con esta función asignada realice consultas de anticipos ya sea de todos los empleados o de uno en específico.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e Ítem Anticipo.		
Poscondiciones:	Lista de Anticipos de Sueldo individual o general.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Consultar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Buscar Anticipo". 3. El usuario ingresa las fechas desde y hasta donde desea realizar la consulta. 4. El usuario elige el tipo de consulta que va a realizar: los anticipos de todos los empleados o los anticipos de un solo empleado, si elige la primera opción va directamente al paso 11. 5. Si el usuario elige la segunda opción el sistema actualiza la pantalla "Buscar Anticipo" de acuerdo a este tipo de consulta. 6. El usuario selecciona la opción [Buscar Empleado] de la pantalla web "Buscar Anticipo". 7. El sistema muestra el panel modal "Empleado". 8. El usuario busca el Empleado para consultar sus Anticipos de Sueldo utilizando (invocando) el caso de uso <u>Consultar Empleado</u>. 9. El usuario selecciona el Empleado que busca del panel modal "Empleado". 10. El sistema carga los datos del Empleado seleccionado en la pantalla web "Buscar Anticipo" y automáticamente cierra el panel modal "Empleado". 11. El usuario elige la opción [Buscar] de la pantalla web "Buscar Anticipo". 12. El sistema verifica que las fechas ingresadas estén correctas. 13. El sistema muestra los Anticipos en la tabla de resultados de la pantalla web "Buscar Anticipo". 14. El caso de uso finaliza. 			
FLUJO ALTERNO:			
<p>A. FECHAS INCORRECTAS</p> <ol style="list-style-type: none"> A.12. El sistema muestra un mensaje de fechas ingresadas incorrectas. A.13. El caso de uso continúa en el paso 3 del flujo normal de eventos. 			

Tabla 81. Descripción Caso de Uso Consultar Anticipo

➤ CASO DE USO: Generar Rol Individual

NOMBRE: Generar Rol Individual	CÓDIGO	OP024
NOMBRE PROGRAMADO	EcrearPagoIndividual.xhtml	

GUARDAR
BORRAR

Empleado

DNI *

Cargo

Fecha Inicio: Jul 1, 2012

Fecha Fin: Jul 31, 2012



Sueldo Básico	Horas Totales	Ingresos			egresos		aportePatronal	Sueldo Líquido
		totalIngresos	Anticipos	Multas	Aporte Personal less	totalEgresos		

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com

Tabla 82. Pantalla Generar Rol Individual

NOMBRE: Panel Modal Empleado	CÓDIGO	PM16
NOMBRE PROGRAMADO	EmodalEmpleadoPagoInd.xhtml	

BU SCAR
BORRAR

DNI *

Nombres *

Apellidos *

Estado

Foto	DNI * *	Nombres * Apellidos * *	Email	Telefono	Celular	Acciones
No data						

Tabla 83. Panel Modal Empleado

Nombre:	Generar Rol Individual		Código: 027
Requerimiento Funcional:	RF015		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario poder generar roles de pagos de manera individual.		
Descripción:	Al usuario que tiene a su cargo esta función generar roles de pago de un solo empleado ósea individuales.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e ítem Rol de Pagos.		
Poscondiciones:	Rol de Pago Individual generado.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Rol Pago Individual] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Generar Rol Individual". 3. El usuario selecciona la opción [Buscar Empleado]. 4. El sistema muestra el panel modal "Empleado". 5. El usuario busca el Empleado para Generar el Rol de Pago individual utilizando (invocando) el caso de uso <u>Consultar Empleado</u>. 6. El usuario selecciona el Empleado que busca del panel modal "Empleado". 7. El sistema carga los datos del Empleado seleccionado en la pantalla web "Generar Rol Individual" y automáticamente cierra el panel modal "Empleado". 8. El usuario ingresa las fechas desde y hasta para generar el Rol. 9. El usuario elige la opción [Actualizar] de la pantalla web "Generar Rol Individual". 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema valida que las fechas ingresadas estén correctas. 12. El sistema muestra la información correspondiente al pago del Empleado en la pantalla web "Generar Rol Individual". 13. El caso de uso finaliza. 			
FLUJO ALTERNO			
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.10. El sistema muestra mensaje en los campos obligatorios vacíos.</p> <p>A.11. El caso de uso continúa en el paso 3 del flujo normal de evento.</p> <p>B. FECHAS INCORRECTAS</p> <p>B.11. El sistema muestra un mensaje de fechas de consulta incorrectas.</p> <p>B.12. El caso de uso continúa en el paso 8 del flujo normal de eventos.</p>			

Tabla 84. Descripción Caso de Uso Generar Rol Individual

➤ CASO DE USO: Generar Rol General

NOMBRE: Generar Rol General		CÓDIGO	OP025
NOMBRE PROGRAMADO		EcrearPagoGeneral.xhtml	

GUARDAR IMPRIMIR CANCELAR

Fecha Inicio: Jun 1, 2012

Fecha Fin: Jun 30, 2012

Criterio: Todos

GENERAR BORRAR

DNI *	Nombres *Apellidos *	Básico	horasT	Dias Labor	valorDia	totalIngresos	Multas	Anticipos	aportacionless	totalEgresos	Suel
No data											

Tabla 85. Pantalla Generar Rol General

Nombre:	Generar Rol General		Código: 028	
Requerimiento Funcional:	RF014			
Tipo de Caso de Uso:	Sistema		Actores:	Usuario
Objetivo:	Permitir al usuario con el rol asignado generar roles de pagos generales.			
Descripción:	Al usuario a quien se ha designado esta función genere roles de pagos de todos los empleados o de un determinado departamento de la organización o empresa.			
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Empleados e ítem Rol de Pagos.			
Poscondiciones:	Rol de Pago General generado.			
FLUJO NORMAL:				
<ol style="list-style-type: none"> 1. El usuario elige la opción [Rol Pago General] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla "Crear Pago General". 3. El usuario ingresa los datos para generar el Rol de Pagos General. 4. El usuario elige si va a general el Rol de Pago de todos los empleados si es así va directamente al paso 7. 				

5. Si el usuario elige generar el Rol de Pago de un departamento el sistema actualiza la pantalla web "Crear Pago General" y muestra los departamentos disponibles.
6. El usuario selecciona el Departamento del cual quiere generar el Rol.
7. El usuario selecciona el botón [Generar] de la Pantalla web "Crear Pago General".
8. El sistema valida que las fechas ingresadas estén correctas.
9. El sistema carga los Empleados con sus respectivos datos Sueldos (tomando en cuenta egresos, ingresos, horas trabajadas entre otros) en la tabla Pagos Generales de la pantalla web "Crear Pago General".
10. El caso de uso finaliza.

FLUJO ALTERNO:

A. FECHAS INCORRECTAS

- A.8. El sistema muestra un mensaje de fechas de consulta incorrectas.
- A.9. El caso de uso continúa en el paso 3 del flujo normal de eventos.

Tabla 86. Descripción Caso de Uso Generar Rol General

3.2.3. MODELO DE INTERACCIÓN

3.2.3.1. DIAGRAMA DE SECUENCIA 007: Crear Empleado (DS007)

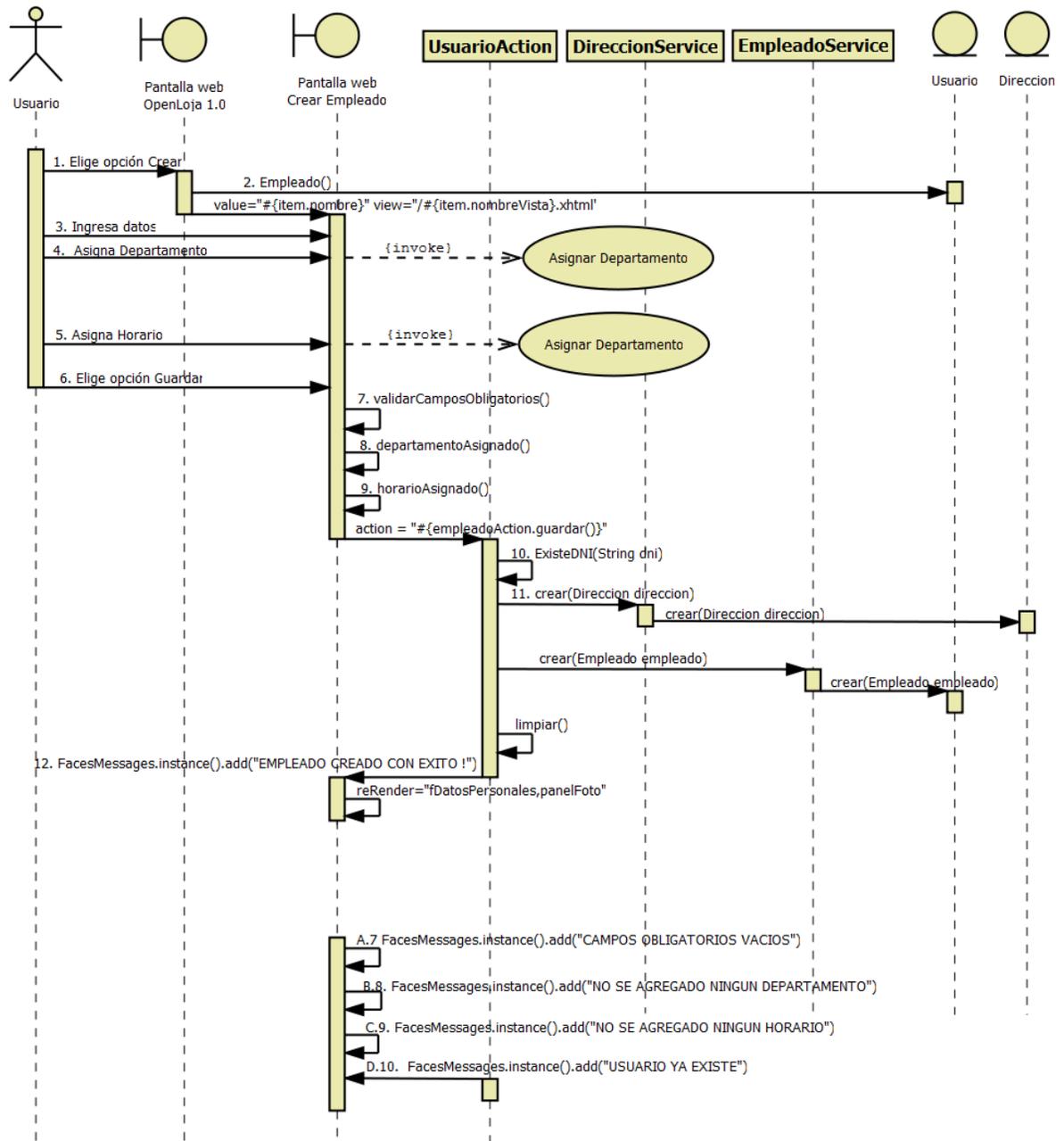


Diagrama 17. Secuencia Crear Empleado Versión Final

3.2.3.2. DIAGRAMA DE SECUENCIA 008: Modificar Empleado

(DS008)

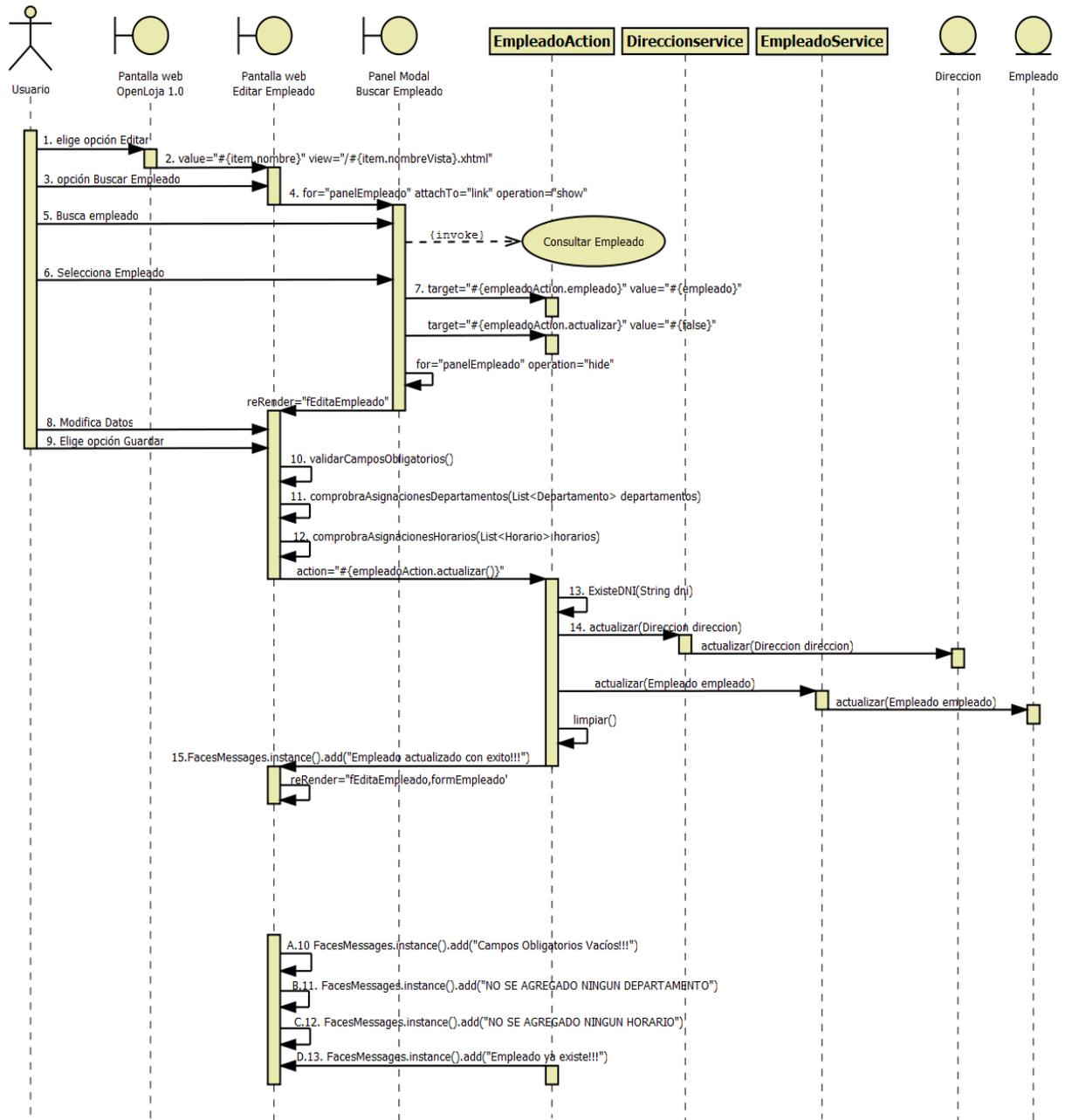


Diagrama 18. Secuencia Modificar Empleado Versión Final

3.2.3.3. DIAGRAMA DE SECUENCIA 009: Consultar Empleado (DS009)

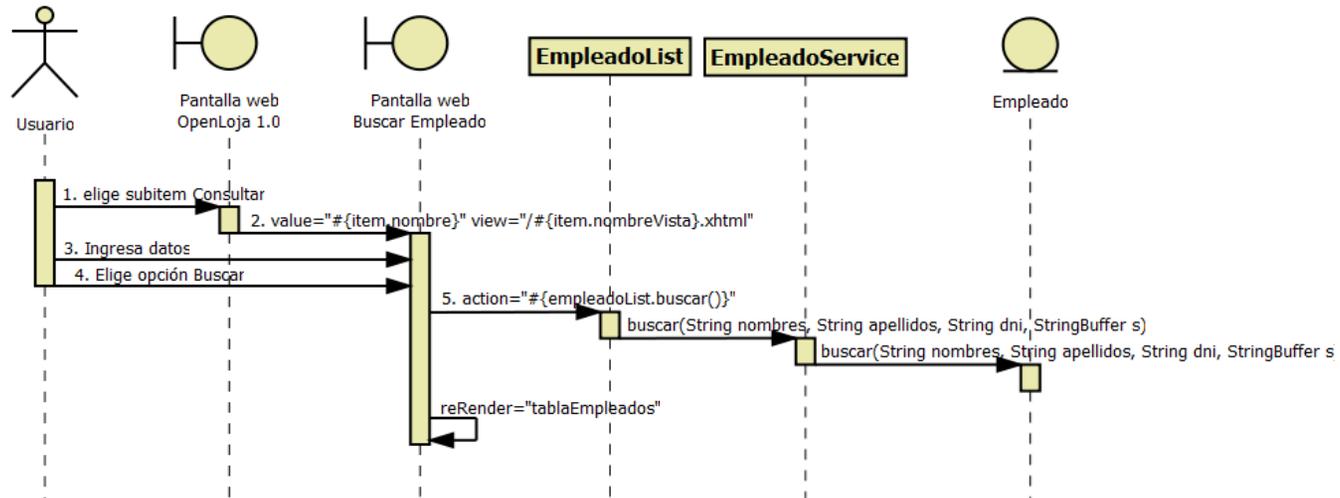


Diagrama 19.Secuencia Consultar Empleado Versión Final

3.2.3.4. DIAGRAMA DE SECUENCIA 010: Crear Departamento (DS010)

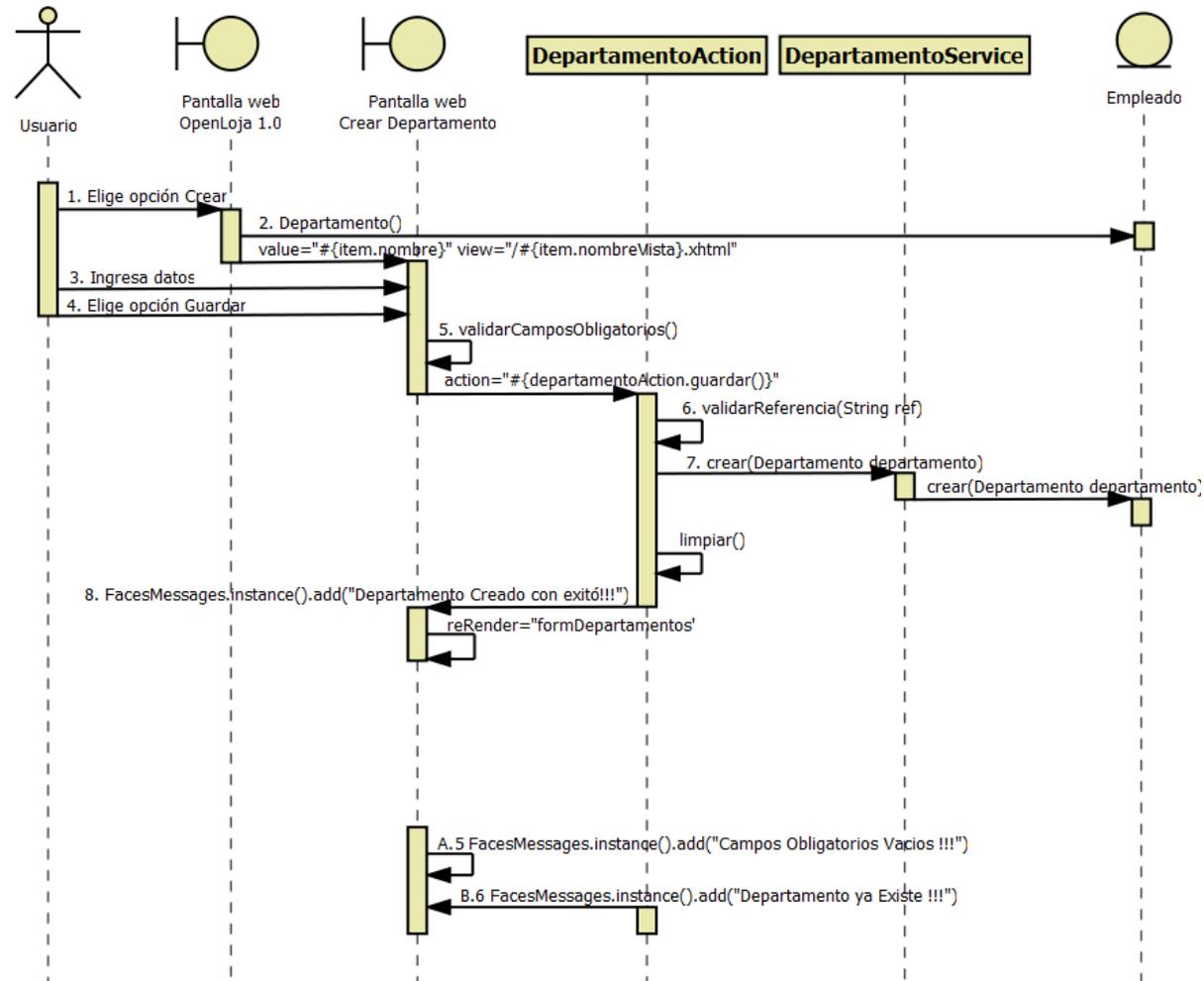


Diagrama 20. Secuencia Crear Departamento Versión Final

3.2.3.5. DIAGRAMA DE SECUENCIA 011: Asignar Departamento (DS011)

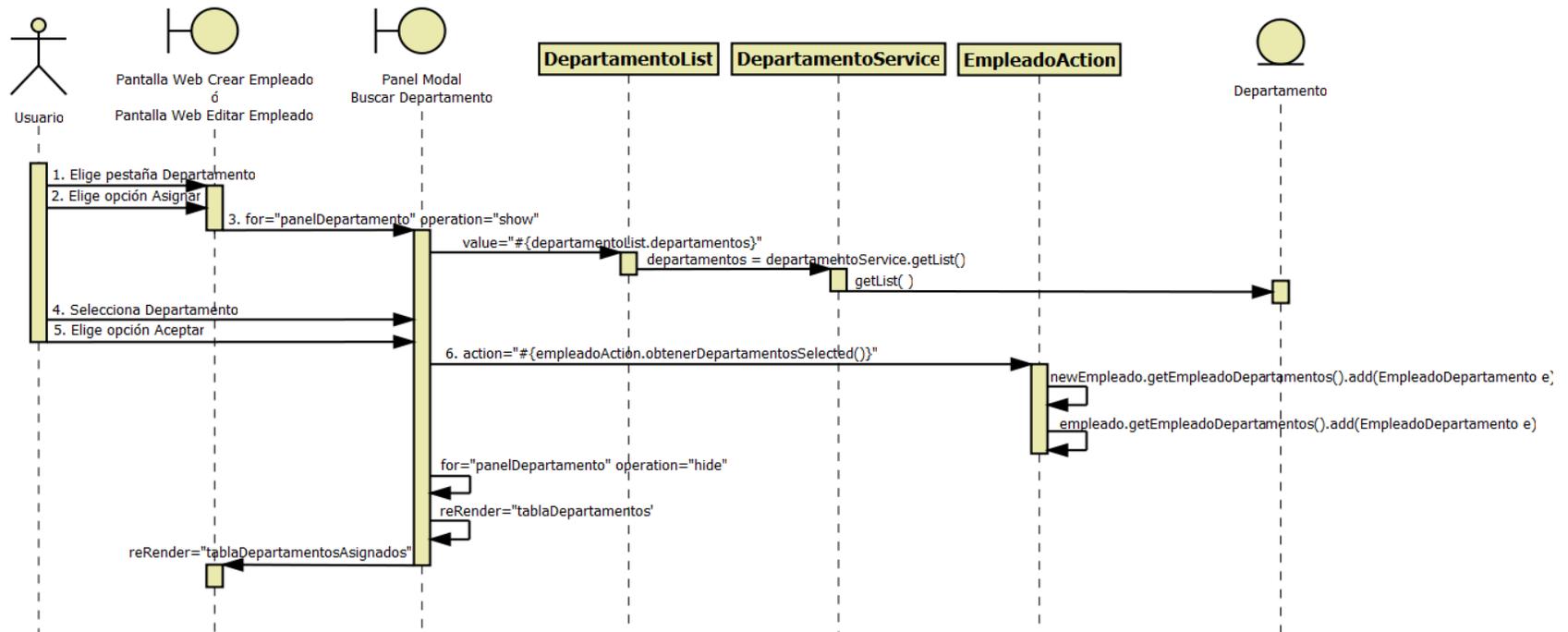


Diagrama 21. Secuencia Asignar Departamento Versión Final

3.2.3.6. DIAGRAMA DE SECUENCIA 012: Modificar Departamento (DS012)

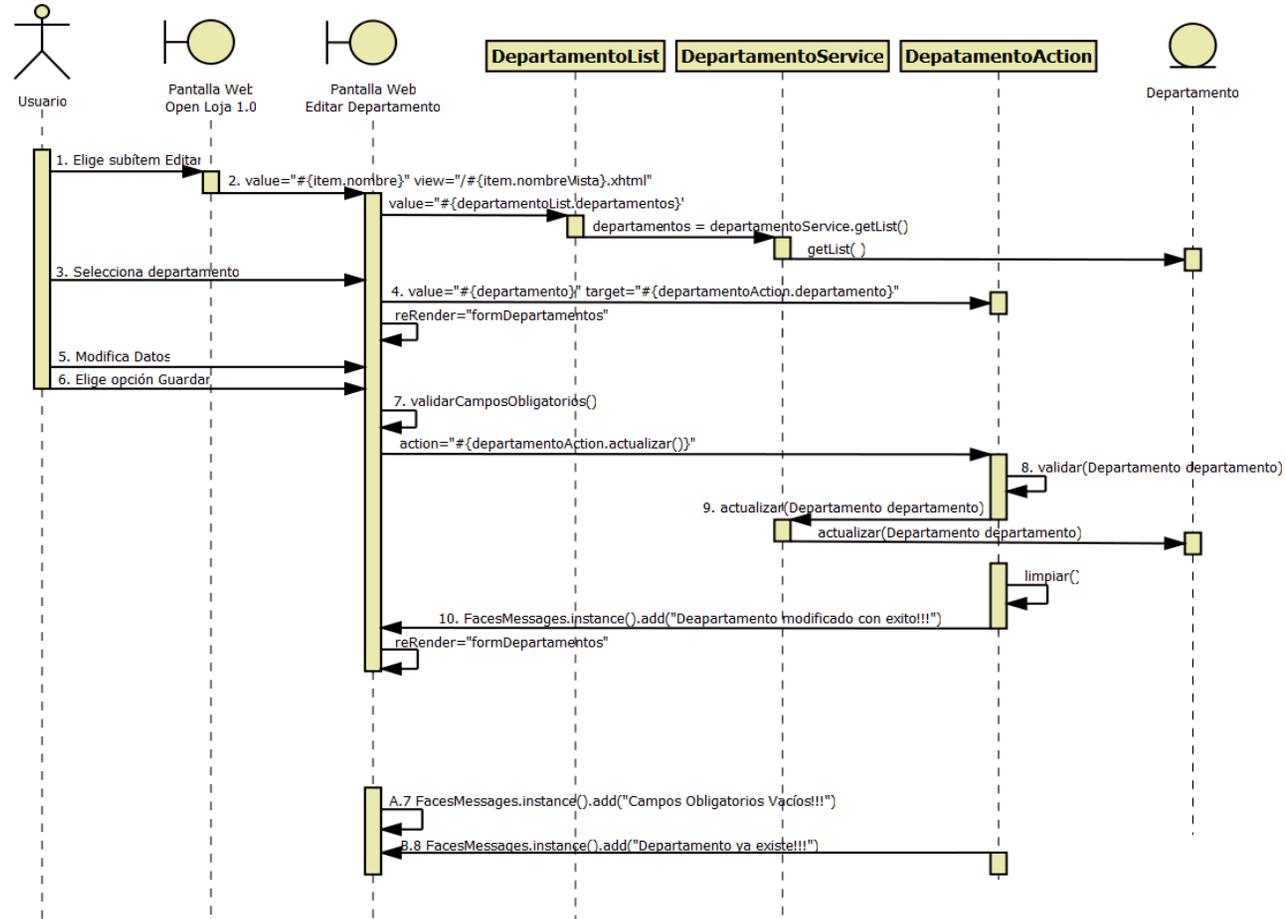


Diagrama 22. Secuencia Modificar Departamento Versión Final

3.2.3.7. DIAGRAMA DE SECUENCIA 013: Consultar Departamento

(DS013)

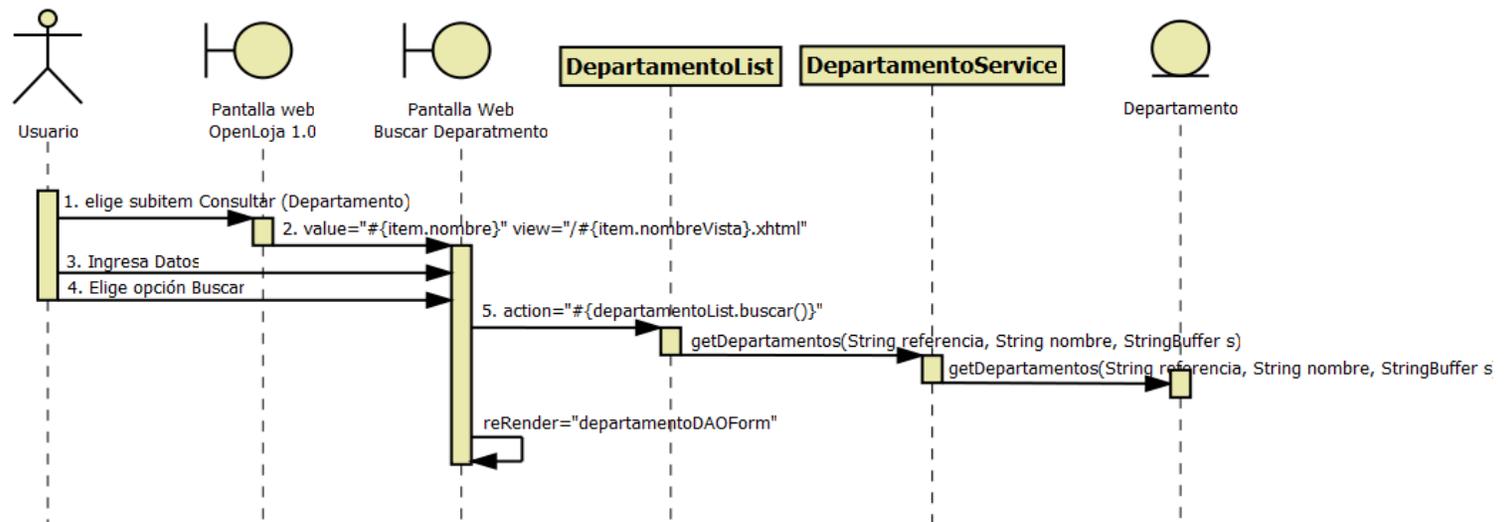


Diagrama 23. Secuencia Consultar Departamento

3.2.3.8. DIAGRAMA DE SECUENCIA 014: Crear Horario

(DS014)

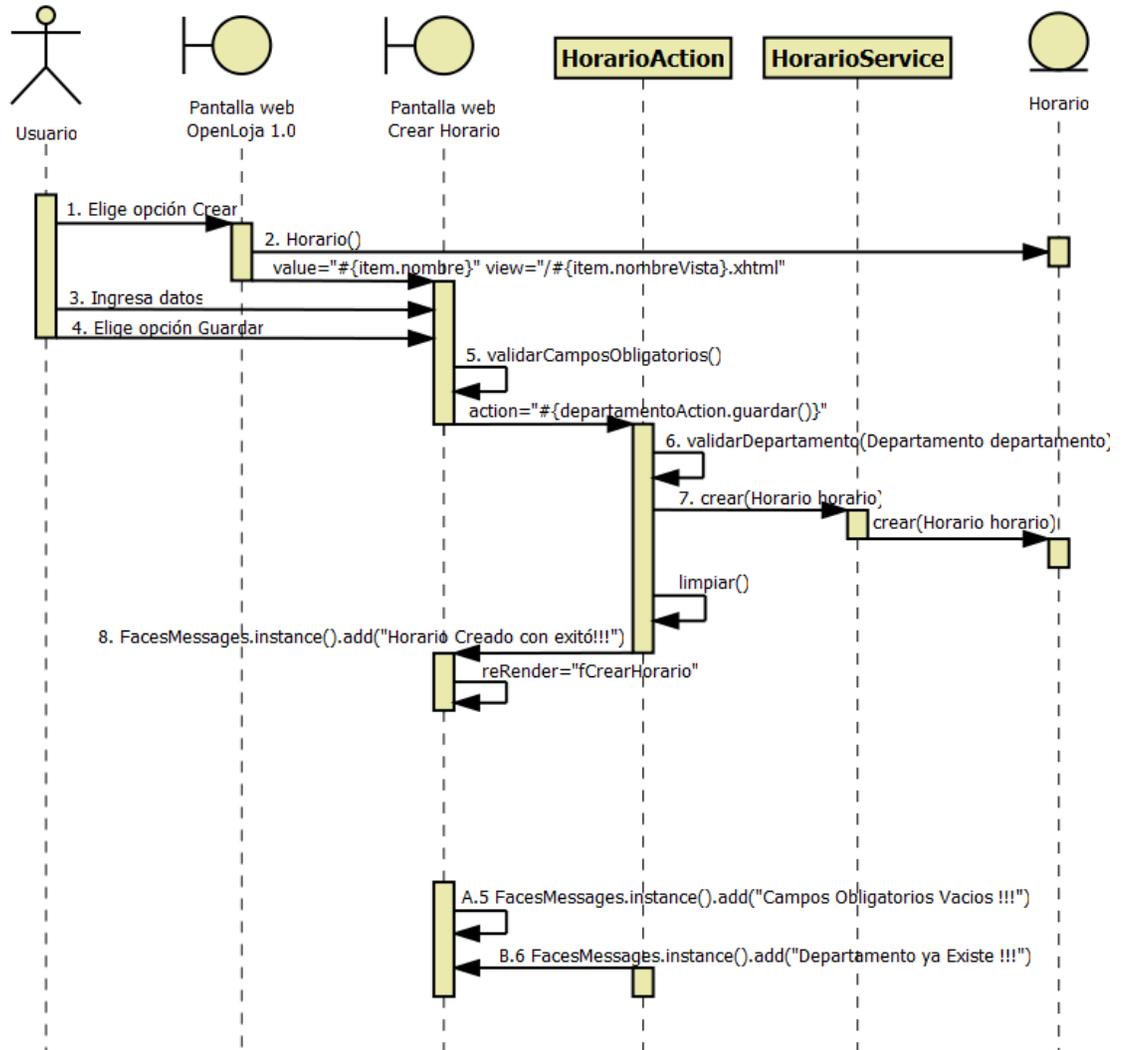


Diagrama 24.Secuencia Crear Horario Versión Final

3.2.3.9. DIAGRAMA DE SECUENCIA 015: Asignar Horario (DS015)

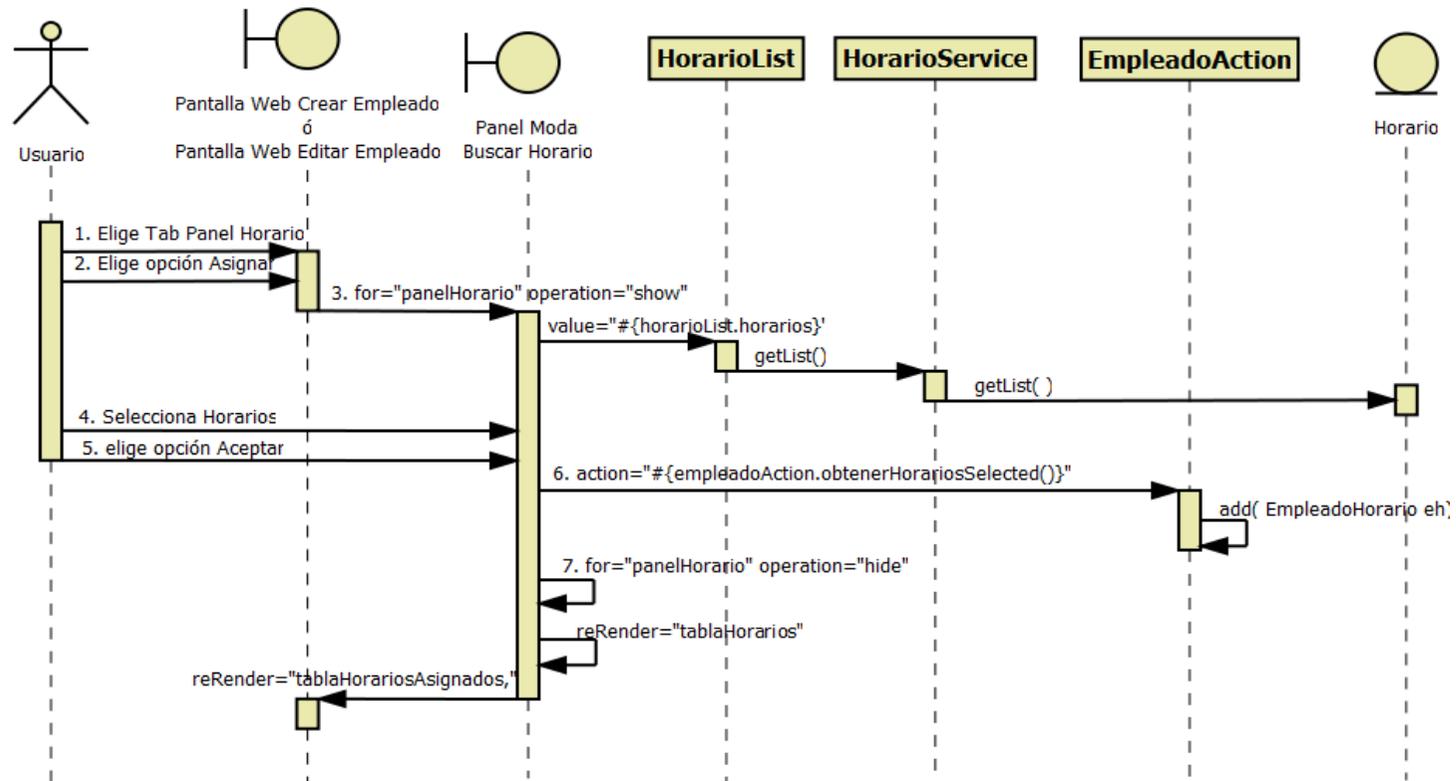


Diagrama 25. Secuencia Asignar Horario Versión Final

3.2.3.10. DIAGRAMA DE SECUENCIA 016: Registro Entrada Salida (DS016)

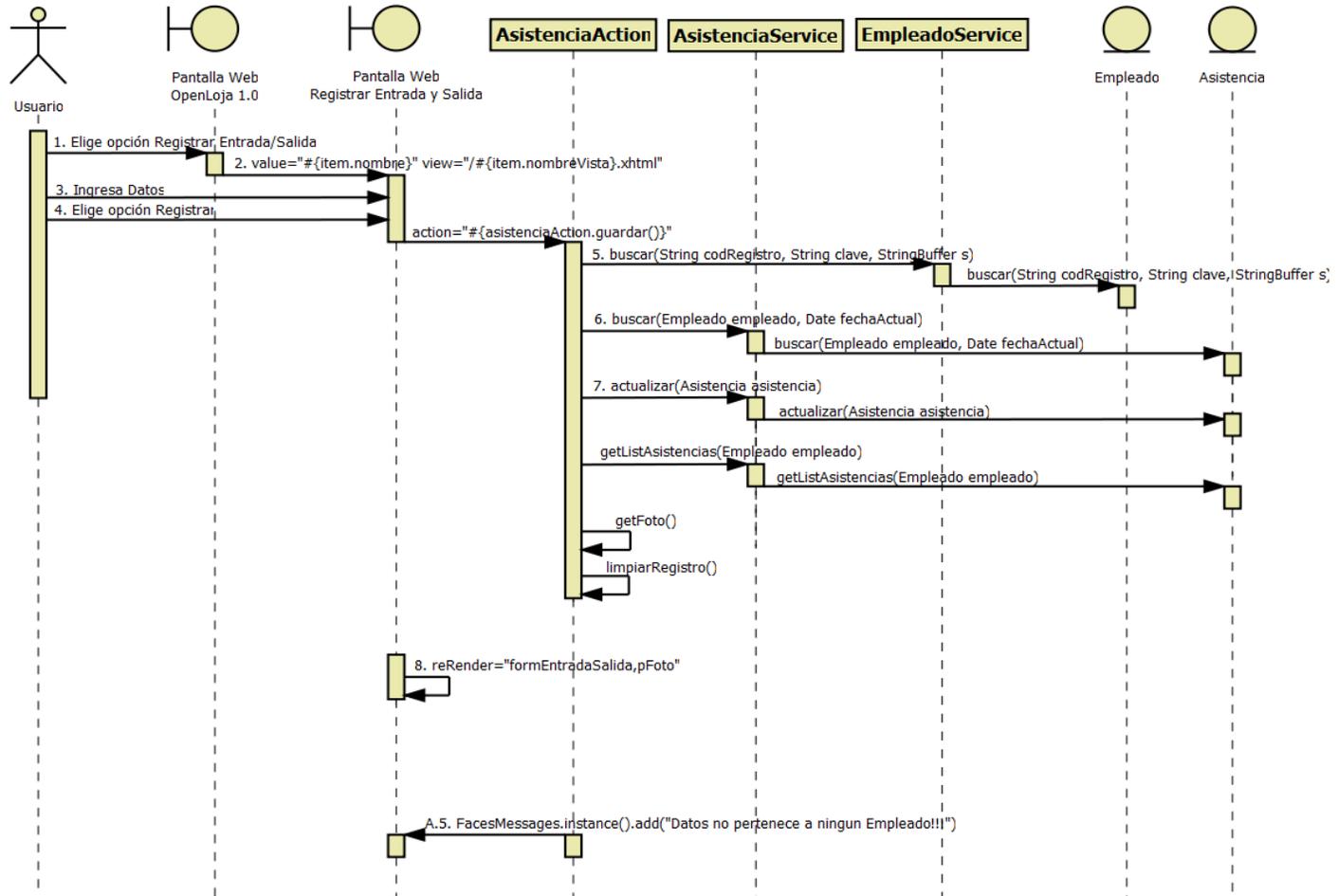


Diagrama 26. Secuencia Registro Entrada Salida Versión Final

3.2.3.11. DIAGRAMA DE SECUENCIA 017: Consultar Asistencias (DS017)

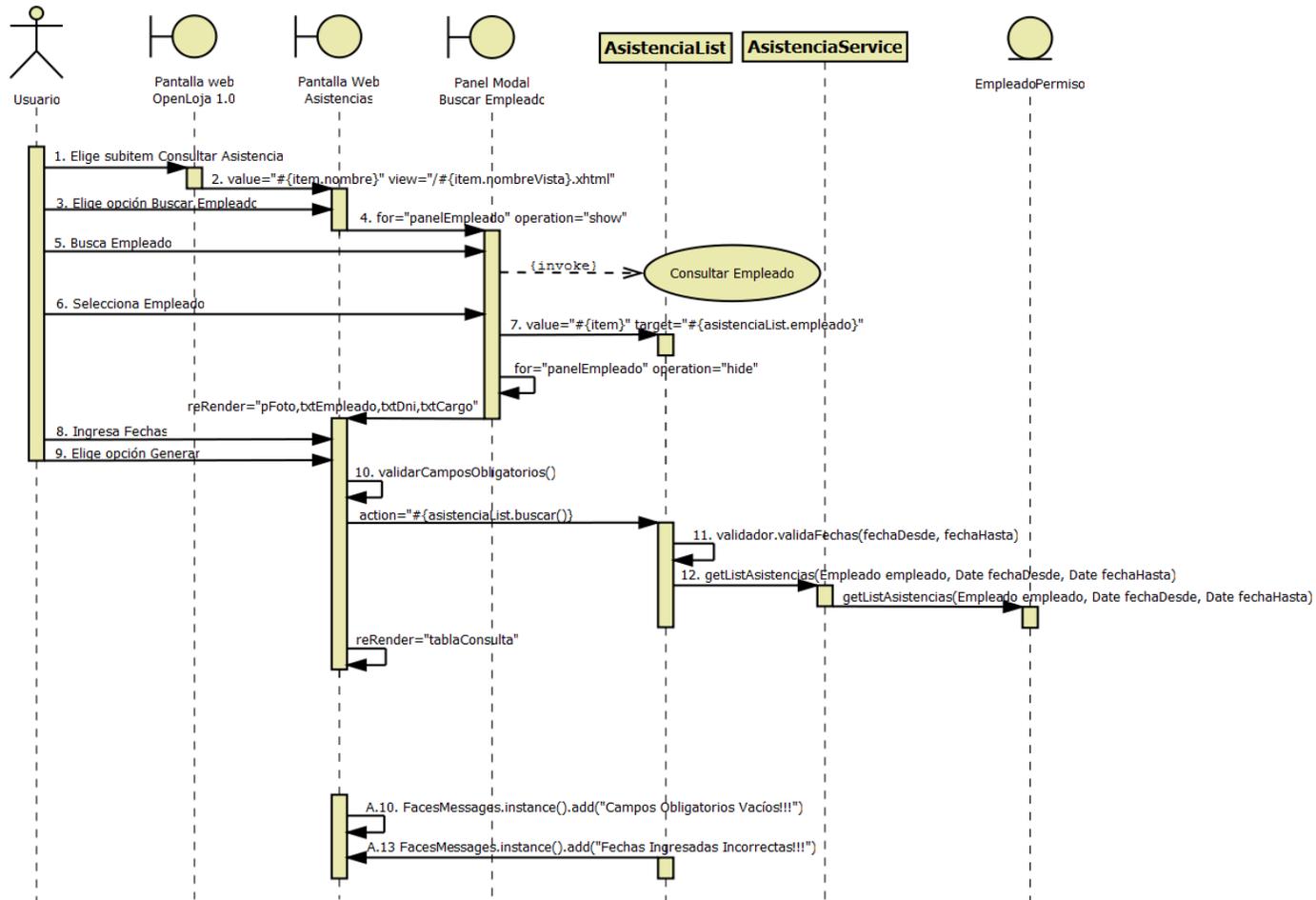


Diagrama 27. Secuencia Consultar Asistencias Versión Final

3.2.3.12. DIAGRAMA DE SECUENCIA 018: Crear Permiso (DS018)

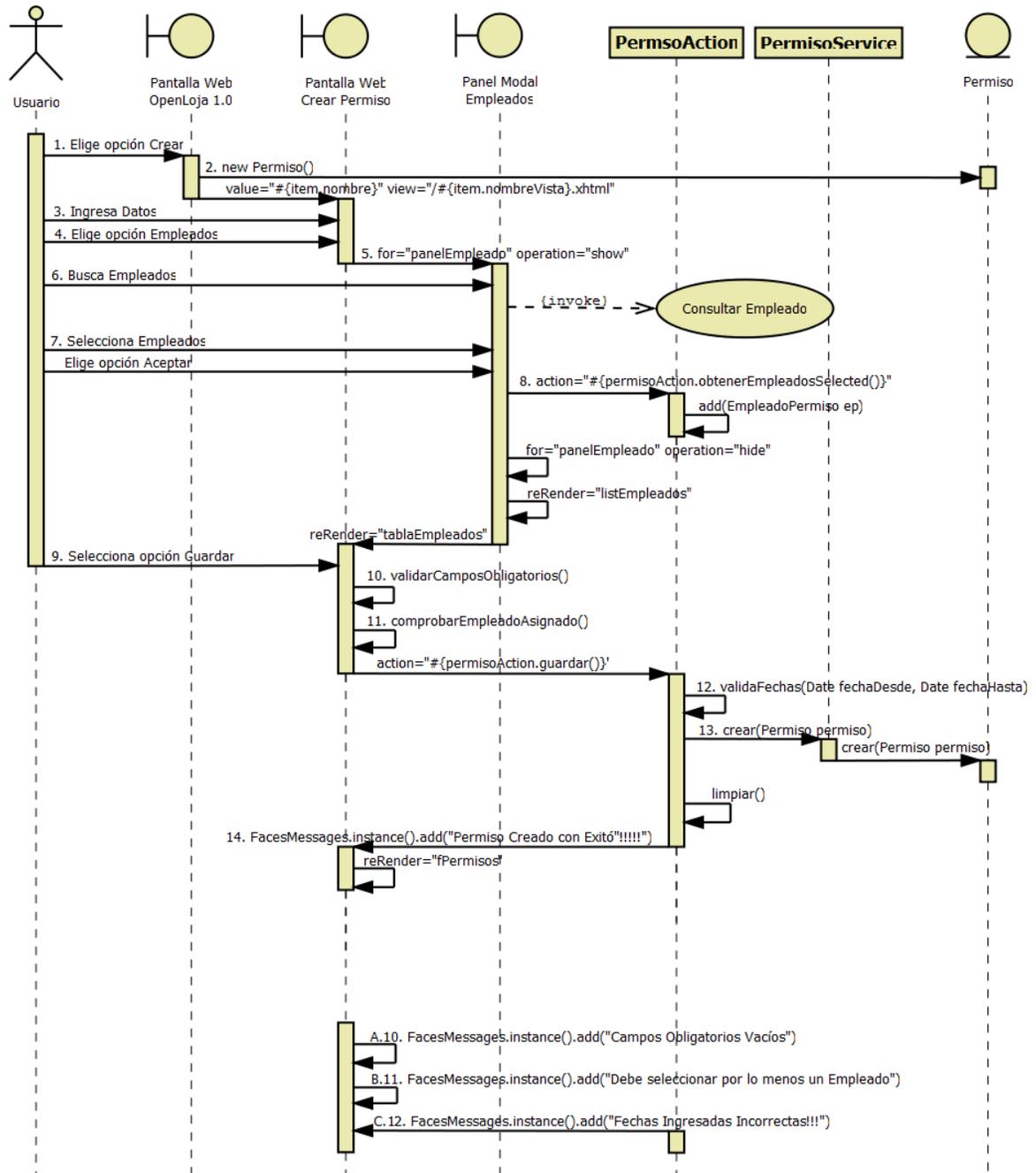


Diagrama 28. Secuencia Crear Permiso Versión Final

3.2.3.13. DIAGRAMA DE SECUENCIA 019: Modificar Permiso

(DS019)

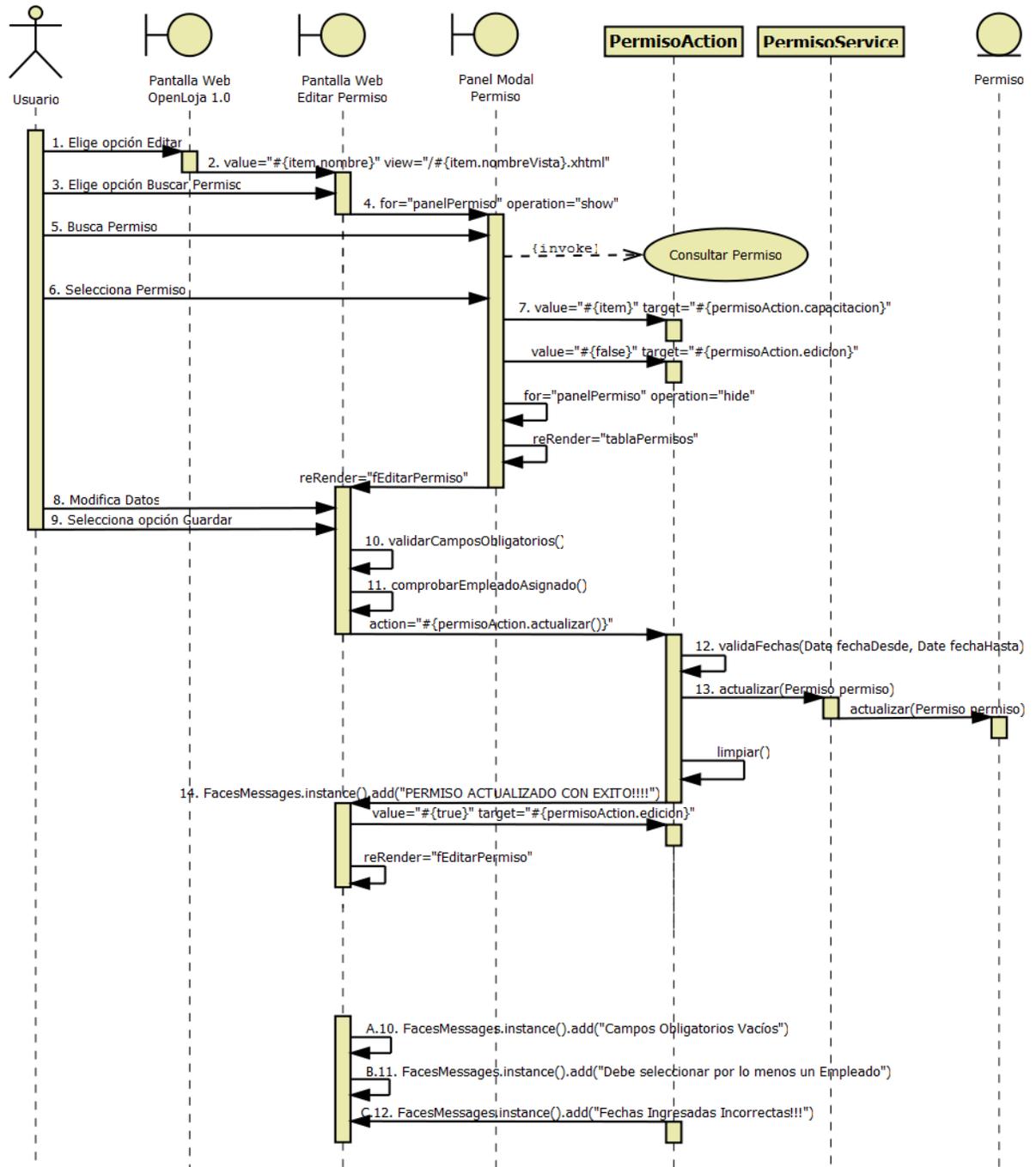


Diagrama 29. Secuencia Modificar Permiso Versión Final

3.2.3.14. DIAGRAMA DE SECUENCIA 020: Consultar Permiso

(DS020)

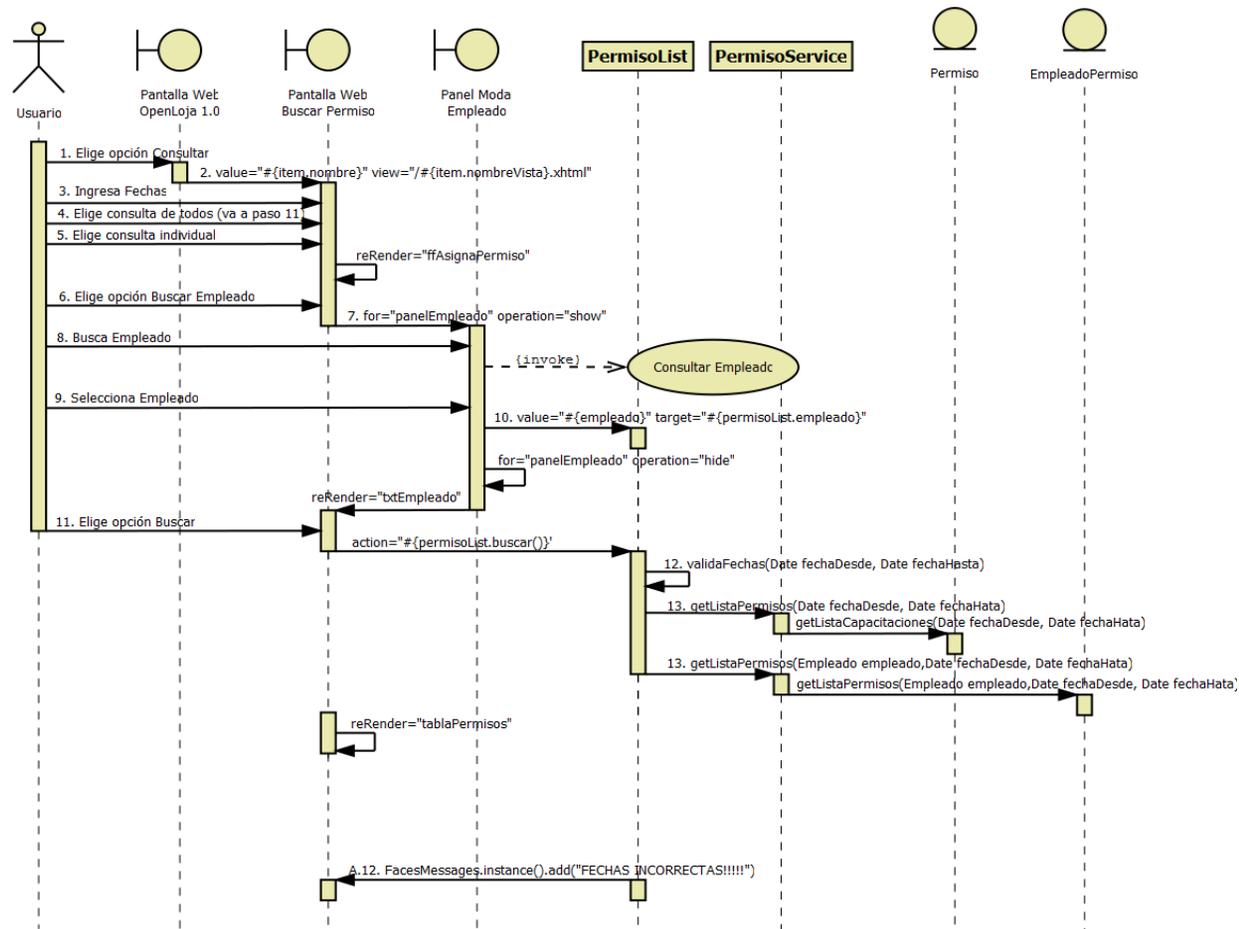


Diagrama 30. Secuencia Consultar Permiso Versión Final

3.2.3.15. DIAGRAMA DE SECUENCIA 021: Crear Capacitación

(DS021)

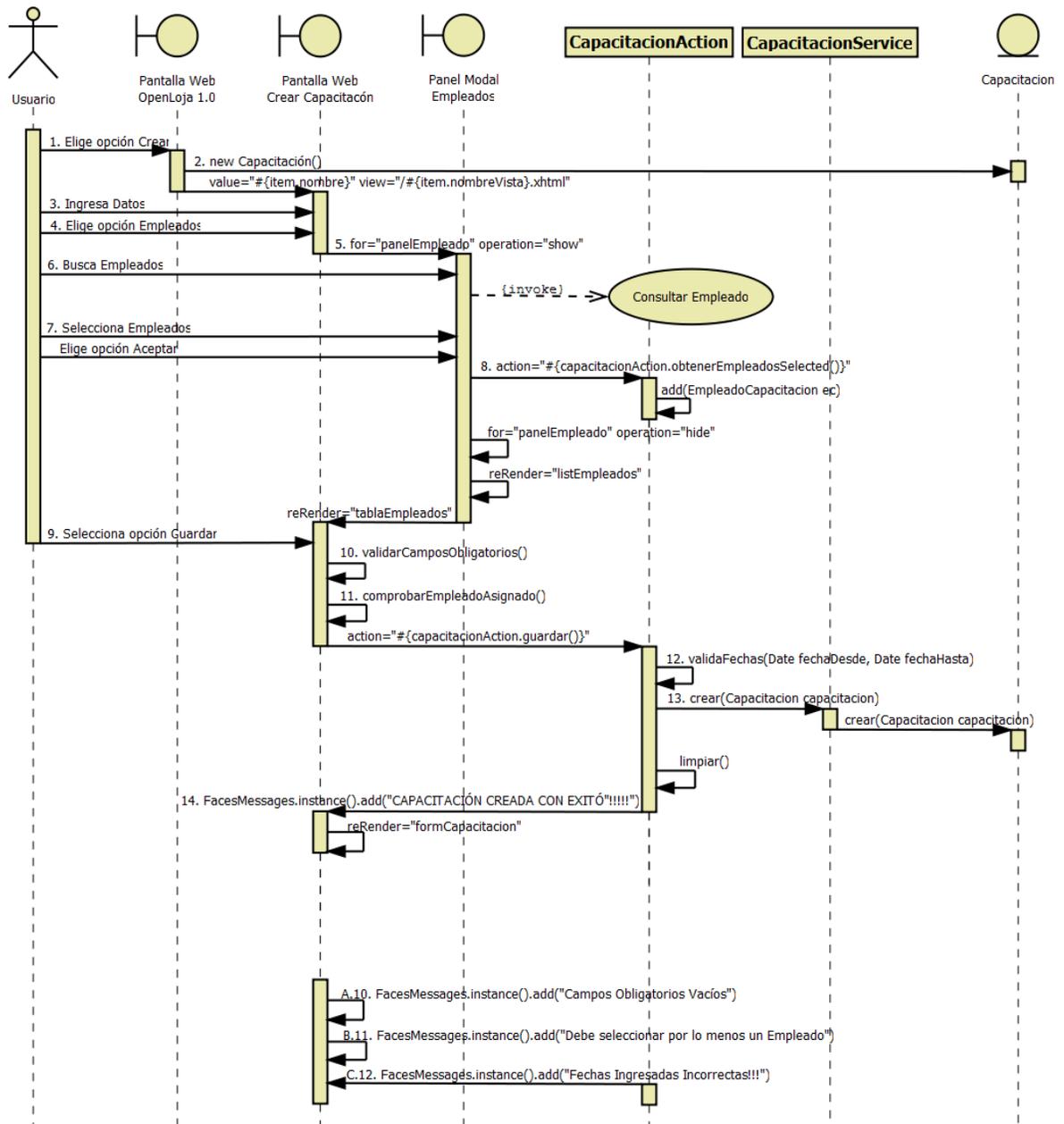


Diagrama 31. Secuencia Crear Capacitación Versión Final

3.2.3.16. DIAGRAMA DE SECUENCIA 022: Modificar Capacitación (DS022)

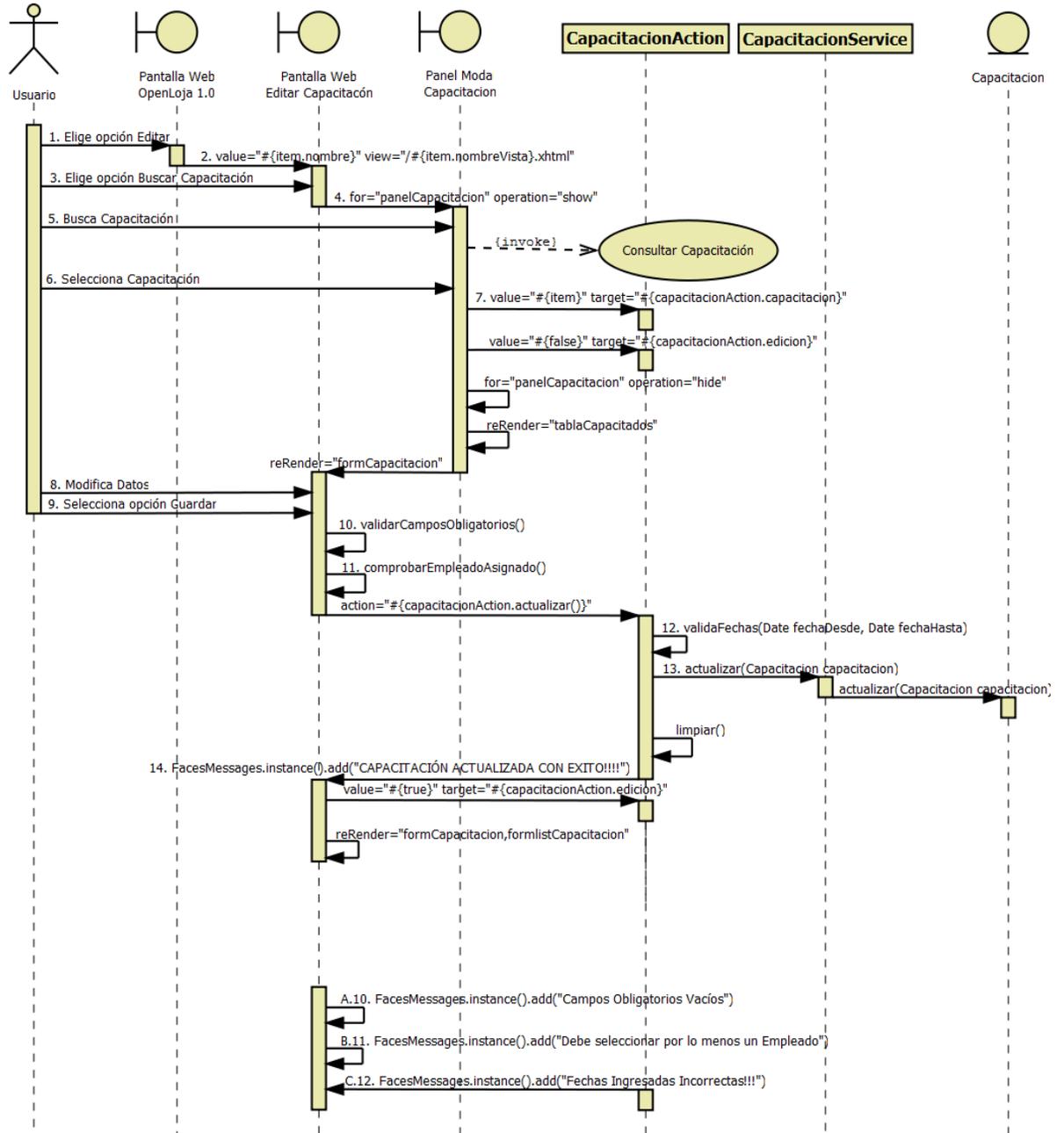


Diagrama 32. Secuencia Modificar Capacitación Versión Final

3.2.3.17. DIAGRAMA DE SECUENCIA 023: Consultar Capacitación (DS023)

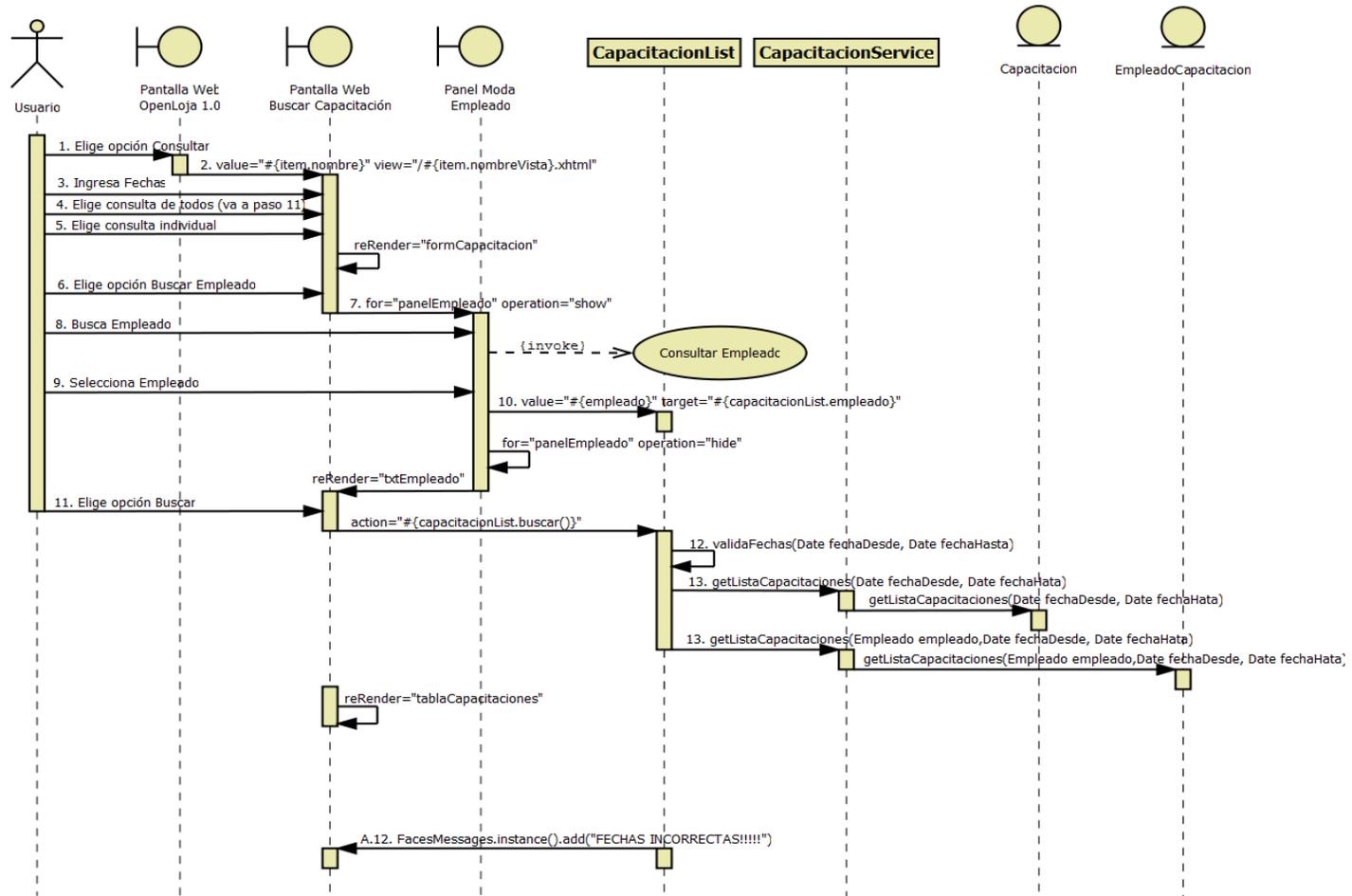


Diagrama 33. Secuencia Consultar Capacitación Versión Final

3.2.3.18. DIAGRAMA DE SECUENCIA 024: Crear Anticipo

(DS024)

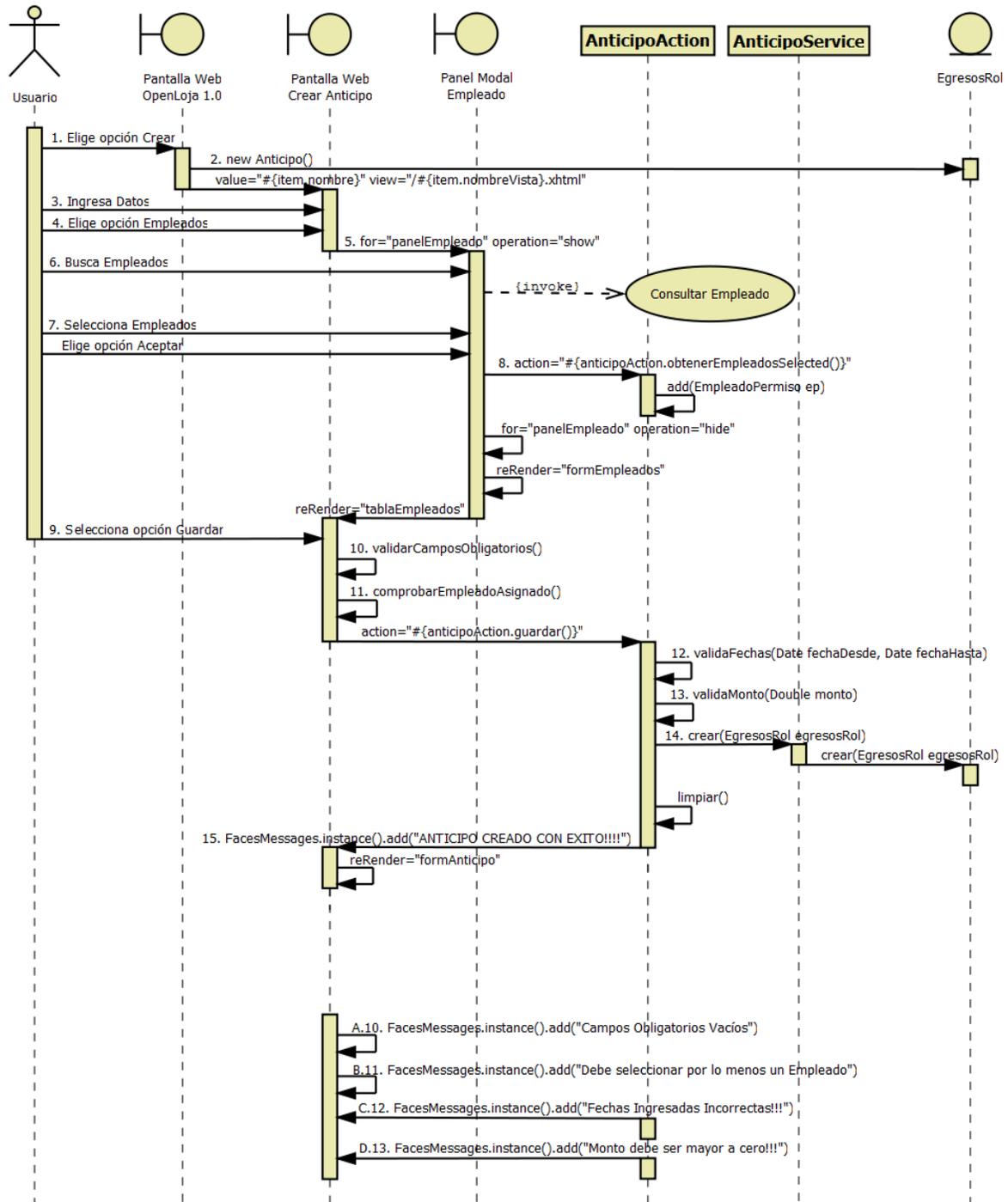


Diagrama 34. Secuencia Crear Anticipo Versión Final

3.2.3.19. DIAGRAMA DE SECUENCIA 025: Modificar Anticipo

(DS025)

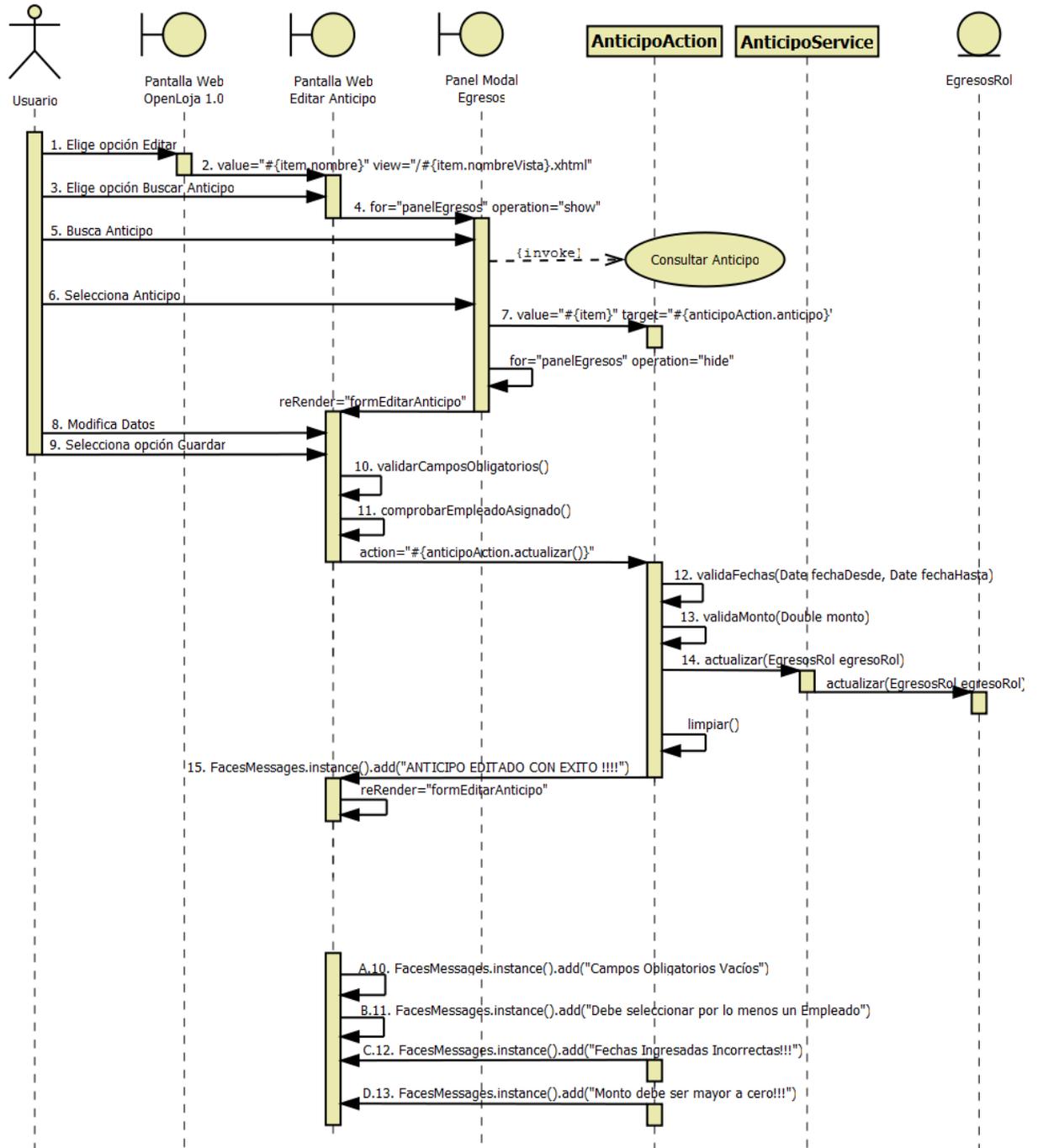


Diagrama 35. Secuencia Modificar Anticipo Versión Final

3.2.3.20. DIAGRAMA DE SECUENCIA 026: Consultar Anticipo (DS026)

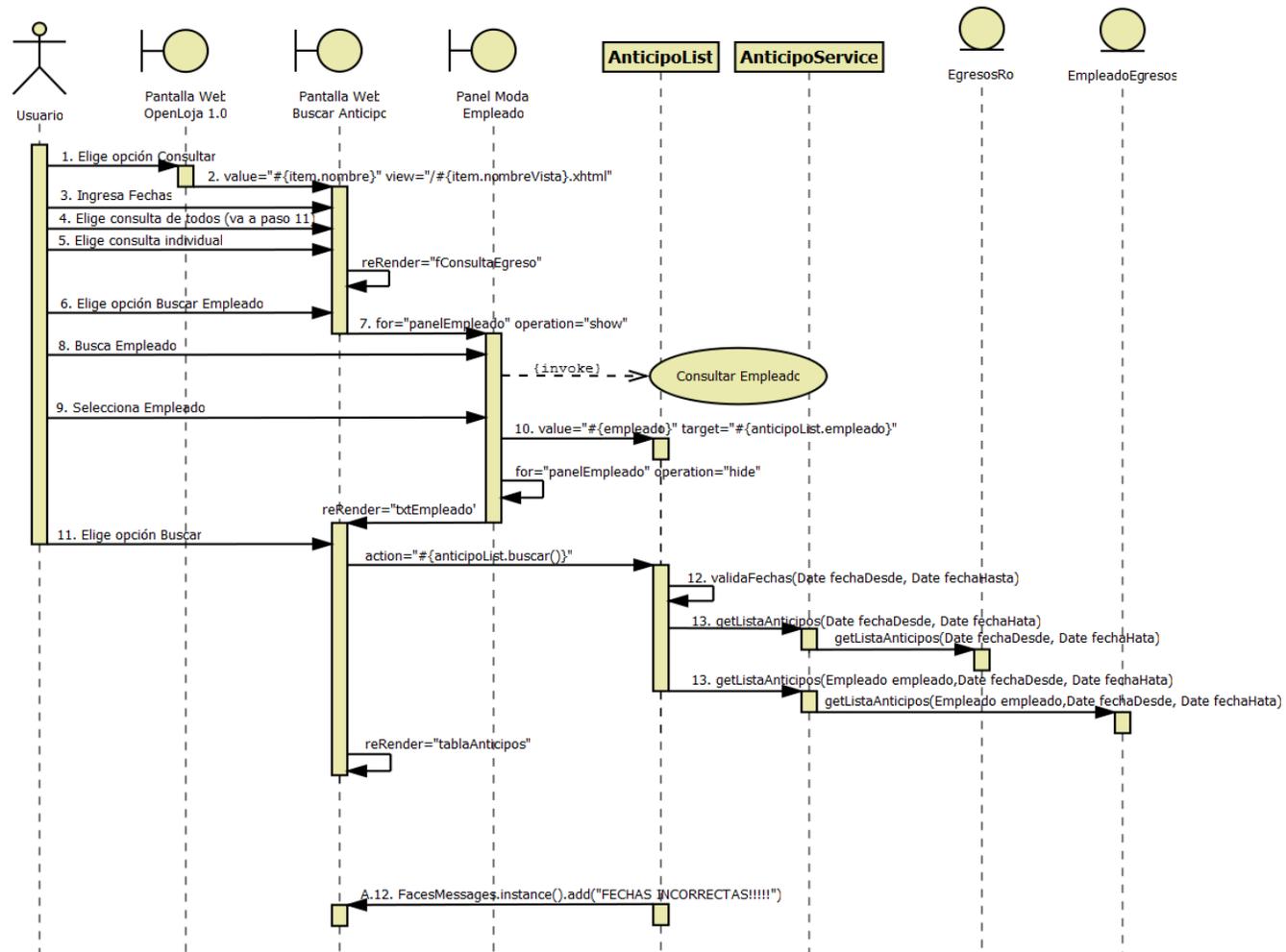


Diagrama 36. Secuencia Consultar Anticipo Versión Final

3.2.3.21. DIAGRAMA DE SECUENCIA 027: Generar Rol Pago Individual (DS027)

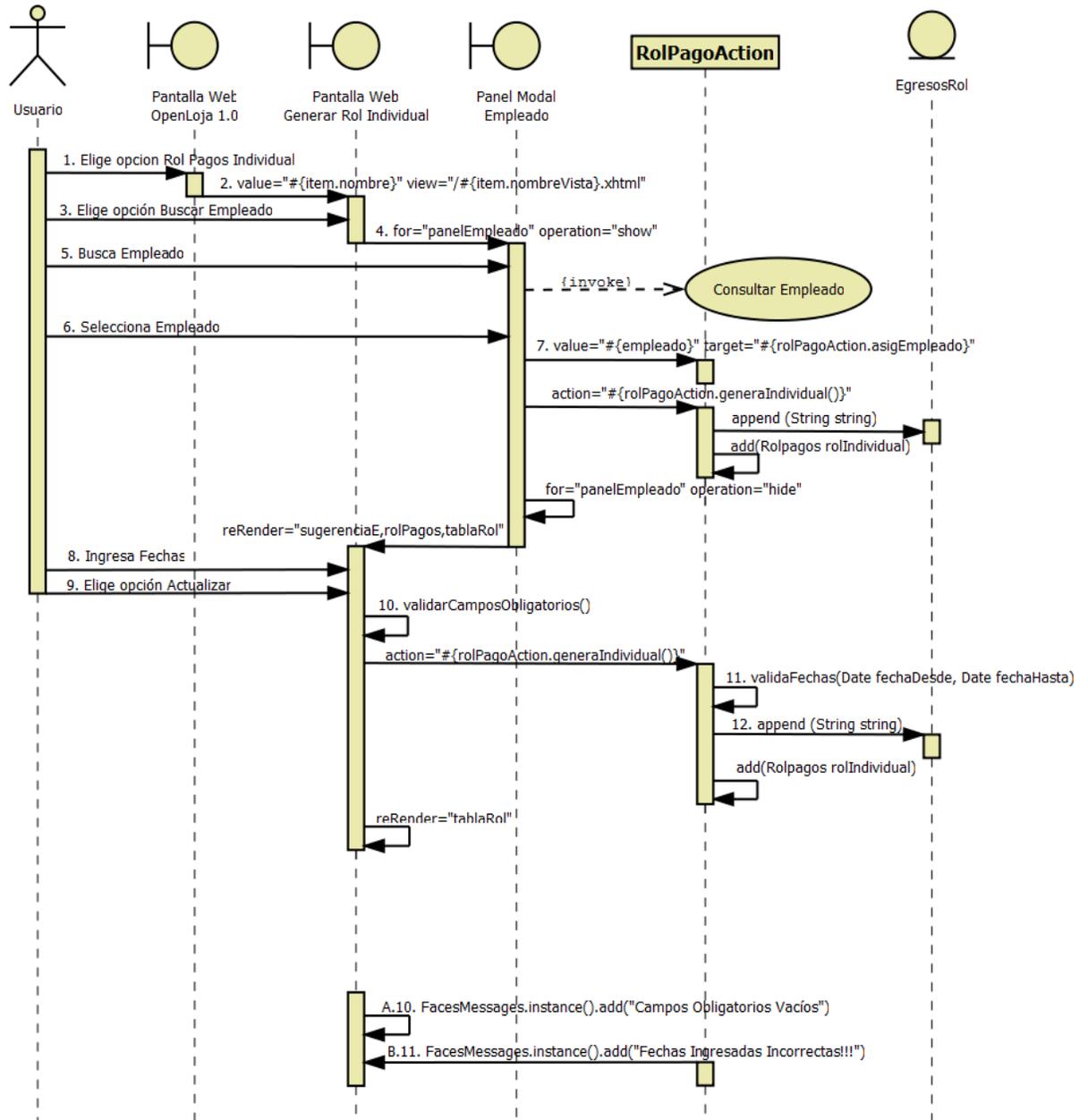


Diagrama 37. Secuencia Generar Rol Pago Individual Versión Final

3.2.3.22. DIAGRAMA DE SECUENCIA 028: Generar Rol General

(DS028)

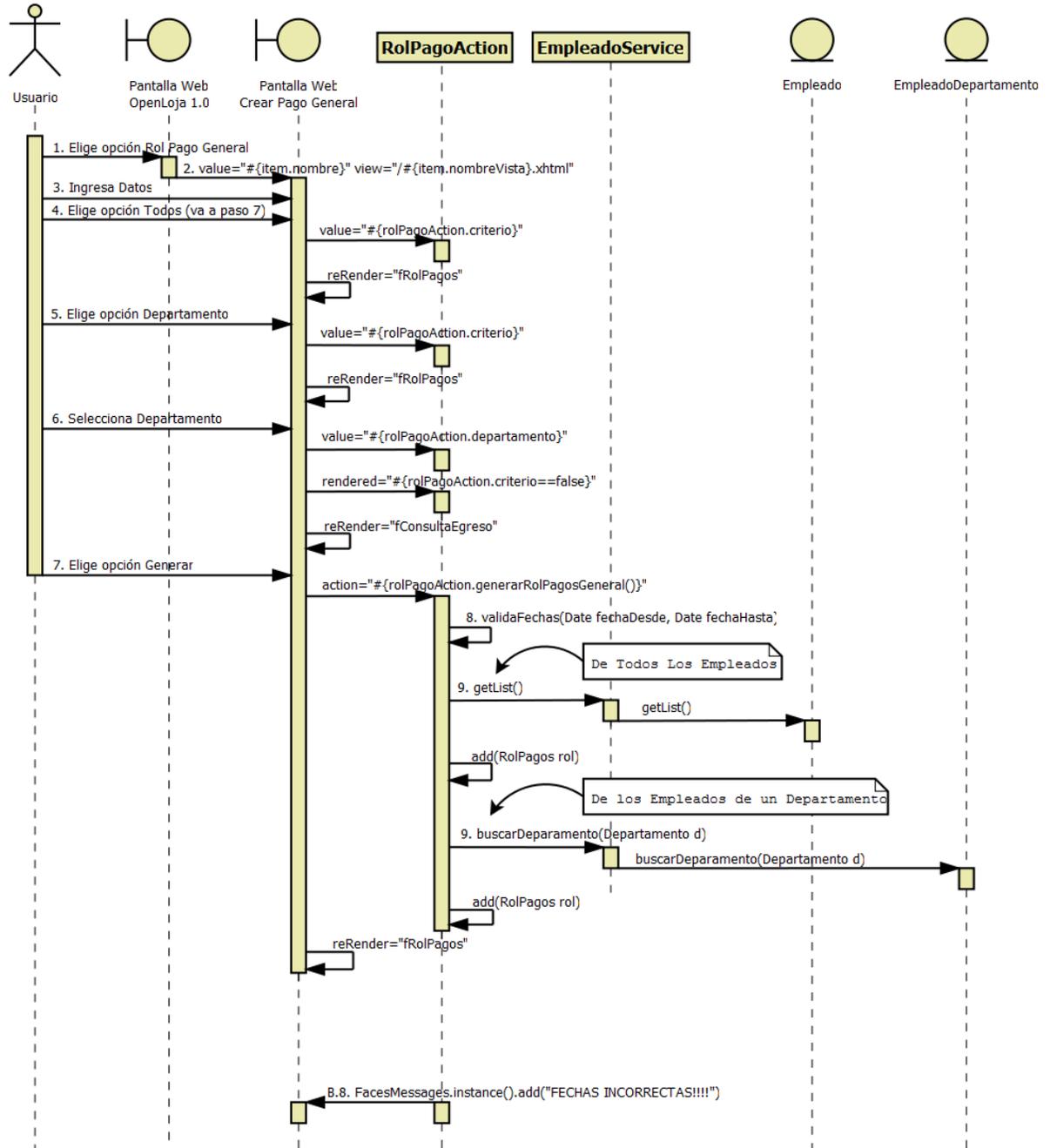


Diagrama 38. Secuencia Generar Rol Pago General Versión Final

3.2.4. DIAGRAMA DE CLASES FINAL

3.2.4.1. CLASES MODELO

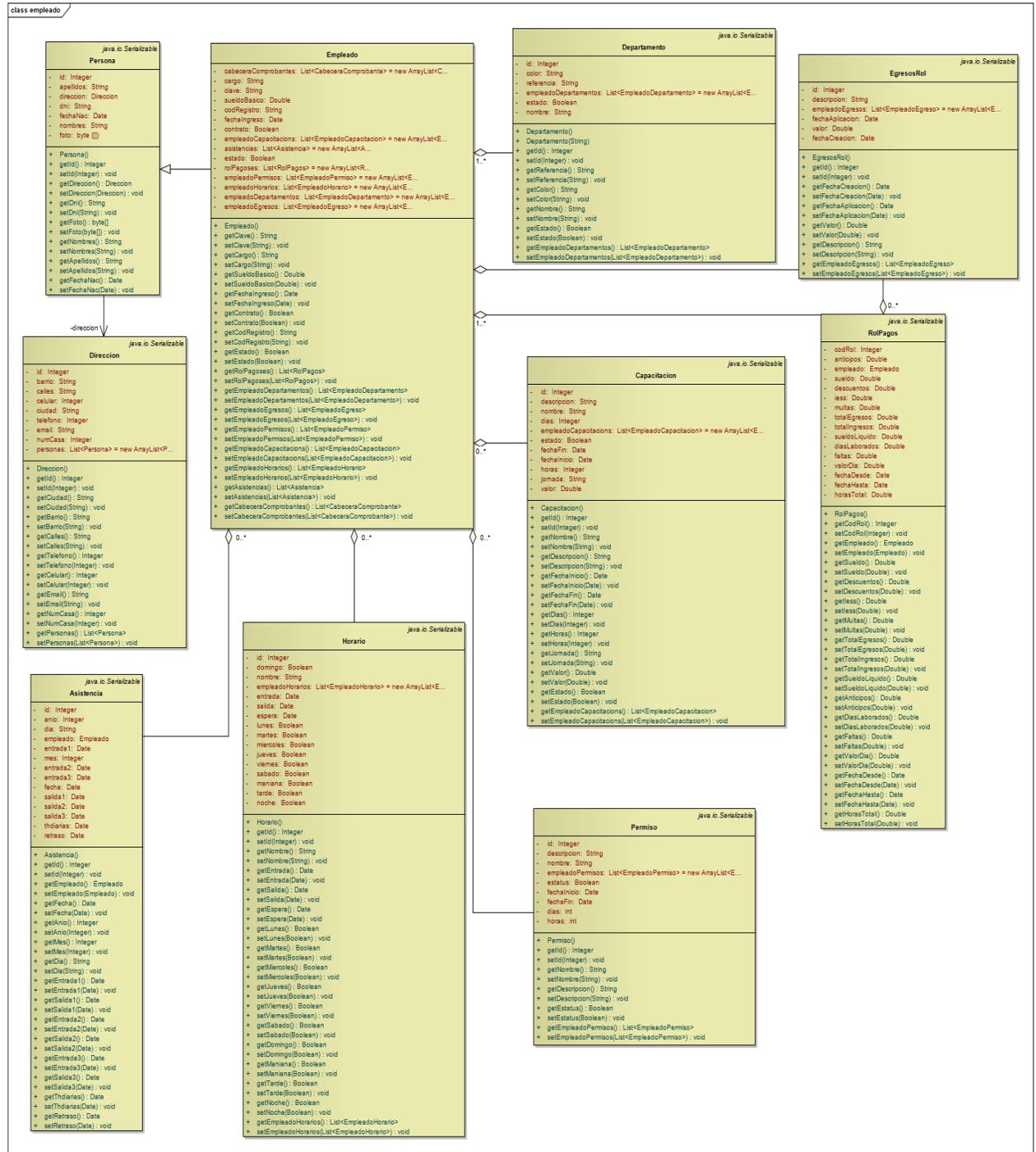


Diagrama 39. Clases Modelo: Gestión de Empleados Versión Final

3.2.4.2. CLASES DAO

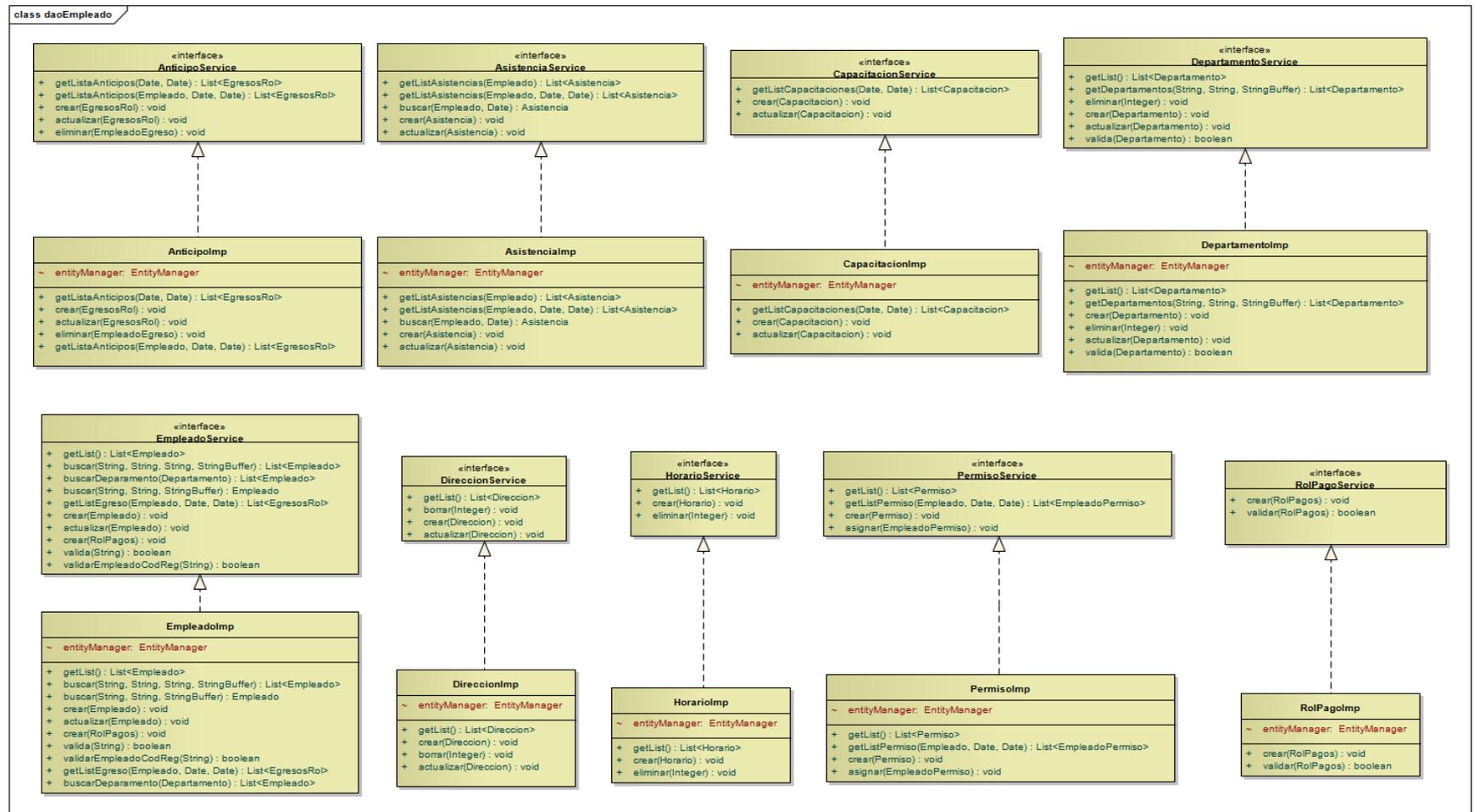


Diagrama 40. Clases DAO: Gestión de Empleados Versión Final

3.2.4.4. CLASES VISTAS

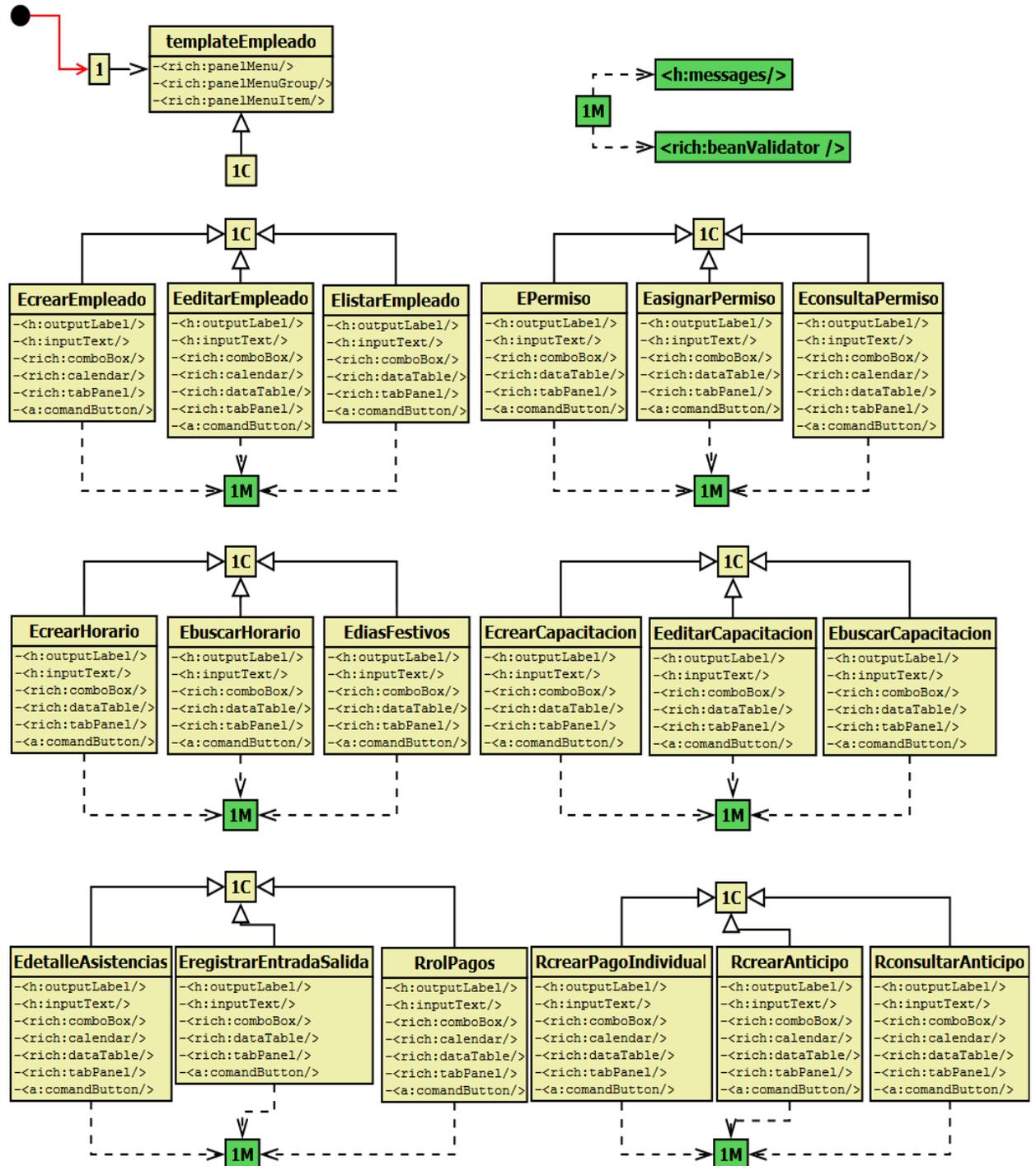


Diagrama 42. Vistas de Empleados Versión Final

3.2.5. DISEÑO DE DATOS

3.2.5.1. MODELO ENTIDAD-RELACIÓN

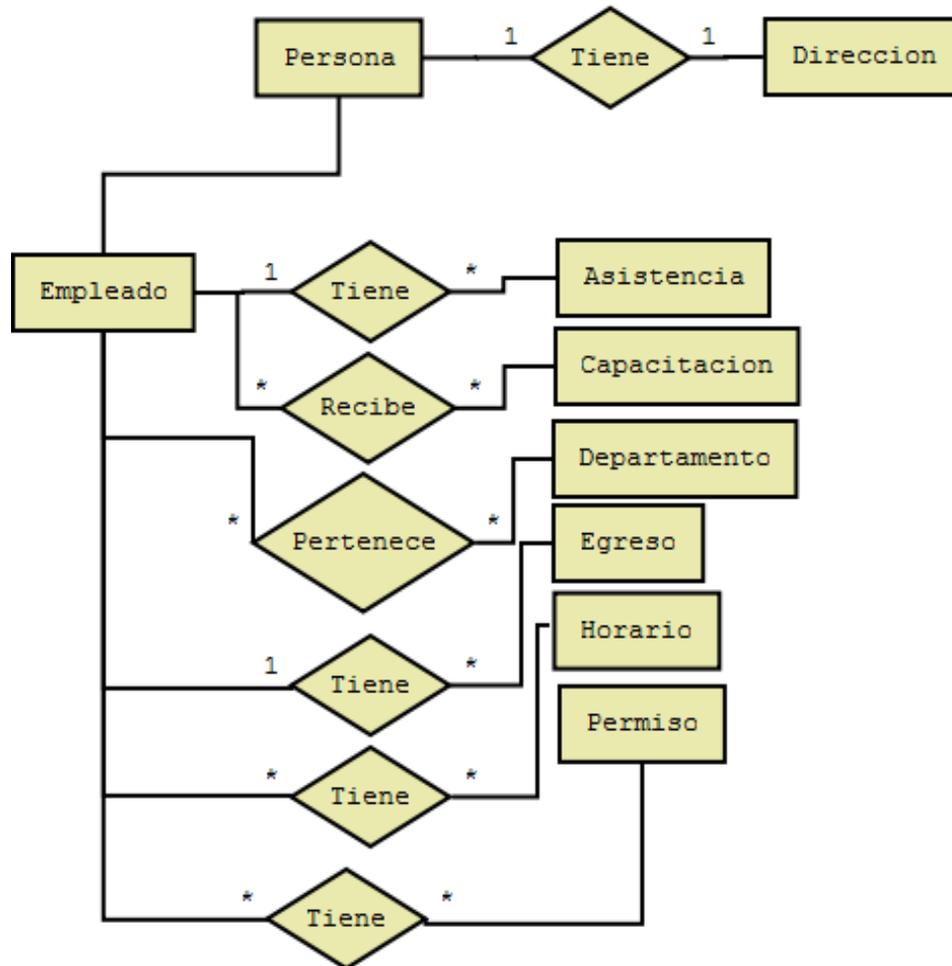


Diagrama 43. Modelo Entidad-Relación: Gestión de Empleado Versión Final

3.2.5.2. MODELO CONCEPTUAL

PERSONA

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
Dni	varchar	no		
Foto	blob	si		

Nombres	varchar	no		
Apellidos	varchar	no		
fechaNac	date	si		
idDireccion	Integer	no		

Tabla 87. Modelo Conceptual: Persona

DIRECCIÓN

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
Ciudad	varchar	no		
Barrio	varchar	si		
Calles	varchar	si		
Teléfono	Integer	no		
Celular	Integer	si		
Email	varchar	si		
numCasa	Integer	si		

Tabla 88. Modelo Conceptual: Dirección

EMPLEADO

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
Clave	varchar	no		
sueldoBasico	boolean	no		
fechaIngreso	varchar	si		
Contrato	boolean	no		
codRegistro	varchar	no		
Estado	boolean	no		

Tabla 89. Modelo Conceptual: Empleado

CAPACITACIÓN

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
Nombre	varchar	no		
descripcion	varchar	si		
fechaInicio	date	no		
fechaFin	date	no		
Días	Integer	si		
Horas	Integer	si		
Jornada	varchar	no		
Valor	double	si		
Estado	boolean	no		

Tabla 90. Modelo Conceptual: Capacitación

EMPLEADO_CAPACITACIÓN

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
codEmpleado	Integer	no		
codCapacitacion	Integer	no		

Tabla 91. Modelo Conceptual: Empleado_Capacitación

HORARIO

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
Nombre	varchar	no		
Entrada	time	no		
Salida	time	no		
Espera	time	no		
Lunes	boolean	si		
Martes	boolean	si		
Miércoles	boolean	si		
Jueves	boolean	si		
Viernes	boolean	si		
Sábado	boolean	si		
Domingo	boolean	si		

Tabla 92. Modelo Conceptual: Horario

EMPLEADO_HORARIO

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
codEmpleado	Integer	no		
codHorario	Integer	no		

Tabla 93. Modelo Conceptual: Empleado_Horario

DEPARTAMENTO

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
Referencia	varchar	no		
Nombre	varchar	no		
Código	varchar	no		

Tabla 94. Modelo Conceptual: Departamento

EMPLEADO_DEPARTAMENTO

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
codEmpleado	Integer	no		
codDepartamento	Integer	no		

Tabla 95. Modelo Conceptual: Empleado_Departamento

PERMISO

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
Nombre	varchar	no		
descripción	varchar	si		
Estatus	boolean	no		
fechaInicio	date	no		
fechaFin	date	no		
Días	Integer	si		
Horas	Integer	si		
Minutos	Integer	si		

Tabla 96. Modelo Conceptual: Permiso

EMPLEADO_PERMISO

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
codEmpleado	Integer	no		
codPermiso	varchar	no		

Tabla 97. Modelo Conceptual: Empleado_Permiso

ASISTENCIA

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
Fecha	date	no		
Anio	Integer	no		
Mes	Integer	no		
Dia	Integer	no		
entrada_1	time	no		
salida_1	time	no		
entrada_2	time	no		
salida_2	time	no		
entrada_3	time	no		
salida_3	time	no		
tHDiarias	time	no		
Retraso	time	si		
codEmpleado	Integer	no		

Tabla 98. Modelo Conceptual: Asistencia

EGRESOS_ROL

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
Nombre	varchar	no		
codEmpleado	Integer	no		
fechaCreación	date	no		

fechaAplicación	date	no		
Valor	double	no		
descripcion	varchar	si		

Tabla 99. Modelo Conceptual: Egresos_Rol

ROL_PAGOS

ATRIBUTO	TIPO	NULO	PK	PK
codRol	Integer	no		
Sueldo	double	no		
antiguedad	double	no		
descuentos	double	no		
less	double	no		
Multas	double	no		
totalEgreso	double	si		
totalIngreso	double	no		
sueldoLiquido	double	no		
Faltas	Integer	no		
Anticipo	double	no		
diasLaborados	double	no		
valorDia	double	no		
fechaDesde	date	no		
fechaHasta	date	no		
horasTotales	double	no		
codEmpleado	integer	no		

Tabla 100. Modelo Conceptual: Rol_pagos



GESTIÓN DE CLIENTES

3.3. MÓDULO DE GESTIÓN DE CLIENTES

3.3.1. MODELO DE DOMINIO

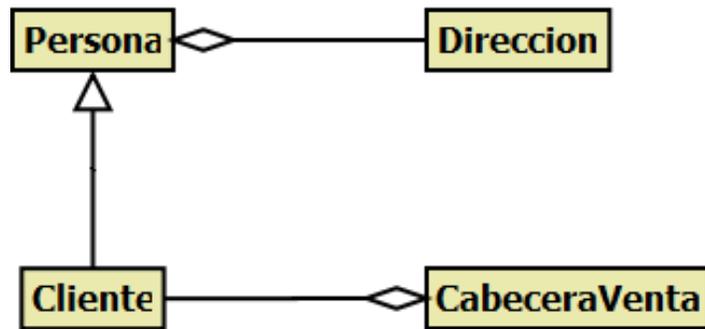


Diagrama 45. Módulo de Clientes Versión Final

3.3.2. MODELO DE CASOS USO DEL SISTEMA

3.3.2.1. DIAGRAMA DE CASOS DE USO

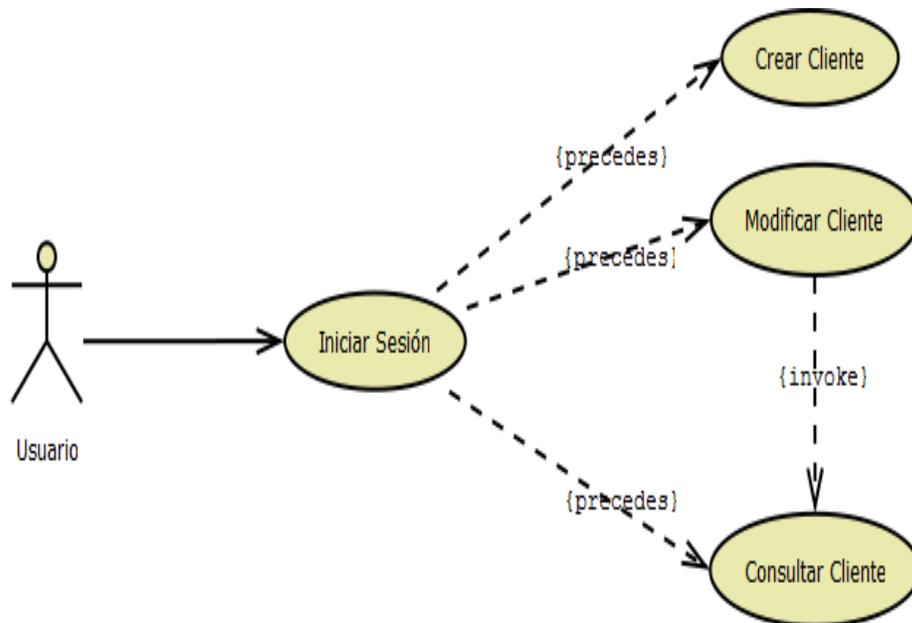


Diagrama 46. Gestión Cliente Versión Final

3.3.2.2. DESCRIPCIÓN DE CASOS DE USO

➤ CASO DE USO: Crear Cliente

NOMBRE: Crear Cliente		CÓDIGO	OP026
NOMBRE PROGRAMADO	CcrearCliente.xhtml		

GUARDAR BUSCAR BORRAR CANCELAR

DNI *

Nombres *

Apellidos *

creditoMax

Estado



Datos Domiciliario

Ciudad

Barrio

Calles

Número Casa

Telefono

Celular

Email

Tabla 101. Pantalla Crear Cliente

Nombre:	Crear Cliente	Código: 029
Requerimiento Funcional:	RF007	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado crear Clientes.	
Descripción:	Qué el usuario al cual se ha designado esta tarea o función pueda crear clientes cuando lo solicite o requiera.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Clientes e ítem Cliente.	
Poscondiciones:	Cliente creado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Crear] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea el Cliente y muestra la pantalla web "Crear Cliente". 3. El usuario ingresa los datos correspondientes al Cliente creado. 4. El usuario selecciona el botón [Guardar] de la pantalla web "Crear Cliente". 5. El sistema valida que los campos obligatorios no estén vacíos. 6. El sistema verifica mediante el DNI que el Cliente creado no coincida con otro ya 		

<p>existente.</p> <ol style="list-style-type: none"> 7. El sistema guarda los datos del nuevo Cliente. 8. El sistema muestra un mensaje de Cliente se ha creado con éxito. 9. El caso de uso finaliza.
FLUJO ALTERNO:
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <ol style="list-style-type: none"> A.5. El sistema muestra un mensaje de campos obligatorios vacíos. A.6. El caso de uso continúa en el paso 3 del flujo normal de evento. <p>B. CLIENTE EXISTENTE</p> <ol style="list-style-type: none"> B.6. El sistema muestra un mensaje de Cliente ya existe. B.7. El caso de uso continúa en el paso 3 del flujo normal de evento.

Tabla 102. Descripción Caso de Uso Crear Cliente

➤ **CASO DE USO: Modificar Cliente**

NOMBRE: Editar Cliente	CÓDIGO	OP027
NOMBRE PROGRAMADO	CeditarCliente.xhtml	

Tabla 103. Pantalla Editar Cliente

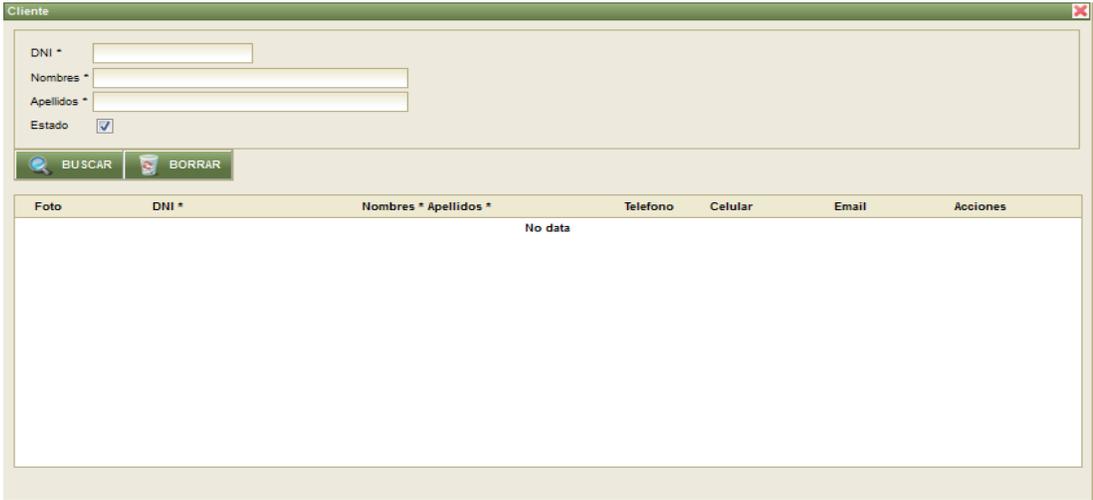
NOMBRE: Panel Modal Cliente		CÓDIGO	PM17
NOMBRE PROGRAMADO	buscarClienteEditar.xhtml		
			

Tabla 104. Panel Modal Cliente

Nombre:	Modificar Cliente	Código: 030
Requerimiento Funcional:	RF008	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado modificar un cliente previamente creado.	
Descripción:	Al usuario con esta función asignada pueda realizar modificaciones de clientes previamente creados.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Clientes e ítem Cliente.	
Poscondiciones:	Datos de Cliente seleccionado modificados.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Editar Cliente". 3. El usuario elige la opción [Buscar] de la pantalla web "Editar Cliente". 4. El sistema muestra el panel Modal "Cliente". 5. El usuario busca el Cliente que desea modificar para lo cual utiliza (invoca) el caso de uso <u>Consultar Cliente</u>. 6. El usuario selecciona el Cliente que busca del panel modal "Cliente". 7. El sistema carga los datos del Cliente seleccionado en la pantalla web "Editar Cliente" y automáticamente cierra el panel modal "Cliente". 8. El usuario modifica los datos del Cliente seleccionado. 9. El usuario elige la opción [Guardar] de la pantalla web "Editar Cliente". 		

10. El sistema valida que los campos obligatorios no estén vacíos.
11. El sistema verifica mediante el DNI que el Cliente modificado no coincida con otro ya existente.
12. El sistema guarda los nuevos datos del Cliente seleccionado para editar.
13. El sistema muestra un mensaje de Cliente se ha modificado con éxito.
14. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACIOS

- A.10. El sistema muestra mensaje en los campos obligatorios vacíos.
- A.11. El caso de uso continúa en el paso 3 del flujo normal de evento.

B. CLIENTE EXISTENTE

- B.11. El sistema muestra un mensaje de Cliente ya existe.
- B.12. El caso de uso continúa en el paso 10 del flujo normal de evento.

Tabla 105. Descripción Caso de Uso Modificar Cliente

➤ **CASO DE USO: Consultar Cliente**

NOMBRE: Buscar Cliente		CÓDIGO	OP028
NOMBRE PROGRAMADO	ClistarCliente.xhtml		

CREAR
EDITAR
CANCELAR

DNI *

Nombres *

Apellidos *

Estado

BUSCAR
BORRAR

Foto	DNI *	Nombres * Apellidos *	Telefono	Celular	Email	Accio
No data						

Tabla 106. Pantalla Buscar Cliente

Nombre:	Consultar Cliente		Código: 031
Requerimiento Funcional:	RF009		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario poder consultar un cliente previamente creado.		
Descripción:	Al usuario realizar consultas de clientes que han sido creados con anterioridad.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Clientes e ítem Cliente.		
Poscondiciones:	Listado de Clientes.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Consultar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Buscar Cliente". 3. El usuario ingresa los datos del Cliente que desea consultar. 4. El usuario elige la opción [Buscar] de la pantalla web "Buscar Cliente". 5. El sistema muestra los Clientes que coinciden con la búsqueda en la pantalla web "Buscar Cliente". 6. El caso de uso finaliza. 			

Tabla 107. Descripción Consultar Cliente

3.3.3. MODELO DE INTERACCIÓN

3.3.3.1. DIAGRAMA DE SECUENCIA 029: Crear Cliente

(DS029)

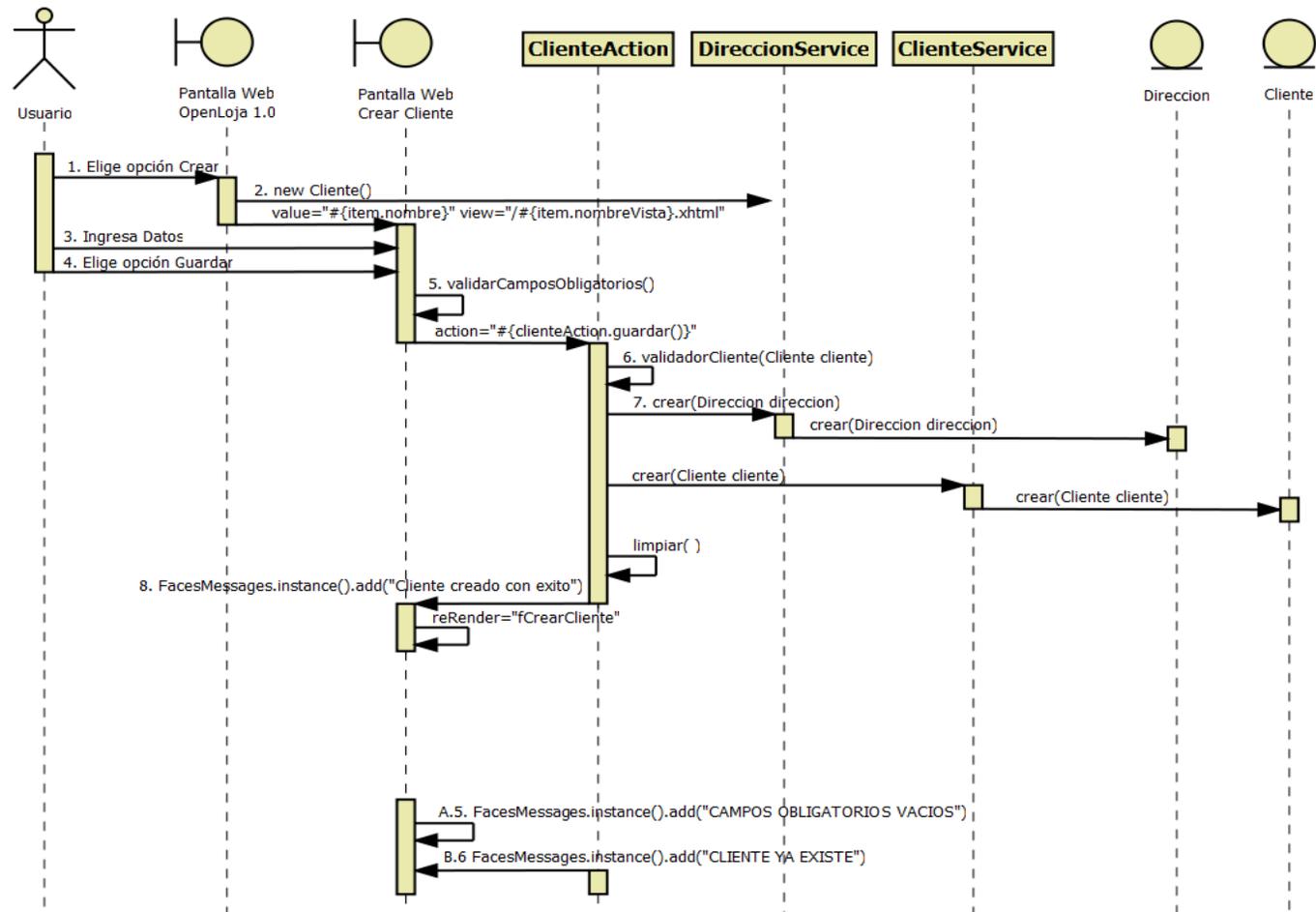


Diagrama 47. Secuencia Crear Cliente Versión Final

3.3.3.2. DIAGRAMA DE SECUENCIA 030: Modificar Cliente (DS030)

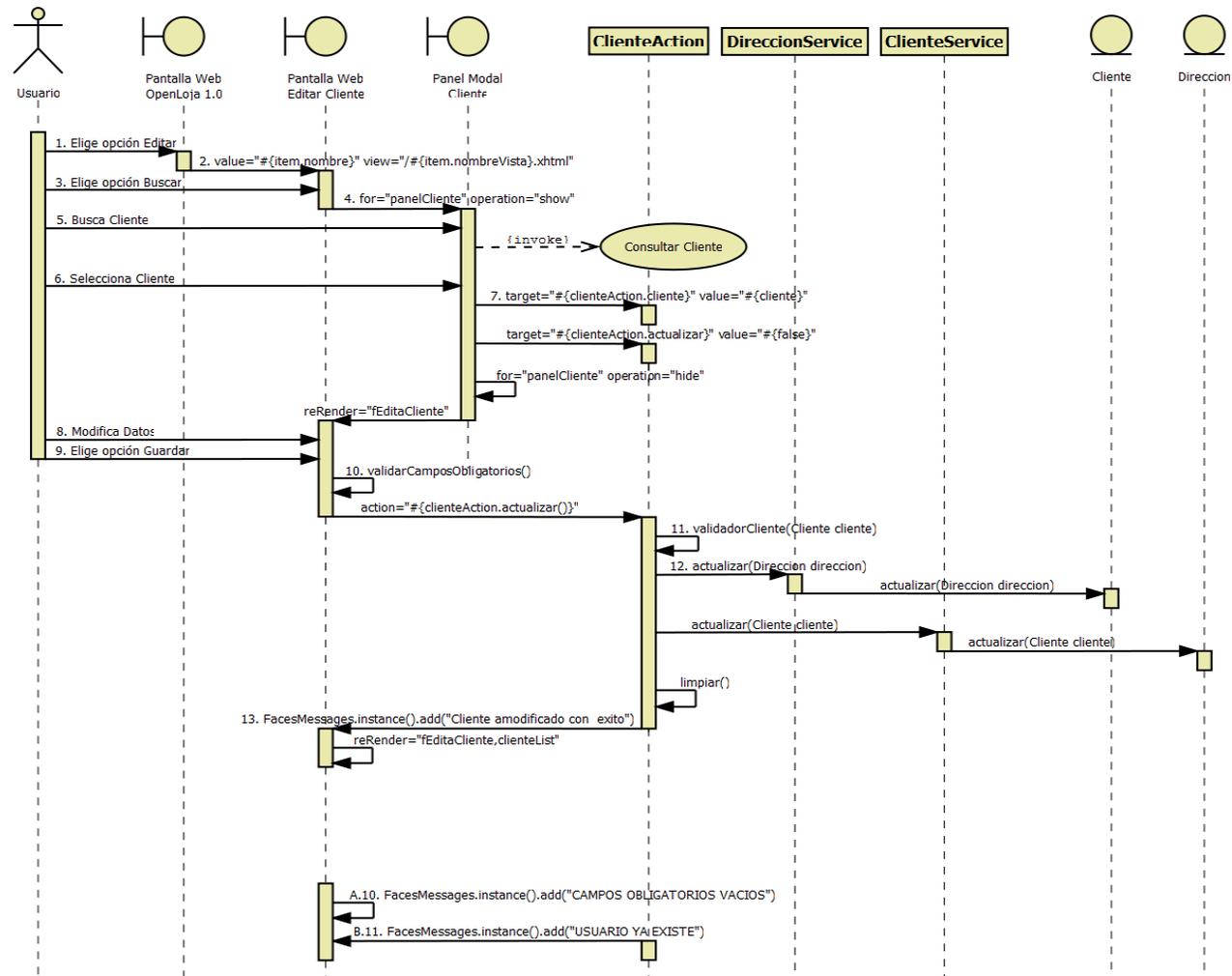


Diagrama 48. Secuencia Modificar Cliente Versión Final

3.3.3.3. DIAGRAMA DE SECUENCIA 031: Consultar Cliente (DS031)

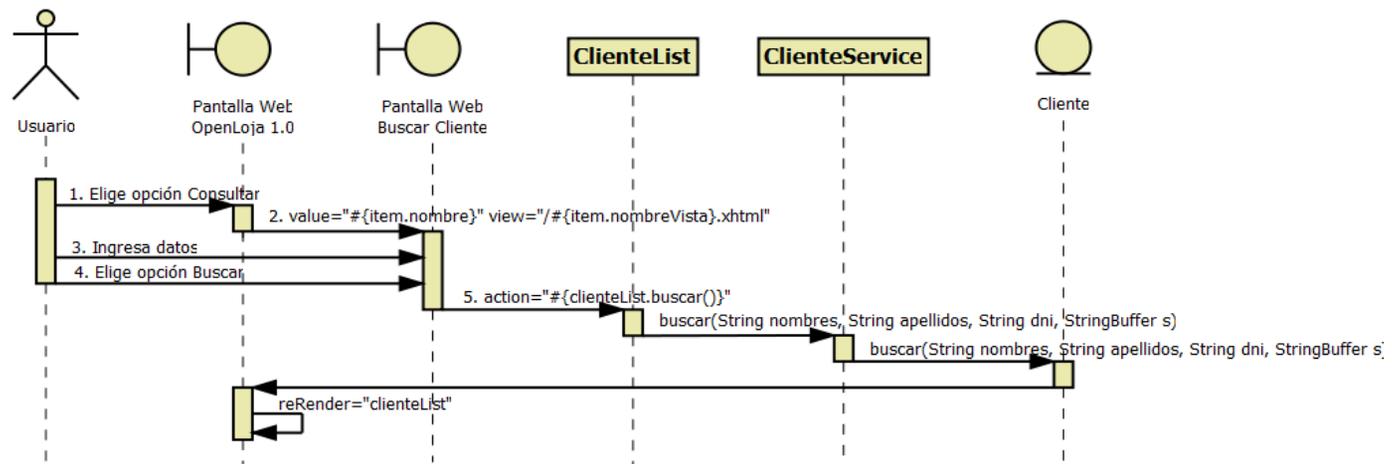


Diagrama 49. Secuencia Consultar Cliente Versión Final

3.3.4. DIAGRAMA DE CLASES FINAL

3.3.4.1. CLASES MODELO

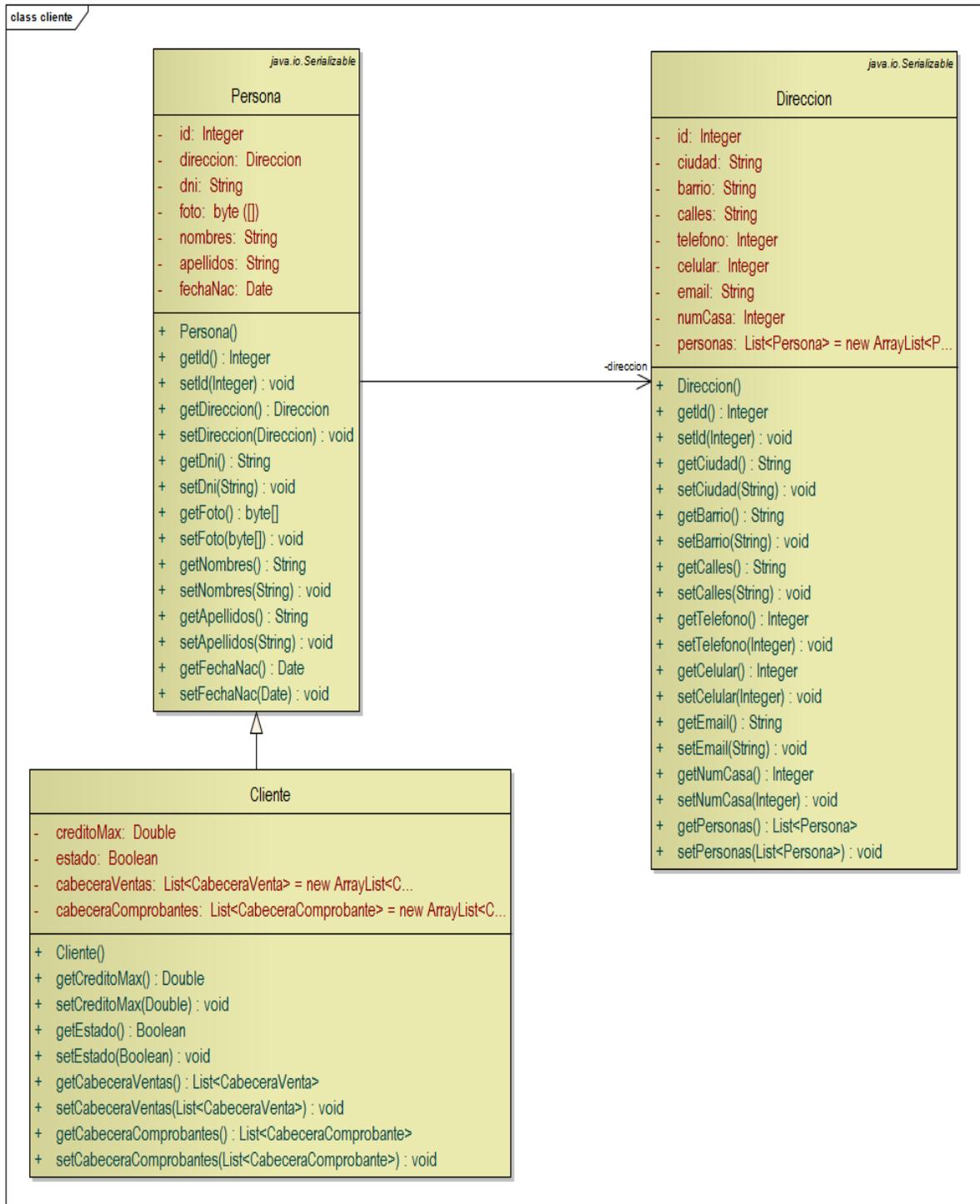


Diagrama 50. Clases Modelo: Gestión de Clientes Versión Final

3.3.4.2. CLASES DAO

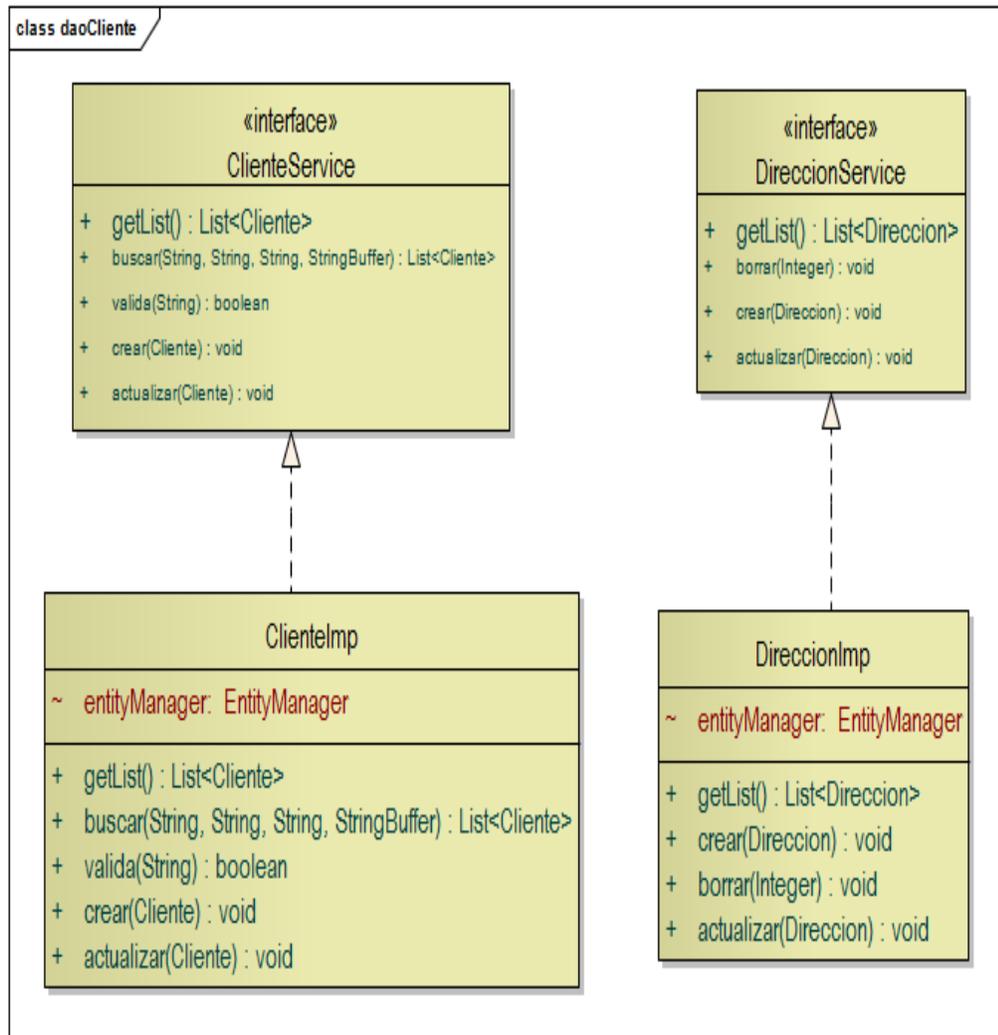


Diagrama 51. Clases DAO: Gestión de Clientes Versión Final

3.3.4.3. CLASES CONTROLADOR

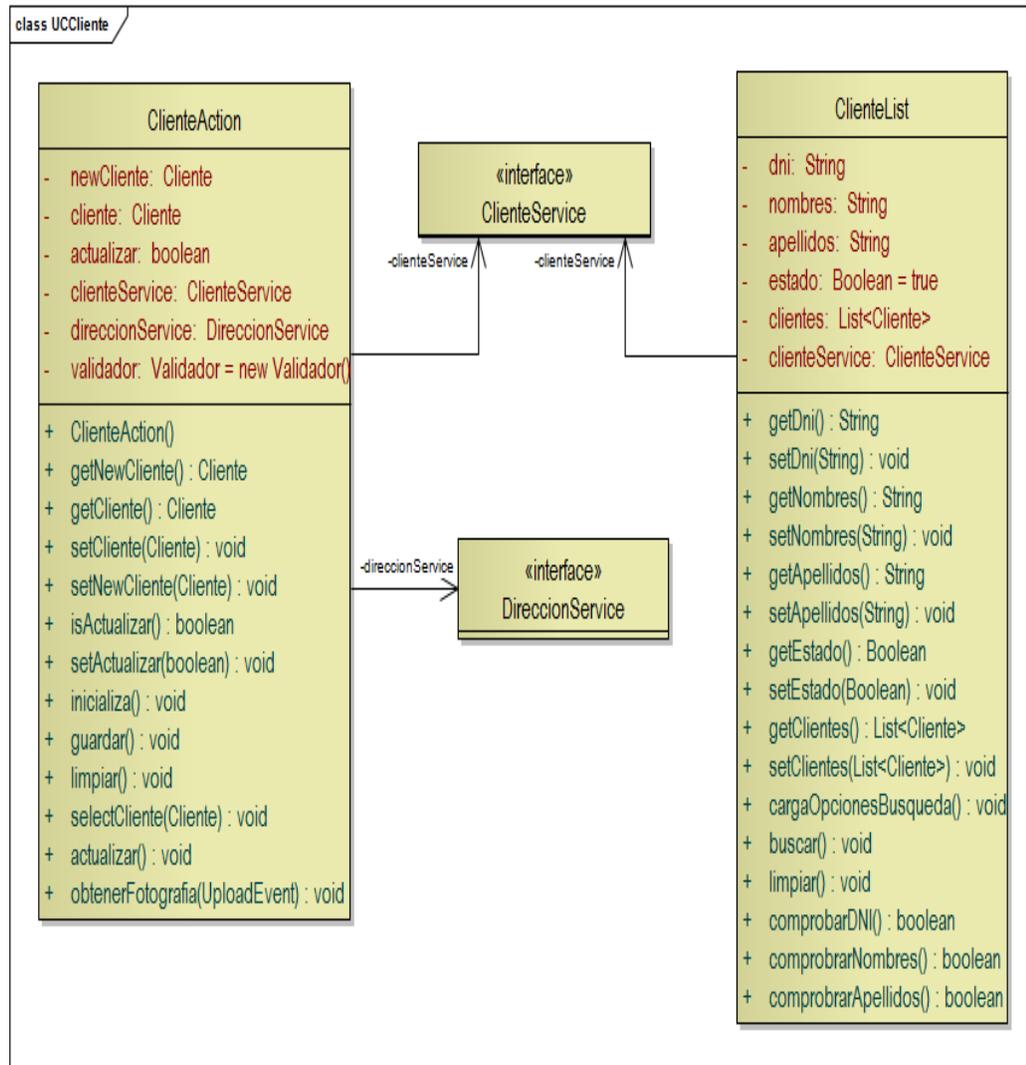


Diagrama 52. Clases Controlador: Gestión de Clientes Versión Final

3.3.4.4. CLASES VISTA

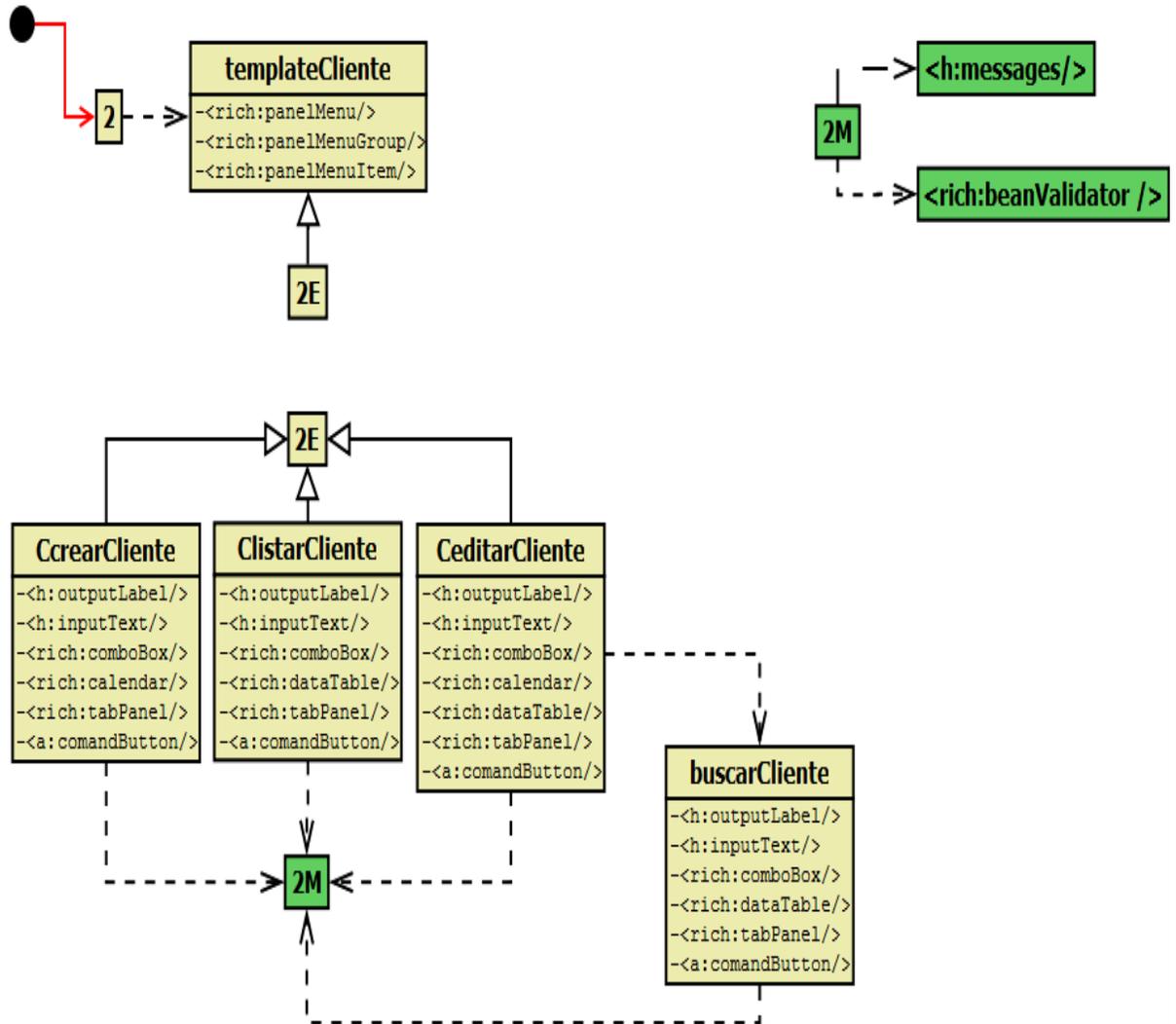


Diagrama 53. Vistas de Cliente Versión Final

3.3.5. DISEÑO DE DATOS

3.3.5.1. MODELO ENTIDAD-RELACIÓN

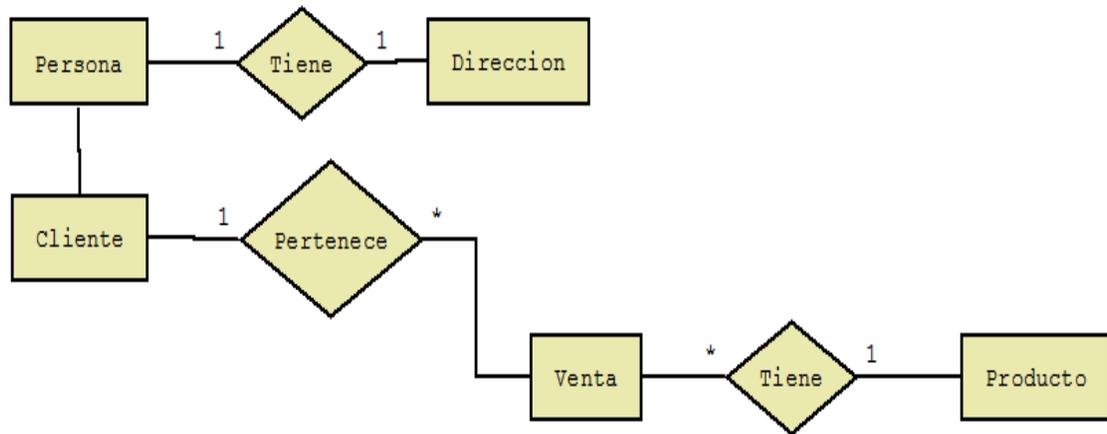


Diagrama 54. Modelo Entidad-Relación: Gestión de Clientes y Ventas Versión Final

3.3.5.2. MODELO CONCEPTUAL

PERSONA

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no	🔑	
Dni	varchar	no		
Foto	blob	si		
Nombres	varchar	no		
Apellidos	varchar	no		
fechaNac	date	si		
idDireccion	Integer	no		🔑

Tabla 108. Modelo Conceptual: Persona

DIRECCIÓN

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no	🔑	
Ciudad	varchar	no		
Barrio	varchar	si		

Tabla 109. Modelo Conceptual: Dirección

CLIENTE

ATRIBUTO	TIPO	NULO	PK	FK
Id	Integer	no		
creditoMax	double	no		
Estado	boolean	no		

Tabla 110. Modelo Conceptual: Cliente

3.3.5.3. MODELO RELACIONAL

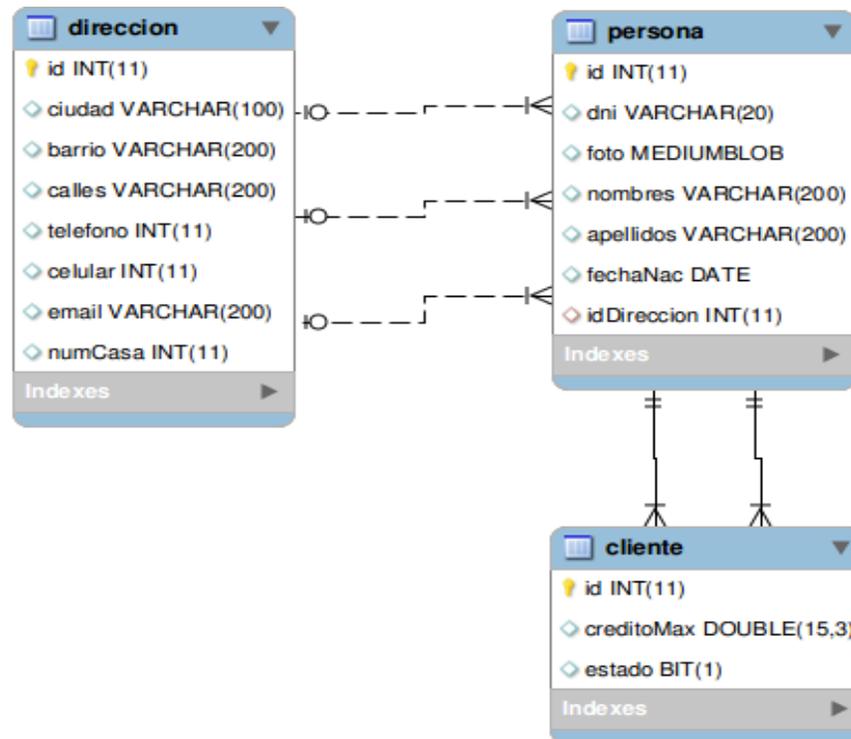


Diagrama 55. Modelo Relacional: Gestión de Clientes Versión Final



GESTIÓN DE PROVEEDORES

3.4. MÓDULO DE GESTIÓN DE PROVEEDORES

3.4.1. MODELO DE DOMINIO

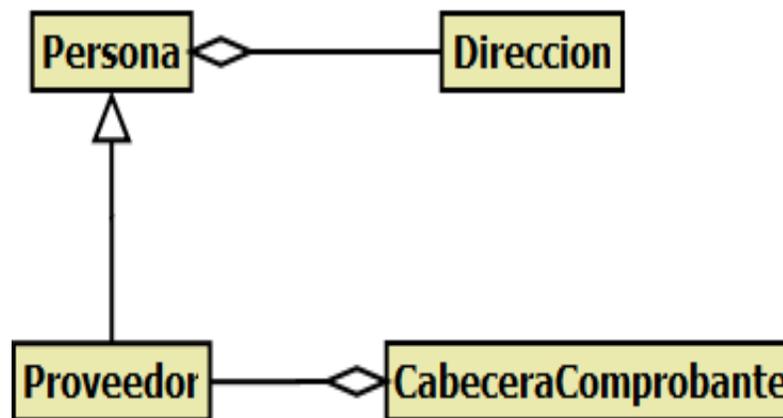


Diagrama 56. Módulo de Proveedores Versión Final

3.4.2. MODELO DE CASOS DE USO DEL SISTEMA

3.4.2.1. DIAGRAMA DE CASOS DE USO

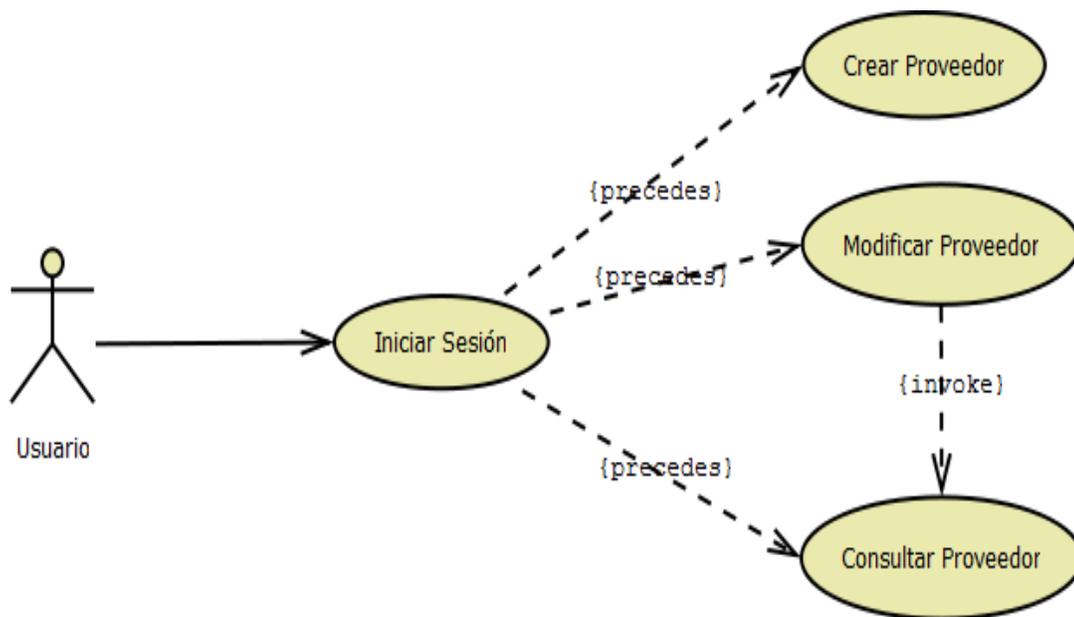


Diagrama 57.Gestión Proveedor Versión Final

3.4.2.2. DESCRIPCIÓN DE CASOS DE USO

➤ CASO DE USO: Crear Proveedor

NOMBRE: Crear Proveedor	CÓDIGO	OP029
NOMBRE PROGRAMADO	PcrearProveedor.xhtml	

GUARDAR
BORRAR
CANCELAR

DNI *

tipo: EMPRESA

Nombre* Comercial

Nombre* Fiscal

URL

Estado



datosDomiciliarios observaciones

Ciudad:

Barrio:

Calles:

Telefono:

Celular:

Email:

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com

Tabla 111. Pantalla Crear Proveedor

Nombre:	Crear Proveedor	Código: 032
Requerimiento Funcional:	RF007	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado pueda crear proveedores.	
Descripción:	Qué el usuario con esta función asignada pueda crear proveedores cuando este lo requiera o necesite.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Proveedores e ítem Proveedor.	
Poscondiciones:	Proveedor Creado.	
FLUJO NORMAL:		
1. El usuario elige la opción [Crear] de la pantalla web principal "OpenLoja 1.0".		

2. El sistema crea el Proveedor y muestra la pantalla web “Crear Proveedor”.
3. El usuario elige el Tipo de Proveedor (Lista desplegable Tipo) el cual puede ser una Empresa o Una Persona Natural en la pantalla web “Crear Proveedor”.
4. El sistema actualiza la pantalla web “Crear Proveedor” y muestra los datos a ingresar según el tipo de Proveedor que el usuario elige.
5. El usuario ingresa los datos correspondientes al Proveedor creado.
6. El usuario selecciona la opción [Guardar] de la pantalla web “Crear Proveedor”.
7. El sistema valida que los campos obligatorios no estén vacíos.
8. El sistema verifica mediante el DNI que el Proveedor creado no coincida con otro ya existente.
9. El sistema guarda los datos del nuevo Proveedor.
10. El sistema muestra un mensaje Proveedor se ha creado con éxito.
11. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACIOS

- A.7. El sistema muestra mensajes en los campos obligatorios vacíos.
- A.8. El caso de uso continúa en el paso 5 del flujo normal de evento.

B. PROVEEDOR EXISTENTE

- B.8. El sistema muestra un mensaje de Proveedor ya existe.
- B.9. El caso de uso continúa en el paso 5 del flujo normal de evento.

Tabla 112. Descripción Caso de Uso Crear Proveedor

➤ CASO DE USO: Modificar Proveedor

NOMBRE: Crear Proveedor	CÓDIGO	OP030
NOMBRE PROGRAMADO	PeditarProveedor.xhtml	

GUARDAR
BUSCAR
BORRAR
CANCELAR

DNI *

tipo

URL

Estado



datosDomiciliarios

Ciudad

Barrio

Calles

Telefono

Celular

eamil

Realizado por: Yazber Romero & Jorge Tapia yazb_37@hotmail.com

Tabla 113. Pantalla Modificar Proveedor

NOMBRE: Panel Modal Proveedor	CÓDIGO	PM18
NOMBRE PROGRAMADO:	buscarProveedor.xhtml	

PROVEEDOR
CANCELAR

DNI *

nombreComercial

nombreFiscal

tipo

Estado



datosDomiciliarios

Ciudad

Barrio

Calles

Telefono

Celular

eamil

Foto	DNI * *	nombreComercial *	nombreFiscal *	Email	Telefono	Celular	Acciones
No data							

Tabla 114. Panel Modal Proveedor

Nombre:	Modificar Proveedor		Código: 033
Requerimiento Funcional:	RF007		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario con este rol asignado modificar un Proveedor previamente creado.		
Descripción:	Al usuario que se le ha designado esta función realizar modificaciones de proveedores previamente creados.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Proveedores e ítem Proveedor.		
Poscondiciones:	Proveedor seleccionado editado.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla "Editar Proveedor". 3. El usuario selecciona la opción [Buscar] de la pantalla web "Buscar Proveedor". 4. El sistema muestra el panel modal "Proveedor". 5. El usuario busca el Proveedor que desea modificar para lo cual utiliza (invoca) el caso de uso <u>Consultar Proveedor</u>. 6. El usuario selecciona el Proveedor que busca del panel modal "Proveedor". 7. El sistema carga los datos del Proveedor seleccionado en la pantalla web "Editar Proveedor" y automáticamente cierra el panel modal "Proveedor". 8. El usuario modifica los datos del Proveedor seleccionado. 9. El usuario elige la opción [Guardar] de la pantalla web "Editar Proveedor". 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica mediante el DNI que el Proveedor modificado no coincida con otro ya existente. 12. El sistema guarda los nuevos datos del Proveedor seleccionado para editar. 13. El sistema muestra un mensaje de Proveedor se ha modificado con éxito. 14. El caso de uso finaliza. 			
FLUJO ALTERNO:			
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.10. El sistema muestra un mensaje de campos obligatorios vacíos.</p> <p>A.11. El caso de uso continúa en el paso 10 del flujo normal de evento.</p> <p>B. PROVEEDOR EXISTENTE</p> <p>B.11. El sistema muestra un mensaje de Proveedor ya existe.</p> <p>B.12. El caso de uso continúa en el paso 10 del flujo normal de evento.</p>			

Tabla 115. Descripción Caso de Uso Modificar Proveedor

➤ CASO DE USO: Consultar Proveedor

NOMBRE: Buscar Proveedor	CÓDIGO	OP031
NOMBRE PROGRAMADO:	PlistarProveedor.xhtml	

GUARDAR
BORRAR
CANCELAR

DNI *

nombreComercial

nombreFiscal

tipo

Estado

BUSCAR
BORRAR

Foto	DNI * *	nombreComercial *	nombreFiscal *	Email	Telefono	Celular
No data						

Realizado por: Yazber Romero & Jorge Tapia
yazb_87@hotmail.com

Tabla 116. Pantalla Consultar Proveedor

Nombre:	Consultar Proveedor	Código: 034
Requerimiento Funcional:	RF08	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado consultar proveedores.	
Descripción:	Al usuario con esta función otorgada realizar consultas de proveedores previamente creados.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Proveedores e ítem Proveedor.	
Poscondiciones:	Listado de Proveedores.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Consultar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Buscar Proveedor". 3. El usuario elige el <u>tipo de Proveedor</u> (Lista desplegable Tipo) que va a buscar el cual puede ser una Empresa o Una Persona Natural de la pantalla web "Buscar 		

Proveedor”.

4. El sistema actualiza la pantalla web “Buscar Proveedor” y muestra los datos a ingresar según el tipo de Proveedor que el usuario elige.
5. El usuario ingresa los datos correspondientes al Proveedor que desea consultar.
6. El usuario elige la opción [Buscar] de la pantalla web “Buscar Proveedor”.
7. El sistema muestra las coincidencias de la búsqueda en la Tabla Proveedores de la pantalla web “Buscar Cliente”.
8. El caso de uso finaliza.

Tabla 117. Descripción Caso de Uso Consultar Proveedor

3.4.3. MODELO DE INTERACCIÓN

3.4.3.1. DIAGRAMA DE SECUENCIA 032: Crear Proveedor (DS032)

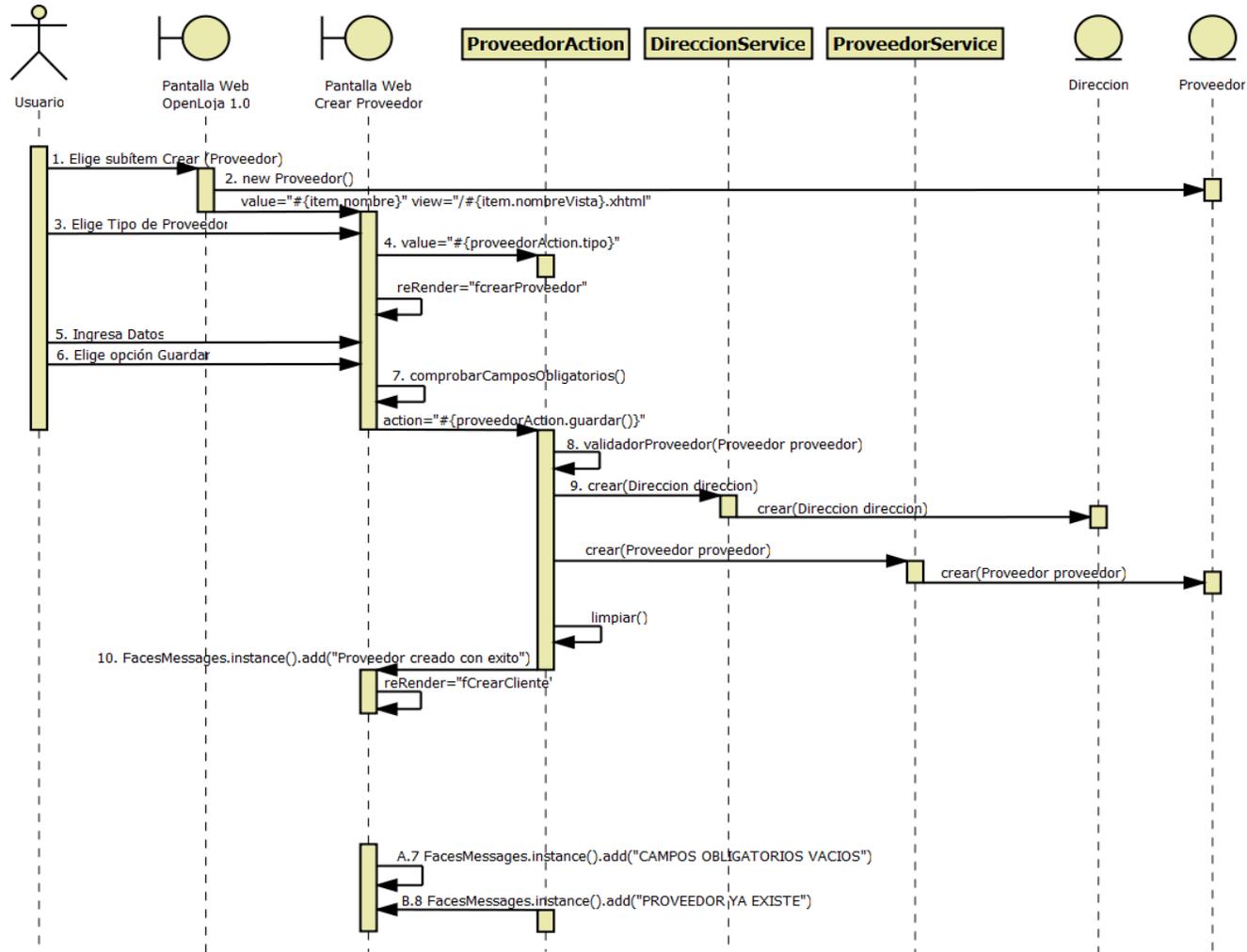


Diagrama 58. Secuencia Crear Proveedor Versión Final

3.4.3.2. DIAGRAMA DE SECUENCIA 033: Modificar Proveedor (DS033)

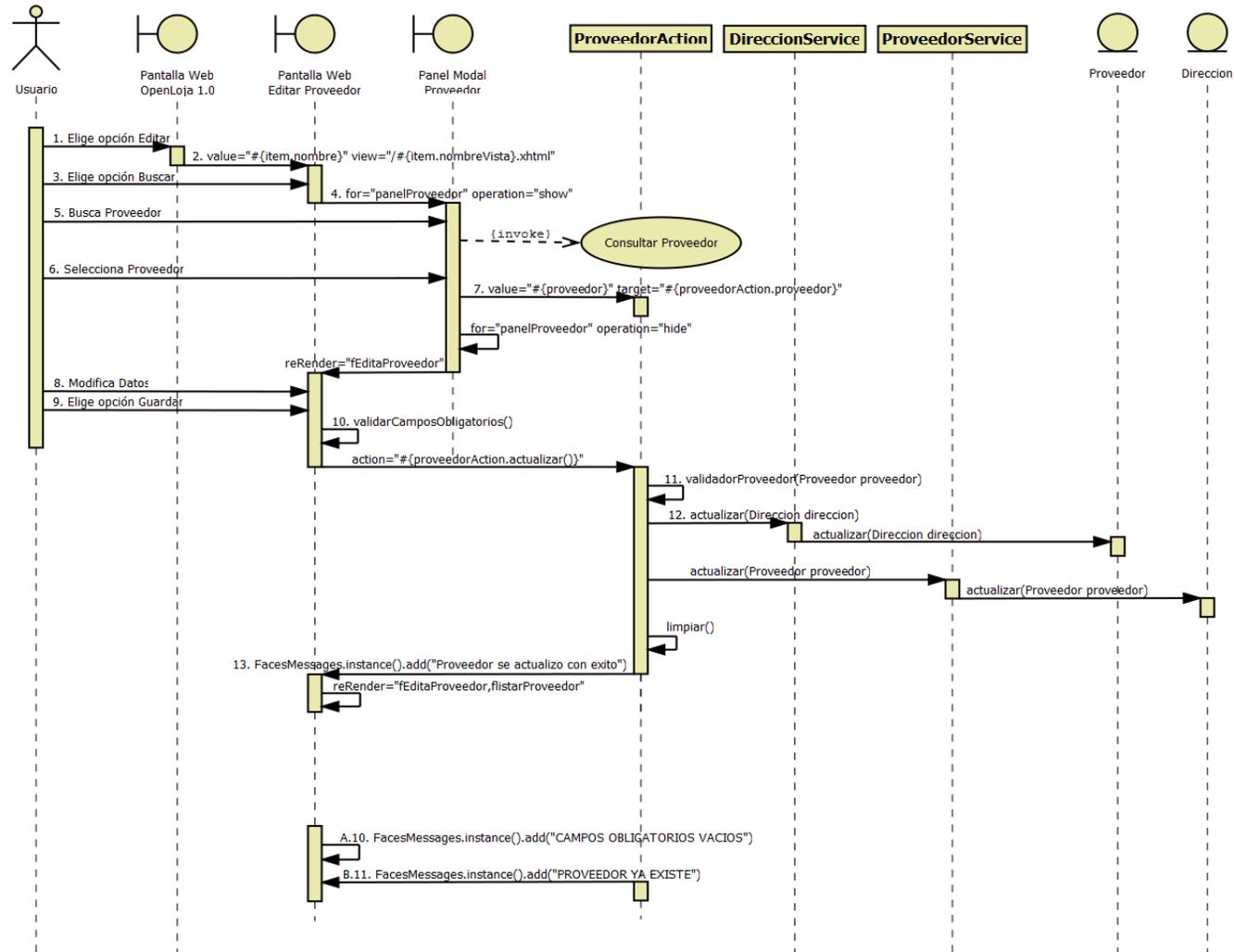


Diagrama 59. Secuencia Modificar Proveedor Versión Final

3.4.3.3. DIAGRAMA DE SECUENCIA 034: Consultar Proveedor (DS034)

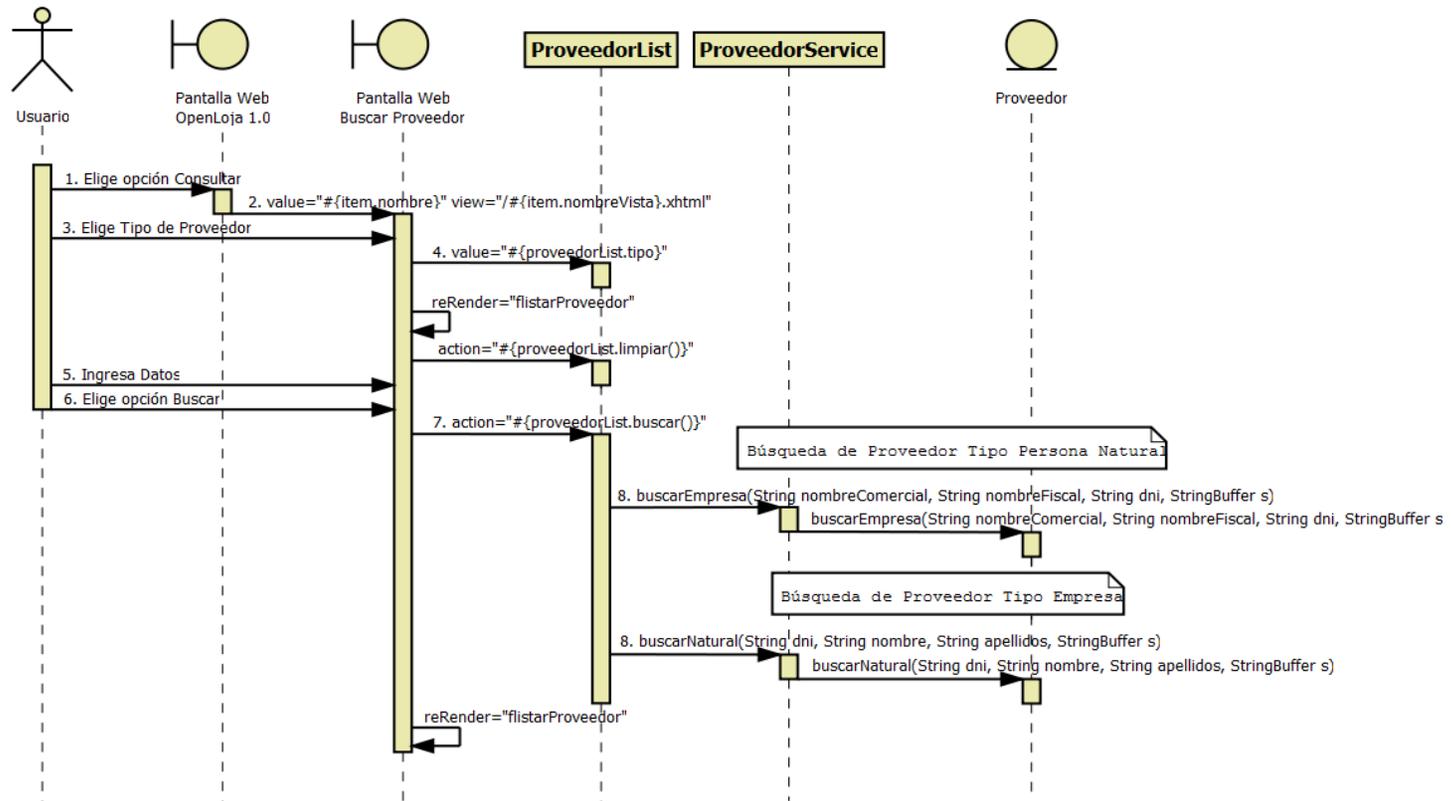


Diagrama 60. Secuencia Consultar Proveedor Versión Final

3.4.4. DIAGRAMA DE CLASES FINAL

3.4.4.1. CLASES MODELO

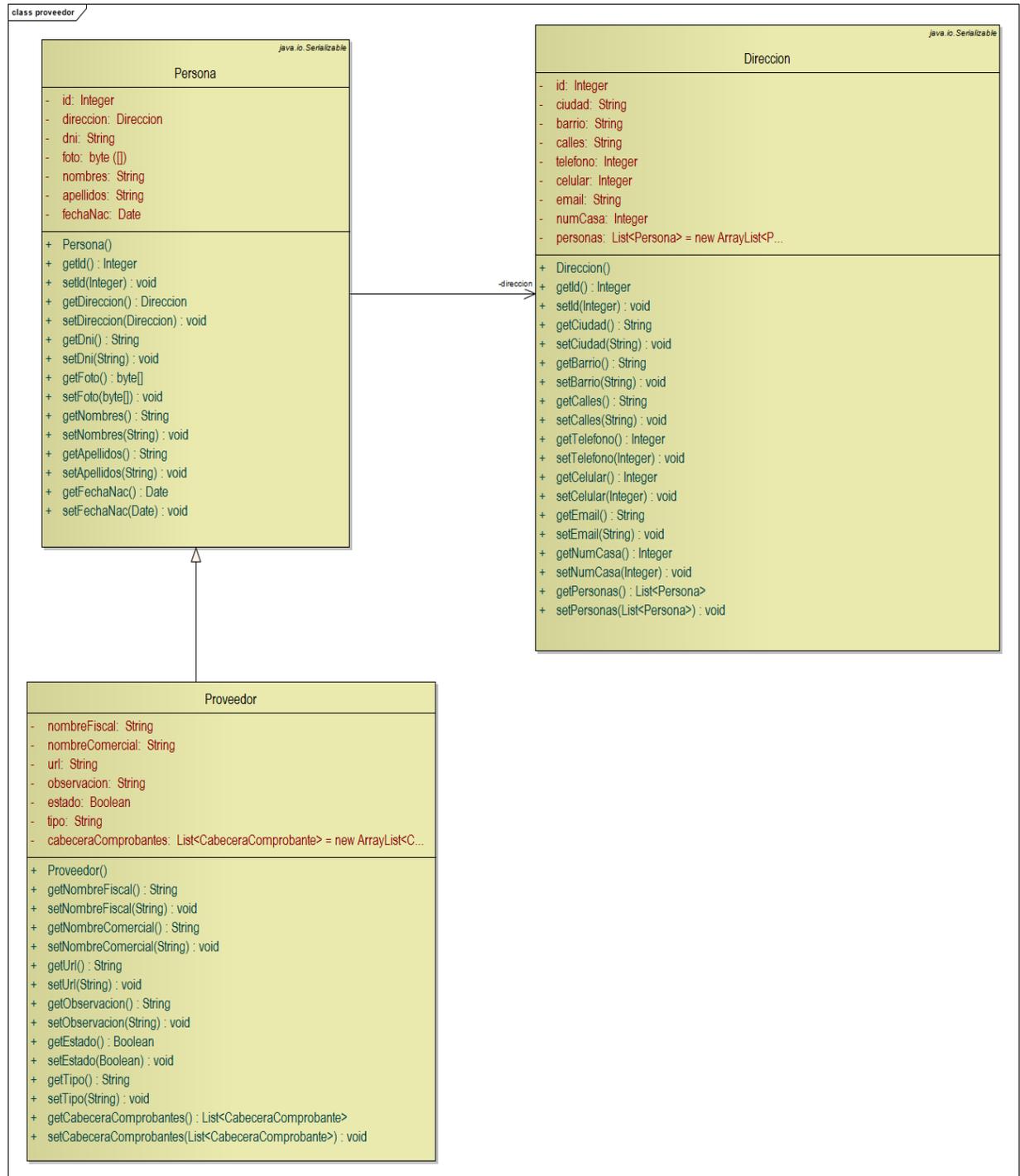


Diagrama 61. Clases Modelo: Gestión de Proveedores Versión Final

3.4.4.2. CLASES DAO

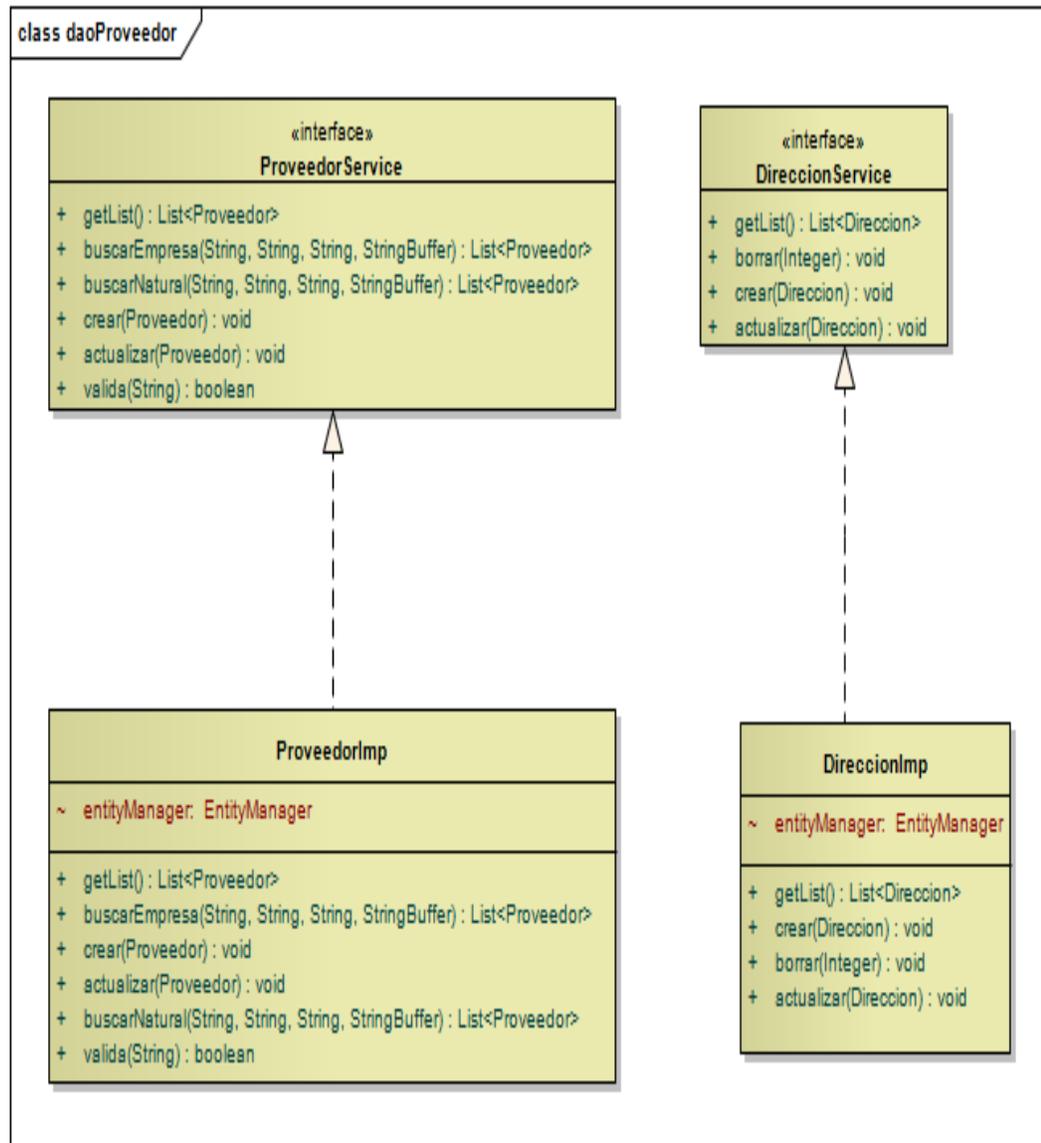


Diagrama 62. Clases DAO: Gestión de Proveedores Versión Final

3.4.4.3. CLASES CONTROLADOR

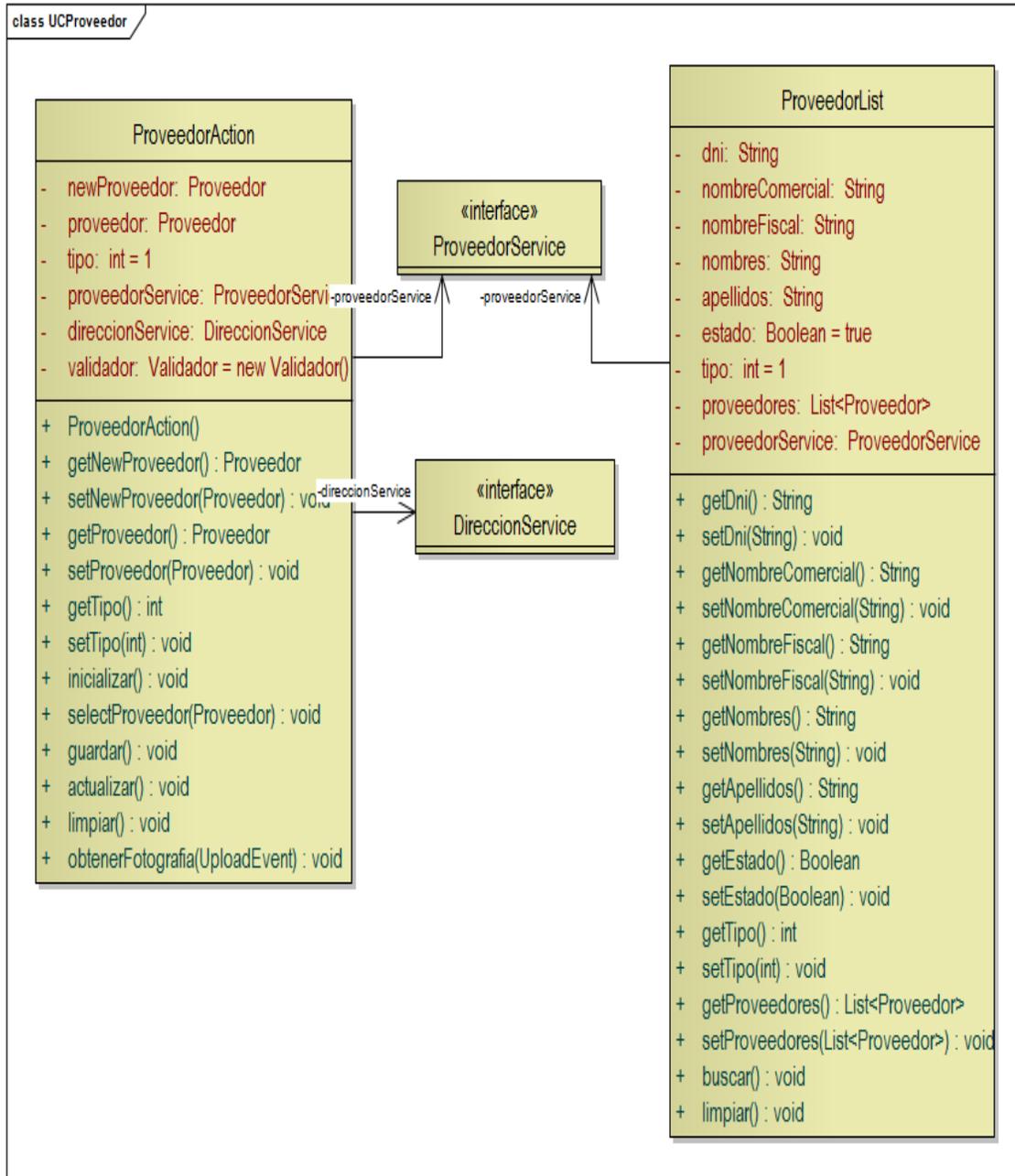


Diagrama 63. Clases Controlador: Gestión de Proveedor Versión Final

3.4.4.4. CLASES VISTA

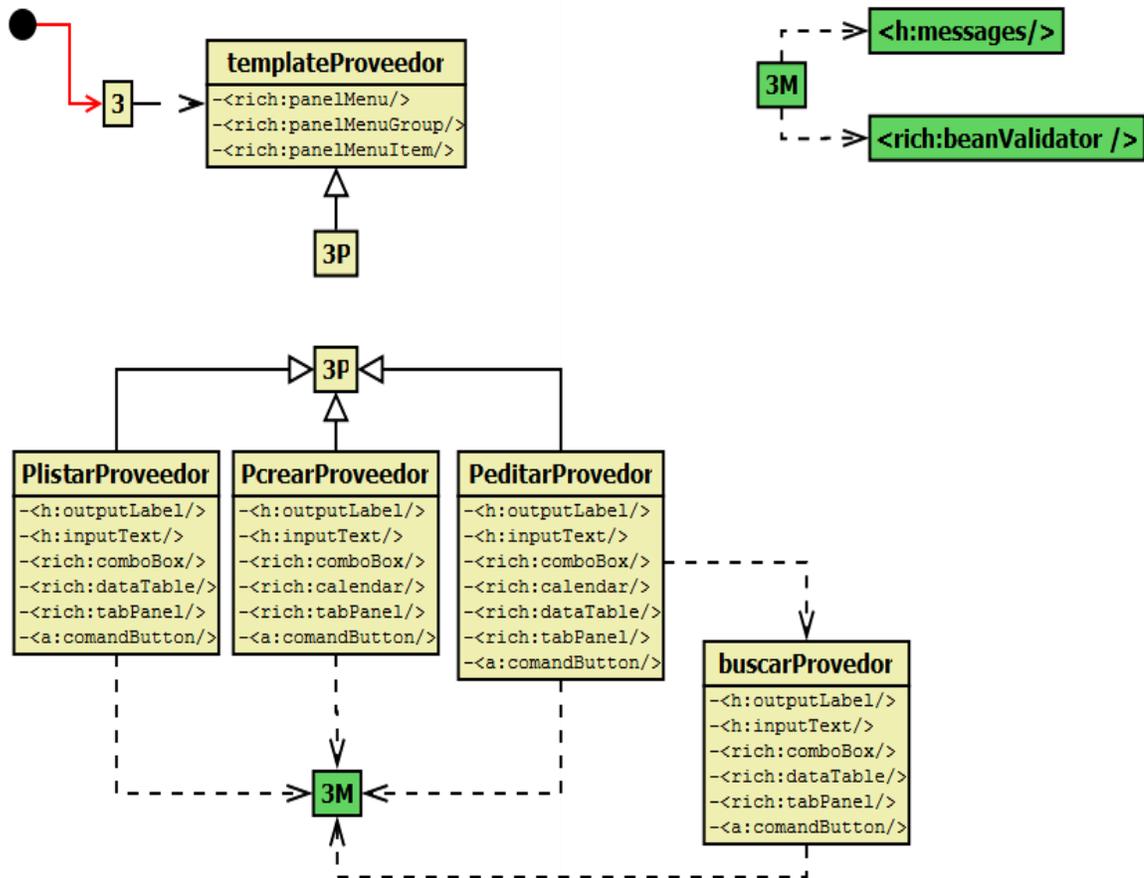


Diagrama 64. Vistas de Proveedor Versión Final

3.4.5. DISEÑO DE DATOS

3.4.5.1. MODELO ENTIDAD-RELACIÓN

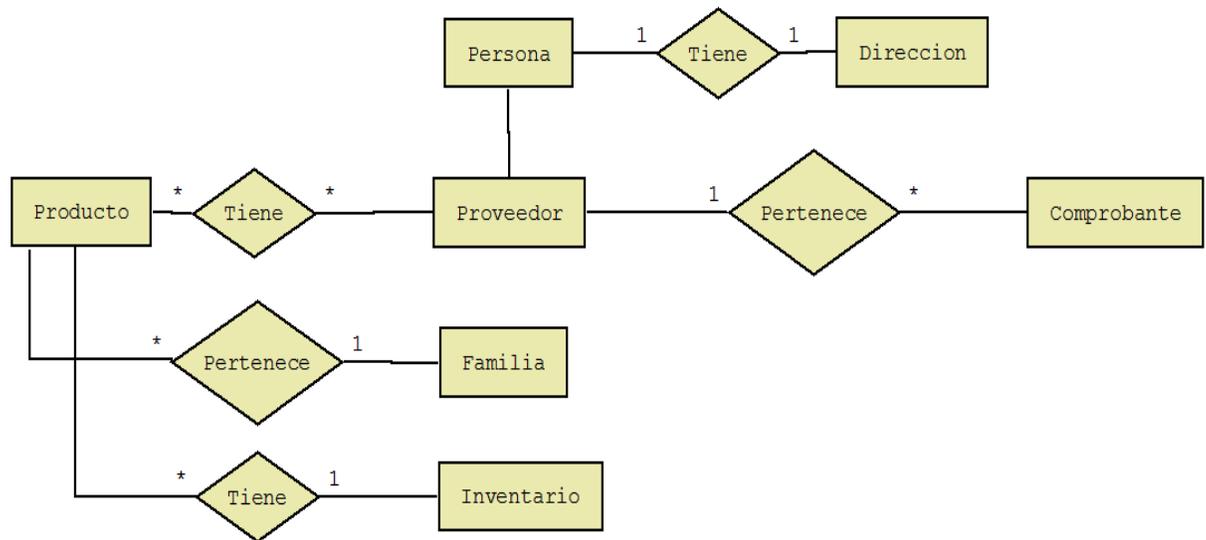


Diagrama 65. Modelo Entidad-Relación: Gestión de Proveedores Versión Final

3.4.5.2. MODELO CONCEPTUAL

PERSONA

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
dni	Varchar	no		
foto	Blob	si		
nombres	Varchar	no		
apellidos	Varchar	no		
fechaNac	Date	si		
idDireccion	Integer	no		

Tabla 118. Modelo Conceptual: Persona

DIRECCIÓN

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
ciudad	Varchar	no		
barrio	Varchar	si		
calles	Varchar	si		

teléfono	Integer	no		
celular	Integer	si		
email	Varchar	si		
numCasa	Integer	si		

Tabla 119. Modelo Conceptual: Dirección

PROVEEDOR

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no	🔑	
nombreFiscal	Varchar	no		
nombreComercial	Varchar	no		
url	Varchar	si		
estado	Boolean	no		

Tabla 120. Modelo Conceptual: Proveedor

3.4.5.3. MODELO RELACIONAL

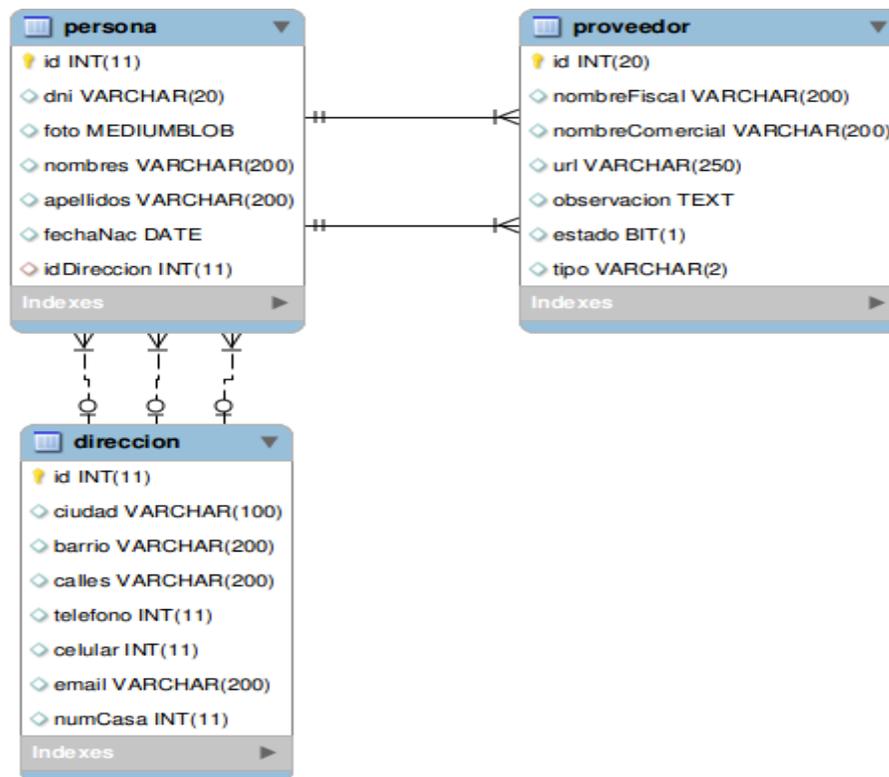


Diagrama 66. Modelo Relacional: Gestión de Proveedores Versión Final



GESTIÓN DE BODEGA

3.5. MÓDULO DE GESTIÓN DE BODEGA

3.5.1. MODELO DE DOMINIO

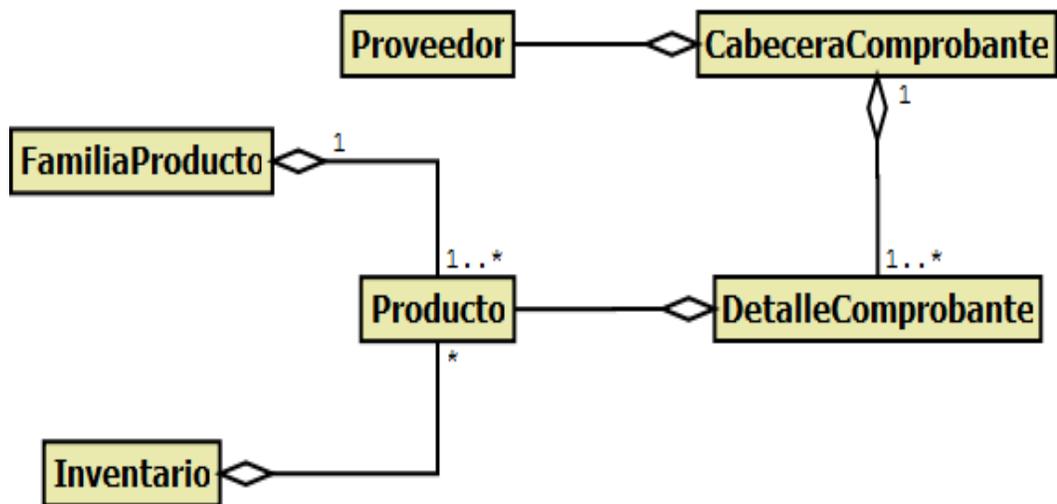


Diagrama 67. Módulo de Bodega Versión Final

3.5.2. MODELO DE CASOS DE USO DEL SISTEMA

3.5.2.1. DIAGRAMA DE CASOS DE USO

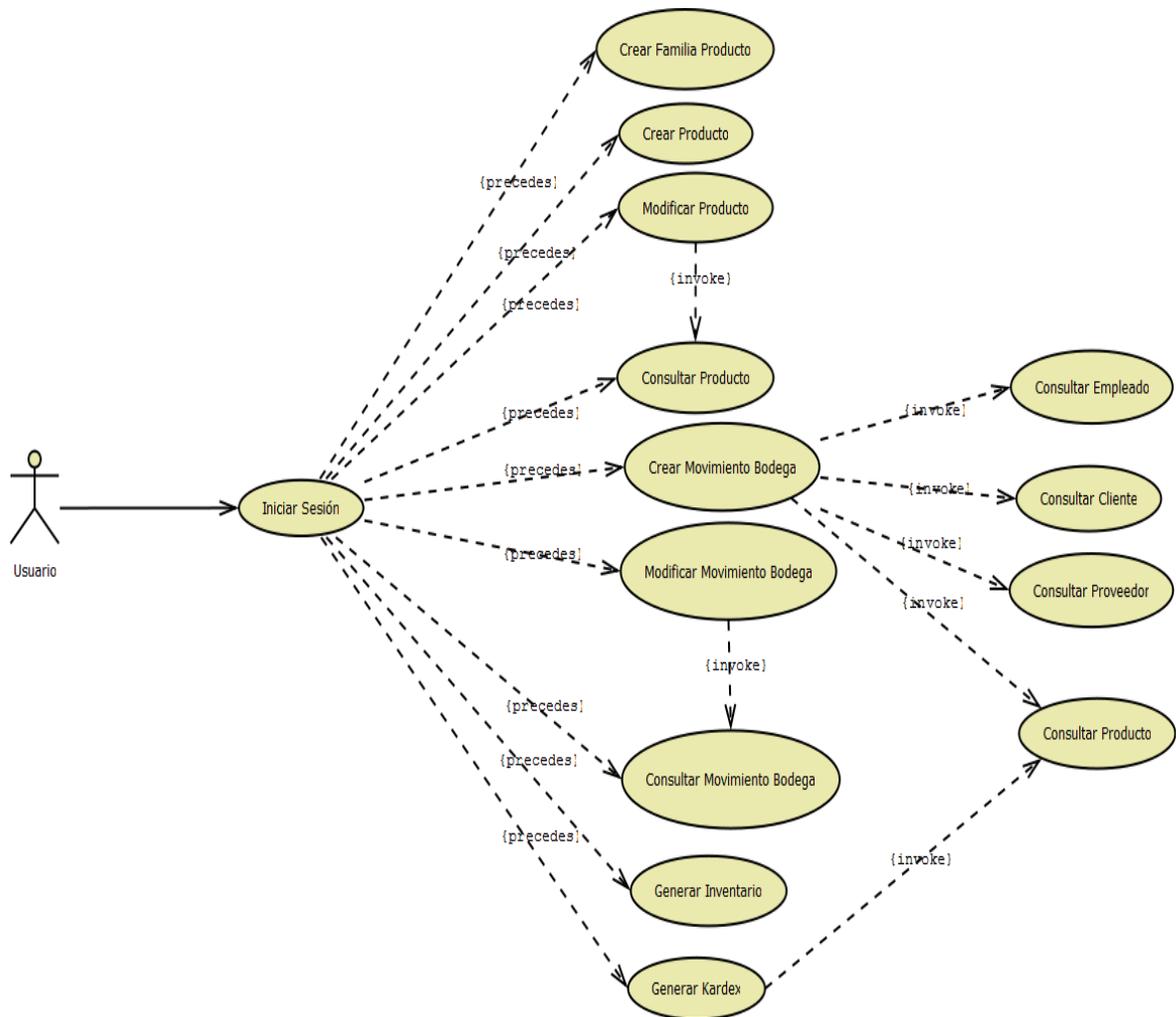


Diagrama 68.Gestión Bodega Versión Final

3.5.2.2. DESCRIPCIÓN DE CASOS DE USO

➤ CASO DE USO: Crear Familia Producto

NOMBRE: Crear Familia		CÓDIGO	OP032
NOMBRE PROGRAMADO	BcrearFamilia.xhtml		

GUARDAR BORRAR

Código Familia

Color

Nombre*

Descripcion:

Color	Código	Nombre*	Acciones
<input type="text"/>	<input type="text"/>	<input type="text"/>	
	001	ADAPTADORES	
	002	SONIDO	
	003	BATERIAS	
	004	BUSES DE DATOS	
	005	CABLES	
	006	CAMARAS	
	007	CARGADORES	
	008	CARTUCHOS	
	009	CASE	
	0010	CD /DVD	
	0011	CELULARES	
	0012	CINTAS	

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com

Tabla 121. Crear Familia Producto

Nombre:	Crear Familia Producto		Código: 035
Requerimiento Funcional:	RF016		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario con este rol asignado crear familias de productos.		
Descripción:	Qué el usuario con esta función asignada pueda crear familias de productos.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Bodega e ítem Producto.		
Poscondiciones:	Familia de Producto creada.		
FLUJO NORMAL:			

1. El usuario elige la opción [Crear Familia Producto] de la pantalla web principal "OpenLoja 1.0".
2. El sistema crea la Familia de Producto y muestra la pantalla web "Crear Familia".
3. El usuario ingresa los datos correspondientes a la Familia de Producto creada.
4. El usuario selecciona el botón [Guardar] de la pantalla web "Crear Familia".
5. El sistema valida que los campos obligatorios no estén vacíos.
6. El sistema verifica mediante el código que la Familia de Producto creada no coincida con otra ya existente.
7. El sistema los datos de la nueva Familia de Producto creada.
8. El sistema muestra un mensaje de Familia de Producto se ha creado con éxito.
9. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACIOS

- A.5. El sistema muestra un mensaje de campos obligatorios vacíos.
- A.6. El caso de uso continúa en el paso 3 del flujo normal de evento.

B. FAMILIA DE PRODUCTO EXISTENTE

- B.5. El sistema muestra un mensaje de Familia de Producto ya existe.
- B.6. El caso de uso continúa en el paso 3 del flujo normal de evento.

Tabla 122. Descripción Caso de Uso Crear Familia Producto

➤ CASO DE USO: Crear Producto

NOMBRE: Crear Producto	CÓDIGO	OP033
NOMBRE PROGRAMADO	BcrearProducto.xhtml	

GUARDAR
BORRAR
CANCELAR

cod:

Nombre*

Familia: ADAPTADORES ▾

tipo: Unidad ▾

stockMinimo:



datosAdicionales

Dimensión Horizontal

Dimensión Vertical

DETALLE

Realizado por: Yazber Romero & Jorge Tapia
yazb_87@hotmail.com

Tabla 123. Pantalla Crear Producto

Nombre:	Crear Producto	Código: 036
Requerimiento Funcional:	RF007	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado crear productos.	
Descripción:	Qué el usuario al cual se ha designado esta función pueda crear productos.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Bodega e ítem Producto.	
Poscondiciones:	Producto creado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Crear] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea el Producto y muestra la pantalla web "Crear Producto". 3. El usuario ingresa los datos correspondientes al Producto creado. 4. El usuario selecciona el botón [Guardar] de la pantalla web "Crear Producto". 		

5. El sistema valida que los campos obligatorios no estén vacíos.
6. El sistema verifica mediante el código que el Producto creado no coincida con otro ya existente.
7. El sistema guarda los datos del nuevo Producto.
8. El sistema muestra un mensaje de Producto se ha creado con éxito.
9. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACIOS

- A.5. El sistema muestra un mensaje de campos obligatorios vacíos.
- A.6. El caso de uso continúa en el paso 3 del flujo normal de evento.

B. PRODUCTO EXISTENTE

- B.5. El sistema muestra un mensaje de producto ya existe.
- B.6. El caso de uso continúa en el paso 3 del flujo normal de evento.

Tabla 124. Descripción de Caso de Uso Crear Producto

➤ CASO DE USO: Modificar Producto

NOMBRE: Editar Producto	CÓDIGO	OP034
NOMBRE PROGRAMADO	BeditarProducto.xhtml	

GUARDAR BU SCAR BORRAR CANCELAR

cod:

Nombre*

Familia

tipo

stockMinimo:

datosAdicionales

Dimensión Horizontal

Dimensión Vertical

DETALLE

Tabla 125. Pantalla Editar Producto

NOMBRE: Panel Modal Producto	CÓDIGO	PM19
NOMBRE PROGRAMADO	BlistarProducto.xhtml	

BU SCAR BORRAR

Código

Nombre*

Familia

Estado

Foto	codProducto	Nombre*	Familia	dimensionX	dimensionY	Acciones
No data						

Tabla 126. Panel Modal Producto

Nombre:	Modificar Producto	Código: 037
Requerimiento Funcional:	RF007	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado modificar un producto previamente creado.	
Descripción:	Al usuario con esta función asignada modificar productos que han sido creados con anterioridad.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Bodega e ítem Producto.	
Poscondiciones:	Producto seleccionado modificado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Editar Producto". 3. El usuario selecciona el botón [Buscar] de la pantalla web "Editar Producto". 4. El sistema muestra el panel modal "Producto". 5. El usuario busca el Producto que desea modificar para lo cual utiliza (invoca) el caso de uso <u>Consultar Producto</u>. 6. El usuario selecciona el Producto que busca del panel modal "Producto". 7. El sistema carga los datos del Producto seleccionado en la pantalla web "Editar Producto" y automáticamente cierra el panel modal "Producto". 8. El usuario modifica los datos del Producto seleccionado. 9. El usuario elige la opción [Guardar] de la pantalla web "Crear Producto". 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica mediante el código que el Producto modificado no coincida con otro ya existente. 12. El sistema guarda los nuevos datos del Producto editado. 13. El sistema muestra un mensaje de Producto se ha modificado con éxito. 14. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.10. El sistema muestra mensaje en los campos obligatorios vacíos.</p> <p>A.11. El caso de uso continúa en el paso 8 del flujo normal de evento.</p> <p>B. PRODUCTO EXISTENTE</p> <p>B.11. El sistema muestra un mensaje de producto ya existe.</p> <p>B.12. El caso de uso continúa en el paso 8 del flujo normal de evento.</p>		

Tabla 127. Descripción Caso de Uso Modificar Producto

➤ CASO DE USO: Consultar Producto

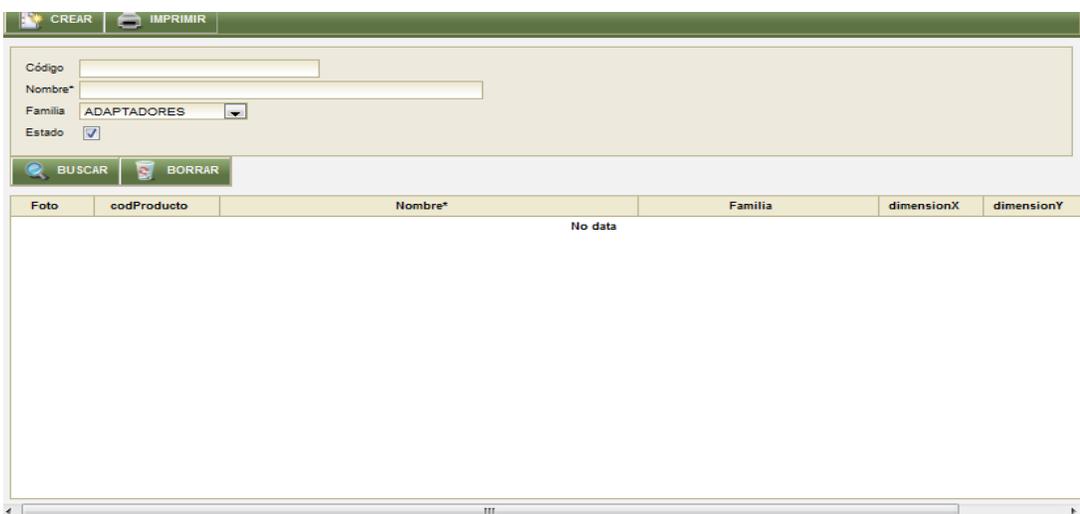
NOMBRE: Buscar Producto	CÓDIGO	OP035
NOMBRE PROGRAMADO	BlistarProducto.xhtml	
		

Tabla 128. Pantalla Buscar Producto

Nombre:	Consultar Producto	Código: 038
Requerimiento Funcional:	RF008	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado consultar productos.	
Descripción:	Al usuario que se le ha asignado esta función realizar consultas de productos previamente creados.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Bodega e ítem Producto.	
Poscondiciones:	Listado de Productos.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Consultar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Buscar Producto". 3. El usuario ingresa los datos del Producto que desea consultar. 4. El usuario elige la opción [Buscar] de la pantalla web "Buscar Producto". 5. El sistema muestra los productos que coinciden con la búsqueda en la Tabla de Productos de la pantalla web "Buscar Producto". 6. El caso de uso finaliza. 		

Tabla 129. Descripción Caso de Uso Consultar Producto

➤ CASO DE USO: Crear Movimiento Bodega

NOMBRE: Movimientos Bodega	CÓDIGO	OP036
NOMBRE PROGRAMADO	BcrearMovimiento.xhtml	

GUARDAR BUSCAR CANCELAR

Cliente <input type="text"/> DNI * <input type="text"/>	numeroDocumento <input type="text"/> Tipo Doc <input type="text" value="INGRESO COMPRA"/> Referencia <input type="text"/>	Fecha <input type="text" value="Jun 20, 2012"/> Tipo Movimiento <input type="text" value="Ingreso"/>
--	---	---

Carrito

Código	Producto	Cantidad	precioU	P. Total	Accio
No data					

comentarios <input style="width: 95%;" type="text"/>	Subtotal <input type="text" value="0.0"/> desc <input type="text" value="0.0"/> % <input type="text" value="0.0"/> impuesto <input type="text" value="12.0"/> % <input type="text" value="0.0"/>
--	--

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.co

Tabla 130. Pantalla Movimientos Bodega

NOMBRE: Panel Modal Proveedor	CÓDIGO	PM20
NOMBRE PROGRAMADO	buscarProveedorComprobante.xhtml	

PROVEEDOR X

DNI * <input type="text"/> Nombres * <input type="text"/> Apellidos * <input type="text"/>	tipo <input type="text" value="PERSONA NATURAL"/> Estado <input checked="" type="checkbox"/>
--	---

BUSCAR BORRAR

DNI * *	Nombres * Apellidos *	Email	Telefono	Celular	Acciones
No data					

Tabla 131. Panel Modal Proveedor

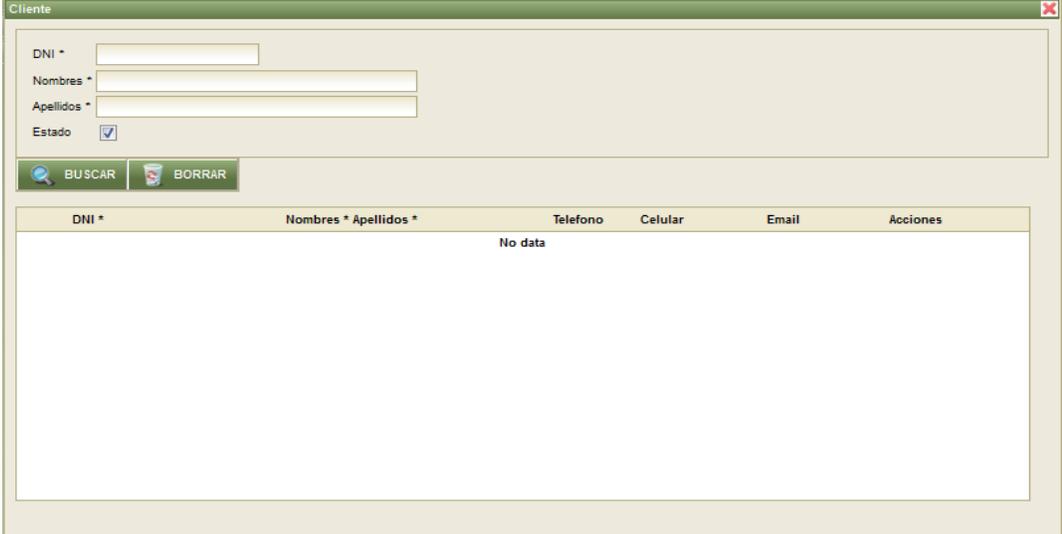
NOMBRE: Panel Modal Cliente	CÓDIGO	PM21
NOMBRE PROGRAMADO	buscarClienteComprobante.xhtml	
		

Tabla 132. Panel Modal Cliente

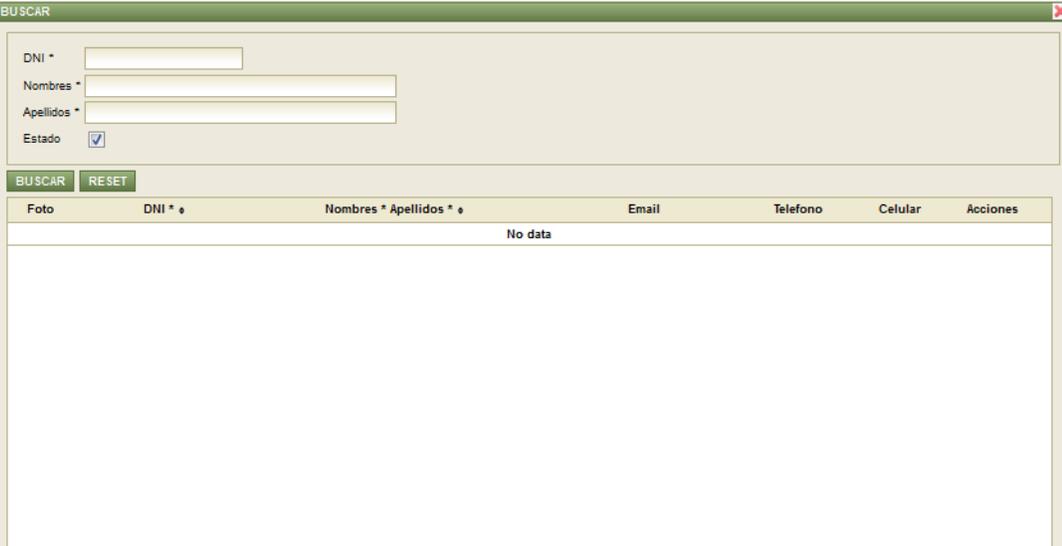
NOMBRE: Panel Modal Empleado	CÓDIGO	PM22
NOMBRE PROGRAMADO	buscarEmpleadoComprobante.xhtml	
		

Tabla 133. Panel Modal Empleado

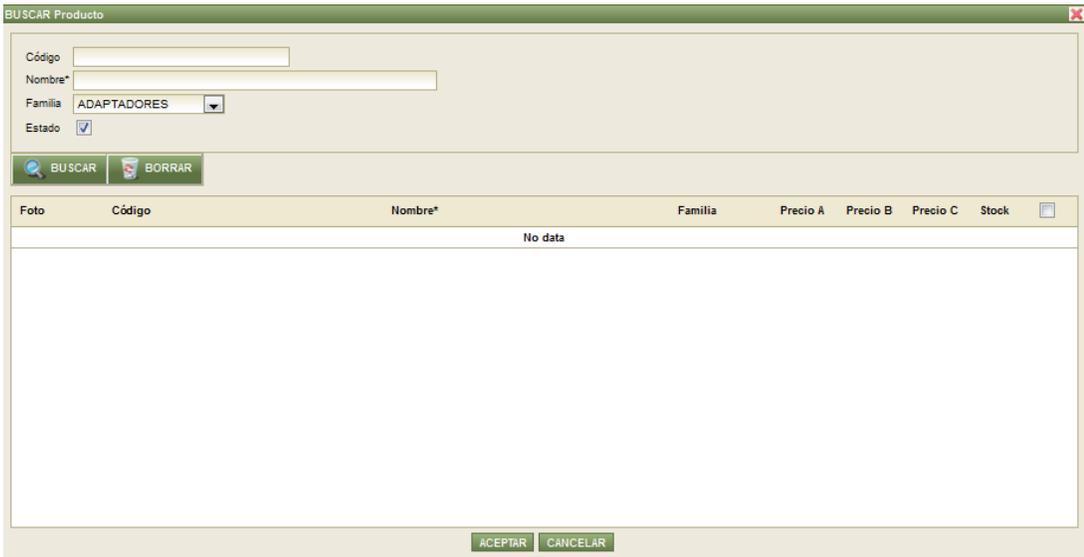
NOMBRE: Panel Modal Producto	CÓDIGO	PM23
NOMBRE PROGRAMADO	EmodalComprobanteInventario.xhtml	
		

Tabla 134. Panel Modal Producto

Nombre:	Crear Movimiento Bodega	Código: 039
Requerimiento Funcional:	RF017 – RF018	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario asignado con este rol realizar movimientos de Bodega (Ingresos o Egresos).	
Descripción:	Qué el usuario asignado con esta función pueda realizar ingresos y egresos de Bodega denominados movimientos.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Bodega e ítem Movimientos.	
Poscondiciones:	Registro de Ingreso o Egreso de Productos.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Crear] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea el movimiento y muestra la pantalla web "Movimientos Bodega". 3. El usuario elige el Tipo de movimiento: ingreso o egreso, según lo que vaya a realizar el sistema actualiza la pantalla web "Movimientos Bodega". 4. El usuario elige el <u>Tipo de documento</u> por el cual se realiza el movimiento (devoluciones, ajustes de inventario, Ventas, Compras) y de acuerdo a este el sistema actualiza inmediatamente la pantalla web "Movimientos Bodega". 5. El usuario elige la opción [Buscar] (va a buscar un Empleado, Cliente o Proveedor considerando los 2 pasos anteriores) de la pantalla web "Movimientos Bodega". 		

6. El sistema muestra el panel modal para realizar la búsqueda (“Cliente” ó “Empleado” ó “Proveedor”) tomando en cuenta la elección de los pasos 4 y 5.
7. El usuario busca: un empleado, cliente o proveedor según sea el caso para esto invoca el caso correspondiente.
8. El usuario selecciona el empleado, cliente o proveedor del panel modal mostrado para realizar la búsqueda.
9. El sistema carga los datos correspondientes a la selección anterior en la pantalla web “Movimientos Bodega” y automáticamente cierra el panel modal mostrado para realizar la búsqueda.
10. El usuario ingresa los datos correspondientes al Movimiento de Bodega creado.
11. El usuario selecciona el botón [Agregar Ítem] de la pantalla web “Movimientos Bodega”.
12. El sistema muestra el panel modal “Producto”.
13. El usuario busca los productos involucrados con el movimiento creado para lo cual utiliza (invoca) el caso de uso Consultar Producto.
14. El usuario selecciona los productos que busca del panel Modal “Producto”.
15. El sistema carga los datos de los productos seleccionados en la tabla de Productos de la pantalla web “Movimientos Bodega” y automáticamente cierra el panel modal “Producto”.
16. El usuario elige la opción [Guardar] de la pantalla web “Movimientos Bodega”.
17. El sistema valida que los campos obligatorios no estén vacíos.
18. El sistema verifica si hay productos asignados para el Movimiento.
19. El sistema verifica que el stock en Bodega de los Productos involucrados en el Movimiento sea suficiente para realizarse el mismo.
20. El sistema guarda los datos del nuevo Movimiento de Bodega.
21. El sistema muestra un mensaje de Movimiento se ha realizado con éxito.
22. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACÍOS

- A.17. El sistema muestra mensajes de campos obligatorios vacíos.
- A.18. El caso de uso continúa en el paso 10 del flujo normal de evento.

B. PRODUCTOS NO ASIGNADOS

- B.18. El sistema muestra un mensaje de No hay Productos asignados a Movimiento.
- B.19. El caso de uso continúa en el paso 11 del flujo normal de eventos.

C. STOCK INSUFICIENTE

- C.19. El sistema muestra un mensaje de Stock Insuficiente.
- C.20. El caso de uso continúa en el paso 11 del flujo normal de eventos.

Tabla 135. Descripción Caso de Uso Crear Movimiento Bodega

➤ CASO DE USO: Modificar Movimiento Bodega

NOMBRE: Editar Movimiento		CÓDIGO	OP037
NOMBRE PROGRAMADO		BeditarMovimiento.xhtml	

ANULAR
BUSCAR
CANCELAR

Proveedor

DNI *

numeroDoc

Tipo Doc

Referencia

Fecha

Tipo Movimiento

Carrito

Código	Producto	Cantidad	precioU	P. Total
No data				

comentarios

Subtotal

desc %

impuesto %

Total

Realizado por: Yazber Romero & Jorge Tapia
yazb_87@hotmail.c

Tabla 136. Pantalla Editar Movimiento

NOMBRE: Panel Modal Movimiento		CÓDIGO	PM24
NOMBRE PROGRAMADO		BmodalComprobante.xhtml	

GENERAR
BORRAR

Proveedor

Fecha Inicio

Fecha Fin

referencia

Movimiento

Fecha	referencia	Proveedor	Cliente	Empleado	subTotal	IVA	Total	tipo	Acciones
No data									

Tabla 137. Panel Modal Movimiento

Nombre:	Editar Movimiento Bodega	Código: 040
Requerimiento Funcional:	RF019	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario editar movimientos de Bodega (Ingresos y Egresos) creados anteriormente.	
Descripción:	Qué el usuario previamente autenticado con privilegios pertinentes pueda modificar ingresos y egresos de Bodega denominados movimientos que ya han sido creados.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Bodega e ítem Movimientos.	
Poscondiciones:	Movimiento de Bodega anulado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Editar Movimiento". 3. El usuario elige la opción [Buscar] de la pantalla web "Editar Movimiento". 4. El sistema muestra el panel modal "Movimiento". 5. El usuario busca el Movimiento que desea modificar para lo cual utiliza (invoca) el caso de uso <u>Consultar Movimiento</u>. 6. El usuario selecciona el Movimiento que busca del panel modal "Movimiento". 7. El sistema carga los datos del Movimiento seleccionado en la pantalla web "Editar Movimiento" y automáticamente cierra el panel modal "Movimiento". 8. El usuario selecciona el botón [Anular] de la pantalla web "Editar Movimiento". 9. El sistema muestra un mensaje de confirmación para anular el Movimiento. 10. El usuario elige la opción [Aceptar] para confirmar la anulación del Movimiento. 11. El sistema anula el movimiento seleccionado el cual ya no se podrá utilizar. 12. El sistema muestra un mensaje de Movimiento anulado con éxito 13. El caso de uso finaliza. 		

Tabla 138. Descripción Caso de Uso Editar Movimiento Bodega

➤ CASO DE USO: Consultar Movimiento Bodega

NOMBRE: Buscar Movimiento		CÓDIGO	OP038
NOMBRE PROGRAMADO		BlistarMovimiento.xhtmll	

CREAR
CANCELAR

Proveedor:

Fecha Inicio: Jun 1, 2012

Fecha Fin: Jun 30, 2012

referencia:

Movimiento: Ingreso

GENERAR
BORRAR

Fecha	referencia	Proveedor	Cliente	Empleado	subTotal	IVA	Total
No data							

Realizado por: Yazber Romero & Jorge Tapia
yazb_87@hotmail.com

Tabla 139. Pantalla Buscar Movimiento

Nombre:	Consultar Movimiento Bodega	Código: 041
Requerimiento Funcional:	RF020	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado consultar Movimientos de Bodega.	
Descripción:	Qué el usuario con esta función designada pueda consultar ingresos o egresos de Bodega denominados movimientos.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Bodega e ítem Movimientos.	
Poscondiciones:	Listado de Movimientos de Bodega.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Consultar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Buscar Movimiento". 		

3. El usuario elige el tipo de Movimientos de Bodega (Lista desplegable Tipo) que desea consultar: ingresos o egresos.
4. El sistema actualiza la pantalla web "Buscar movimiento" de acuerdo al Tipo de movimiento elegido en el paso anterior.
5. El usuario ingresa los datos del Movimiento que va a consultar.
6. El usuario elige la opción [Generar] de la pantalla web "Buscar Movimiento".
7. El sistema valida que las fechas ingresadas estén correctas.
8. El sistema muestra los Movimientos que coinciden con la búsqueda en la Tabla Movimientos de la pantalla web "Buscar Movimiento".
9. El caso de uso finaliza.

FLUJO ALTERNO:

A. FECHAS INCORRECTAS

- A.7. El sistema muestra un mensaje de fechas incorrectas.
- A.8. El caso de uso continúa en el paso 5 del flujo normal de eventos.

Tabla 140. Descripción Caso de Uso Consultar Movimiento Bodega

➤ CASO DE USO: Generar Inventario

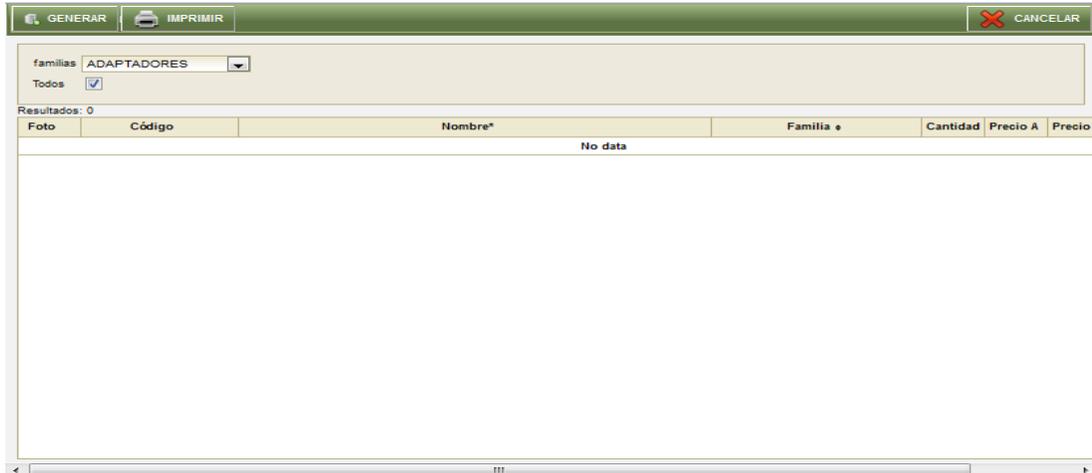
NOMBRE: Generar Inventario		CÓDIGO	OP039
NOMBRE PROGRAMADO	BlistarInventario.xhtml		
			

Tabla 141. Pantalla Generar Inventario

Nombre:	Generar Inventario	Código: 042
Requerimiento Funcional:	RF021	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol generar Inventarios.	
Descripción:	Qué el usuario con esta función asignada pueda generar Inventarios cuando lo necesite o se lo soliciten.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Bodega e ítem Reportes.	
Poscondiciones:	Inventario General generado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Inventario] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Generar Inventario". 3. El usuario marca la opción <u>Todos</u> o elige una familia de Productos de la cual desea generar el inventario. 4. El usuario elige la opción [Generar] de la pantalla web "Generar Inventarios". 5. El sistema muestra el inventario general con todos los productos existentes si el usuario eligió todos o solo con los que pertenecen a una familia en particular que haya elegido el mismo. 6. El caso de uso finaliza. 		

Tabla 142. Descripción Caso de Uso Generar Inventario

Nombre:	Generar Kárdex		Código: 043
Requerimiento Funcional:	RF022		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario con este rol asignado generar Kárdex de producto.		
Descripción:	Qué el usuario con esta función asignada pueda generar Kárdex de los productos existentes en Bodega cuando lo requiera.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Bodega e ítem Reportes.		
Poscondiciones:	Kárdex de Producto elegido generado.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Kárdex] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Generar Kárdex". 3. El usuario selecciona la opción [Buscar Producto] de la pantalla web "Generar Kárdex". 4. El sistema muestra el panel modal "Producto". 5. El usuario busca el Producto del cual va a generar el Kárdex para lo cual utiliza (invoca) el caso de uso <u>Consultar Producto</u>. 6. El usuario selecciona el Producto que busca del panel modal "Producto". 7. El sistema carga los datos del Producto seleccionado en la pantalla web "Generar Kárdex" y automáticamente cierra el panel modal "Producto". 8. El usuario ingresa la fecha Inicio y fecha Fin para generar el Kárdex. 9. El usuario selecciona el botón [Generar] de la pantalla web "Generar Kárdex". 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica que las fechas ingresadas sean correctas. 12. El sistema muestra los datos producto seleccionado (egresos e ingresos) en la tabla de resultados de la pantalla web "Generar Kárdex". 13. El caso de uso finaliza. 			
FLUJO ALTERNO:			
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.10. El sistema muestra mensajes en los campos obligatorios vacíos.</p> <p>A.11. El caso de uso continúa en el paso 3 del flujo normal de evento</p> <p>B. FECHAS INCORRECTAS.</p> <p>B.11. El sistema muestra mensaje de fechas ingresadas incorrectas.</p> <p>B.12. El caso de uso continúa en el paso 8 del flujo normal de evento.</p>			

Tabla 145. Descripción Caso de Uso Generar Kárdex

3.5.3. MODELO DE INTERACCIÓN

3.5.3.1. DIAGRAMA DE SECUENCIA 035: Crear Familia de Producto

(DS035)

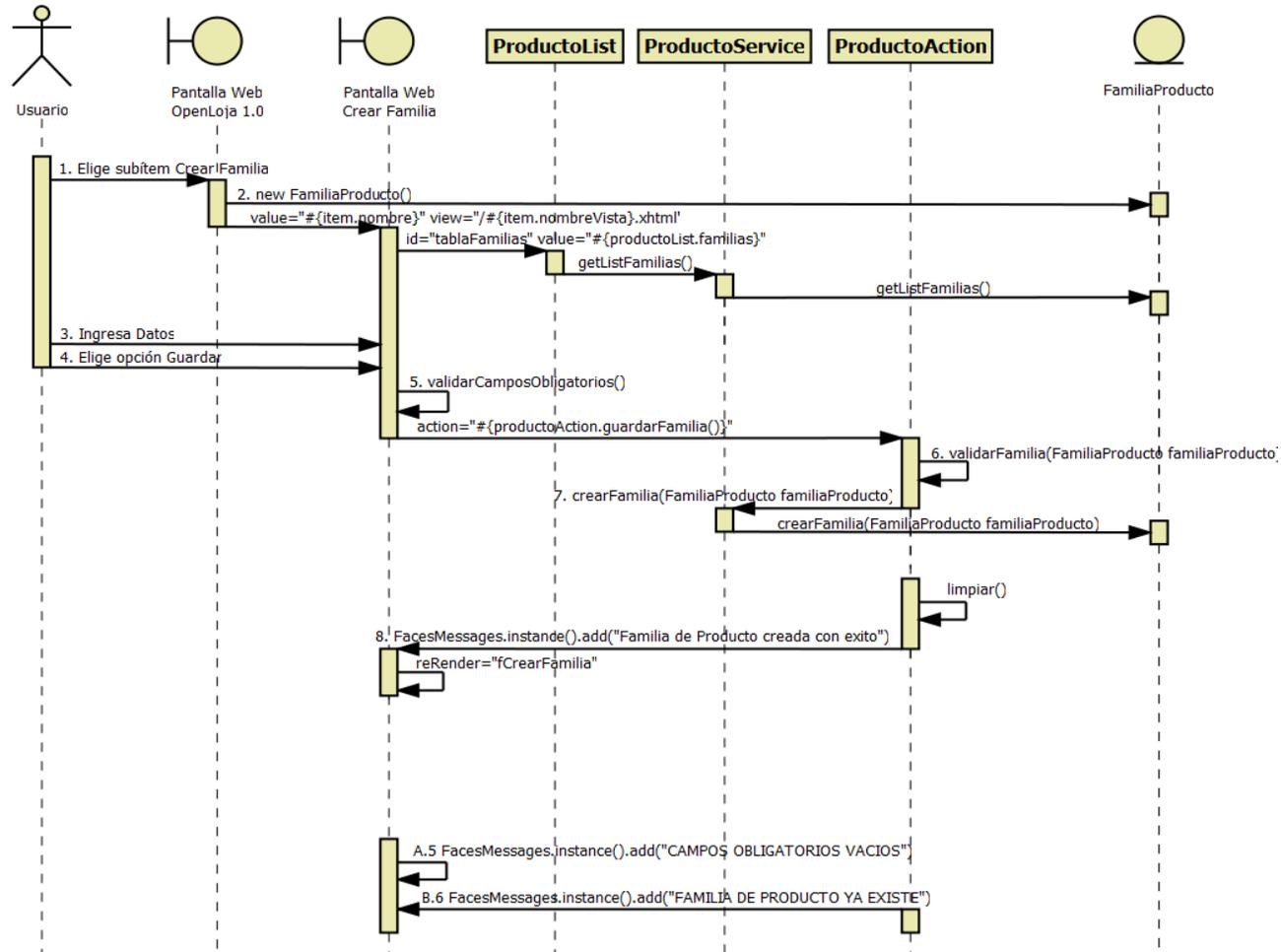


Diagrama 69. Secuencia Crear Familia de Producto Versión Final

3.5.3.2. DIAGRAMA DE SECUENCIA 036: Crear Producto (DS036)

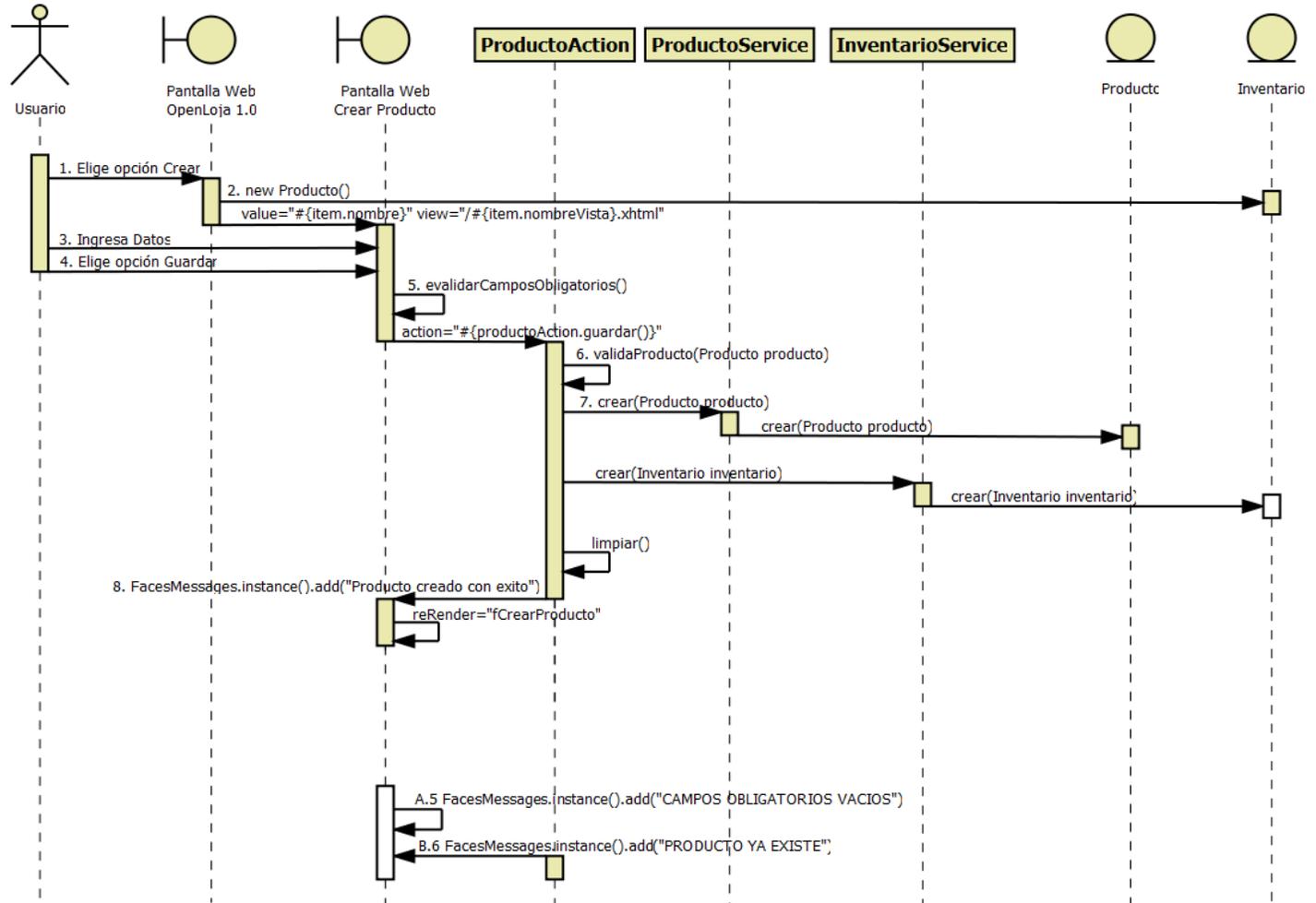


Diagrama 70. Secuencia Crear Producto Versión Final

3.5.3.3. DIAGRAMA DE SECUENCIA 037: Modificar Producto (DS037)

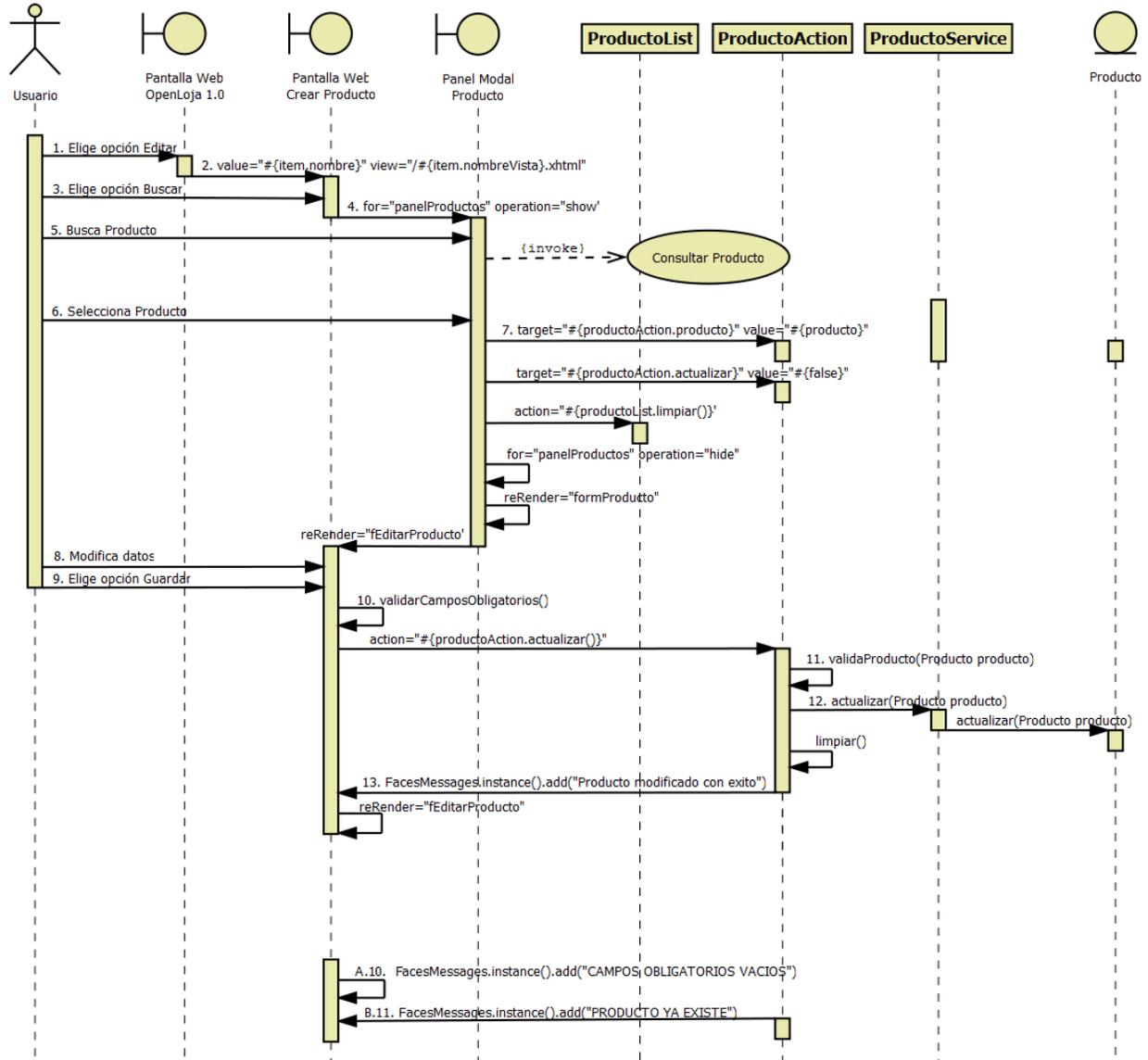


Diagrama 71. Secuencia Modificar Producto Versión Final

3.5.3.4. DIAGRAMA DE SECUENCIA 038: Consultar Producto (DS038)

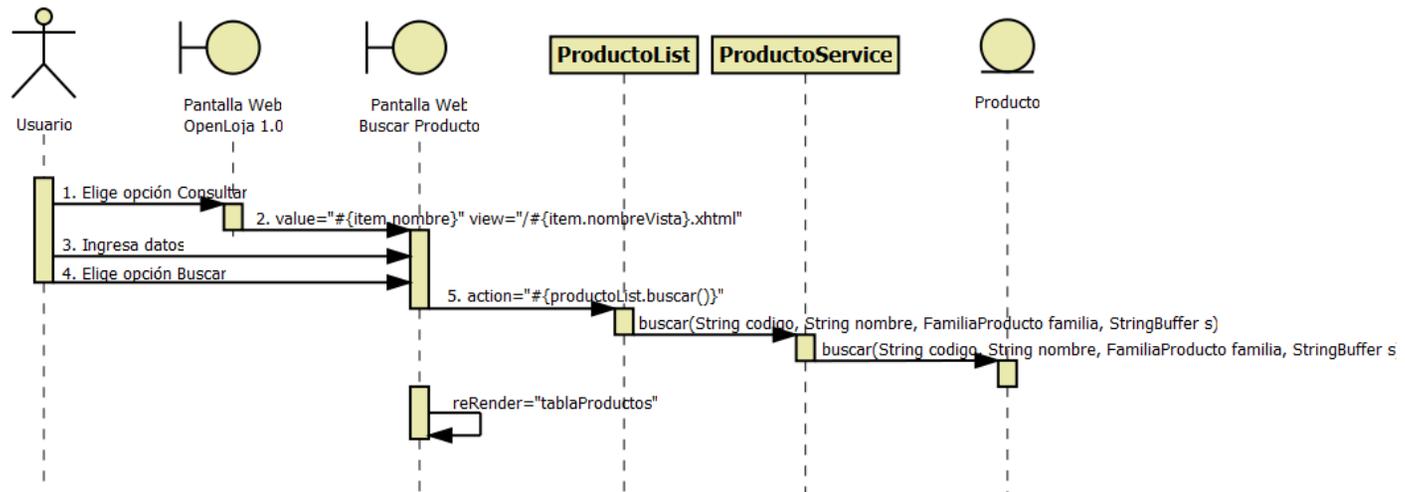


Diagrama 72. Secuencia Consultar Producto Versión Final

3.5.3.5. DIAGRAMA DE SECUENCIA 039: Crear Movimiento Bodega(DS039)

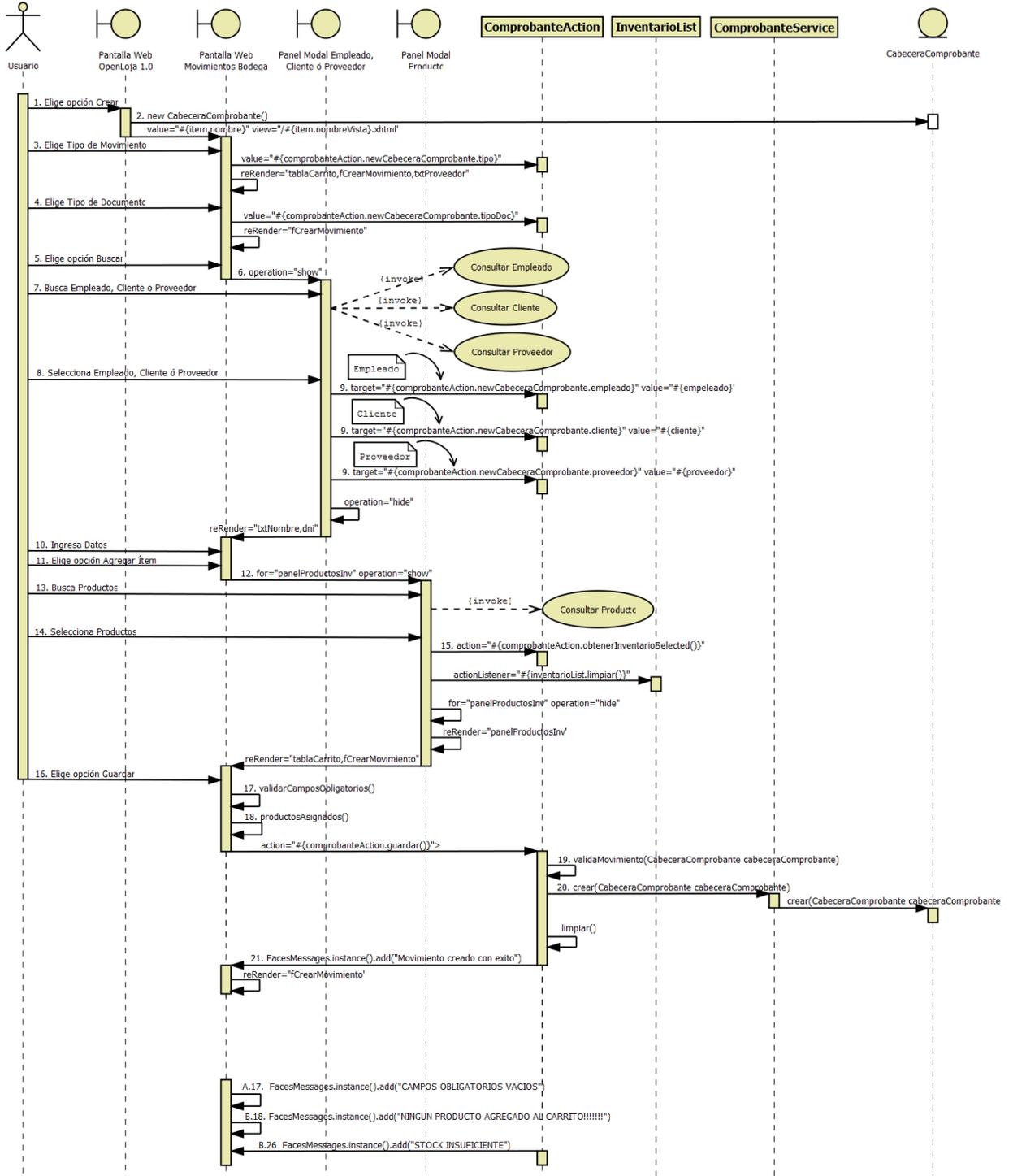


Diagrama 73. Secuencia Crear Movimiento Bodega Versión Final

3.5.3.6. DIAGRAMA DE SECUENCIA 040: Modificar Movimiento (DS040)

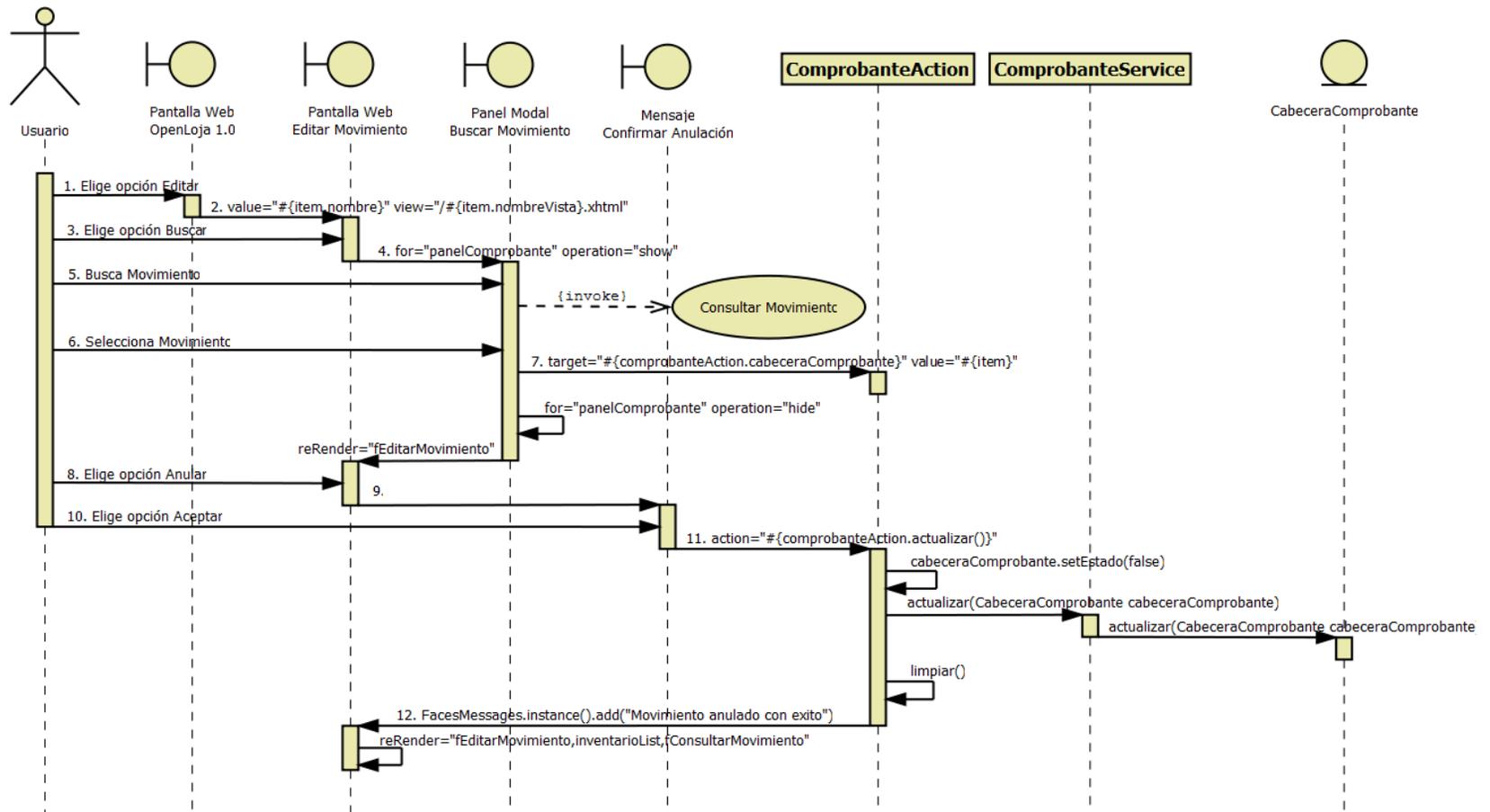


Diagrama 74. Secuencia Editar Movimiento Bodega Versión Final

3.5.3.7. DIAGRAMA DE SECUENCIA 041: Consultar Movimientos Bodega (DS041)

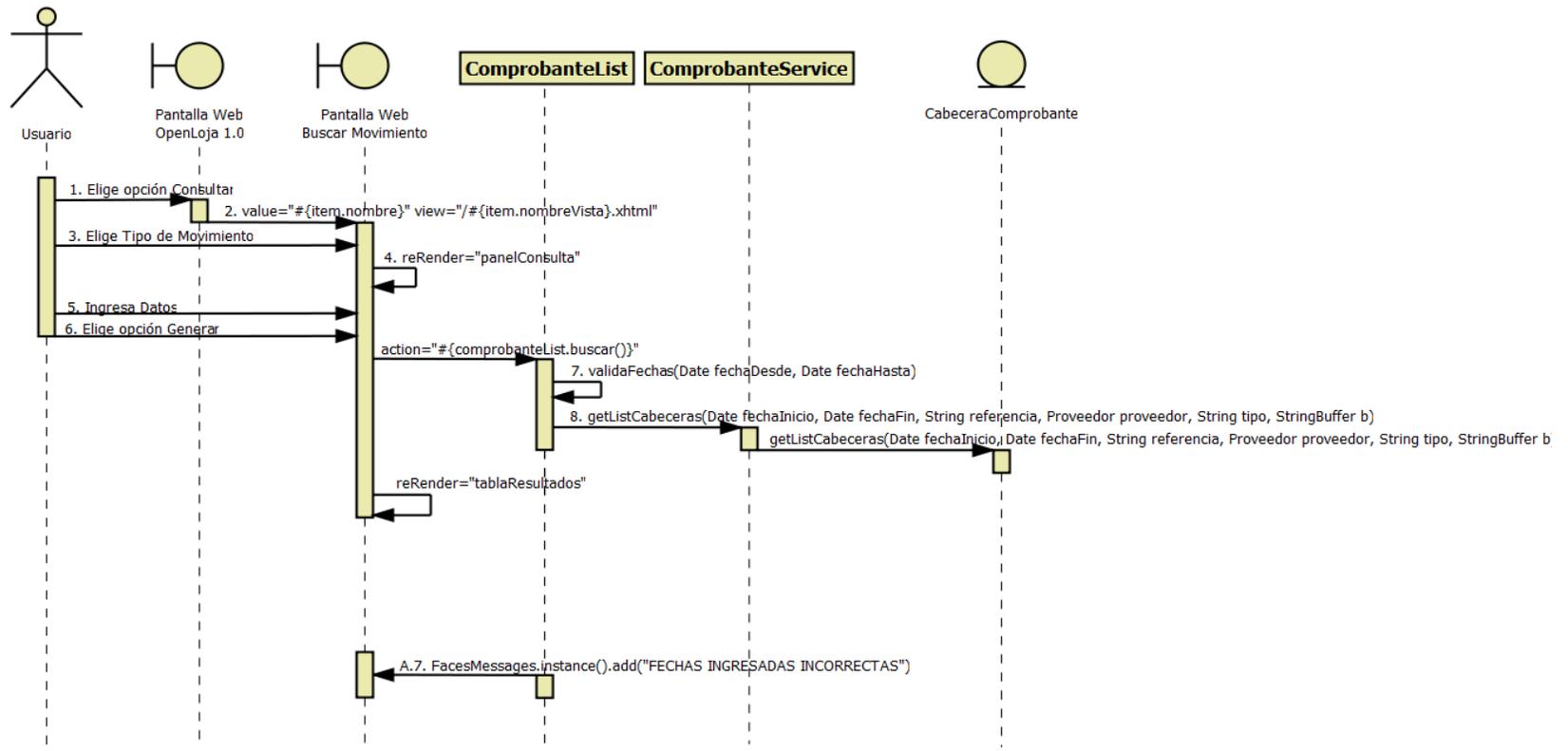


Diagrama 75. Secuencia Consultar Movimientos Bodega Versión Final

3.5.3.8. DIAGRAMA DE SECUENCIA 042: Generar Inventario (DS042)

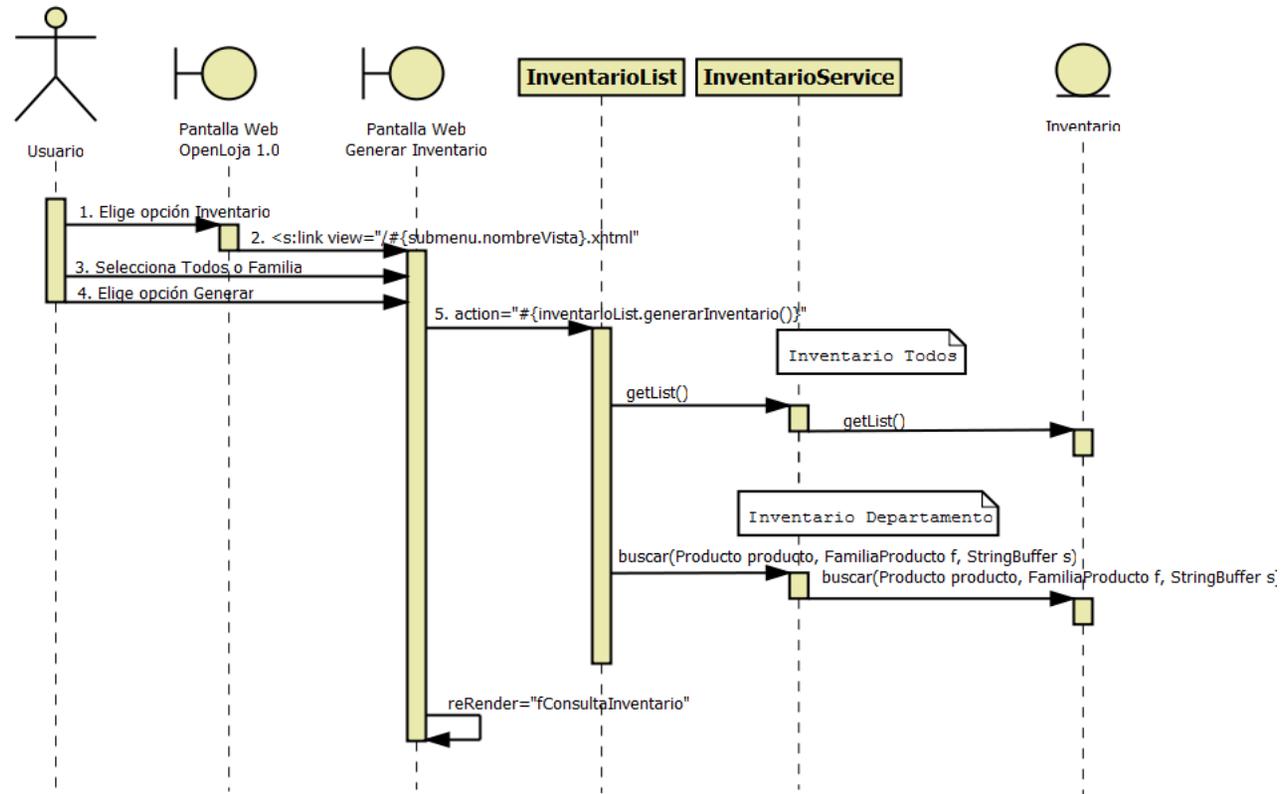


Diagrama 76. Secuencia Generar Inventario Versión Final

3.5.3.9. DIAGRAMA DE SECUENCIA 043: Generar Kárdex (DS043)

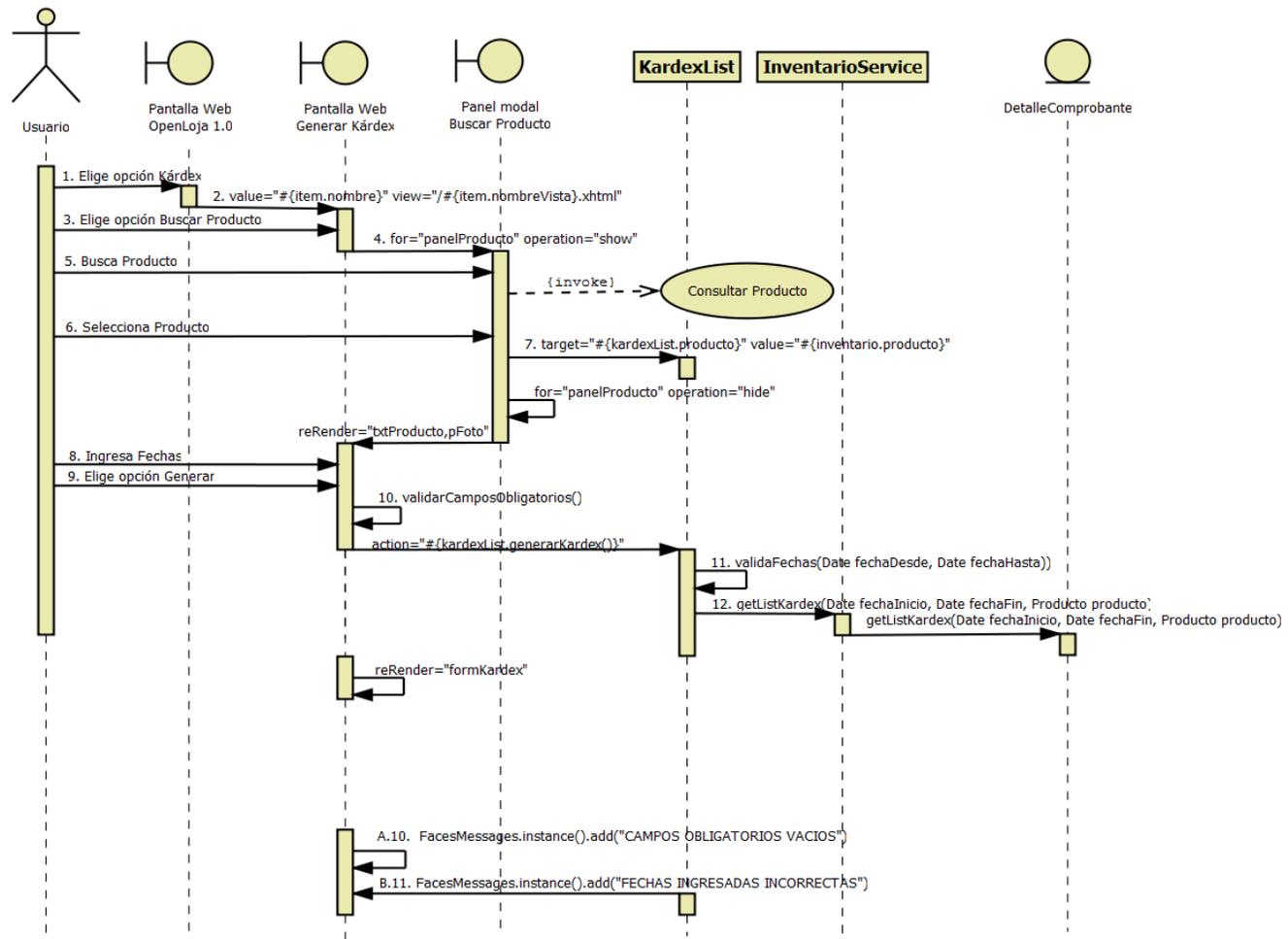


Diagrama 77. Secuencia Generar Kárdex Versión Final

3.5.4. DIAGRAMA DE CLASES FINAL

3.5.4.1. CLASES MODELO



Diagrama 78. Clases Modelo: Gestión de Bodega Versión Final

3.5.4.2. CLASES DAO

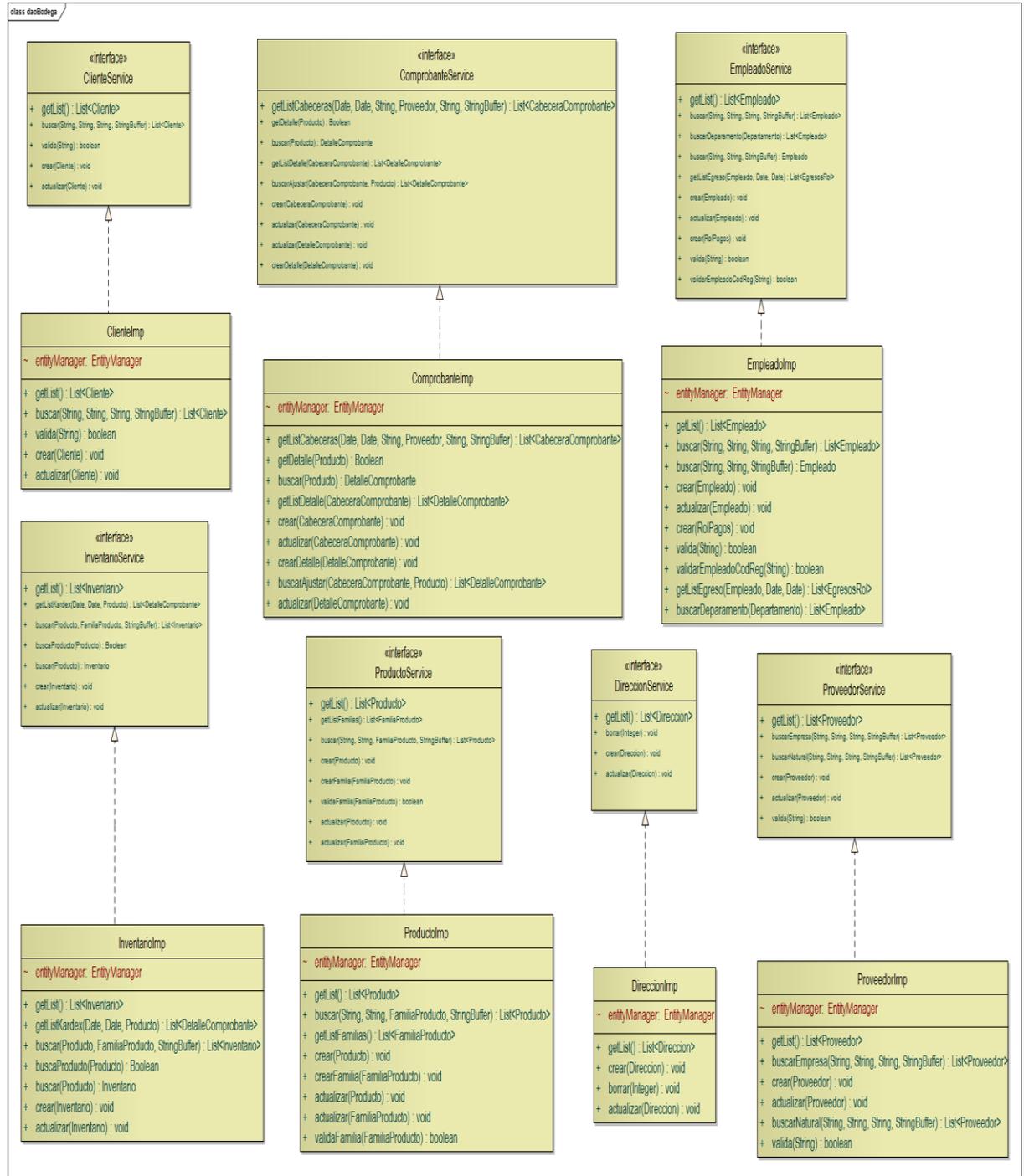


Diagrama 79. Clases DAO: Gestión de Bodega Versión Final

3.5.4.3. CLASES CONTROLADOR

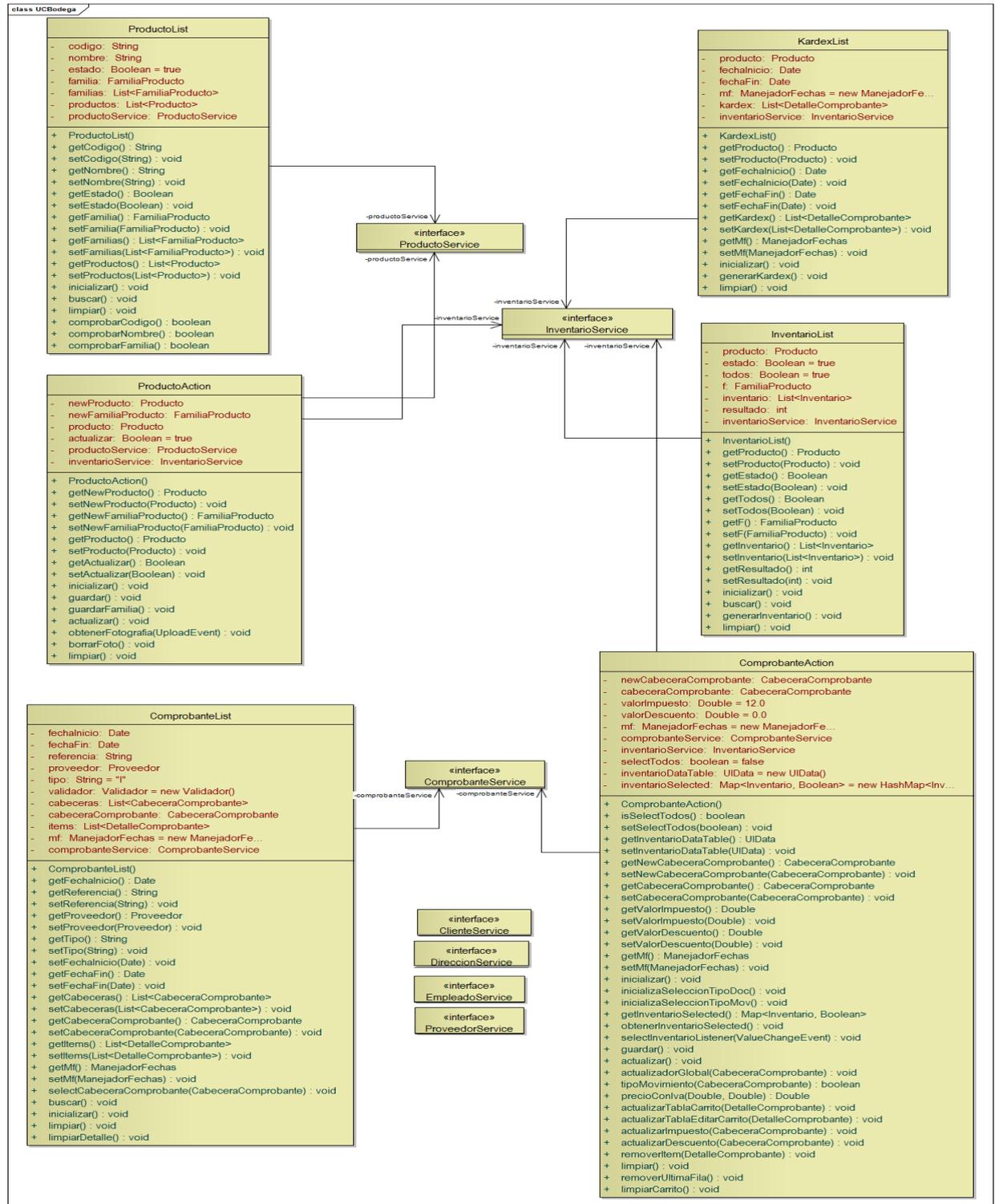


Diagrama 80. Clases Controlador: Gestión de Bodega Versión Final

3.5.4.4. CLASES VISTA

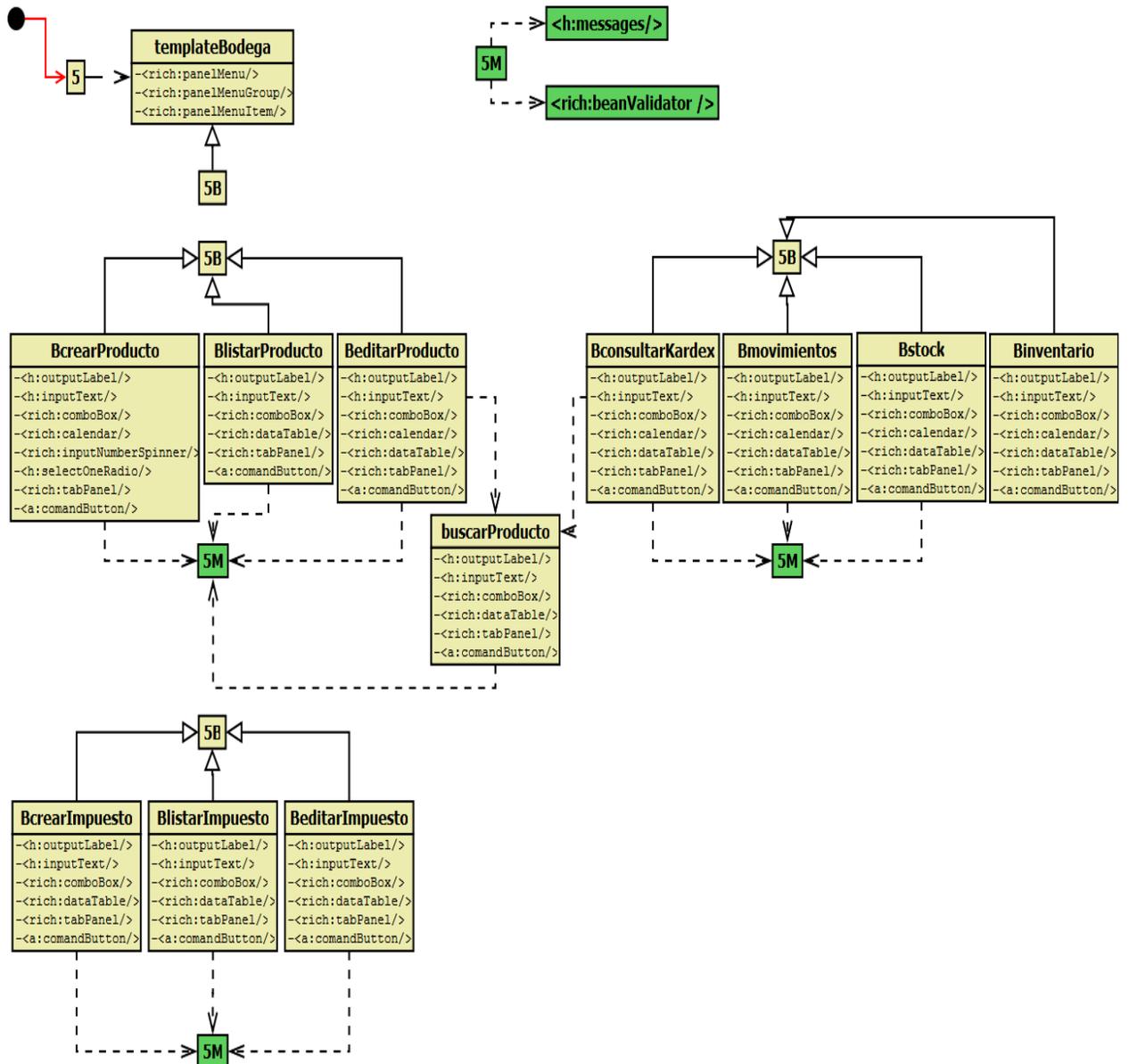


Diagrama 81. Vistas de Bodega Versión Final

3.5.5. DISEÑO DE DATOS

3.5.5.1. MODELO ENTIDAD-RELACIÓN

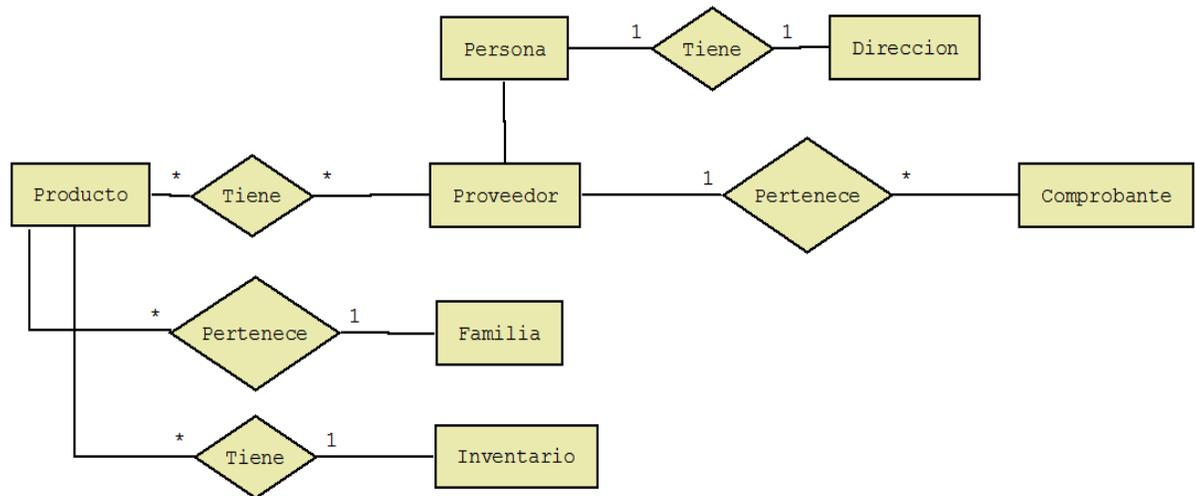


Diagrama 82. Modelo Entidad-Relación: Gestión de Bodega Versión Final

3.5.5.2. MODELO CONCEPTUAL

PERSONA

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	No	🔑	
dni	varchar	No		
foto	blob	Si		
nombres	varchar	No		
apellidos	varchar	No		
fechaNac	date	Si		
idDireccion	Integer	No		🔑

Tabla 146. Modelo Conceptual: Persona

DIRECCIÓN

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	No	🔑	
ciudad	varchar	No		
barrio	varchar	Si		
calles	varchar	Si		
teléfono	Integer	No		

celular	Integer	Si		
email	varchar	Si		
numCasa	Integer	Si		

Tabla 147. Modelo Conceptual: Dirección

PROVEEDOR

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	No		
nombreFiscal	varchar	No		
nombreComercial	varchar	No		
url	varchar	Si		
estado	boolean	No		

Tabla 148. Modelo Conceptual: Proveedor

PRODUCTO

ATRIBUTO	TIPO	NULO	PK	FK
codProducto	varchar	No		
foto	blob	Si		
nombre	varchar	No		
unidadMedida	varchar	No		
stockMin	Integer	Si		
dimensionH	Integer	Si		
dimensionV	Integer	Si		
codFamilia	Integer	No		
descripcion	varchar	Si		
estado	boolean	No		

Tabla 149. Modelo Conceptual: Producto

FAMILIA_PRODUCTO

ATRIBUTO	TIPO	NULO	PK	FK
id	numérico	No		
codAsociacion	cadena	No		
nombre	cadena	No		
descripcion	cadena	Si		

Tabla 150. Modelo Conceptual: Familia_Producto

PRODUCTO_PROVEEDOR

ATRIBUTO	TIPO	NULO	PK	FK
id	numérico	No		
codProducto	numérico	No		
codProveedor	numérico	No		

Tabla 151. Modelo Conceptual: Producto_Proveedor

CABECERA_COMPROBANTE

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	No		
numero	varchar	No		
referencia	varchar	No		
fecha	date	No		
tipo	boolean	No		
subtotal	double	No		
total	double	No		
codProveedor	Integer	No		

Tabla 152. Modelo Conceptual: Cabecera_Comprobante

DETALLE_COMPROBANTE

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	No		
idCabecera	Integer	No		
cantidad	Integer	No		
precioU	double	No		
precioTotal	double	No		
saldoCantidad	double	No		
saldoTotal	double	No		
codProducto	Integer	No		

Tabla 153. Modelo Conceptual: Detalle_Comprobante

INVENTARIO

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	No		
codProducto	Integer	No		
cantidad	Integer	No		
precioA	double	No		
precioB	double	No		
precioC	double	No		

Tabla 154. Modelo Conceptual: Inventario

3.5.5.3. MODELO RELACIONAL

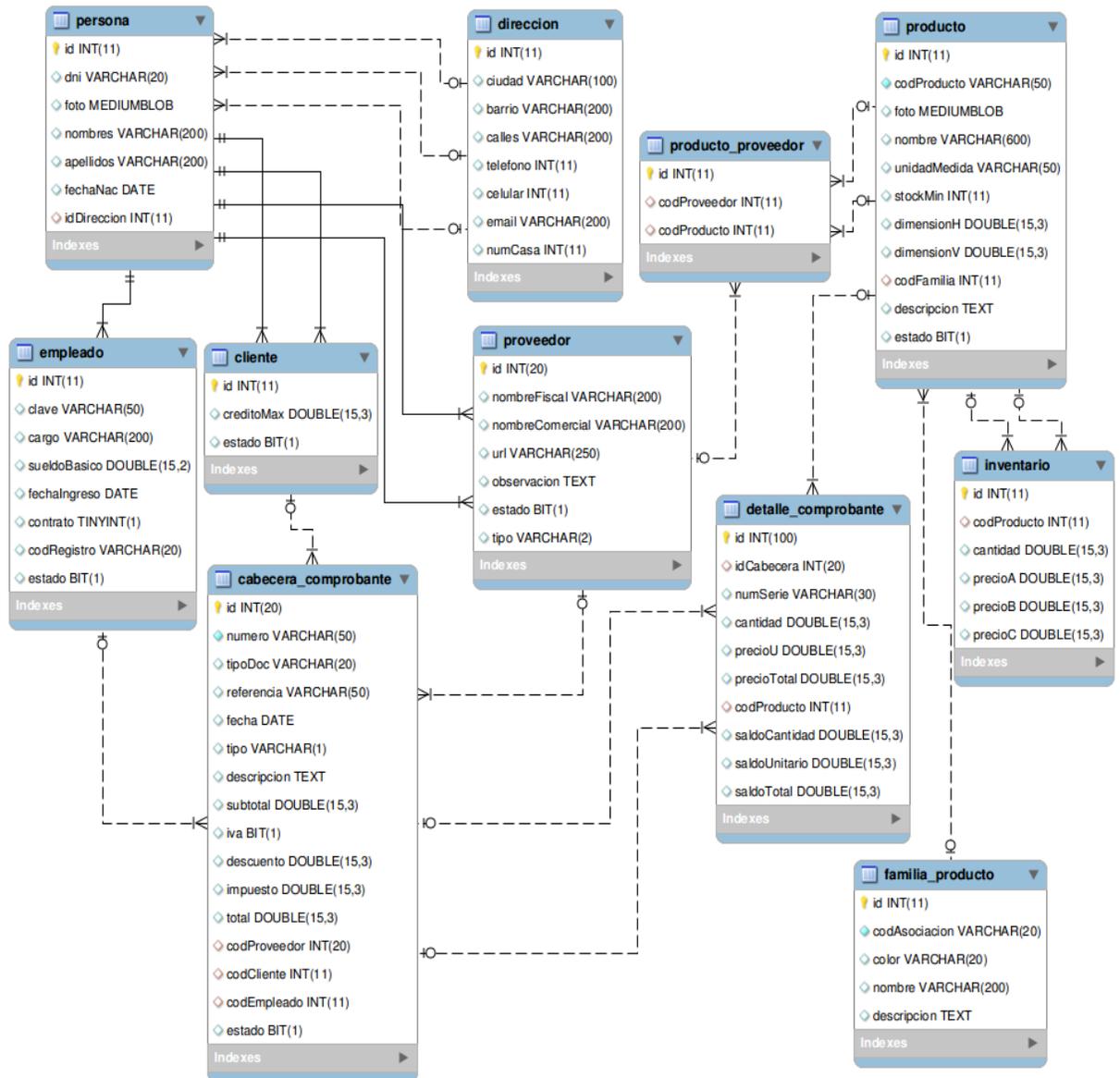


Diagrama 83. Modelo Relacional: Gestión de Bodega Versión Final



GESTIÓN DE VENTAS

3.6. MÓDULO DE GESTIÓN DE VENTAS

3.6.1. MODELO DE DOMINIO

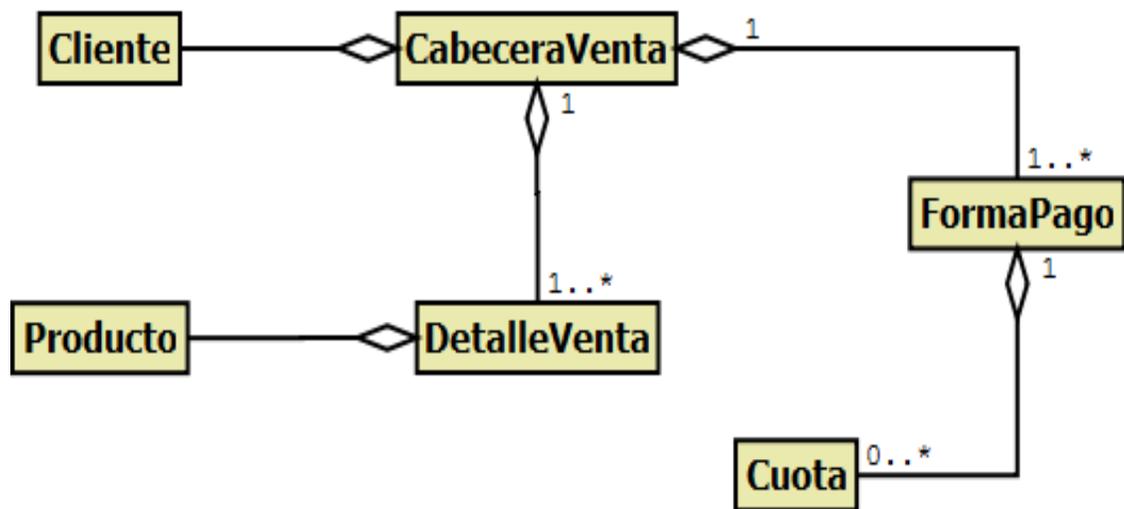


Diagrama 84. Módulo de Ventas Versión Final

3.6.2. MODELO DE CASOS DE USO DEL SISTEMA

3.6.2.1. DIAGRAMA DE CASOS DE USO

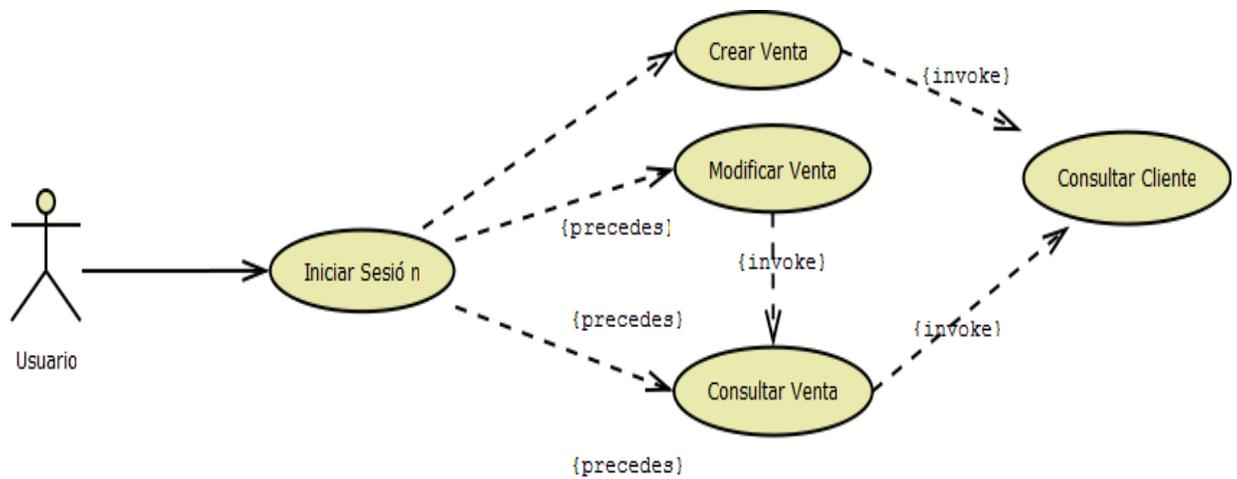


Diagrama 85. Gestión Ventas Versión Final

3.6.2.2. DESCRIPCIÓN DE CASOS DE USO

➤ CASO DE USO: Crear Venta

NOMBRE: Crear Venta		CÓDIGO	OP041
NOMBRE PROGRAMADO	VcrearVenta.xhtml		

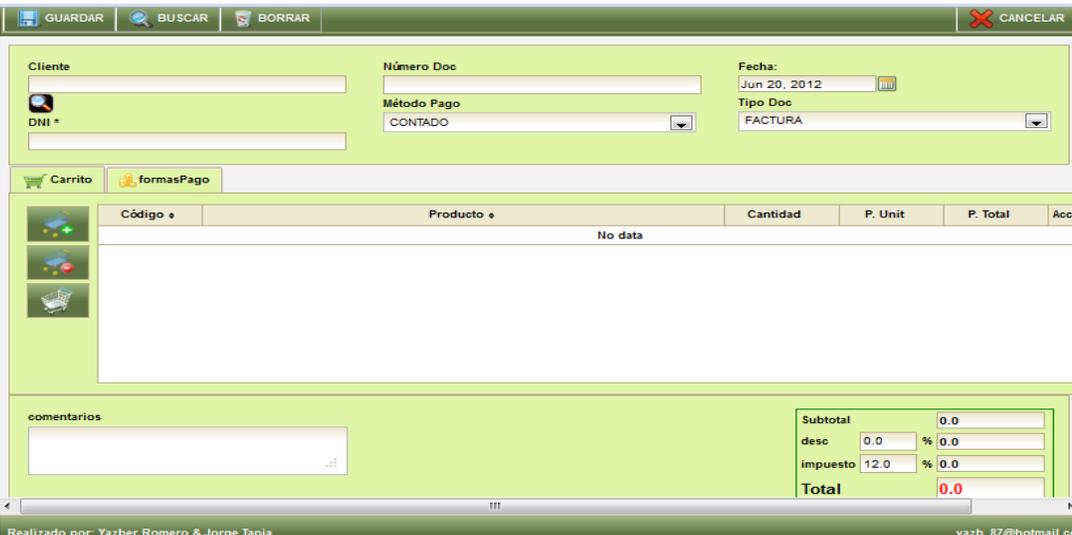


Tabla 155. Pantalla Crear Venta

NOMBRE: Panel Modal Cliente		CÓDIGO	PM26
NOMBRE PROGRAMADO	buscarCliente.xhtml		



Tabla 156. Panel Modal Cliente

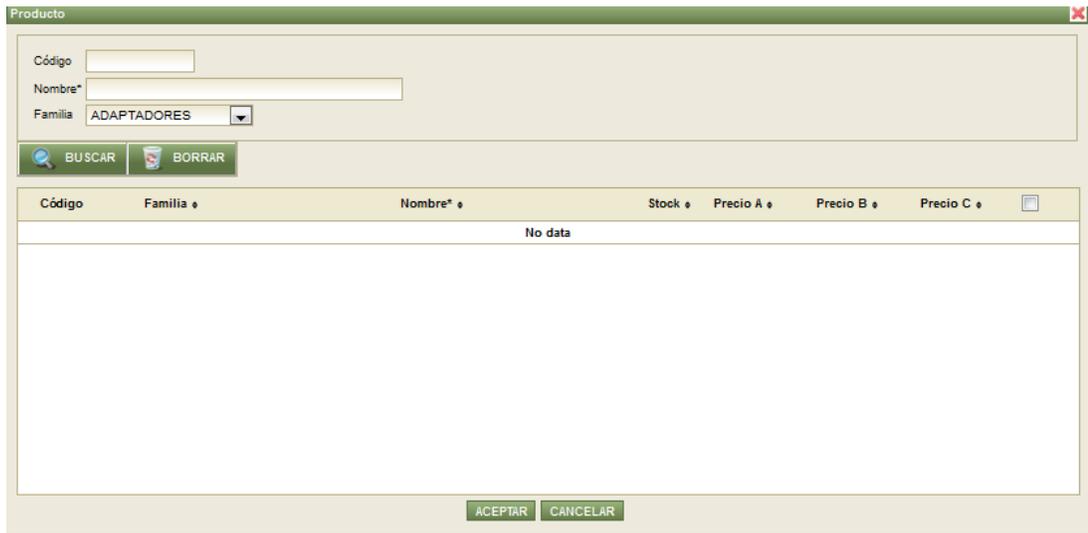
NOMBRE: Panel Modal Producto	CÓDIGO	PM27
NOMBRE PROGRAMADO	buscarProductoVenta.xhtml	
		

Tabla 157. Panel Modal Producto

Nombre:	Crear Venta	Código: 044
Requerimiento Funcional:	RF024	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado realizar venta de productos.	
Descripción:	Qué el usuario con esta función asignada pueda realizar ventas emitiendo sus respectivos comprobantes	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Ventas e ítem Ventas.	
Poscondiciones:	Venta realizada con su con su respectivo comprobante emitido.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Facturar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea la Venta y muestra la pantalla web "Crear Venta". 3. El usuario elige el tipo de comprobante que va a emitir: Factura o Nota de Venta en la pantalla web "Crear Venta". 4. El usuario selecciona el botón [Buscar Cliente] de la pantalla web "Crear Venta". 5. El sistema muestra el panel modal "Cliente". 6. El usuario busca el Cliente que está asociado con la venta para lo cual utiliza (invoca) el caso de uso <u>Consultar Cliente</u>. 7. El usuario selecciona el Cliente que busca del panel modal "Cliente". 		

8. El sistema carga los datos del Cliente seleccionado en la pantalla web "Crear Venta" y automáticamente cierra el panel modal "Cliente".
9. El usuario elige la forma de pago que puede ser a crédito o al contado en la pantalla web "Crear Venta"
10. El sistema actualiza la pestaña Formas de Pago de la pantalla web "Crear Venta".
11. El usuario elige la opción [Agregar Ítem] de la pantalla web "Crear Venta".
12. El sistema muestra el panel modal "Producto".
13. El usuario busca los productos que están asociados con la venta para lo cual utiliza (invoca) el caso de uso Consultar Producto.
14. El usuario selecciona los Productos que busca del panel modal "Producto".
15. El usuario elige la opción [Aceptar] del panel modal "Producto".
16. El sistema carga los datos de los Productos seleccionados en la pantalla web "Crear Venta" generando automáticamente los costos de la Venta y cierra el panel modal "Producto".
17. El usuario ingresa los datos correspondientes a la Venta creada.
18. El usuario selecciona la pestaña Formas de pago de la pantalla web "Crear Venta".
19. El usuario ingresa los datos correspondientes al pago de la Venta.
20. El usuario selecciona el botón [Guardar] de la pantalla web "Crear Venta".
21. El sistema valida que los campos obligatorios no estén vacíos.
22. El sistema verifica que existan Ítems asignados a la Venta.
23. El sistema valida que el Stock en Bodega de los Productos asociados con la Venta sea suficiente para realizar la misma.
24. El sistema guarda los datos de la nueva Venta realizada.
25. El sistema muestra un mensaje de Venta realizada con éxito.
26. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACIOS

- A.21. El sistema muestra mensajes en los campos obligatorios vacíos.
- A.22. El caso de uso continúa en el paso 3 del flujo normal de evento.

B. ÍTEMS NO ASIGNADOS

- B.22. El sistema muestra un mensaje de No se ha asignado ningún ítem.
- B.23. El caso de uso continúa en el paso 11 del flujo normal de eventos.

C. STOCK INSUFICIENTE

- C.23. El sistema muestra un mensaje de Stock insuficiente para realizar venta.
- C.24. El caso de uso continúa en el paso 11 del flujo normal de eventos.

Tabla 158. Descripción Caso de Uso Crear Venta

➤ CASO DE USO: Modificar Venta

NOMBRE: Editar Venta		CÓDIGO	OP042
NOMBRE PROGRAMADO		VanularVenta.xhtml	

ANULAR
BUSCAR
BORRAR
CANCELAR

Cliente	Número Doc	Fecha:
<input type="text"/>	<input type="text"/>	<input type="text"/>
DNI *	Método Pago	Tipo Doc
<input type="text"/>	CONTADO	FACTURA

Código	Producto	Cantidad	P. Unit	P. T
No data				

comentarios	<table style="width: 100%;"> <tr> <td>Subtotal</td> <td><input type="text"/></td> </tr> <tr> <td>desc 0.0 %</td> <td><input type="text"/></td> </tr> <tr> <td>impuesto 12.0 %</td> <td><input type="text"/></td> </tr> <tr> <td>Total</td> <td><input type="text"/></td> </tr> </table>	Subtotal	<input type="text"/>	desc 0.0 %	<input type="text"/>	impuesto 12.0 %	<input type="text"/>	Total	<input type="text"/>
Subtotal	<input type="text"/>								
desc 0.0 %	<input type="text"/>								
impuesto 12.0 %	<input type="text"/>								
Total	<input type="text"/>								

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com

Tabla 159. Pantalla Editar Venta

NOMBRE: Panel Modal Comprobante		CÓDIGO	PM28
NOMBRE PROGRAMADO			

Cliente	✖
---------	---

Cliente	
Fecha Inicio	Jun 1, 2012
Fecha Fin	Jun 30, 2012
Tipo Doc	FACTURA

BUSCAR	BORRAR
--------	--------

Fecha	Factura	Cliente	tipoPago	Subtotal	Impuestos	Total	Acci
No data							

Tabla 160. Panel Modal Comprobante

Nombre:	Editar Venta	Código: 045
Requerimiento Funcional:	RF025	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado realizar anulaciones de comprobantes de ventas.	
Descripción:	Qué el usuario con esta tarea designada pueda anular comprobantes de ventas.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Ventas e ítem Ventas.	
Poscondiciones:	Comprobante de Venta Anulado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Editar Venta". 3. El usuario elige la opción [Buscar] de la pantalla web "Editar Venta". 4. El sistema muestra el panel modal "Comprobante". 5. El usuario busca el Comprobante que desea editar para lo cual utiliza (invoca) el caso de uso <u>Consultar Comprobante</u>. 6. El usuario selecciona el Comprobante del panel modal "Comprobante" 7. El sistema carga los datos del Comprobante seleccionado en la pantalla web "Editar Venta" y automáticamente cierra el panel modal "Comprobante". 8. El usuario elige la opción [Anular] de la pantalla web "Editar Venta". 9. El sistema muestra el panel modal de confirmación de anulación. 10. El usuario elige la opción [Aceptar] del panel modal de confirmación de anulación. 11. El sistema anula el Comprobante seleccionado el cual no se podrá volver a utilizar. 12. El sistema muestra un mensaje de comprobante anulado con éxito. 13. El caso de uso finaliza. 		

Tabla 161. Descripción Caso de Uso Editar Venta

➤ CASO DE USO: Consultar Venta

NOMBRE: Buscar Venta	CÓDIGO	OP043
NOMBRE PROGRAMADO	VlistarVentas.shtml	

CREAR IMPRIMIR CANCELAR

Cliente

Fecha Inicio

Fecha Fin

Tipo Doc

BUSCAR BORRAR

Fecha	Factura	Cliente	tipoPago	Subtotal	Impuestos	Total	Acc
No data							

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com

Tabla 162. Pantalla Buscar Venta

NOMBRE: Panel Modal Cliente	CÓDIGO	PM29
NOMBRE PROGRAMADO	VmodalCliente.shtml	

BUSCAR BORRAR

Cliente

DNI *

Nombres *

Apellidos *

BUSCAR BORRAR

DNI *	Nombres * Apellidos *	Telefono	Celular	Email	Acciones
No data					

Tabla 163. Panel Modal Cliente

Nombre:	Consultar Ventas	Código: 046
Requerimiento Funcional:	RF026	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado realizar consultas de ventas realizadas.	
Descripción:	Qué el usuario a quien se ha otorgado esta función pueda consultar Ventas con sus respectivos comprobantes.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Ventas e ítem Ventas.	
Poscondiciones:	Listado de Ventas con sus respectivos comprobantes.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Buscar Venta".} 3. El usuario elige la opción [Buscar Cliente] de la pantalla web "Buscar Cliente". 4. El sistema muestra el panel modal "Cliente". 5. El usuario busca el Cliente del cual va a consultar las Ventas utilizando (invocando) el caso de uso <u>Consultar Cliente</u>. 6. El usuario selecciona el Cliente que busca del panel modal "Cliente". 7. El sistema carga los datos del Cliente seleccionado en la pantalla web "Buscar Venta" y automáticamente cierra el panel modal "Clientes". 8. El usuario ingresa los datos para realizar la consulta. 9. El usuario elige la opción [Buscar] de la pantalla web "Buscar Venta". 10. El sistema verifica que las fechas ingresadas estén correctas. 11. El sistema muestra los comprobantes que coinciden con la búsqueda en la tabla de resultados de la pantalla web "Buscar Venta". 12. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>A. FECHAS INCORRECTAS</p> <p>A.10. El sistema muestra mensaje de fechas ingresadas incorrectas.</p> <p>A.11. El caso de uso continúa en el paso 4 del flujo normal de evento</p>		

Tabla 164. Descripción Caso de Uso Consultar Ventas

3.6.3. MODELO DE INTERACCIÓN

3.6.3.1. DIAGRAMA DE SECUENCIA 044: Crear Venta (DS044)

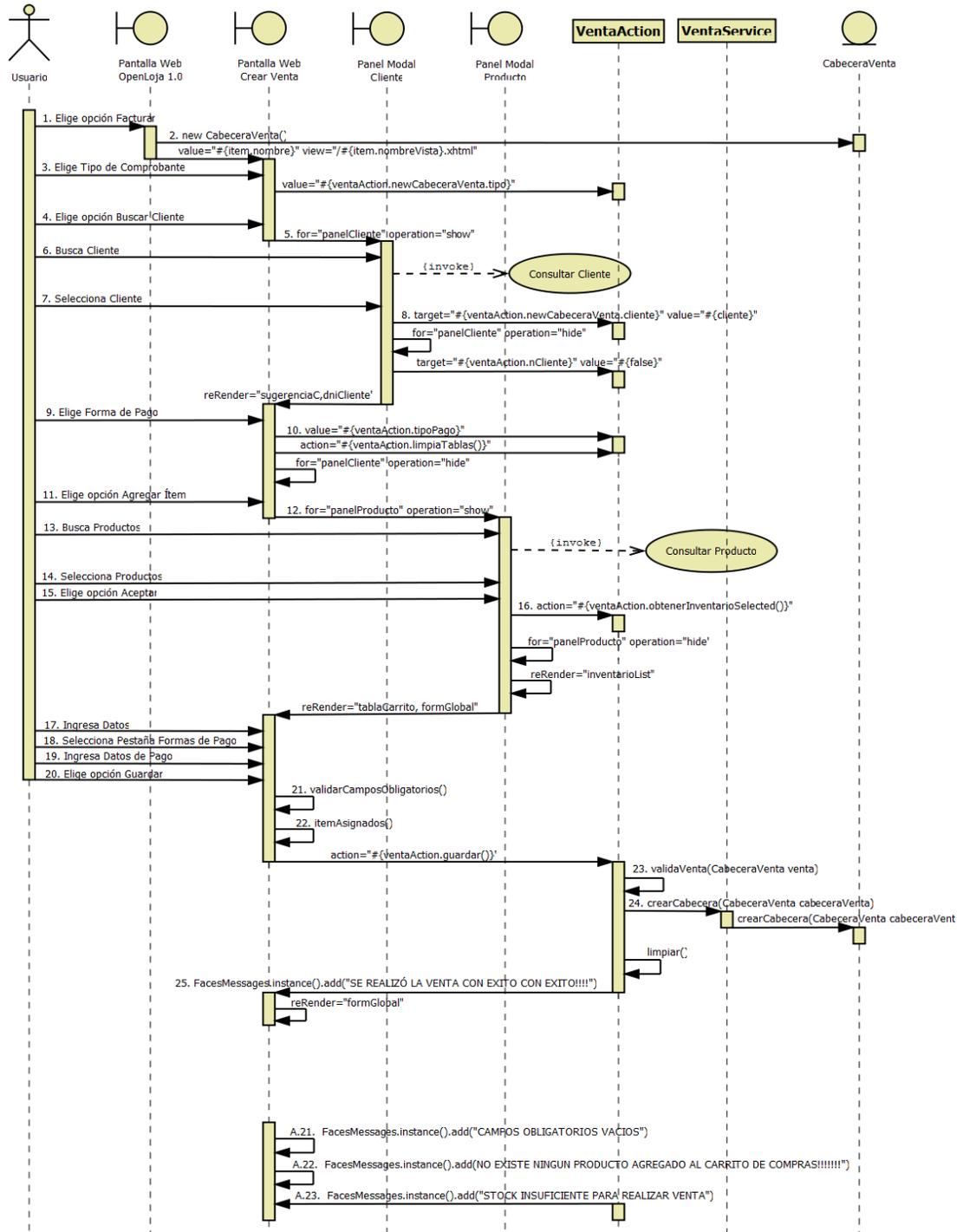


Diagrama 86. Secuencia Crear Venta Versión Final

3.6.3.2. DIAGRAMA DE SECUENCIA 045: Modificar Venta (DS045)

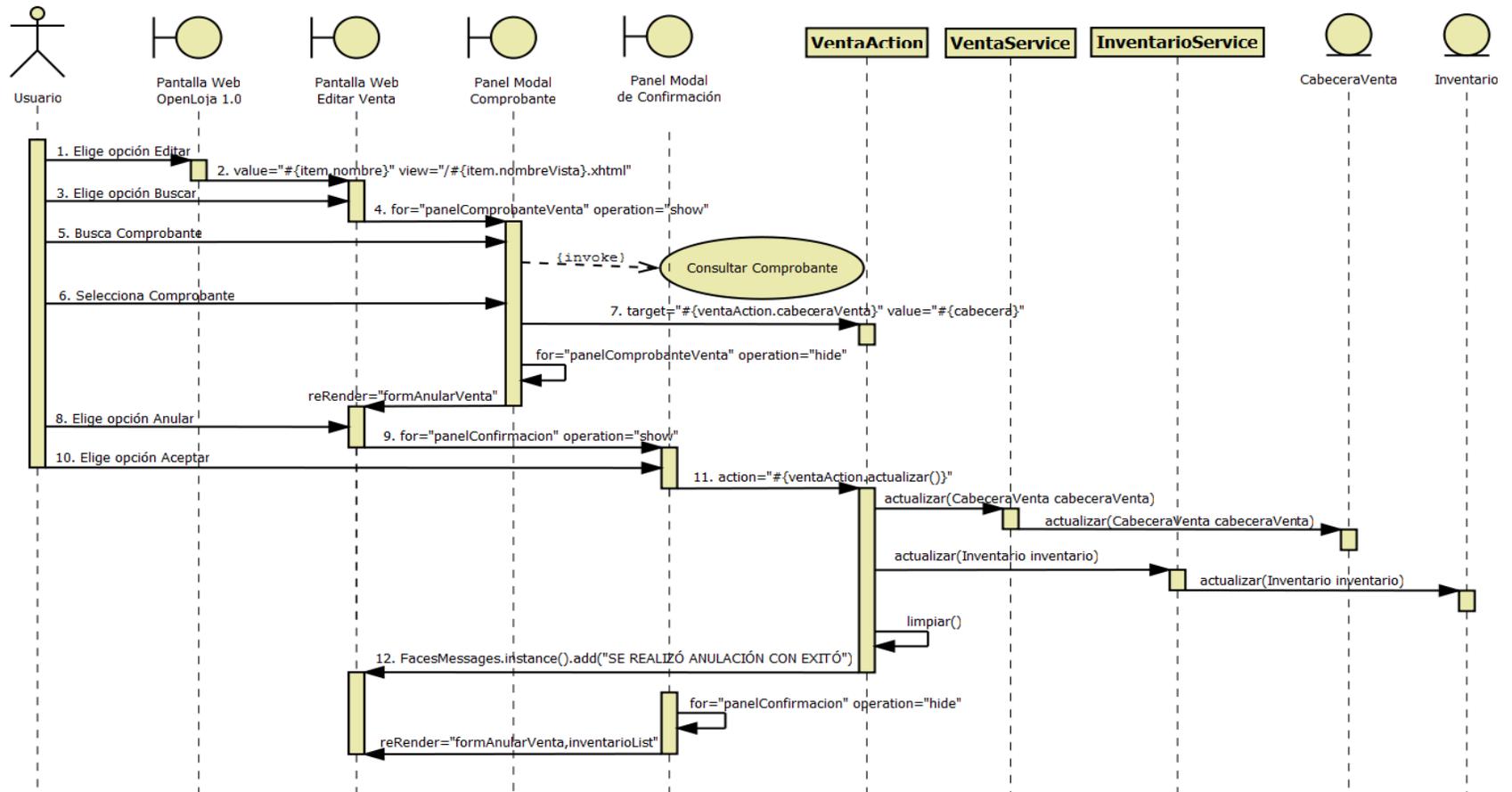


Diagrama 87. Secuencia Modificar Venta Versión Final

3.6.3.3. DIAGRAMA DE SECUENCIA 046: Consultar Venta

(DS046)

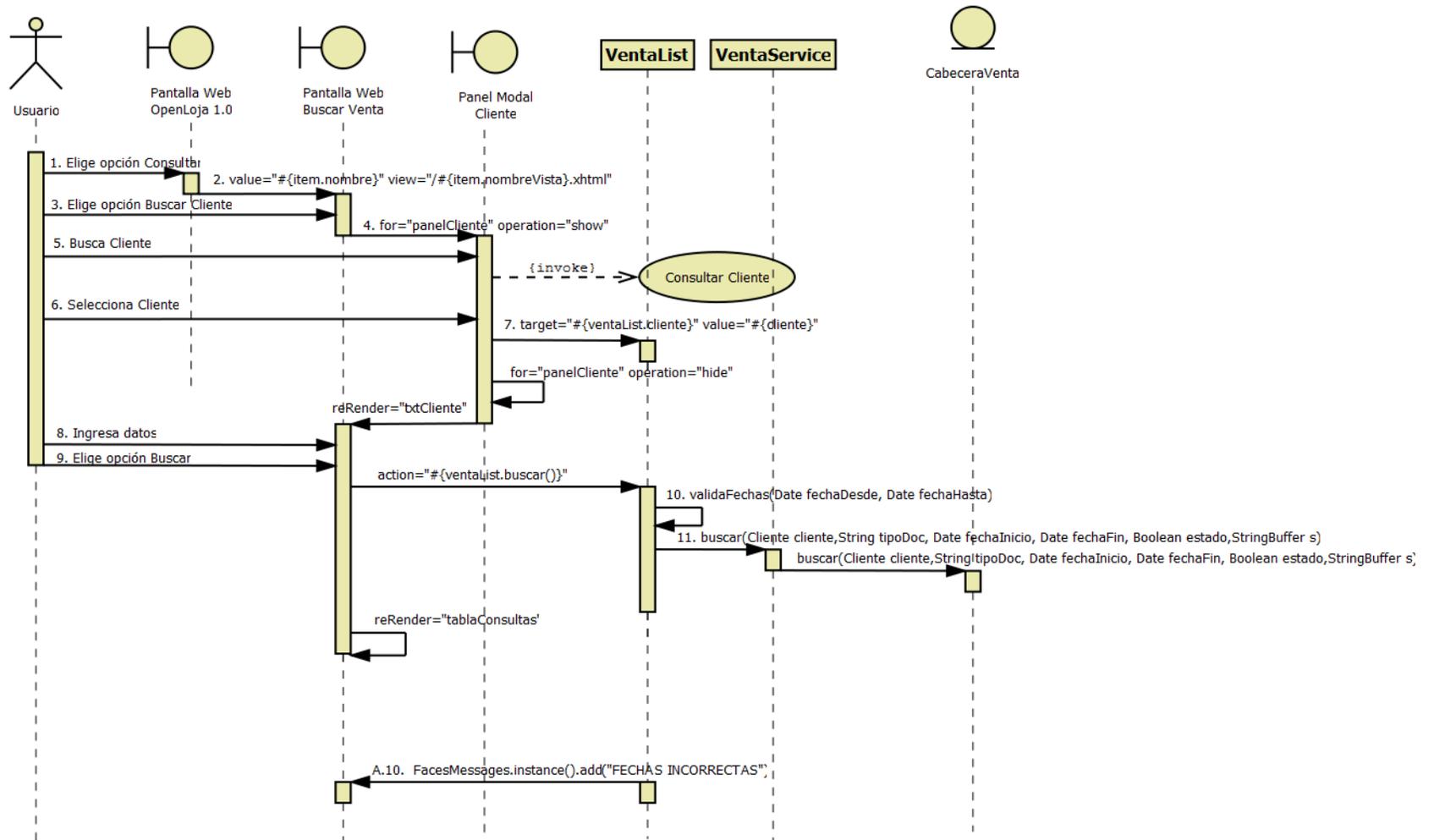


Diagrama 88. Secuencia Consultar Venta Versión Final

3.6.4. DIAGRAMA DE CLASES FINAL

3.6.4.1. CLASES MODELO

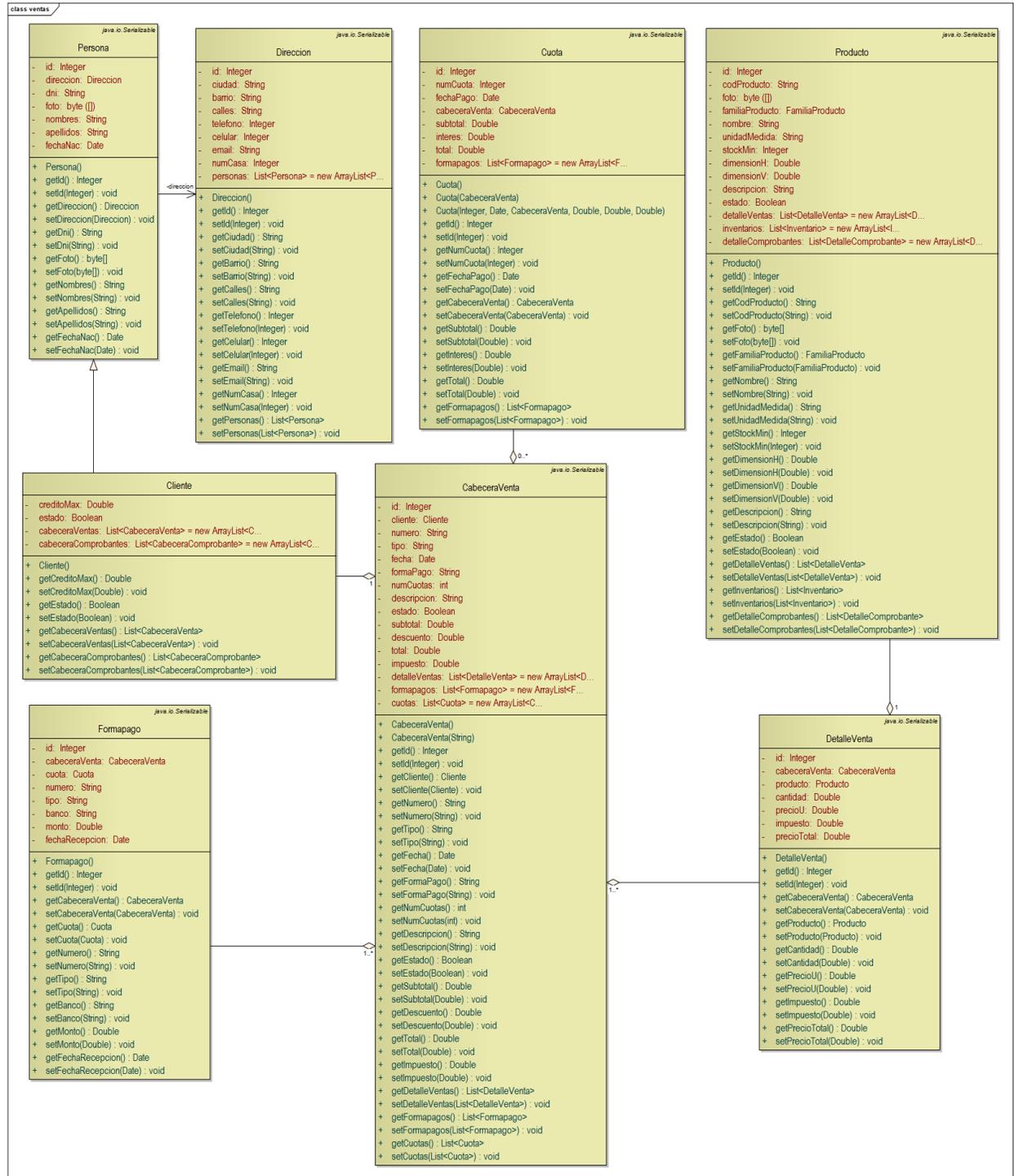


Diagrama 89. Clases Modelo: Gestión de Ventas Versión Final

3.6.4.2. CLASES DAO

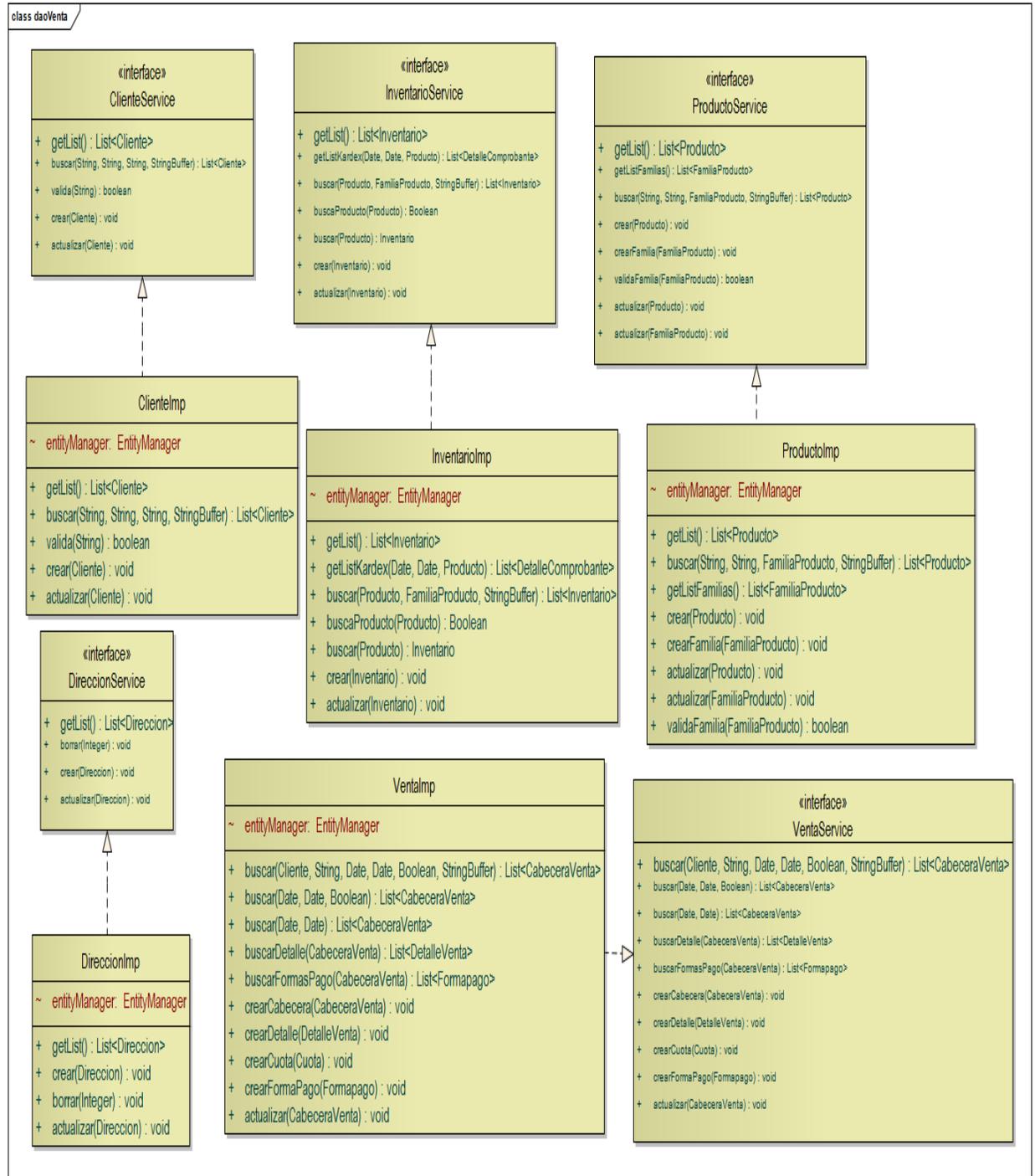


Diagrama 90. Clases DAO: Gestión de Ventas Versión Final

3.6.4.3. CLASES CONTROLADOR

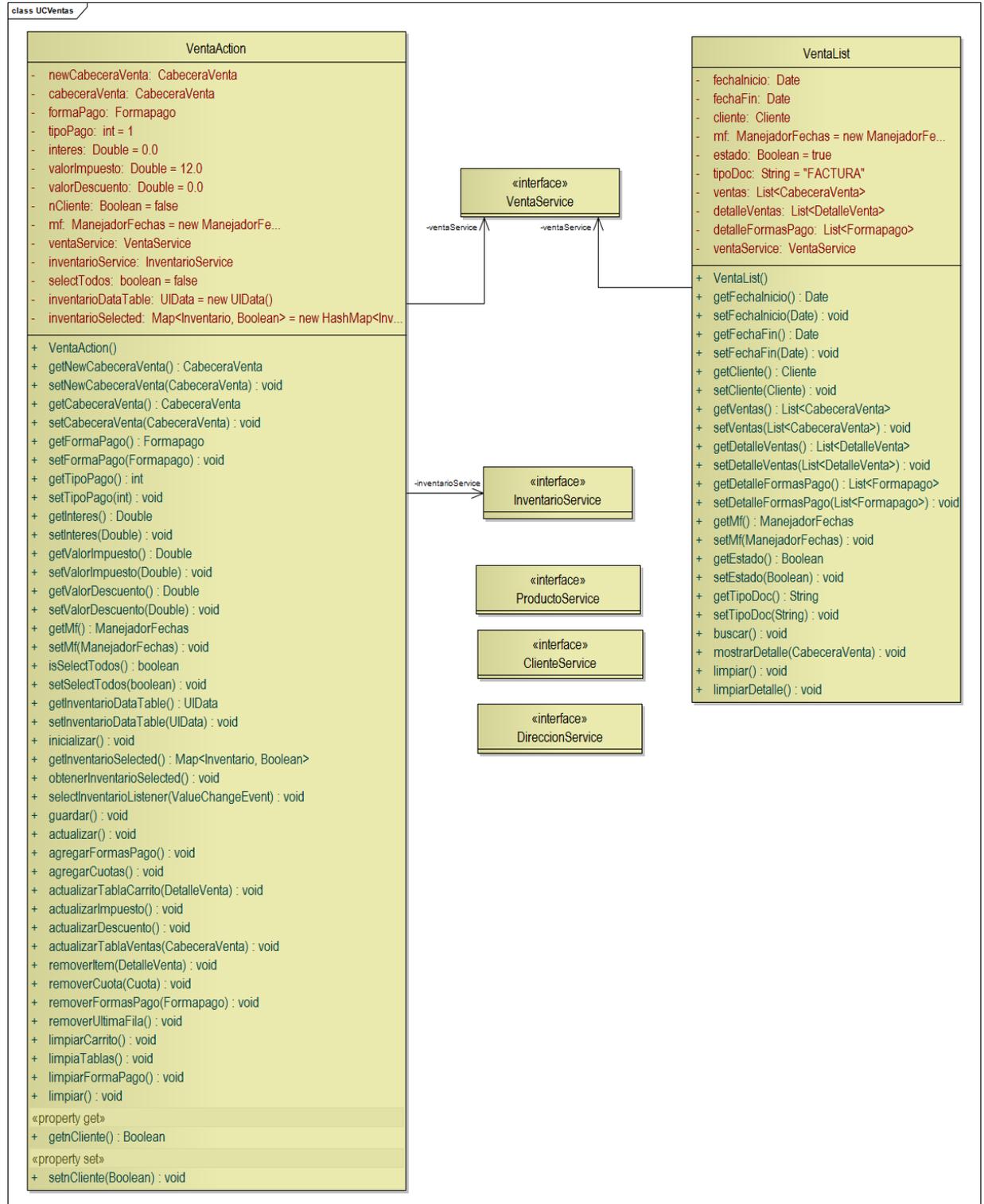


Diagrama 91. Clases Controlador: Gestión de Ventas Versión Final

3.6.4.4. CLASES VISTA

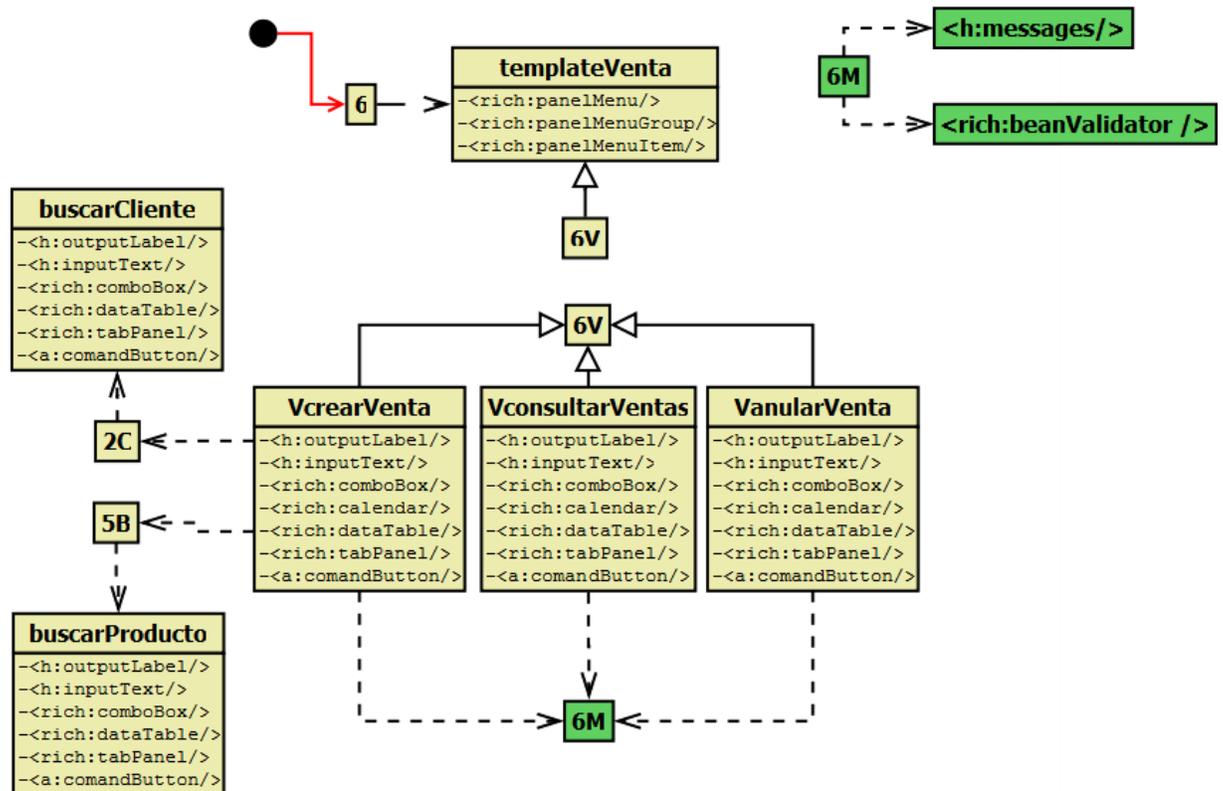


Diagrama 92. Vistas de Ventas Versión Final

3.6.5. DISEÑO DE DATOS

3.6.5.1. MODELO ENTIDAD-RELACIÓN

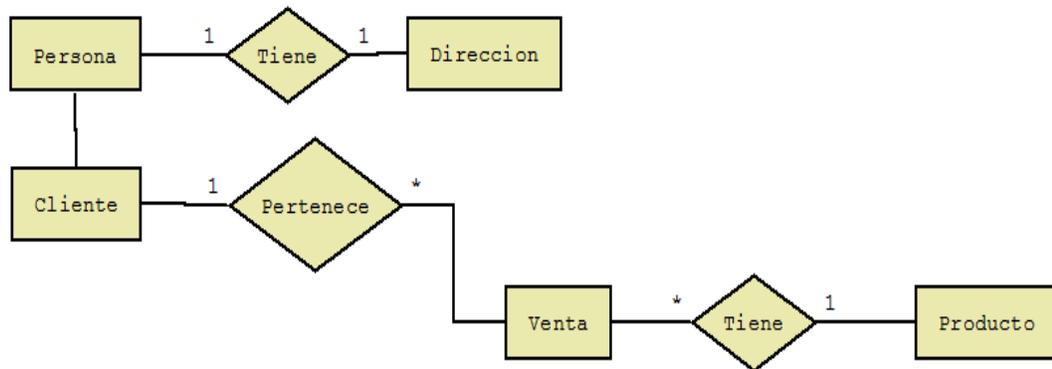


Diagrama 93. Modelo Entidad-Relación: Gestión de Clientes y Ventas Versión Final

3.6.5.2. MODELO CONCEPTUAL

PERSONA

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
dni	Varchar	no		
foto	Blob	si		
nombres	Varchar	no		
apellidos	Varchar	no		
fechaNac	Date	si		
idDireccion	Integer	no		

Tabla 165. Modelo Conceptual: Persona

DIRECCIÓN

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
ciudad	Varchar	no		
barrio	Varchar	si		
calles	Varchar	si		
teléfono	Integer	no		
celular	Integer	si		
email	Varchar	si		

numCasa	Integer	si		
---------	---------	----	--	--

Tabla 166. Modelo Conceptual: Dirección

PRODUCTO

ATRIBUTO	TIPO	NULO	PK	FK
codProducto	Varchar	no		
foto	Blob	si		
nombre	Varchar	no		
unidadMedida	Varchar	no		
stockMin	Integer	si		
dimensionH	Integer	si		
dimensionV	Integer	si		
codFamilia	Integer	no		
descripcion	Varchar	si		
estado	Boolean	no		

Tabla 167. Modelo Conceptual: Producto

FAMILIA_PRODUCTO

ATRIBUTO	TIPO	NULO	PK	FK
id	Numérico	no		
codAsociacion	Cadena	no		
nombre	Cadena	no		
descripcion	Cadena	si		

Tabla 168. Modelo Conceptual: Familia_Producto

PRODUCTO_PROVEEDOR

ATRIBUTO	TIPO	NULO	PK	FK
id	Numérico	no		
codProducto	Numérico	no		
codProveedor	Numérico	no		

Tabla 169. Modelo Conceptual: Producto_Proveedor

INVENTARIO

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
codProducto	Integer	no		
cantidad	Integer	no		
precioA	Doblé	no		
precioB	Doblé	no		
precioC	Doblé	no		

Tabla 170. Modelo Conceptual: Inventario

CABECERA_VENTA

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
numero	Varchar	no		
tipo	Boolean	no		
fecha	Date	no		
formaPago	Varchar	no		
estado	Boolean	no		
descripcion	Varchar	si		
subtotal	Double	no		
total	Double	no		
codCliente	Integer	no		

Tabla 171. Modelo Conceptual: Cabecera_Venta

DETALLE_VENTA

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
idCabecera	Integer	no		
cantidad	Integer	no		
precioU	Double	no		
precioTotal	Double	no		
codProducto	Integer	no		

Tabla 172. Modelo Conceptual: Detalle_Venta

3.6.5.3. MODELO RELACIONAL

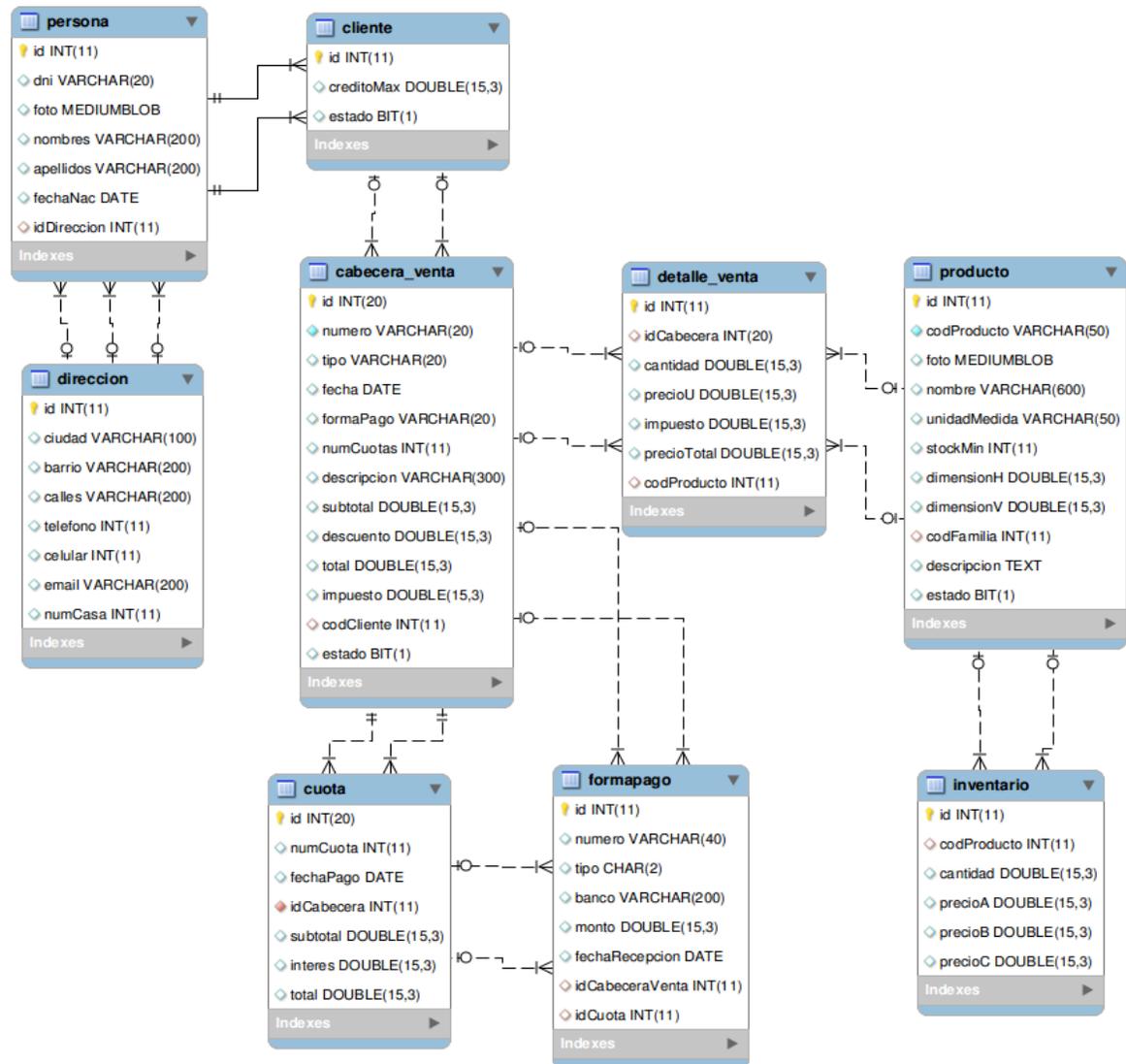


Diagrama 94. Modelo Relacional: Gestión de Ventas Versión Final



GESTIÓN DE COMUNICACIÓN

3.7. MÓDULO DE GESTIÓN DE COMUNICACIÓN

3.7.1. MODELO DE DOMINIO

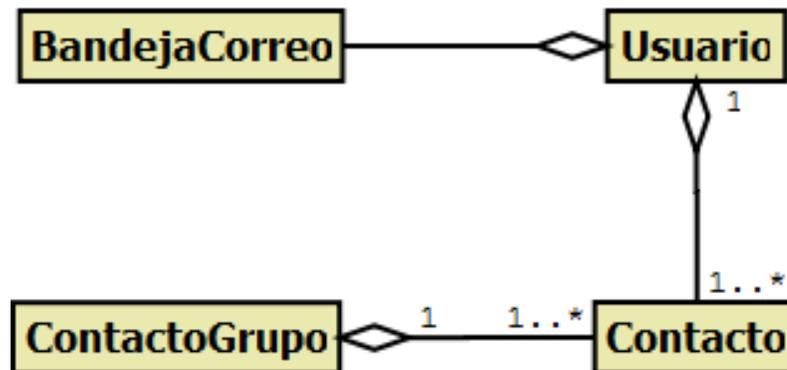


Diagrama 95. Módulo de Comunicación Versión Final

3.7.2. MODELO DE CASOS DE USO DEL SISTEMA

3.7.2.1. DIAGRAMA DE CASOS DE USO

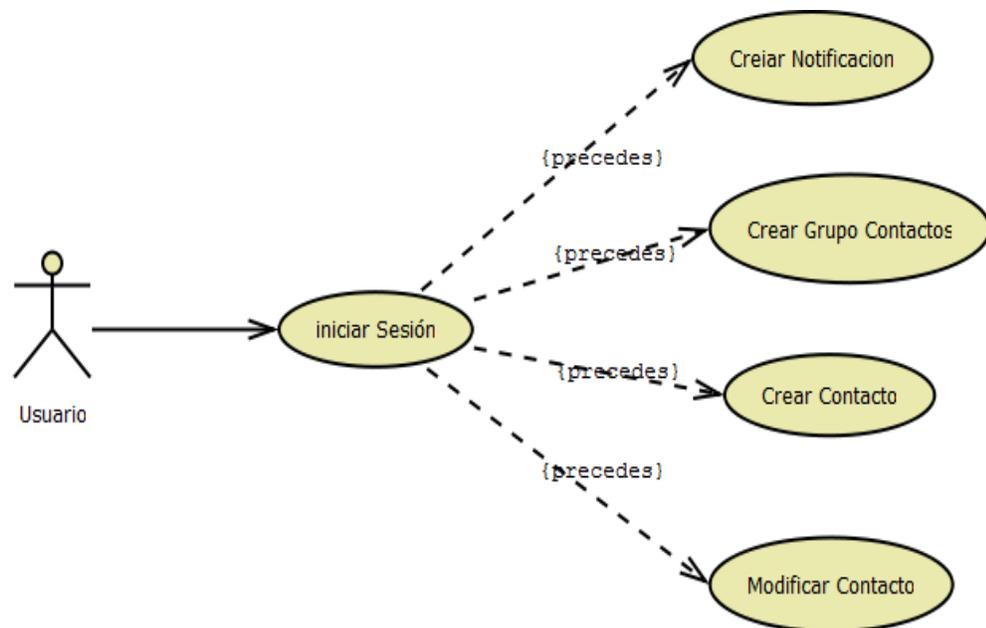


Diagrama 96. Gestión Comunicación Versión Final

3.7.2.2. DESCRIPCIÓN DE CASOS DE USO

➤ CASO DE USO: Crear Notificación

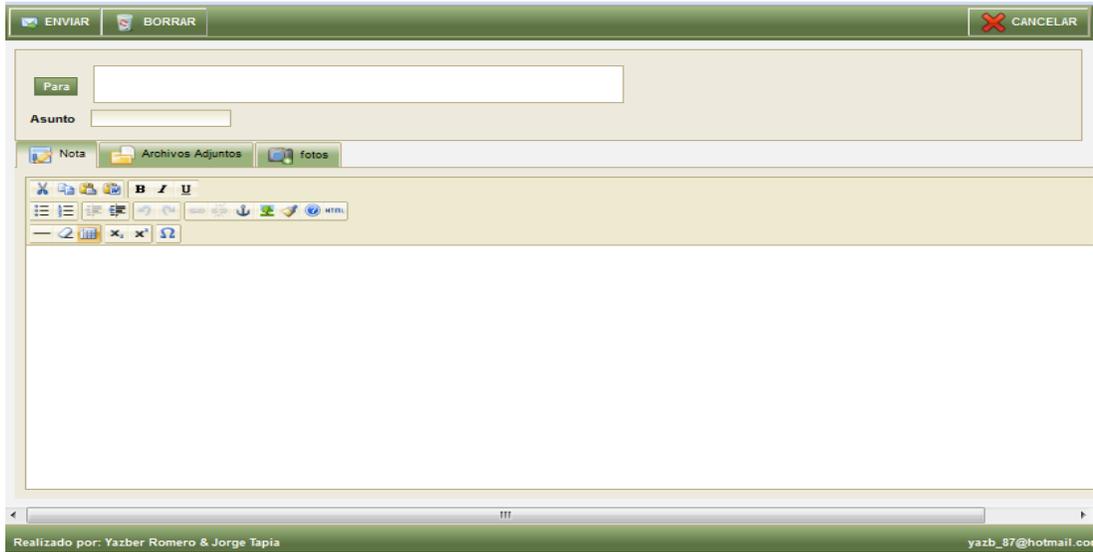
NOMBRE: Crear Notificación	CÓDIGO	OP044
NOMBRE PROGRAMADO	lcrearEmail.xhtml	
		

Tabla 173. Pantalla Crear Notificación

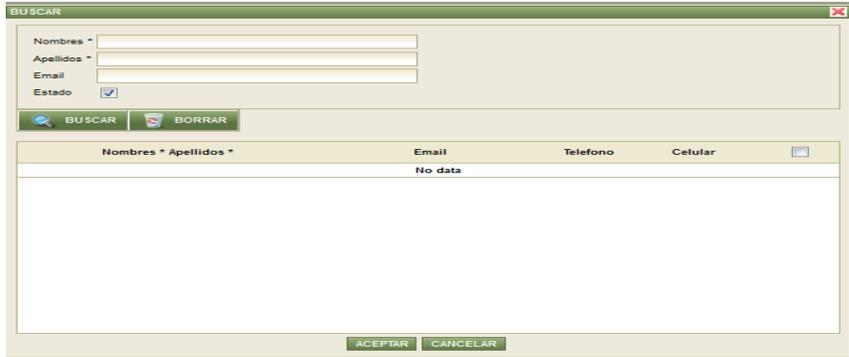
NOMBRE: Panel Modal Contacto	CÓDIGO	PM30
NOMBRE PROGRAMADO	lbandejaNotificacion.xhtml	
		

Tabla 174. Panel Modal Contacto

Nombre:	Crear Notificación	Código: 047
Requerimiento Funcional:	RF027	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol enviar notificaciones vía email a clientes, empleados, proveedores y contactos	
Descripción:	Qué el usuario el cual tenga asignado esta función pueda enviar notificaciones vía email a clientes, empleados, proveedores y contactos	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Comunicación e ítem Bandeja Notificaciones.	
Poscondiciones:	Notificación enviada.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Notificar] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea la Notificación y muestra la pantalla web "Crear Notificación". 3. El usuario selecciona el botón [Para] de la pantalla web "Crear Notificación". 4. El sistema muestra el panel modal "Contacto". 5. El usuario selecciona los Contactos que desea notificar. 6. El usuario elige la opción [Agregar] del panel modal "Contacto" 7. El sistema carga los Contactos seleccionados en la tabla de Contactos de la pantalla web "Crear Notificación" y automáticamente cierra el panel modal "Contacto". 8. El usuario ingresa datos y la información correspondiente de la notificación creada. 9. El usuario selecciona el botón [Enviar]. 10. El sistema valida que los campos obligatorios no estén vacíos. 11. El sistema verifica que hayan Contactos asignados para enviar la Notificación. 12. El sistema envía la notificación a los contactos agregados. 13. El sistema muestra un mensaje de notificación enviada correctamente. 14. El caso de uso finaliza. 		
FLUJO ALTERNO:		
<p>A. CAMPOS OBLIGATORIOS VACIOS</p> <p>A.10. El sistema muestra mensajes en los campos obligatorios vacíos.</p> <p>A.11. El caso de uso continúa en el paso 3 del flujo normal de evento.</p> <p>B. CONTACTO NO ASIGNADO</p> <p>B.11. El sistema muestra un mensaje de Debe asignar contactos.</p> <p>B.12. El caso de uso continúa en el paso 3 del flujo normal de eventos.</p>		

Tabla 175. Descripción Caso de Uso Crear Notificación

➤ CASO DE USO: Crear Grupo Contacto

NOMBRE: Crear Grupo		CÓDIGO	OP045
NOMBRE PROGRAMADO	lcrearGrupo.xhtml		

GUARDAR
BORRAR
CANCELAR

Nombre* Grupo:

Color:

Nota:

Color	Nombre*	Acc
■	Amigos	■
■	Clientes	■
■	Empeados	■
■	Proveedores	■
■	trtryt	■

Realizado por: Yazber Romero & Jorge Tapia
yazb_87@hotmail.com

Tabla 176. Pantalla Crear Grupo

Nombre:	Crear Grupo Contactos	Código: 048
Requerimiento Funcional:	RF028	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado crear grupos de contactos.	
Descripción:	Qué el usuario con esta función asignada pueda crear grupos de contactos cuando lo necesite.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Comunicación e ítem Agenda	
Poscondiciones:	Grupo de Contactos creado.	
FLUJO NORMAL:		
1. El usuario elige la opción [Grupo] de la pantalla web principal "OpenLoja 1.0".		

2. El sistema crea el Grupo de Contactos y muestra la pantalla web “Crear Grupo”.
3. El usuario ingresa los datos correspondientes al Grupo de Contactos creado
4. El usuario selecciona el botón [Guardar] de la pantalla web “Crear Grupo”.
5. El sistema valida que los campos obligatorios no estén vacíos.
6. El sistema verifica mediante el nombre que Grupo de Contactos creado no coincida con otro ya existente.
7. El sistema guarda los datos del Grupo de Contactos creado.
8. El sistema muestra un mensaje de Grupo de Contactos se ha creado con éxito.
9. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACIOS

- A.5. El sistema muestra un mensaje de campos obligatorios vacíos.
- A.6. El caso de uso continúa en el paso 3 del flujo normal de evento.

B. GRUPO DE CONTACTOS EXISTENTE

- B.6. El sistema muestra un mensaje de Grupo de Contactos ya existe.
- B.7. El caso de uso continúa en el paso 3 del flujo normal de evento.

Tabla 177. Descripción Caso de Uso Crear Grupo Contactos

➤ CASO DE USO: Crear Contacto

NOMBRE: Crear Contacto		CÓDIGO	OP046
NOMBRE PROGRAMADO	lcrearContacto.xhtml		

GUARDAR BORRAR CANCELAR

<p>Nombres * <input type="text"/></p> <p>Apellidos * <input type="text"/></p> <p>Email <input type="text"/></p> <p>Compañía <input type="text"/></p> <p>Grupo Amigos</p> <p>Nota <input style="width: 100%;" type="text"/></p> <p>Estado <input checked="" type="checkbox"/></p>	
---	---

Datos Domiciliario

Ciudad

Barrio

Calles

Telefono

Celular

Realizado por: Yazber Romero & Jorge Tapia yazb_87@hotmail.com

Tabla 178. Pantalla Crear Contacto

Nombre:	Crear Contacto	Código:	049
Requerimiento Funcional:	RF028		
Tipo de Caso de Uso:	Sistema	Actores:	Usuario
Objetivo:	Permitir al usuario con este rol asignado crear Contactos.		
Descripción:	Qué el usuario con esta función asignada pueda crear Contactos cuando lo necesite.		
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Comunicación e ítem Agenda		
Poscondiciones:	Contacto creado.		
FLUJO NORMAL:			
<ol style="list-style-type: none"> 1. El usuario elige la opción [Crear Contacto] de la pantalla web principal "OpenLoja 1.0". 2. El sistema crea el Contacto y muestra la pantalla web "Crear Contacto". 			

3. El usuario ingresa los datos correspondientes al Contacto creado
4. El usuario selecciona el botón [Guardar] de la pantalla web "Crear Contacto".
5. El sistema valida que los campos obligatorios no estén vacíos.
6. El sistema verifica mediante los Nombres y Apellidos que el Contacto creado no coincida con otro ya existente.
7. El sistema guarda los datos del Contacto creado.
8. El sistema muestra un mensaje de Contacto se ha creado con éxito.
9. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACIOS

- A.5. El sistema muestra un mensaje de campos obligatorios vacíos.
- A.6. El caso de uso continúa en el paso 3 del flujo normal de evento.

B. CONTACTO EXISTENTE

- B.6. El sistema muestra un mensaje de Grupo de Contactos ya existe.
- B.7. El caso de uso continúa en el paso 3 del flujo normal de evento.

Tabla 179. Descripción Caso de Uso Crear Contacto

➤ CASO DE USO: Modificar Contacto

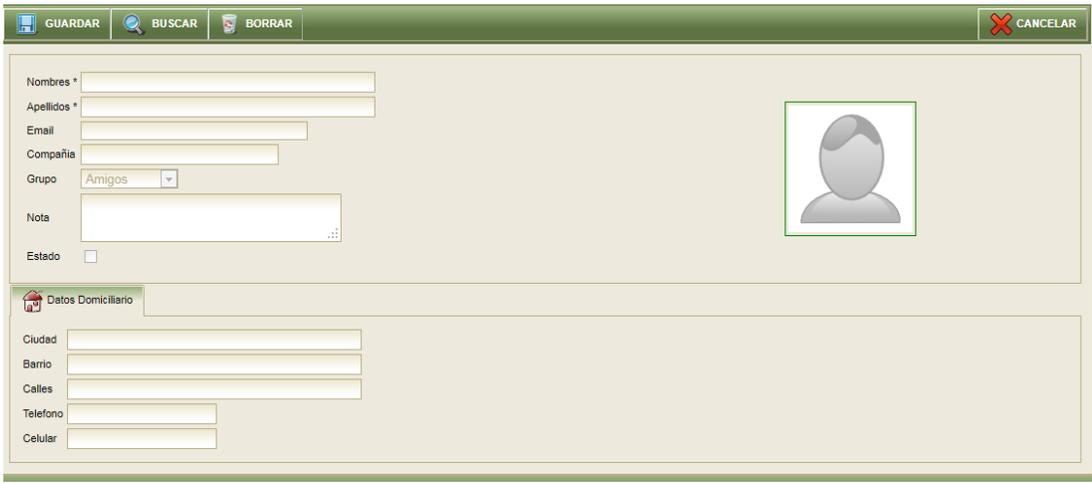
NOMBRE: Editar Contacto		CÓDIGO	OP047
NOMBRE PROGRAMADO	leditarContacto.xhtml		
			

Tabla 180. Pantalla Editar Contacto

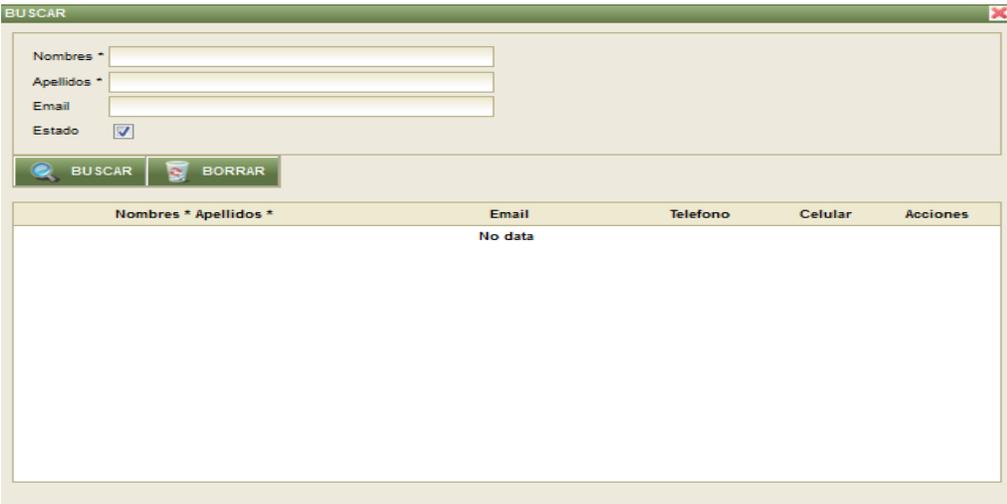
NOMBRE: Panel Modal Contacto		CÓDIGO	PM31
NOMBRE PROGRAMADO	leditarContacto.xhtml		
			

Tabla 181. Panel Modal Contacto

Nombre:	Modificar Contacto	Código: 050
Requerimiento Funcional:	RF029	
Tipo de Caso de Uso:	Sistema	Actores: Usuario
Objetivo:	Permitir al usuario con este rol asignado editar Contactos.	
Descripción:	Qué el usuario con esta función asignada pueda editar Contactos cuando lo necesite.	
Precondiciones:	El usuario debe estar logueado. El usuario debe ubicarse en la ruta Menú Aplicación, Submenú Gestión de Comunicación e ítem Agenda	
Poscondiciones:	Contacto seleccionado modificado.	
FLUJO NORMAL:		
<ol style="list-style-type: none"> 1. El usuario elige la opción [Editar Contacto] de la pantalla web principal "OpenLoja 1.0". 2. El sistema muestra la pantalla web "Editar Contacto". 3. El usuario elige la opción [Buscar Contacto] de la pantalla web "Editar Contacto". 4. El sistema muestra el Panel modal "Contacto". 5. El usuario ingresa los datos del Contacto que busca para editar. 6. El usuario elige la opción [Buscar] del panel modal "Contacto". 7. El sistema muestra los Contactos que coinciden con la búsqueda en la Tabla de resultados del panel modal "Contacto". 8. El usuario selecciona el Contacto que busca del panel modal "Contacto". 9. El sistema carga los datos del Contacto seleccionado en la pantalla web "Editar Contacto" y automáticamente cierra el panel modal "Contacto". 10. El usuario modifica los datos correspondientes al Contacto seleccionado. 11. El usuario selecciona el botón [Guardar] de la pantalla web "Editar Contacto". 		

12. El sistema valida que los campos obligatorios no estén vacíos.
13. El sistema verifica mediante los Nombres y Apellidos que el Contacto modificado no coincida con otro ya existente.
14. El sistema guarda los nuevos datos del Contacto seleccionado.
15. El sistema muestra un mensaje de Contacto se ha modificado con éxito.
16. El caso de uso finaliza.

FLUJO ALTERNO:

A. CAMPOS OBLIGATORIOS VACIOS

- A.12. El sistema muestra un mensaje de campos obligatorios vacíos.
- A.13. El caso de uso continúa en el paso 3 del flujo normal de evento.

B. CONTACTO EXISTENTE

- B.13. El sistema muestra un mensaje de Grupo de Contactos ya existe.
- B.14. El caso de uso continúa en el paso 10 del flujo normal de evento.

Tabla 182. Descripción Caso de Uso Modificar Contacto

3.7.3. MODELO DE INTERACCIÓN

3.7.3.1. DIAGRAMA DE SECUENCIA 047: Crear Notificación (DS047)

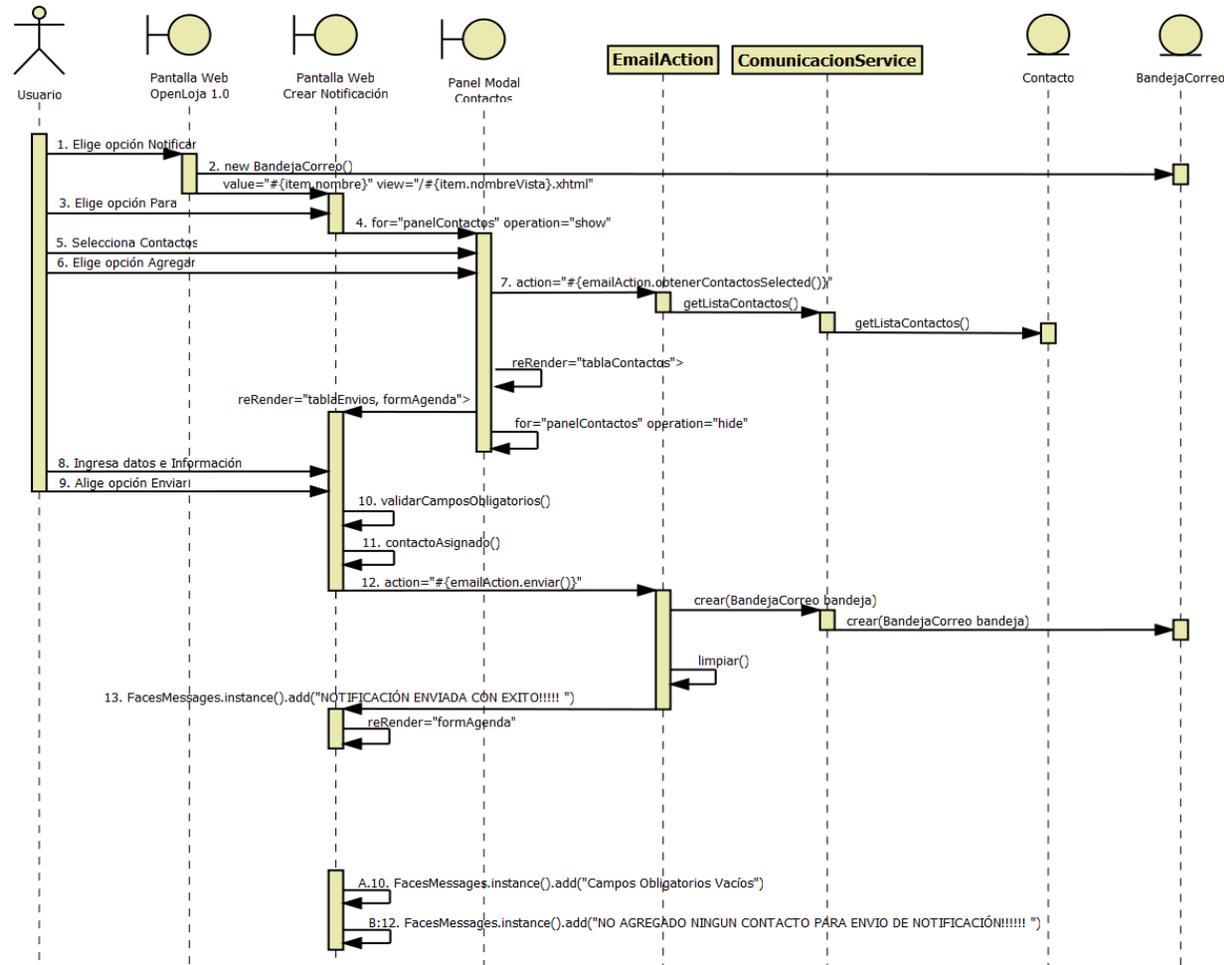


Diagrama 97. Secuencia Crear Notificación Versión Final

3.7.3.2. DIAGRAMA DE SECUENCIA 048: Crear Grupo contacto (DS048)

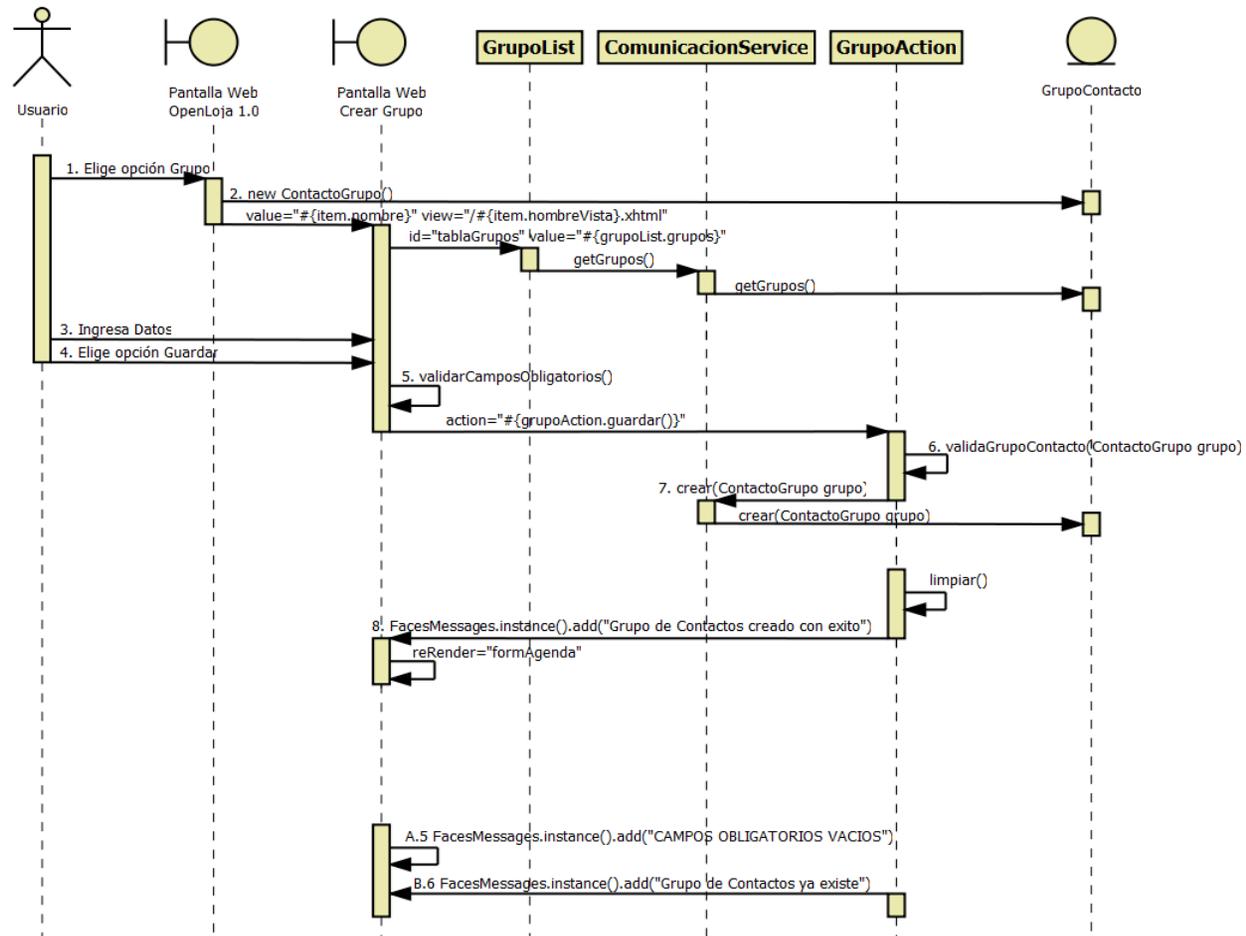


Diagrama 98. Secuencia Crear Grupo Contacto Versión Final

3.7.3.3. DIAGRAMA DE SECUENCIA 049: Crear Contacto (DS049)

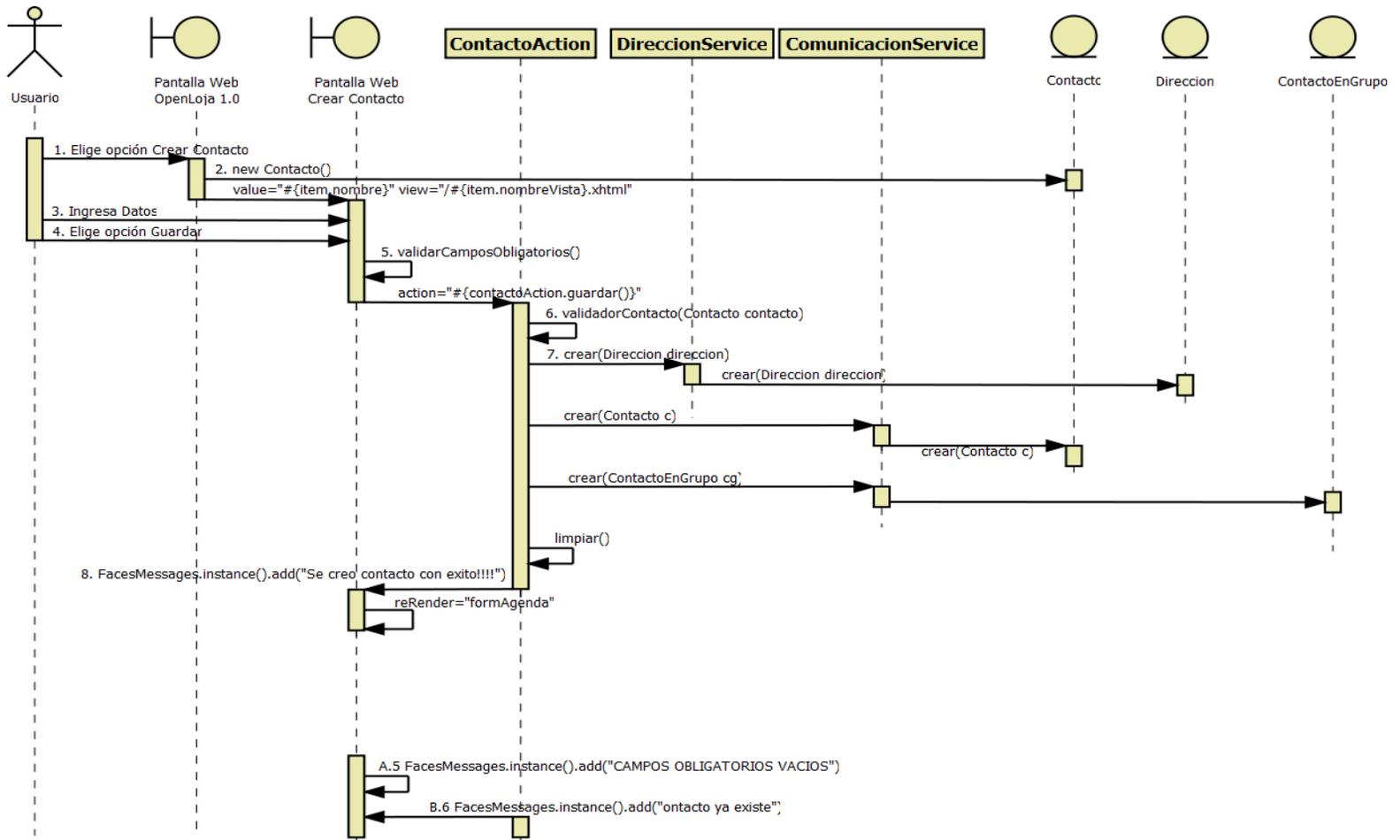


Diagrama 99. Secuencia Crear Contacto Versión Final

3.7.3.4. DIAGRAMA DE SECUNCIA 050: Modificar Contacto (DS050)

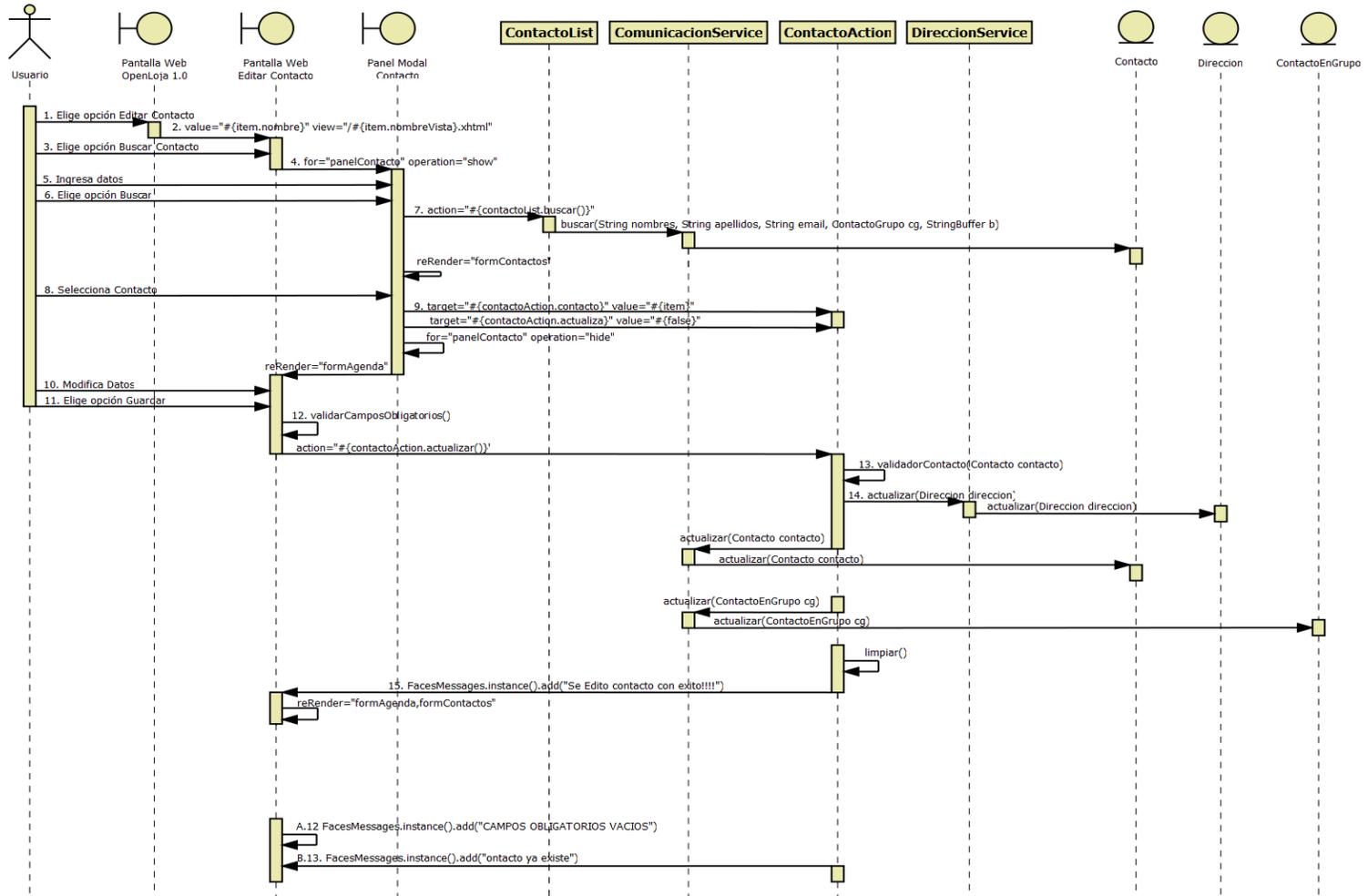


Diagrama 100. Secuencia Modificar Contacto Versión Final

3.7.4. DIAGRAMA DE CLASES FINAL

3.7.4.1. CLASES DE MODELO

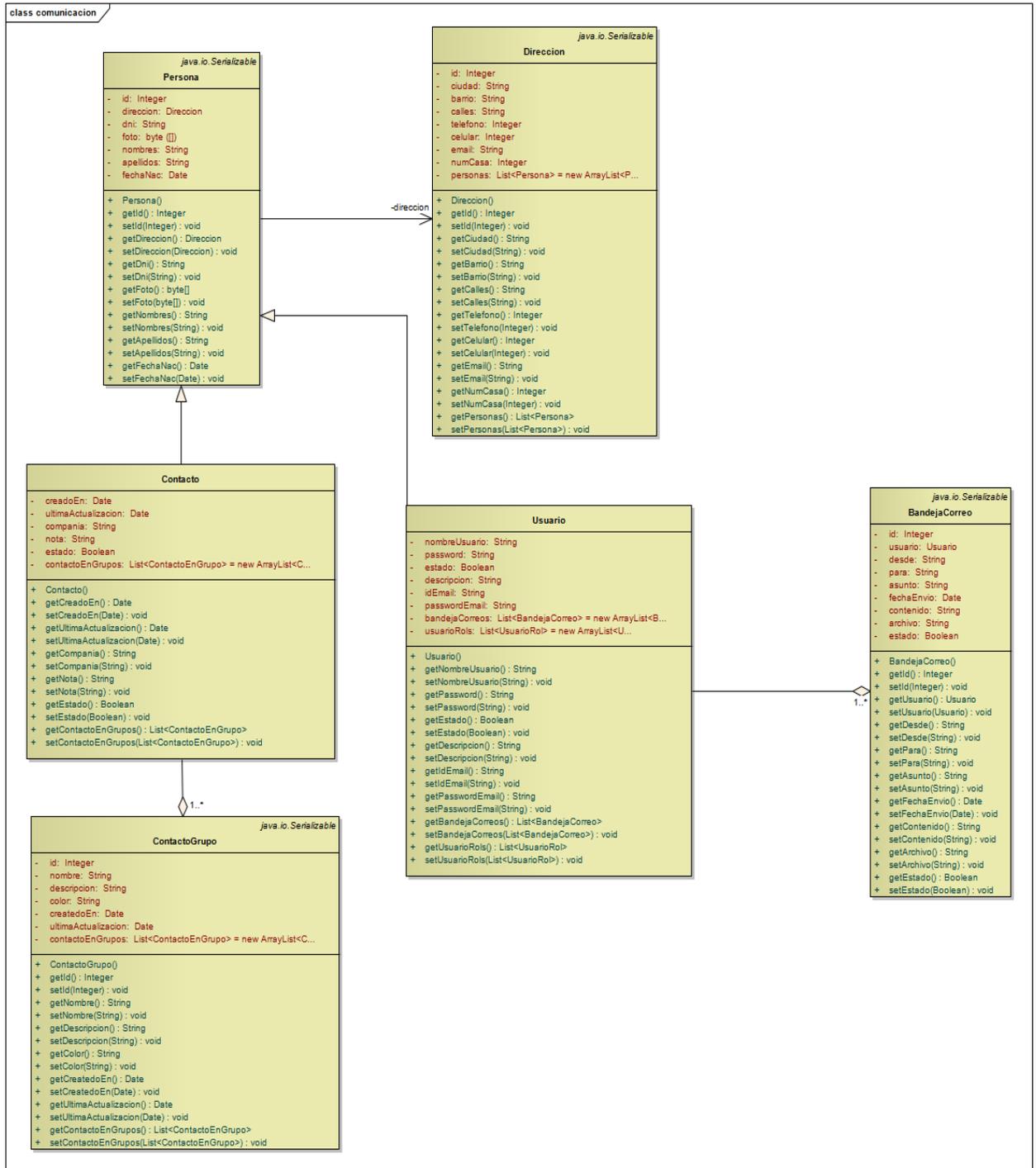


Diagrama 101. Clases Modelo: Gestión de Comunicación Versión Final

3.7.4.2. CLASES DAO

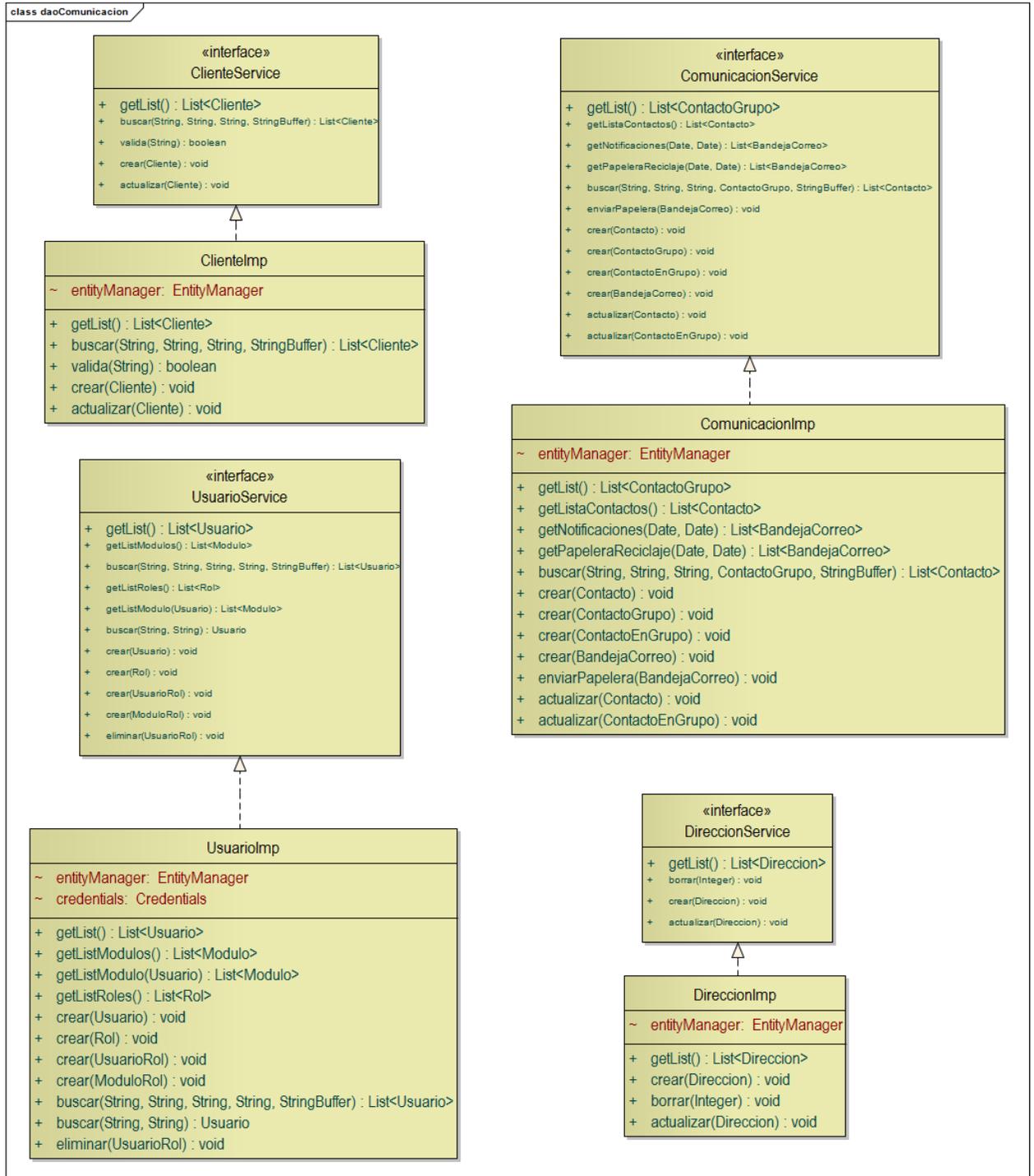


Diagrama 102. Clases DAO: Gestión de Comunicación Versión Final

3.7.4.3. CLASES CONTROLADOR

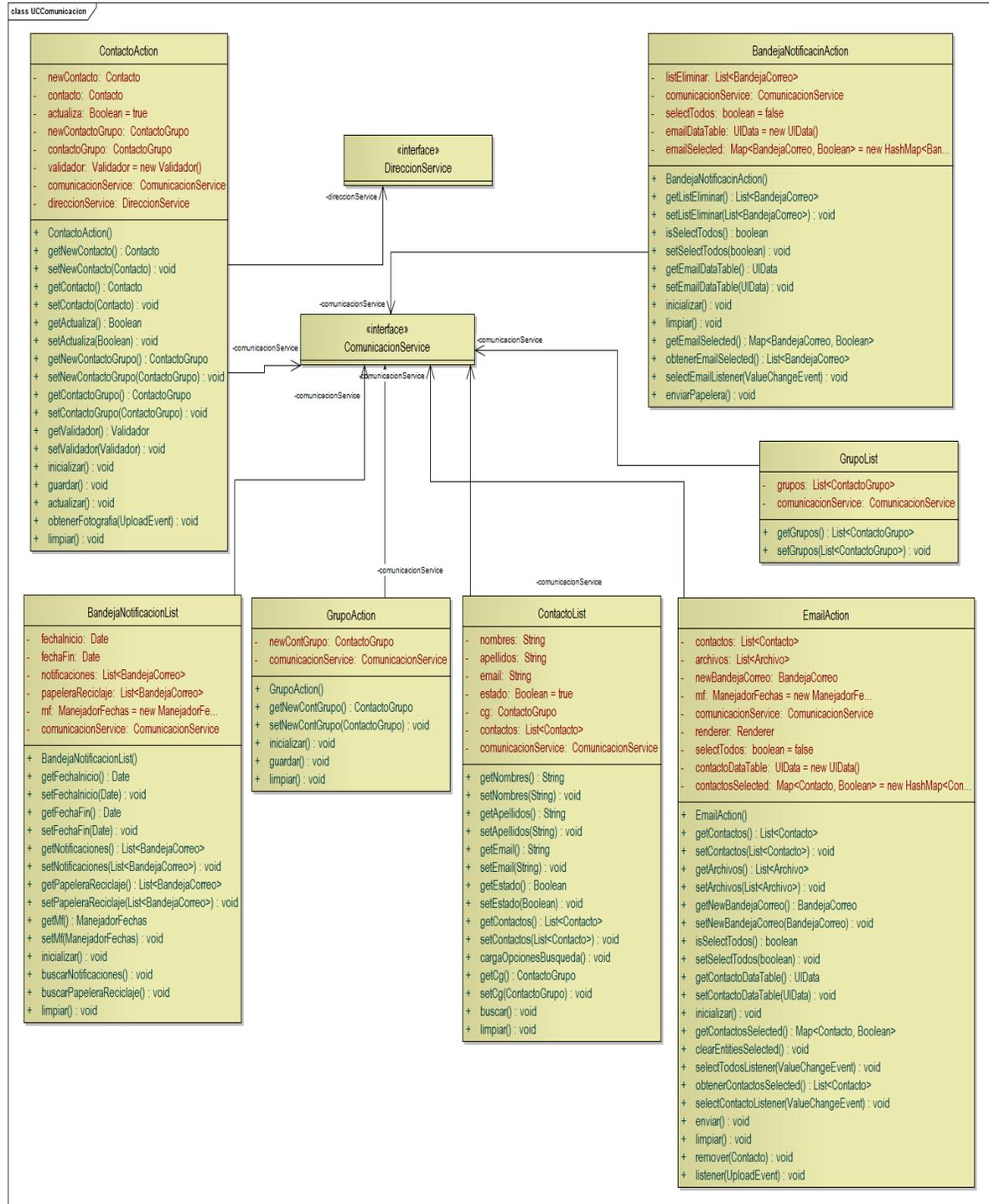


Diagrama 103. Clases Controlador: Gestión de Comunicación Versión Final

3.7.4.4. CLASES VISTA

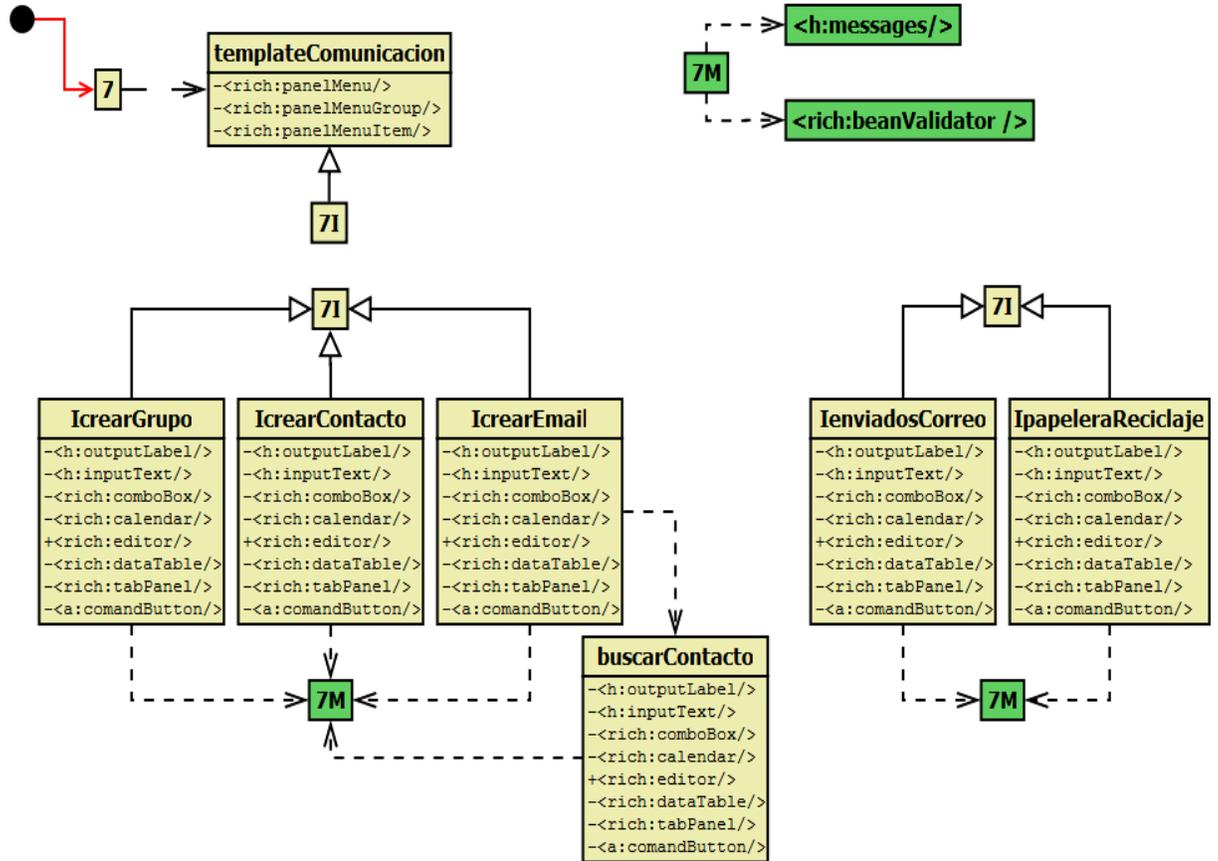


Diagrama 104. Vistas de Comunicación Versión Final

3.7.5. DISEÑO DE DATOS

3.7.5.1. MODELO ENTIDAD-RELACIÓN

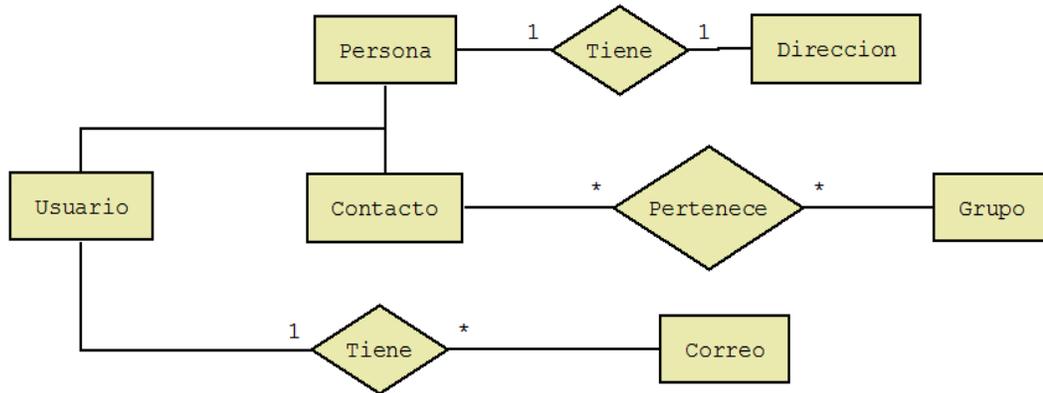


Diagrama 105. Modelo Entidad-Relación: Gestión de Comunicación Versión Final

3.7.5.2. MODELO CONCEPTUAL

PERSONA

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
dni	Varchar	no		
foto	Blob	si		
nombres	Varchar	no		
apellidos	Varchar	no		
fechaNac	Date	si		
idDireccion	Integer	no		

Tabla 183. Modelo Conceptual: Persona

DIRECCIÓN

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
ciudad	Varchar	no		
barrio	Varchar	si		
calles	Varchar	si		
teléfono	Integer	no		
celular	Integer	si		
email	Varchar	si		
numCasa	Integer	si		

Tabla 184. Modelo Conceptual: Dirección

CONTACTO

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
creadoEn	Date	no		
compania	Varchar	si		
nota	Varchar	si		
estado	Boolean	no		

Tabla 185. Modelo Conceptual: Contacto

CONTACTO_GRUPO

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	No		
nombre	Varchar	No		
descripcion	Varchar	Si		
creadoEn	Date	No		
ultimaActualizacion	Date	Si		

Tabla 186. Modelo Conceptual: Contacto_Grupo

CONTACTO_EN_GRUPO

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
codGrupo	Integer	no		
codContacto	Integer	no		

Tabla 187. Modelo Conceptual: Contacto_En_Grupo

BANDEJA_CORREO

ATRIBUTO	TIPO	NULO	PK	FK
id	Integer	no		
desde	Varchar	no		
para	Varchar	no		
asunto	Varchar	si		
fechaEnvio	Date	no		
contenido	Varchar	si		
codUsuario	Integer	no		
archivo	Varchar	si		
estado	Boolean	no		

Tabla 188. Bandeja_Correo

3.7.5.3. MODELO RELACIONAL

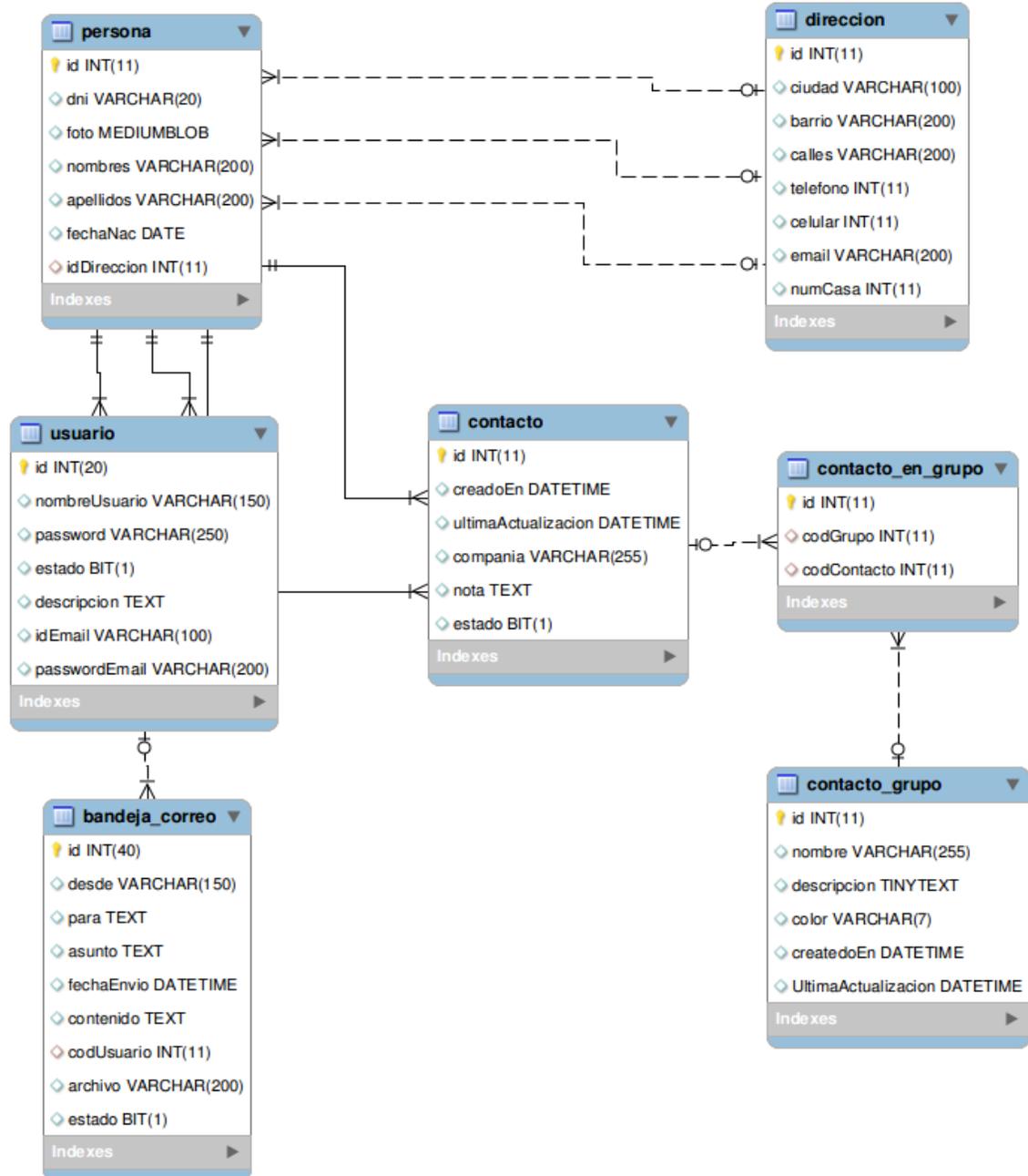


Diagrama 106. Modelo Relacional: Gestión Comunicación Versión Final

4. DISEÑO DE LA ARQUITECTURA

4.1. DIAGRAMA DE COMPONENTES

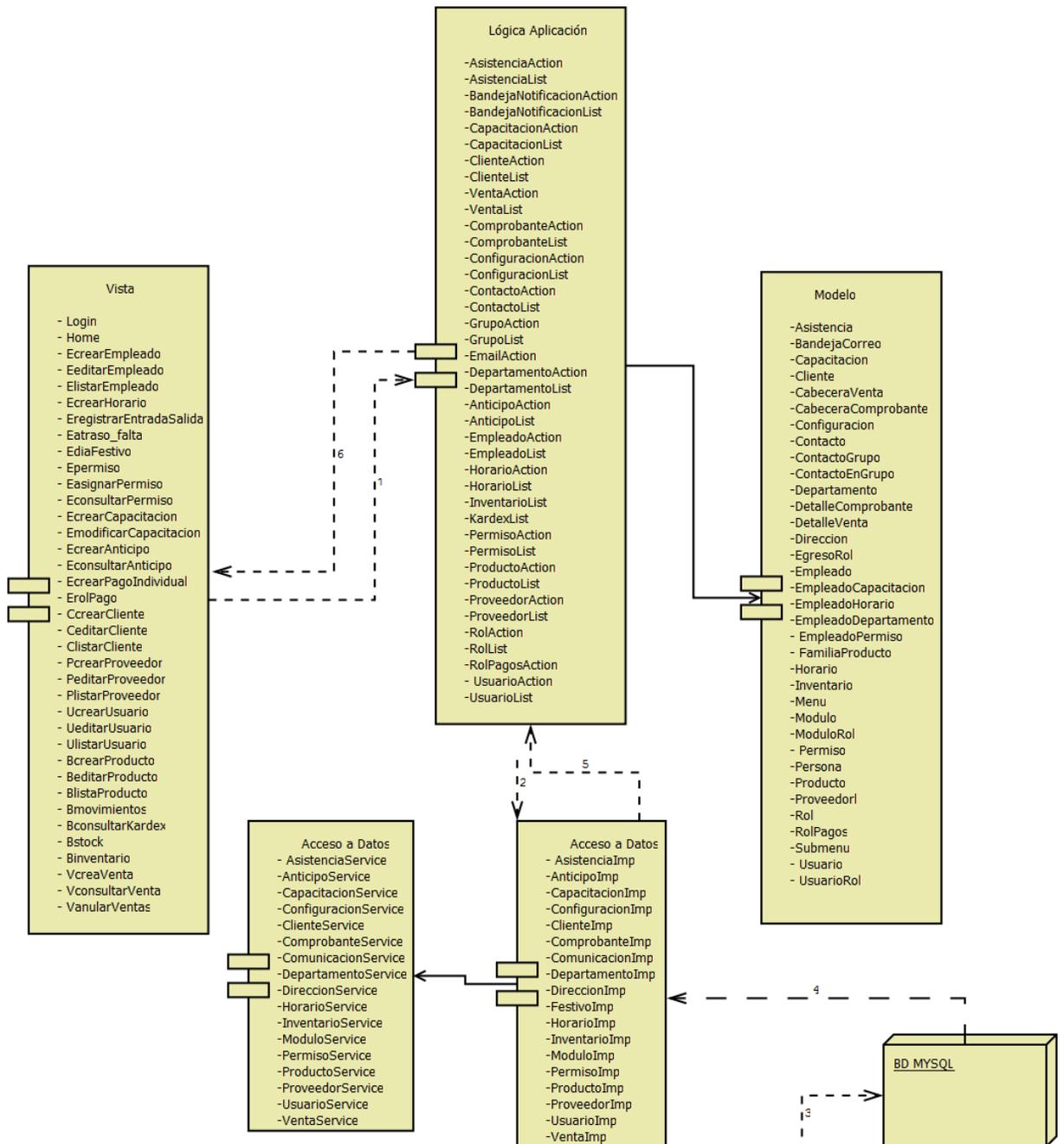


Diagrama 107. Componentes Versión Final

4.2. DIAGRAMA DE DESPLIEGUE

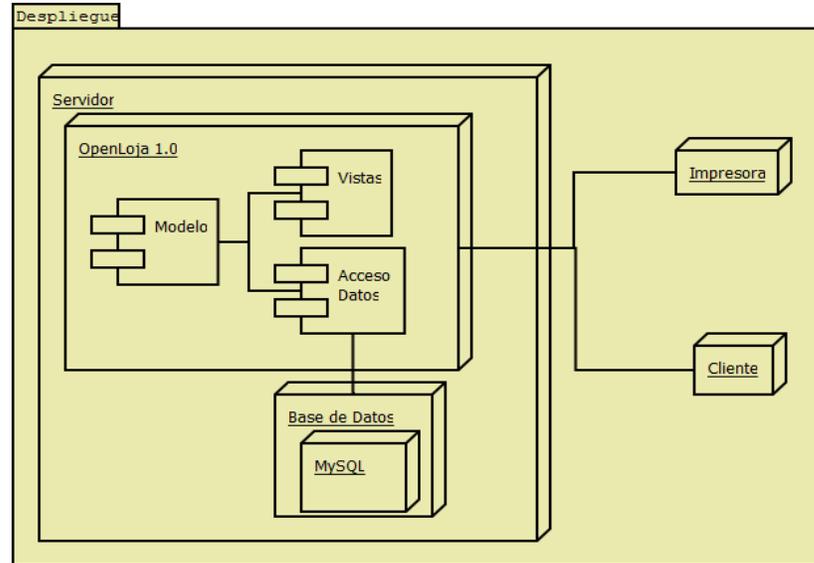


Diagrama 108. Despliegue Versión Final

4.3. DIAGRAMA DE PAQUETES

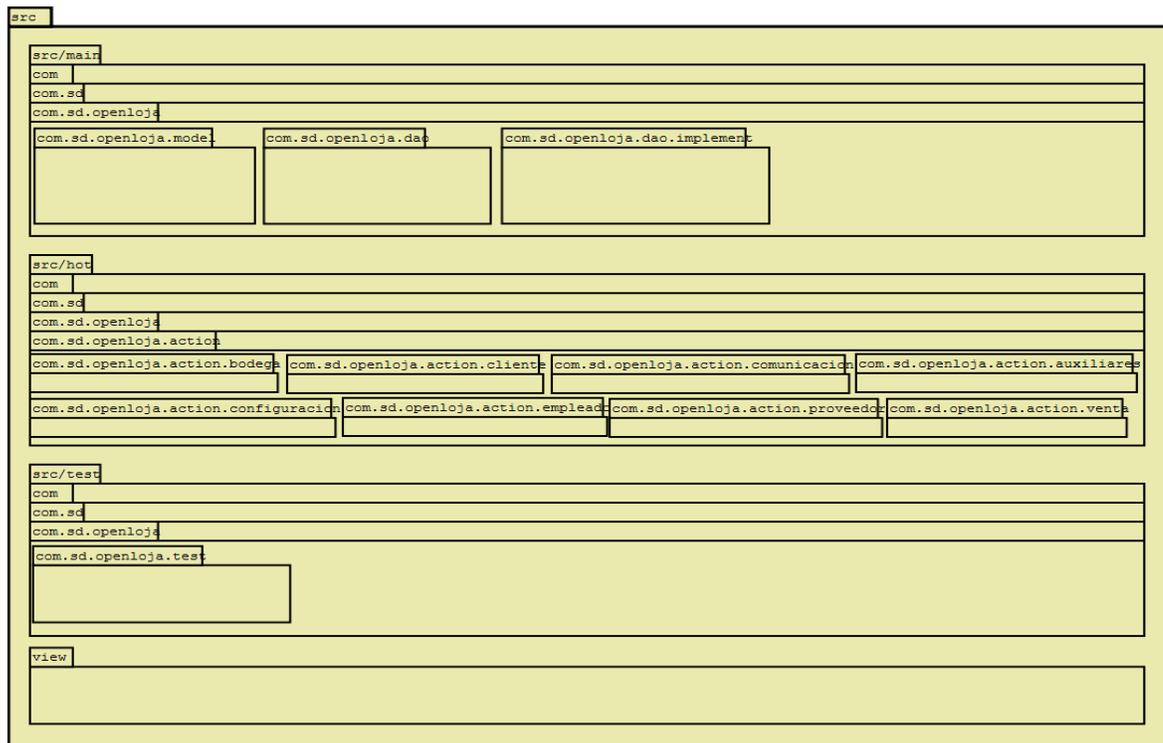


Diagrama 109. Paquetes Versión Final

4.4. MODELO DE LA ARQUITECTURA

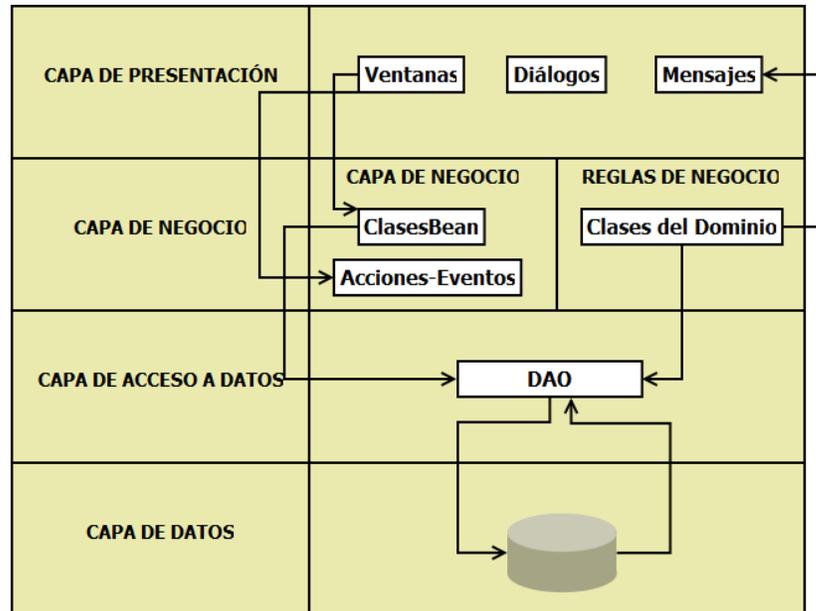


Diagrama 110. Modelo de la Arquitectura Versión Final



DISEÑO DE NAVEGACIÓN

5. DISEÑO NAVEGACIÓN

5.1. INTRODUCCIÓN

Actualmente la navegación es considerada un paso crítico en el diseño de aplicaciones. Un modelo navegación es construido como una vista sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegación provee una vista subjetiva del diseño conceptual. A continuación se describe como es el despliegue de la aplicación, indicando la navegabilidad que se les ofrecerá a los usuarios en la aplicación web. La aplicación se compone de estas cuatro zonas bien diferenciadas:

- **Cabecera.** Se mostrarán aspectos básicos y relevantes de la aplicación así como el menú de acceso a los diferentes módulos de la aplicación en codependencia con el rol asignado.
- **Menú.** Contendrá un menú específico relacionado con el módulo previamente seleccionado por el usuario que le permita al usuario realizar las operaciones que se necesitan en cada momento.
- **Bloque principal.** Representa la zona donde se cargara cada formulario que interactuará principalmente, mostrándose todo tipo de información.
- **Pie de página.** Contendrá información general sobre la aplicación.

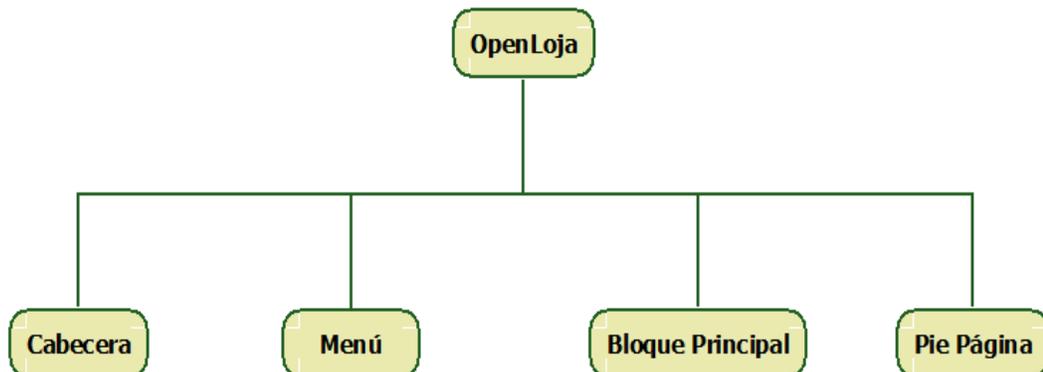


Ilustración 9. Zonas de Navegación

Seguidamente describiremos cada uno de los bloques, indicamos cada uno de los diagramas específicos.

5.2. CABECERA

Ubicada en la parte superior del navegador, contendrá el menú general de cada módulo, además un botón Logout para poder salir de la aplicación en cada momento.

El sistema muestra en un esquema de las partes que forman parte de la cabecera.

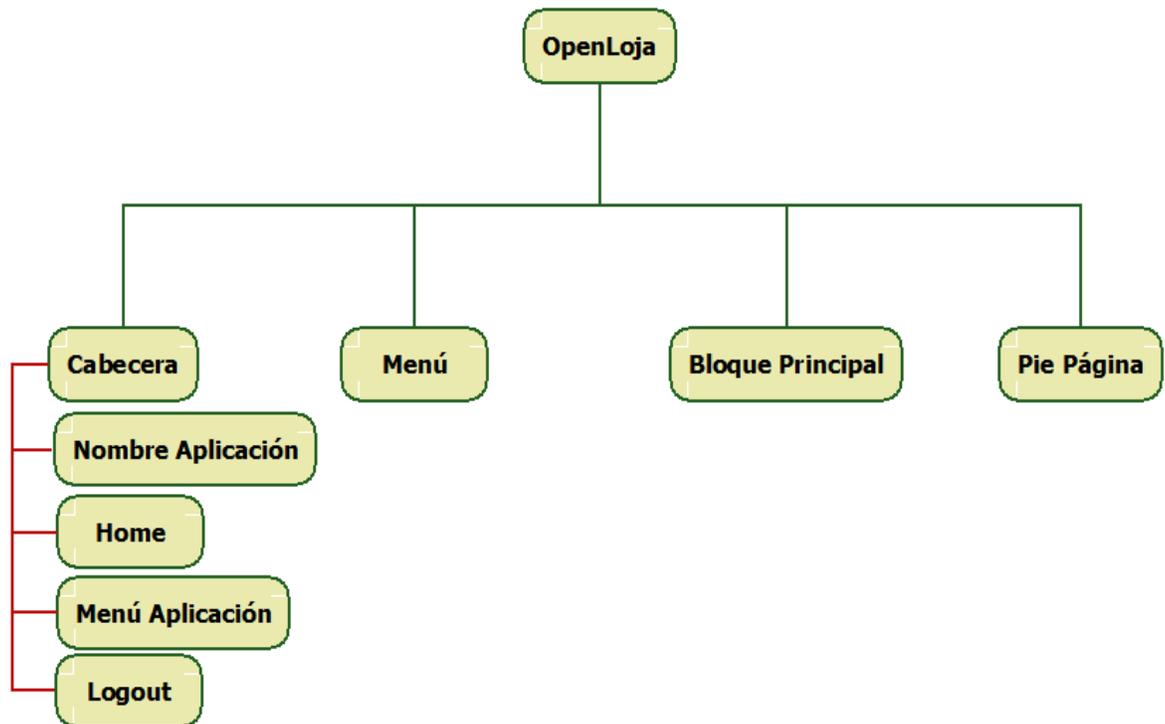


Ilustración 10. Descripción de Cabecera

- **Nombre Aplicación.**-Nombre asignado a la aplicación.
- **Home.**- Permite acceder a la pantalla principal del sistema.
- **Menú Aplicación.**- Contiene cada uno de los accesos a los diferentes módulos de la aplicación dependiendo del rol de usuario.
- **Logout.**- Opción disponible en todo momento para cerrar sesión de trabajo.



Ilustración 11. Menú de la Aplicación

5.3. MENÚ

El menú ubicado en la parte izquierda contiene las distintas opciones correspondientes a las diversas operaciones que tiene acceso el usuario, presenta una lista de accesos específicos.

El sistema muestra en un esquema de las partes que forman parte del menú.

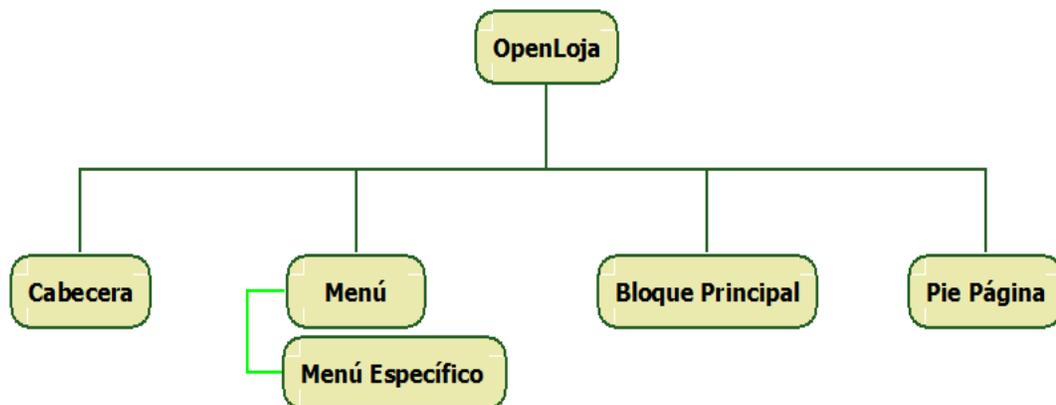


Ilustración 12. Descripción de Menú

- **Menú Específico.-** La parte descrita como menú específico de la aplicación es una de las secciones claves que permite al usuario poder acceder a las diversas operaciones que puede realizar el usuario para poderlas visualizar es necesario seleccionar

previamente de menú aplicación el módulo al que usted tiene acceso, previa asignación de su administrador.

Para entender mejor en un ejemplo a continuación se muestra el conjunto de opciones que se dispone si se a seleccionada el módulo de Gestión de Empleados descrito en la sección anterior.



Ilustración 13. Ejemplo Menú Empleado

5.4. BLOQUE PRINCIPAL.

Ubicada en la parte derecha, junto a la sección de menú, y cuya proporción visual es la que predomina de las secciones anteriores, para tener una idea más clara de esta sección la hemos detallado en su respectivo diagrama y descripción pertinente.

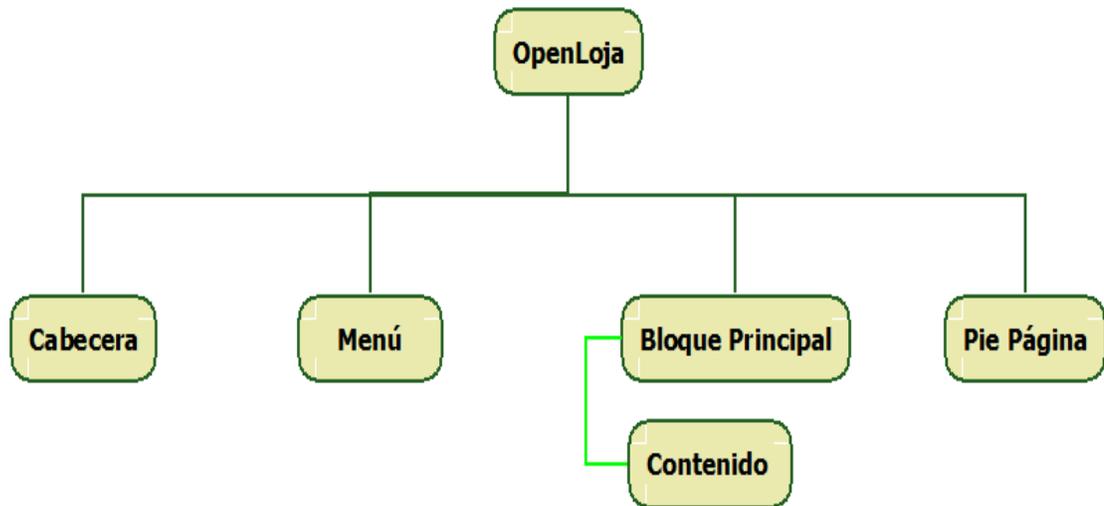


Ilustración 14. Descripción de Bloque Principal

- **Contenido.-** El bloque principal o también denominado cuerpo, tiene como objetivo proveer de un contenedor de formularios que permiten visualizar, y que son respuesta de la acción seleccionada de la sección del menú específico indicado en la sección anterior.



Ilustración 15. Ejemplo de Contenedor de Formularios

5.5. PIE DE PÁGINA

Esta sección está dedicada de manera específica a brindar información general de la aplicación ubicada en la parte inferior, puede contener información que detalla la autoría de la aplicación, email de contacto.

Realizado por: Yazber Romero & Jorge Tapia

yazb_87@hotmail.com

Ilustración 16. Pie de Página



PLAN DE VALIDACIÓN

6. PLAN DE VALIDACIÓN

6.1. INTRODUCCIÓN

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un desarrollo software. Básicamente es una fase más dentro del desarrollo del software consistente en probar las aplicaciones construidas.

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

Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

“El testing puede probar la presencia de errores pero no la ausencia de ellos”
(Dijkstra)

Para el desarrollo del presente plan de validación hemos creído conveniente dividirlo en varios puntos entre los que tenemos:

- Pruebas de Interfaces Gráficas.
- Pruebas de Funcionalidad.
- Pruebas de Seguridad.

Para lo cual hemos realizado encuestas tanto al encargado o administrador del sistema, como a los encargados de los departamentos que abarca el presente proyecto que son 5 en total. A continuación se detalla cada uno de los puntos:

6.2. PRUEBAS DE INTERFACES GRÁFICAS.

Durante el desarrollo de esta etapa del plan se pretende realizar un seguimiento con respecto a la respuesta visual de cada uno de los componentes que hemos hecho uso para el desarrollo del presente proyecto.

Tomando en consideración lo anteriormente expuesto se realizan pruebas sobre cada uno de los diferentes formularios involucrados, En este aspecto, algunas de las validaciones más importantes deben ser las siguientes:

- **Campos obligatorios:** se debe validar que en los formularios sean ingresados todos aquellos campos que sean necesarios; éstos deben ser marcador de alguna manera (usualmente con un asterisco) que permita a los usuarios entender la obligatoriedad de ingresar información en ellos; adicionalmente, debe indicarse tal condición en forma explícita.
- **Validaciones:** es necesario realizar en algunos puntos estratégicos del formulario validaciones que permitan tener consistencia en ingreso de datos.

Para corroborar estos dos puntos se procedió a realizar dentro de ambas encuestas la siguiente pregunta:

¿El sistema indica qué campos son obligatorios de ingresar, así como también valida los datos que usted ingresa?

Si () No ()

Los resultados obtenidos de esta interrogante son los siguientes:



Ilustración 17. Gráfica de Campos Obligatorios

Es decir el 100% de los encuestados que vienen hacer 5, considera que el sistema cumple con estos dos puntos.

- **Multiplataforma:** se debe comprobar que los formularios funcionan en diferentes navegadores, en diferentes sistemas operativos.

La validación de este punto en lo que respecta a la utilización del sistema en diferentes navegadores se la hizo mediante la siguiente pregunta que de igual manera consta en ambas encuestas tanto en la del administrador como en la del personal.

¿El sistema se puede ejecutar y responde normalmente en cualquier navegador como mozilla firefox, internet explorer, google chrome, entre otros?

Si () No ()

La cual arrojo los siguientes resultados:

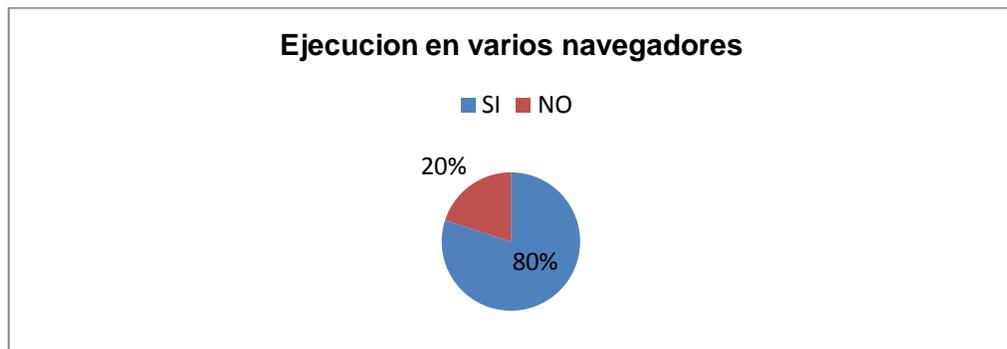


Ilustración 18. Gráfica de Ejecución en Varios navegadores

Lo que equivale a que el 80% de los encuestados que vienen hacer 4 opinan que el sistema cumple con este punto, mientras que el 20% que viene a ser 1 de los encuestado opina que no.

Mientras en lo que respecta a que si el sistema es multiplataforma, es decir que funciona en varios sistemas operativos, se elaboro y se propuso solo en la encuesta correspondiente al administrador del programa la siguiente pregunta:

¿El sistema diseñado para su empresa puede ser instalado tanto en Linux como en Windows?

Si () No ()

En donde el usuario, en este caso solo el administrador luego de realizar la implementación y ejecución pudo constatar personalmente que el sistema se pudo ejecutar correctamente en ambos sistemas tanto en Windows como en Linux.

- **Pantalla:** para obtener un diseño visual adecuado será necesario el uso de templates que ayuden al usuario del sistema tener una fácil adaptación de manejo visual simple del software.

Para la validación esta pregunta se procedió a plantear la siguiente pregunta igualmente en ambas encuestas:

¿La interfaz grafica del sistema está diseñada de acuerdo a la función o actividad que vaya a desarrollar?

Si () No ()

De la cual se obtuvo los siguientes resultados:

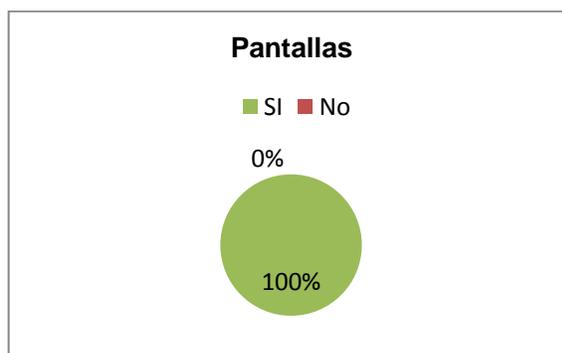


Ilustración 19. Gráfica de Pantallas

Una vez revisadas y analizadas las respuestas que corresponden a la evaluación de la interfaz grafica del sistema, determinamos que el sistema cumple con los siguientes aspectos:

PRUEBAS REALIZADAS	RESULTADOS
Campos obligatorios	Se indica claramente campos obligatorios en cada formulario.
Validaciones	Existen validaciones de ingreso previa realización de alguna operación en la BD.
Pantallas	El manejo de vistas se realiza de manera adecuada gracias al uso de template en cada uno de los módulos.
Navegadores	El sistema se comporta de manera aceptable en

	diferentes navegadores.
Plataformas	El sistema funciona correctamente en diferentes S. O

Tabla 189. Conclusiones de Interfaz Gráfica

6.3. PRUEBAS DE FUNCIONALIDAD

En este sentido, las pruebas se deben hacer sobre diferentes elementos, siendo algunos de los más importantes los siguientes:

6.3.1. PRUEBAS DE SEGURIDAD

El manejo de seguridad permiten tener una idea clara del manejo adecuado de la información involucrada en cada una de los procesos para los cual es necesario citar puntos importantes como:

- **Roles:** Para dar adecuada protección de acceso a cierta partes del sistema se hace uso de los roles de acceso, con lo cual se limita el acceso a ciertas partes y funcionalidades del sistema.

Este punto esta validado en ambas encuestas, mediante la pregunta que se describe a continuación:

¿El sistema le permite ingresar al sistema de acuerdo al rol que se la ha asignado en la empresa?

Si () No ()

De la cual se obtuvo lo siguiente:



Ilustración 20. Gráfica de Roles

El 100% de los encuestados opina que el sistema cumple con este aspecto y pueden acceder al sistema mediante el rol que se les ha asignado en la empresa.

- **Recuperación de Claves:** Un aspecto muy usado en sistemas que se desenvuelven bajo entornos web es el poder recuperar claves de acceso previa introducción de un email válido y que haya sido registrado con anterioridad durante el proceso de creación de usuarios:

Este punto se valida en las dos encuestas realizadas al personal y dice así:

¿Considera adecuada la opción de recuperar su contraseña para ingresar al sistema vía mail en caso que la haya olvidado?

Si () No ()

¿Por qué?

.....

.....

La cual obtuvo los siguientes resultados:



Ilustración 21. Gráfica de Recuperar Contraseña

La totalidad de los encuestados está de acuerdo en que el sistema cuente con ésta opción la cual le permita recuperar sus contraseñas, porque cualquier usuario puede olvidarla.

Además se evaluó otro punto del sistema el cual es importante y tiene que ver con la inactividad del usuario en su cuenta, la cual se cierra luego de un periodo de tiempo determinado por lo que el usuario debe volver a iniciar sesión para ingresar en el sistema; esta pregunta se valida en ambas encuestas mediante la siguiente pregunta:

¿Cree conveniente el cierre de sesión luego de un tiempo de inactividad en su cuenta?

Si () No ()

De la que se obtuvo los siguientes resultados:

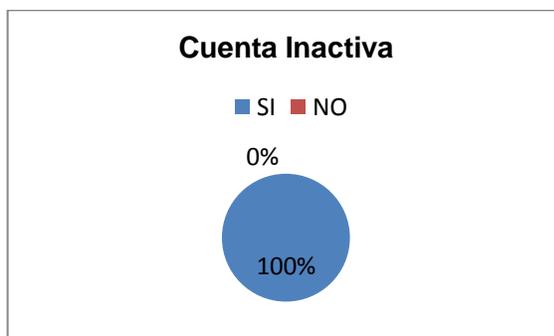


Ilustración 22. Gráfica de Sesión Inactiva

En donde el 100% de los encuestados está de acuerdo con esta opción del sistema ya que hay ocasiones en las que por algún motivo se queda abierta la sesión en su puesto

de trabajo, no hay ningún inconveniente en que alguna persona que no tenga que ver con la empresa pueda acceder a esta ya que se inactiva y no puede hacer uso de la misma.

Una vez obtenidos y analizados estos resultados se determino lo siguiente:

PRUEBAS REALIZADAS	RESULTADOS
Ingreso de Usuario según Rol asignado	Mostrar módulos asignados
Ingreso URL a secciones de la aplicación	Acceso solo con cuenta de Usuario
Recuperación de Claves vía Email	Envío de contraseña a un correo proporcionado
Sesión inactiva durante un periodo de Tiempo	Volver a iniciar Sesión

Tabla 190. Pruebas de Seguridad

6.4. TEST DE EVALUACIÓN DEL SISTEMA

Dentro de esta parte se también se realizaron preguntas para aplicarlas al personal de la empresa las cuales son las siguientes:

De la instalación del sistema, como considera los siguientes aspectos:

INSTALACIÓN	Malo	Regular	Normal	Bueno	Muy Bueno
Facilidad					
Confianza					
Requerimientos					
Estabilización					

Esta pregunta fue aplicada solo al encargado del sistema dentro de la empresa el cual indico lo siguiente en lo que respecta al sistema:

INSTALACIÓN	Malo	Regular	Normal	Bueno	Muy Bueno
Facilidad.				✓	
Confianza.				✓	

Requerimientos.			✓		
Estabilización.			✓		

Tabla 191. Test de Instalación

Otra parte que se válido es la configuración del sistema la cual se hizo mediante la siguiente pregunta igualmente solo realizada al encargado del sistema en la empresa:

En lo que respecta a la configuración del sistema que pudo determinar en relación a los siguientes aspectos:

CONFIGURACIÓN	Malo	Regular	Normal	Bueno	Muy Bueno
Facilidad					
Confianza					
Requerimientos					
Estabilización					

Donde el administrador determino que el sistema se desenvuelve de la siguiente manera:

CONFIGURACIÓN	Malo	Regular	Normal	Bueno	Muy Bueno
Facilidad.				✓	
Procedimientos.				✓	
Estabilización.			✓		
Control.			✓		

Tabla 192. Test de Configuración

Otra parte a considerar en esta parte de la evaluación es la documentación con la que cuenta el sistema la cual se la hizo a ambas partes del personal mediante la siguiente pregunta:

En lo que respecta a la información sobre el manejo del sistema (manuales) como considera los siguientes aspectos:

MANUALES	Malo	Regular	Normal	Bueno	Muy Bueno
Expectativas					
Comprensibles					
Disponibilidad					

Luego de haber revisado todas las encuestas realizadas al personal se puede concluir que el sistema se desenvuelve de la siguiente manera:

MANUALES	Malo	Regular	Normal	Bueno	Muy Bueno
Expectativas.				✓	
Comprensibles.				✓	
Disponibilidad.					✓

Tabla 193. Test de Manuales

Y finalmente se evalúa la capacidad de respuesta del sistema ante las peticiones del usuario, la cual se hizo a ambas partes del personal de la siguiente manera:

Durante su interacción con el sistema en los siguientes aspectos como respondió éste ante sus peticiones:

CAPACIDAD DE RESPUESTA	Malo	Regular	Normal	Bueno	Muy Bueno
Persistencias					
Consultas					
Impresiones					
Validaciones					

Y de donde se puede determinar luego de analizar las encuestas que el sistema se desenvuelve así:

CAPACIDAD DE RESPUESTA	M	R	N	B	MB
Persistencias.				✓	
Consultas.				✓	
Impresiones.			✓		
Validaciones.					✓

Tabla 194. Test de Capacidad de Respuesta

Todas estas pruebas se encuentran validadas mediante las encuestas realizadas al personal de la empresa Lojanet++ las cuales se pueden encontrar en la parte de anexos de éste proyecto.

G. CONCLUSIONES

- ✓ Implementar el sistema Openloja en la empresa Lojanet++, permite que ésta realice de forma automatizada todas las funciones en los departamentos que abarca el sistema, mejorando su productividad y por lo tanto su servicio a la comunidad.
- ✓ El uso de herramientas y software de licencia libre como Ubuntu, Eclipse, Java, Seam, Hibernate, My Sql, entre otros; para el desarrollo del sistema OpenLoja son de gran ayuda ya que gracias a estas el sistema se pudo diseñar correctamente y de manera oportuna.
- ✓ Diseñar el sistema OpenLoja por módulos sirve para mejorar su rendimiento y funcionalidad así como también permite la adaptación de nuevas funciones en caso que el usuario lo requiera o necesite.
- ✓ La integración de nuevas funciones al sistema o el mejoramiento del mismo se puede considerar para la realización de un nuevo proyecto investigativo, por lo tanto puede ser considerado como un tentativo para tema de tesis.
- ✓ La ejecución de pruebas al sistema garantiza que la capacidad de respuesta en la realización de tareas satisface las expectativas del usuario, es decir cumple los requerimientos que la empresa buscaba.
- ✓ Mediante el uso de la metodología ICONIX a través de todas sus etapas, conjuntamente con el Lenguaje Unificado de Modelo (UML), logramos obtener toda la documentación necesaria y así sustentar el presente proyecto.
- ✓ El desarrollo del presente proyecto investigativo permite aumentar y fortalecer nuestros conocimiento adquiridos durante los años de estudio cursados, así como también interactuar en un entorno laboral muy similar a nuestro futuro profesional.
- ✓ En el desarrollo de la investigación se presentaron inconvenientes entre los cuales se puede mencionar el mal desempeño o funcionamiento del sistema al cambiarlo a otra red, para lo cual se considero remplazar el servidor que manejaba la aplicación por uno más actual lo que permite que éste se desenvuelva correctamente en cualquier lugar donde se implemente; así como este inconveniente se superaron todos los demás que se suscitaron durante el trayecto de éste proyecto, de la manera más apropiada, fortaleciendo nuestra capacidad de optar por nuevas alternativas de solución.

H. RECOMENDACIONES

- ✓ Capacitar a los usuarios del sistema a través de manuales, foros o charlas, facilita manejar el sistema de la manera más adecuada.
- ✓ Asignar a una persona como Administrador del sistema, para que cree los usuarios con sus respectivos privilegios.
- ✓ Agregar el módulo de contabilidad para ampliar la funcionalidad de la aplicación, utilizando el mismo estándar de desarrollo.
- ✓ Utilizar una metodología de desarrollo coherente con la aplicación a realizar, que ayude a llevar un proceso ordenado con su respectiva documentación para su posterior sustentación.
- ✓ Realizar pruebas al sistema en distintas redes, para evitar posibles fallos del sistema al momento de cambiar o utilizar el sistema en otro sitio.
- ✓ Utilizar herramientas de generación automática de diagramas para ahorrar tiempo, ya que el diseño de estos es un proceso tedioso y largo, aumentando así el intervalo de tiempo asignado para la construcción del software.

I. BIBLIOGRAFÍA

LIBROS

1. **MINTER, D. y LINWOOD, J.** (2009). PRO HIBERNATE 3. (2ª Ed.). New York: Springer - Verlag.
2. **ORTUÑO, M Á.** (2010). Elección de ERP: Criterios y Costes de implantación de un ERP. (4ª Ed.). Murcia.

RECURSOS DE INTERNET

1. **Fernández, A. M.** (2011). JPA – Hibernate. Recuperado el 23 de Septiembre de 2011 de la página web del Departamento de Informática de la Universidad Nacional de Oviedo: http://www.di.uniovi.es/~alberto_mfa/?Docencia:Cursos:JPA-Hibernate
2. Universidad Politécnica Salesiana Repositorio Digital. (s.f.). Recuperado el 24 de Septiembre del 2011 de http://www.dspace.ups.edu.ec/bitstream/123456789/1616/5/Capitulo_4.pdf
3. **Delgado, A.** (2007). Hibernate. Recuperado el 19 de septiembre del 2011 de <http://www.fing.edu.uy/inco/cursos/ingsoft/pasantiaATAM/Hibernate%20v1.2.pdf>
4. **Gomez, G. Á.** (2008). Implementación de una Aplicación Web utilizando frameworks j2ee. Recuperado el 15 de Agosto del 2011 de <http://www.maia.ub.es/~jaume/TFC/AngelGomezGarcia.pdf>
5. **Coral, H. y Gordillo, L.** (2010). Evaluación de tres implementaciones Java Server Faces 1.2. Recuperado el 23 de Septiembre del 2011 de la pagina web del Repositorio Digital de la Universidad Politécnica del Ejército: <http://repositorio.espe.edu.ec/handle/21000/309>



ANEXOS



J. ANEXOS

1. ANTEPROYECTO



UNIVERSIDAD NACIONAL DE LOJA

AREA DE LA ENERGIA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

INGENIERIA EN SISTEMAS

OPENLOJA 1.0

“Sistema Planificador de Recursos Empresariales (ERP) integrado con módulo de comunicación de clientes para la empresa de servicios informáticos Lojanet++ Cía. Ltda.”

Integrantes:

Vladimir Yazber Romero Gonzaga.

Jorge Luis Tapia Guarnizo.

Loja-Ecuador

2010



1. TEMA

OPENLOJA 1.0

“Sistema Planificador de Recursos Empresariales (ERP) integrado con módulo de comunicación de clientes para la empresa de servicios informáticos Lojanet++ Cía. Ltda.”

2. PROBLEMÁTICA

2.1. Antecedentes.

En la ciudad de Loja durante los últimos años ha venido sufriendo un incremento en la demanda de equipos y servicios informáticos debido a que en la actualidad las empresas y las personas particulares dependen enormemente del uso de la tecnología para realizar la mayor parte de sus actividades, así como también a la ayuda de empresas que se dedican a la comercialización de equipos y suministros informáticos.

Una de estas empresas dedicadas a venta de suministros y prestación de Servicios informáticos de nuestra ciudad es Lojanet ++, la cual ha venido funcionando eficientemente desde sus inicios en 1995, la misma que se encuentra ubicada en las calles 10 de Agosto entre Olmedo y J.J. Peña y que durante los últimos años ha contribuido al desarrollo comercial de la ciudad y provincia, poniendo a disposición de todas las personas interesadas en utilizar sus servicios.

Para lo cual dispone de una gran variedad de equipos vinculados con la tecnología de última generación, de acuerdo a las necesidades de sus clientes y con precios muy accesibles, además de contar promociones, facilidades de pago.

2.2. Situación Problemática.

En las empresas dedicadas a la comercialización de productos y servicios como en la gran mayoría de entidades, es importante la eficiencia y calidad al momento de proporcionar una excelente atención a sus clientes mejorando cada uno de los procesos en el menor tiempo posible.

Específicamente en el caso de Lojanet++, dentro de los ámbitos laborales de la empresa, no existe una correcta administración de los recursos empresariales con los que cuenta.

Esto se debe a que no cuenta con un correcto manejo, almacenamiento y respaldo de toda la información involucrada ya sea directa o indirectamente con los procesos que llevan a cabo dicha institución, así como también a la falta de comunicación eficiente con los clientes que hacen uso de sus servicios.

Por otra parte esta la inadecuada administración de los procesos y tareas que se realizan dentro la empresa ya sean estos ejecutados interna o externamente lo que hace que no se tenga un control adecuado del desempeño laboral de los empleados. Además cabe recalcar que en la actualidad el software con el que cuenta la organización denominado "OPENMARCO" no cubre con todas las necesidades y expectativas que esta requiere entre las cuales se puede destacar:

- El manejo del Rol de Pagos.
- Comunicación directa con clientes.
- Acceso limitado al sistema (solo intranet).

El control y manejo de inventario es un punto a considerar dentro de los aspectos que repercuten en el desarrollo acertado de las actividades en el organismo; la desorganización e incoherencia que se divisan en el manejo del inventario generan una desintegración en compras y ventas que se puedan producir.

Lo cual conlleva que al momento de realizar cualquier tipo de proceso tarde más de normal provocando un gran desperdicio de tiempo y recursos.

2.3. Problema de Investigación.

Por todo lo anteriormente mencionado hemos creído conveniente que el problema a resolver es el siguiente:

"El software de gestión denominado OPENMARCO con que cuenta la empresa de servicios informáticos Lojanet Cía. Ltda.; no cubre adecuadamente con todas las necesidades que se generan en su entorno empresarial; entre los cuales están: manejo de rol de pagos de los empleados, comunicación directa con los clientes y acceso limitado

al sistema solo a través de intranet; a todo esto se asocia también los procesos que no se desarrollan correctamente como: el inadecuado manejo de inventario, compras y ventas”.

2.4. Delimitación.

2.4.1. Espacio

Para el desarrollo del presente proyecto hemos optado por la empresa de ventas de equipos y suministros, y servicios informáticos Lojanet++ lo cual involucrara las áreas de inventario, ventas, compra y gestión de recursos humanos.

2.4.2. Tiempo

El proyecto se lo planea concluir en el lapso de tiempo de un año, desde Noviembre del 2010 a Noviembre del 2011 y al finalizar esto será implantado dentro de la empresa Lojanet++ de la ciudad de Loja.

2.4.3. Unidades de observación

Para el desarrollo del proyecto se necesitara la colaboración de las partes que integran la empresa Lojanet++, las cuales son:

- Empleados.
- Clientes.
- Gerente.
- Proveedores.
- Sistema **OPENLOJA**

3. JUSTIFICACIÓN

Durante el desarrollo de este proyecto, consideramos adecuado el planteamiento del tema, tomando en consideración el auge de las tecnologías de código abierto (**OpenSource**), y la gran necesidad de contar con un sistema que permita una adecuada organización de los recursos empresariales y que le permita a la entidad una correcta comunicación con los entes involucrados en los procesos cotidianos de la empresa, por lo cual hemos creído conveniente justificar el siguiente proyecto en los siguientes ámbitos:

3.1. Académica

La Universidad Nacional de Loja dentro de sus objetivos es la formación de profesionales capaces de desenvolverse en el accionar profesional y consientes de la gran importancia que tiene poner en práctica los conocimientos adquiridos durante el transcurso de la carrera, detallado en el Sistema Académico Modular por Objetos de Transformación (SAMOT); lo cual nos permitirá actuar eficientemente ante cualquier situación que se nos presente durante nuestra futura vida profesional, hemos considerado conveniente justificar el proyecto en lo académico, aportando a la realización profesional, reafirmar sustentos teóricos y ponerlos en práctica, previo a la obtención del título de Ingenieros en Sistemas mediante la ejecución de este proyecto, logrando profesionales con presencia a nivel local y nacional.

3.2. Científico – Técnica

Tomando en consideración el sustento técnico y tecnológico para el normal progreso de proyecto, el tema que ponemos a consideración brindar las facilidades necesarias tanto en hardware como software.

En lo que respecta al hardware el proyecto presta las facilidades para cubrir cada uno de los requerimientos que se presenta durante el desarrollo del proyecto en mención, de igual forma el software brinda la versatilidad necesaria dando que el tema en mención se encuentra enmarcado en los ámbitos del código abierto (OpenSource), brindando el

sustento teórico necesario que facilita a cada uno de los integrantes del grupo de trabajo como foros, salas de aprendizaje entre otros.

3.3. Económica

Durante el progreso de este proyecto, nuestro grupo contará con los recursos necesarios, que serán proporcionados por la empresa involucrada, la misma que es consciente de la inversión que realiza y los resultados a corto plazo a obtener, por otra parte para la elaboración del mismo utilizaremos software de licencia libre, para lo cual no necesitaremos un aporte económico adicional, facilitándonos el cumplimiento de todos los requerimientos que se presenten durante el transcurso del proyecto.

4. OBJETIVOS

4.1. Objetivo General

- Desarrollar un Sistema Planificador de Recursos Empresariales (ERP) integrado a un módulo de comunicación con clientes para la empresa de servicios informáticos Lojanet++.

4.2. Objetivos Específicos

- Analizar los Requerimientos del sistema OPENLOJA
- Diseñar los módulos de bodega, ventas, configuración, RR HH, comunicación del sistema OPENLOJA
- Construir módulos de bodega, ventas, configuración, RR HH, comunicación del sistema OPENLOJA
- Integrar los módulos del sistema OPENLOJA previamente construidos.
- Implementar el sistema OPENLOJA en la empresa Lojanet++.
- Ejecutar el plan de validación al sistema OPENLOJA.

5. MARCO TEÓRICO

5.1. Esquema del Marco Teórico

CAPITULO I

1. SISTEMA DE PLANIFICACIÓN DE RECURSOS EMPRESARIALES(ERP)

- 1.1 Introducción
- 1.2 Definición
- 1.3 ¿Por qué utilizar un ERP?
 - 1.3.1 Competitividad
 - 1.3.2 Control
 - 1.3.3 Integración
- 1.4 Objetivos principales de los sistemas ERP
- 1.5 Características principales de los sistemas ERP
 - 1.5.1 Otros Beneficios
- 1.6 Factores de Éxito/Fracaso en implementación de un ERP
- 1.7 Módulos de ERP
- 1.8 Ejemplos de ERP
 - 1.8.1 Openbravo
 - 1.8.2 OpenXpertya
 - 1.8.3 OpenERP
 - 1.8.4 Compire
- 1.9 Conclusiones de ERP

CAPITULO II

2 JBoss Seam FRAMEWORK

- 2.1 Introducción
- 2.2 Definición
- 2.3 Características
- 2.4 Sistemas

- 2.4.1 Sistema de Seguridad
- 2.4.2 Sistema de Plantillas (templates)
- 2.4.3 Sistema de caché
- 2.5 Testeabilidad
- 2.6 Configuración

CAPITULO III

3 HIBERNATE

- 3.1 Definición
- 3.2 Arquitectura de Hibernate
 - 3.2.1 Gestión de Conexión
 - 3.2.2 Gestión de Transacciones
 - 3.2.3 Mapeo Relacional de Objetos
- 3.3 Características
- 3.4 Introducción Anotaciones

CAPITULO IV

4 RICHFACES

- 4.1 Introducción
- 4.2 Características
- 4.3 Inconvenientes
- 4.4 Requisitos e Incompatibilidades
- 4.5 Interacción con otros componentes
 - 4.5.1 Sun JSF RI
 - 4.5.2 Apache MyFaces
 - 4.5.3 Facelets
 - 4.5.4 JBoss Seam
- 4.6 Recomendaciones de Uso
- 4.7 Enterprise JavaBeans



- 4.7.1 Definición
- 4.7.2 Tipos de Enterprise JavaBeans
- 4.7.3 Funcionamiento de un Enterprise JavaBeans
- 4.7.4 Interfaz "Home"
- 4.7.5 Interfaz Remota
- 4.7.6 Clase de Implementación EJB
- 4.7.7 Correspondencia entre métodos de interfaz y métodos de implementación

CAPITULO I



Ilustración 23. Logo OpenLoja

1. SISTEMA DE PLANIFICACIÓN DE RECURSOS EMPRESARIALES (ERP).

1.1. Introducción

Es indudable que el ambiente competitivo en el que se vive en el ámbito empresarial actualmente, requiere de promover los procesos y actividades de negocio que generan las ventajas competitivas de las compañías ante sus más fuertes competidores.

Por esto, desde hace ya varios años, se ha dado mayor importancia a las Tecnologías de Información y su alineación con las estrategias del negocio para mejorar sus procesos clave de negocio. Prueba de ello, es el incremento tan sustancial de adquisiciones de paquetes de software empresariales tales como el ERP (*Enterprise Resource Planning*), con el cual los directivos de las compañías esperan tener integradas todas las áreas o departamentos de la compañía que apoyan para la generación de sus productos y servicios.

Hoy más que nunca las empresas requieren de herramientas que les proporcionen control y centralización de su información, esto con el fin tomar las mejores decisiones para sus procesos y estrategias de negocios. Los ERP son una solución robusta para aquellas empresas que buscan una solución universal a la centralización de su información.

1.2. Definición.

Ortuño define un ERP de la siguiente manera: “ERP son las siglas en inglés de Enterprise Resource Planning que en español significa planificación de recursos empresariales, es un

sistema de gestión de la información estructurado para satisfacer la demanda de soluciones de gestión empresarial, basado en el ofrecimiento de una solución completa que permite a las empresas evaluar, implementar y gestionar más fácilmente su negocio. Las soluciones ERP se caracterizan por su modularidad, integración de la información (dato único), universalidad, estandarización e interfaces con otras aplicaciones. Son sistemas abiertos y en la mayoría de los casos multiplataforma”. (2008).

1.3. ¿Por qué utilizar un ERP?

Existen tres razones fundamentales por las cuales una empresa se interesa en implantar una solución ERP: aumentar su competitividad, controlar mejor sus operaciones e integrar su información.

1.3.1. Competitividad

Las empresas requieren continuas optimizaciones de sus costos, ya sea de producción, comercialización o administración; por otro lado, deben incrementar constantemente su productividad.

1.3.2. Control

Varias empresas tienen un manejo aislado de la información generada en los distintos departamentos y requieren de una solución global que integre y organice los datos para que en forma accesible apoye la toma de decisiones.

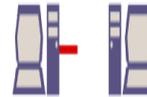
1.3.3. Integración

Es importante integrar la información en las áreas vitales de la empresa como finanzas, distribución y manufactura. En este sentido una de las principales integraciones son entre el back-office y el front-office, es decir, aquellas aplicaciones que apoyan la fuerza de ventas, comercialización y servicio al cliente con las aplicaciones de permiten a las empresas comprar, monitorear, administrar y distribuir productos.

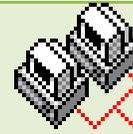
1.4. Objetivos principales de los sistemas ERP

El propósito fundamental de un ERP es otorgar apoyo a los clientes del negocio, tiempos rápidos de respuesta a sus problemas así como un eficiente manejo de información que permita la toma oportuna de decisiones y disminución de los costos totales de operación.

- ✓ Optimización de los procesos empresariales.



- ✓ Acceso a información confiable, precisa y oportuna.



- ✓ La posibilidad de compartir información entre todos los componentes de la organización.



- ✓ Eliminación de datos y operaciones innecesarias.



- ✓ Reducción de tiempos y de los costes de los procesos.



Tabla 195. Objetivos principales de un ERP

1.5. Características principales de los sistemas ERP.

El ERP no es simplemente un software que se compra, instala y usa como Windows o un juego de computadora.

Consiste en una revolución que involucra todos los procesos internos y debe ser precedido de una revaluación de todos los departamentos, sus funciones, mecanismos de

decisión y formas de actuación para lo cual se puede citar las siguientes características que sobresalen en un ERP.

- ✓ Base de datos centralizada
- ✓ Los componentes del ERP interactúan entre sí consolidando todas las operaciones.
- ✓ En un sistema ERP los datos se ingresan sólo una vez y deben ser consistentes, completos y comunes.
- ✓ Las empresas que lo implanten deben modificar alguno de sus procesos para alinearlos con los del sistema ERP.
- ✓ Un sistema ERP incluye un conjunto de aplicaciones ERP o módulos.

1.6. Módulos de ERP.

Los ERP entienden que una empresa es un conjunto de departamentos que se encuentran interrelacionados por la información que comparten y que se genera a partir de sus procesos. Una ventaja de los ERP, tanto económica como técnicamente es que la funcionalidad se encuentra dividida en módulos, los cuales pueden instalarse de acuerdo con los requerimientos del cliente. Ejemplo: ventas, materiales, finanzas, control de almacén, recursos humanos, etc.

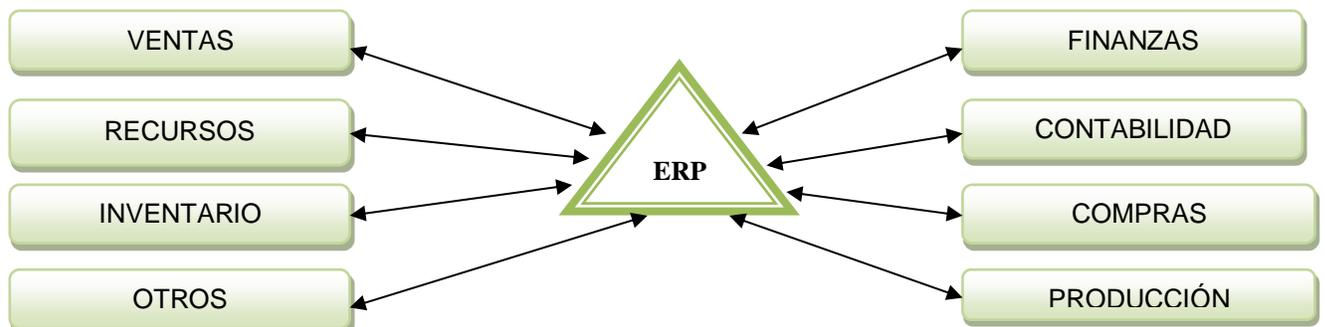


Ilustración 24. Módulos de un ERP

Los ERP están creados para adaptarse a la idiosincrasia de cada empresa. Esto se logra por medio de la configuración o parametrización de los procesos de acuerdo con las salidas que se necesiten de cada uno. Por ejemplo, para controlar inventarios, es posible que una empresa necesite manejar la partición de lotes pero otra empresa no. Los ERP más avanzados suelen incorporar herramientas de programación de 4ª Generación para el desarrollo rápido de nuevos procesos. La parametrización es el valor añadido fundamental que se debe hacer con cualquier ERP para adaptarlo a las necesidades concretas de cada empresa.

1.7. Ejemplos de ERP

1.7.1. Openbravo

Openbravo es el reconocido desarrollador de Openbravo ERP, la solución comercial en software libre y entorno web para las pequeñas y medianas empresas.

La primera alternativa real al software propietario. Su sistema en entorno web de gestión integral de empresas (ERP) ha sido descargado más de 1.750.000 veces y se utiliza en alrededor de 50 países.

El crecimiento de Openbravo es debido a la contribución de su comunidad internacional de usuarios, partners y desarrolladores en constante expansión. El modelo de negocio de la compañía basado en el software libre comercial, elimina el coste de las licencias y ofrece soporte, servicios y mejoras de los productos mediante una suscripción anual. Un creciente catálogo de soluciones y extensiones para su ERP, tanto comerciales como gratuitas, se encuentran disponibles en su marketplace on-line, Openbravo Exchange.

Openbravo, respaldado por inversores internacionales, ha conseguido un récord de financiación en el ámbito de los ERP en software libre de la mano de Amadeus Capital Partners, GIMV, Adara Venture Partners y Sodena.

1.7.2. OpenXpertya

En pocas y concisas palabras, openXpertya es una solución integral para la empresa que engloba ERP y CRM, con integración de servicios en línea de B2B o B2C (en función del tipo de cliente final) e incluso B2E (servicios internos) y con soporte de exportación e

datos (enlaces) al estándar EDI (intercambio electrónico de información entre empresa: facturas, albaranes, pedidos: EDIFACT, estándar mundial de la ONU) y con posibilidad de trabajar con cubos multidimensionales OLAP (análisis exhaustivo de resultados). Todo ello adaptado muy de cerca a nuestra legislación, tanto fiscal, como mercantil, civil, contable, etc.

El propósito de openXpertya es cubrir ampliamente, y muy de cerca, todas aquellas necesidades de gestión que una empresa de tamaño medio o grande podría tener. Es la planificación global de todos los recursos.

La solución openXpertya tiene capacidad multientidad, multiempresa, multicentro, multialmacén, multicaja, etc. Haciendo posible la descentralización de una organización y siendo el tipo de aplicación ideal para una cadena de franquicias, una empresa de distribución, de producción o de servicios

1.7.3. OpenERP

OpenERP como producto no 'pertenece' a ninguno de sus distribuidores, tiene libertad para elegir al proveedor que más le convenga según sus necesidades.

Puede contratar únicamente lo que necesite. Lo habitual es tercerizar todos los procesos de la implantación, pero quizá su empresa ya disponga de algunos recursos y sólo requiera desarrollo de algún módulo específico o formación/soporte técnico de algún módulo oficial.

Al ser software libre, podrá disponer del código para realizar cualquier mejora sobre los módulos ya existentes, o crear uno nuevo adaptado a sus necesidades.

OpenERP dispone de más de 400 módulos, muchos de ellos específicos determinados sectores. Puede comenzar utilizando solamente el módulo de recursos humanos o la contabilidad, e ir integrando más módulos posteriormente.

1.7.4. Compiere

Compiere es una aplicación para negocios de código abierto, ERP y CRM destinada para las empresas de pequeño y mediano tamaño y con una gran expansión en el mercado anglosajón en los últimos años.

Compiere está desarrollada usando J2EE. La aplicación y el código fuente se provee sobre la base de distribución libre bajo una licencia basada en la licencia pública Mozilla. Puede ser configurada y extendida dentro de la aplicación y por medio de la adición de componentes modulares. La documentación y el soporte solo están disponibles mediante pago.

Desde la versión 2.5.2, Compiere es independiente de la base de datos, y existe una infraestructura para la conexión a múltiples bases de datos. La conectividad a las siguientes bases de datos: PostgreSQL, MySQL y Sybase puede estar disponible o en procesos de completarse pero no es soportada oficialmente por Compiere, que continúa soportando únicamente Oracle como base de datos "oficial".

Aunque Compiere está gobernado por una licencia de software libre derivada de la MPL 1.1, la CPL 1.1 (Compiere Public License), realmente es difícil saber cuánto del producto es código abierto y cuánto no, al incluir varias librerías internas cuyo código no se proporciona con el producto e incluso algunas de pago (de terceros en cualquier caso) que realizan funciones centrales en el producto. Asimismo, la propia licencia CPL incluye la posibilidad clara de que la empresa desarrolladora pase partes, o la totalidad del código, a licencia comercial transcurridos dos años de su fecha de lanzamiento.

TABLA COMPARATIVA DE ERPs

ERP Package	Lenguaje	Licencia	Otra Información	Origen
<u>Adempiere</u>	Java Ali	<u>GPL</u>	started as a fork of Compiere	España
<u>BlueErp</u>	PHP, MySQL, PostGreSQL	<u>GPL</u>		
<u>Compiere</u>	Java	<u>GPL/Commercial</u>		
<u>Dolibarr</u>	PHP, MySQL, PostgreSQL	<u>GPL</u>	ERP/CRM for SME, freelancers or foundations	US, Francia, Bélgica, España, India, Argentina...
<u>ERP5</u>	Python, Zope, MySQL	<u>GPL</u>	Basado en Modelo Unificado	Brasil, Francia, Alemania, Japón, Senegal

<u>Fedena</u>	Ruby	<u>Apache License</u>	ERP para escuelas/universidades	India
<u>GNU Enterprise</u>	Python	<u>GPLv3</u>		
<u>HeliumV</u>	Java	<u>AGPL</u>	ERP para pequeños y medianos negocios	Austria, Alemania
<u>JFire</u>	Java	<u>LGPL</u>		
<u>Kuali Foundation</u>	Java			
<u>LedgerSMB</u>	Perl	<u>GPL</u>		
<u>OFBiz</u>	Apache, Java	<u>Apache License 2.0</u>		
<u>Openbravo</u>	Java	Openbravo licencia pública (OBPL), licencia de software libre basada en la licencia pública de Mozilla (MPL)		España
<u>OpenERP</u>	Python, PostgreSQL	<u>GPL</u> , OpenERP Public License	formerly Tiny ERP	Bélgica, India, USA
<u>Opentaps</u>	Java			
<u>Postbooks</u>	C++, JavaScript, PostgreSQL	<u>CPAL</u>	Produced by <u>XTuple</u> , uses <u>Qt</u> framework	
<u>SQL-Ledger</u>	Perl, PostgreSQL	<u>GPL</u>		
<u>Tryton</u>	Python	<u>GPLv3</u>	started as a fork of OpenERP	

Tabla 196. Comparativa de ERPs

1.8. Conclusiones de ERP.

En la actualidad las tecnologías de información juegan un papel importante en las estrategias de negocios, ya que están cambiando la forma en que las empresas realizan sus procesos. Los sistemas de información permiten a las compañías lograr ventajas competitivas de diferentes maneras: coordinando actividades de valor en localidades que se encuentran en una amplia geografía, o también mediante la creación de nuevas interrelaciones entre los negocios, ampliando el alcance de las industrias.

Asimismo le sirve a las empresas para soportar sus estrategias competitivas, ya sea para ir un paso delante de la competencia o reducir las ventajas que la misma pueda presentar.⁹

El ERP es un sistema integral de gestión empresarial que está diseñado para modelar y automatizar la mayoría de procesos en la empresa (área de finanzas, comercial, logística, producción, etc.). Su misión es facilitar la planificación de todos los recursos de la empresa.

Como todo sistema, tiene sus ventajas y sus desventajas, de los beneficios más comunes e importantes podemos mencionar:

- ❖ Solo un sistema para manejar muchos de sus procesos comerciales.
- ❖ Integración entre las funciones de las aplicaciones
- ❖ Reduce los costos de gerencia
- ❖ Incrementa el retorno de inversión
- ❖ Fuente de Infraestructura abierta

De las desventajas podemos mencionar:

- ❖ Son muy caros.
- ❖ Requiere cambios en la compañía y procesos para su instalación.
- ❖ Son complejos y muchas compañías no pueden ajustarse a ellos.
- ❖ Hay pocos expertos en ERP.

Antes de implementar un ERP, es importante que la empresa considere los beneficios que desea para su organización y en base a ello buscar la mejor solución en el mercado.

⁹Kumar y Hillengersberg (2000) definen al *Enterprise Resource Planning* (ERP)

CAPITULO II

2. JBoss Seam FRAMEWORK

2.1. Introducción

JBoss Seam es un framework desarrollado por JBoss, una división de Red Hat. El líder del proyecto es Gavin King, también autor del framework para mapeo objeto relacional Hibernate. Combina a los 2 frameworks Enterprise JavaBeans EJB3 y JavaServerFaces JSF. Se puede acceder a cualquier componente EJB desde la capa de presentación refiriéndote a él mediante su nombre de componente seam.

Seam introduce el concepto de contextos. Cada componente de Seam existe dentro de un contexto. El contexto conversacional por ejemplo captura todas las acciones del usuario hasta que éste sale del sistema o cierra el navegador - inclusive puede llevar un control de múltiples pestañas y mantiene un comportamiento consistente cuando se usa el botón de regresar de el navegador.

Tú puedes automáticamente generar una aplicación web de altas, bajas cambio y modificaciones a partir de una base de datos existente utilizando una herramienta de línea de comandos llamada seam-gen incluida con el framework. El desarrollo WYSIWYG es facilitado a través del uso de las JBoss Tools, que es un conjunto de plug-ins diseñados para el entorno integrado de desarrollo Eclipse. Seam puede ser integrado con las bibliotecas de componentes JSF JBossRichFaces o con ICEsoftICEFaces. Ambas bibliotecas poseen soporte para AJAX.

2.2. Definición

JBossSeam es un framework que integra y unifica los distintos standards de la plataforma Java EE 5.0, pudiendo trabajar con todos ellos siguiendo el mismo modelo de programación.

Ha sido diseñado intentado simplificar al máximo el desarrollo de aplicaciones, basando el diseño en Plain Old Java Objects (POJOs) con anotaciones. Estos componentes se usan

desde la capa de persistencia hasta la de presentación, poniendo todas las capas en comunicación directa.

El núcleo principal de Seam está formado por las especificaciones Enterprise JavaBeans 3 (EJB3) y JavaServer Faces (JSF).

A grandes rasgos podemos definir EJB3 como una arquitectura para un sistema transaccional (como bases de datos) de objetos distribuidos basado en componentes que permite construir aplicaciones portables, reusables y escalables.

JSF es un framework de la capa de presentación que define componentes para el interfaz gráfico y "managedbeans" para la lógica de la aplicación que interactúan a través de un sistema de eventos.

Sin embargo, estos frameworks tienen algunas limitaciones y no han sido concebidos para trabajar juntos (esto pretende resolverse con la futura especificación web beans); tienen distinto tipo de configuraciones (JSF usa archivos XML mientras que EJB3 usa anotaciones), distinto ciclo de vida y no pueden comunicarse directamente a nivel de framework.

Para hacerlos cooperar necesitaríamos escribir "clases fachada" y multitud de código de relleno que se encargase de pasar las llamadas de una capa de la aplicación a otra. Ahí es donde entra en juego Seam.

Seam elimina la barrera existente entre estas tecnologías, permitiendo usar EJBs directamente como "backingbeans" de JSF y lanzar o escuchar eventos web.

2.3. Características.

- Tiene un diseño 100% basado en componentes.
- JBoss Seam ofrece una solución completa para el desarrollo web que abarca todas las capas, desde la de presentación hasta la de integración.
- Sigue un modelo Pull en el que desde una página puede referenciarse cualquier componente que esté asociado a un contexto accesible.
- Proporciona varios componentes para manejar los mensajes multidioma.

- Determina automática el Locale de cada petición y lo usa si está disponible en la aplicación. También permite sobrescribir el Locale detectado y/o establecer uno por defecto. Se pueden configurar los mensajes en archivos de recursos tipo bundle.
- Dispone de una clase que se encarga de parsear automáticamente los formatos horarios al formato establecido.
- Para los mensajes de error o éxito se puede hacer uso de la clase de FacesMessages incluida con JSF.

2.4. Sistemas

2.4.1. Sistema de Seguridad

El Seam Security API es una parte de Seam que proporciona funcionalidades de autenticación y autorización basado en JAAS (Java Authentication and AuthorizationService). Puede usarse el modo sencillo por defecto que se basa en roles o usar el framework JBoss Rules, que ofrece un sistema más poderoso basado en reglas. El sistema se encarga del manejo de errores de autorización o autenticación permitiendo redirigir al usuario a una página determinada.

Las restricciones pueden establecerse mediante el uso de anotaciones. Permite restringir tanto acciones como entidades o páginas o componentes de la página.

2.4.2. Sistema de Plantillas (templates)

Seam viene integrado por defecto con Facelets.

Facelets es un sistema de plantillas para JSF. Permite definir la vista mediante xml en forma de árbol de manera que se pueden definir componentes como composición de otros componentes. Entre sus características destacar que soporta el lenguaje EL, composición a partir de diferentes archivos, decorators y reporte de errores indicando la etiqueta/línea precisa.

2.4.3. Sistema de caché

En total podemos encontrar varios sistemas de caché.

En primer lugar podemos encontrar la importantísima caché de la base de datos.

Después existe otra caché en la solución ORM adoptada.

El contexto de persistencia manejado por Seam actúa como caché de los datos leídos en la conversación actual.

También pueden guardarse datos no transaccionales en el contexto de aplicación.

La aplicación puede almacenar datos en memoria utilizando el sistema JBoss Cache, que ofrece una cache estructurada en forma de árbol, con posibilidad de cluster y transaccional.

2.5. Testeabilidad

Al ser POJOs todos los componentes, los test de unidad son triviales. También se proporcionan anotaciones para facilitar la recreación del entorno.

Para los test de integración proporciona un contenedor de EJBs y declaración de objetos mock mediante anotaciones.

La herramienta SeamGen crea automáticamente test de unidad estándar y test TestNG que simulan peticiones y respuestas JSF para cada acción mediante scripting.

2.6. Configuración

La configuración puede realizarse en su práctica totalidad mediante anotaciones, lo que ayuda a simplificar esta tediosa tarea. Aun así, hay una pequeña cantidad de configuración que debe hacerse con XML (como la relativa a JSF). Afortunadamente, esta configuración es autogenerada por la herramienta SeamGen. En caso de no hacer uso de ella, la configuración puede copiarse de las aplicaciones de ejemplo. Además, el framework sigue la filosofía de “configuración por omisión”: Todos los componentes



vienen con un comportamiento por defecto que solo hace falta redefinir si queremos personalizarlos.

También permite la configuración de los componentes internos del framework a través del archivo seam.properties.

CAPITULO III



Ilustración 25. Logo Hibernate

3. HIBERNATE

3.1. Definición

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL.¹⁰

3.2. Arquitectura De Hibernate

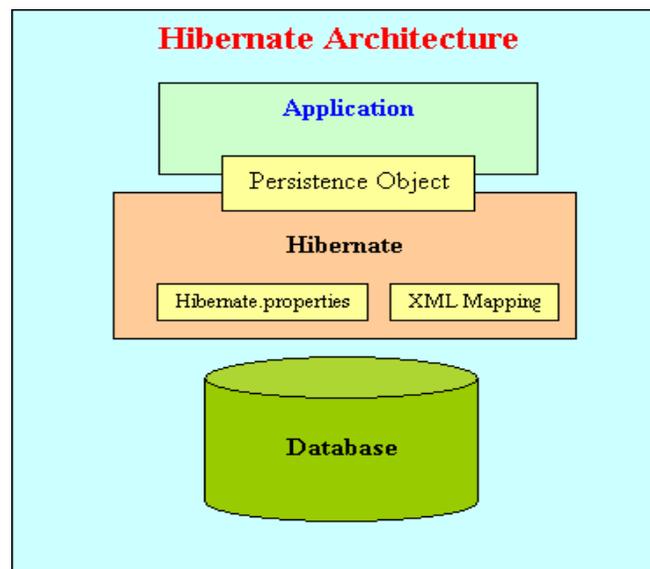


Ilustración 26. Arquitectura de Hibernate

¹⁰ Association for Information Systems (CAIS)

El diagrama anterior muestra que Hibernate está usando la configuración y los datos de base de datos para proporcionar servicios de persistencia (y objetos persistentes) a la aplicación.

Para utilizar Hibernate, es necesario crear clases Java que representa la tabla en la base de datos y el mapa de la variable de instancia en la clase con las columnas de la base de datos. Luego de hibernación se puede utilizar para realizar operaciones en la base de datos como, seleccionar, insertar, actualizar y eliminar los registros de la tabla. Hibernate crea automáticamente la consulta para realizar estas operaciones.

Arquitectura de Hibernate tiene tres componentes principales:

3.2.1. Gestión de Conexión

Management service Conexión Hibernate servicio de gestión de proporcionar una gestión eficiente de las conexiones de base de datos. Databaseconnectionmostexpensive conexión de base de datos es la parte más costosa de interactuar con la base de datos, ya que requiere una gran cantidad de recursos de abrir y cerrar la conexión de base de datos.

3.2.2. Gestión de Transacciones

Gestión de servicios de transacciones ofrecen la posibilidad al usuario para ejecutar más de una base de datos de estados a la vez.

3.2.3. Mapeo Relacional de Objetos

Mapeo relacional de objetos es una técnica de mapeo de la representación de los datos de un modelo de objetos a un modelo de datos relacional. Esta parte de la hibernación se utiliza para seleccionar, insertar, actualizar y eliminar los registros de forma de la tabla subyacente. Cuando pasamos un objeto a un **Session.save ()** método, Hibernate lee el estado de las variables de ese objeto y ejecuta la consulta necesaria.

3.3. Características

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

Hibernate ofrece también un lenguaje de consulta de datos llamado **HQL** (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "criteria").

Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente **HibernateAnnotations** que implementa el estándar JPA, que es parte de esta plataforma.

3.4. Introducción Anotaciones

Tradicionalmente, la comunidad java a estado en una confusión profunda acerca precisamente de qué tipo de meta información debe estar como configuración. J2EE y populares contenedores ligeros han proporcionado descriptores de despliegue basados en XML tanto para cosas que son verdaderamente configurables entre diferentes despliegues del sistema, y para cualquier otro tipo de declaración la cual no puede ser fácilmente expresada en java. Las anotaciones de Java 5 cambian todo esto.

Ejb3 adopta las anotaciones y la "configuración por excepción" como la más fácil manera para proporcionar información a el contenedor en una forma declarativa. Seam extiende las anotaciones proporcionadas por Ejb3 con un conjunto de anotaciones para gestión del estado en forma declarativa y demarcación del contexto también en forma declarativa. Esto permite eliminar las declaraciones de beans gestionados en Jsf(managed beans) y reduce el XML requerido a justamente la información que verdaderamente pertenece a XML(las reglas de navegación Jsf).

CAPITULO IV



Ilustración 27. Logo RichFaces

4. RichFaces

4.1. Introducción

RichFaces es una librería de componentes visuales para JSF, escrita en su origen por Exadel y adquirida por Jboss. Además, RichFaces posee un framework avanzado para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF.

4.2. Características

En este apartado se pretende dar una visión de las características que RichFaces permite:

- Intensificar el conjunto de beneficios JSF al trabajar con Ajax. RichFaces está completamente integrado en el ciclo de vida de JSF. Mientras que otros marcos sólo dan acceso a los managedbean, RichFaces permite acceder al action y al valor del listener, así como invocar a validadores y convertidores durante el ciclo de petición-respuesta de Ajax.
- Añadir capacidad Ajax a aplicaciones JSF. El framework proporciona dos librerías de componentes (Core Ajax y la interfaz de usuario). La librería Core nos permite agregar la funcionalidad Ajax en las páginas que queramos sin necesidad de escribir nada de código JavaScript. RichFaces permite definir eventos en la propia página. Un evento invoca a una petición Ajax, sincronizándose así zonas de la página y componentes JSF después de recibir la respuesta del servidor por Ajax.
- Crear rápidamente vistas complejas basándose en la caja de componentes. La librería UI (Interfaz de usuario) que contiene componentes para agregar características de interfaz de usuario a aplicaciones JSF. Se amplía el framework de

RichFaces incluyendo un gran conjunto de componentes “habilitación de Ajax” que extiende el soporte de la página. Además, los componentes de RichFaces están diseñados para ser usados sin problemas con otras librerías de componentes en la misma página, de modo que existen más opciones para el desarrollo de aplicaciones

- Escribir componentes propios con función soportada por Ajax. El CDK o Kit de Desarrollo de Componentes basado en maven, incluye un generador de código para plantillas JSP utilizando una sintaxis similar. Estas capacidades ayudan a evitar un proceso de rutina de un componente de creación.
- Proporciona un paquete de recursos con clases de aplicación Java. Además de su núcleo, la funcionalidad de Rich Faces para Ajax proporciona un avanzado soporte a la gestión de diferentes recursos: imágenes, código JavaScript y hojas de estilo CSS. El framework de recursos hace posible empaquetar fácilmente estos recursos en archivos jar junto con el código de los componentes personalizados.
- Generar fácilmente recursos binarios sobre la marcha. Los recursos del framework pueden generar imágenes, sonidos, hojas de cálculo de Excel, etc.
- Crear una moderna interfaz de usuario 'look-and-feel' basadas en tecnología de skins. RichFaces proporciona una función que permite definir y administrar fácilmente diferentes esquemas de color y otros parámetros de la interfaz de usuario, con la ayuda de los parámetros del skin. Por lo tanto, es posible acceder a los parámetros del skin desde el código JSP y el código de Java (por ejemplo, para ajustar las imágenes generadas sobre la marcha basadas en la interfaz de usuario). RichFaces viene con una serie de skins predefinidas para empezar, pero también se pueden crear fácilmente skins propios.
- Permite gracias a una herramienta del framework generar casos de prueba para los componentes que estas creando (actions, listeners, etc.).
- Los componentes de la interfaz de usuario de RichFaces vienen preparados para su uso fuera del paquete, así los desarrolladores ahorrarán tiempo y podrán disponer de las ventajas mencionadas para la creación de aplicaciones Web. Como resultado, la experiencia puede ser más rápida y fácil de obtener.
- RichFaces permite definir (por medio de etiquetas de JSF) diferentes partes de una página JSF que se desee actualizar con una solicitud Ajax, proporcionando así varias opciones para enviar peticiones Ajax al servidor.

Entre los elementos que componen RichFaces se encuentran:

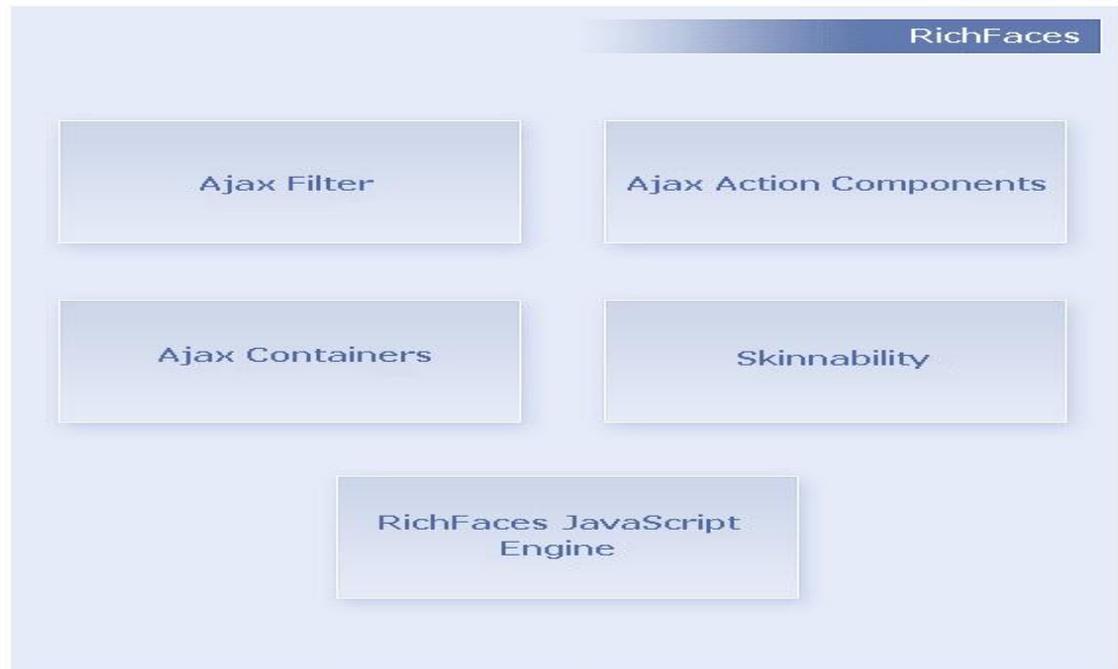


Ilustración 28. Componentes de RichFaces

4.3. Inconvenientes

Como inconvenientes, podríamos decir que:

- ✓ Usando Ajax4JSF tenemos que indicar qué parte de la pantalla tiene que repintarse. No es tan simple como ICEfaces, pero implica tener más control sobre los eventos que se producen en la interfaz de usuario.
- ✓ En las últimas versiones siempre se les cuela alguna "peora", que merma la funcionalidad de algún componente y donde, por ejemplo, funcionaba la subida de ficheros mediante un componente JSF con barra de progreso en Internet Explorer, ahora solo funciona en Firefox. Aunque también es cierto que se detecta y soluciona en la siguiente versión.

4.4. Requisitos e Incompatibilidades

RichFaces fue desarrollado con una arquitectura de código abierto para ser compatible con la más amplia variedad de entornos.

Para poder trabajar con RichFaces es necesario disponer de una aplicación JSF. Además hay que tener en cuenta la versión de los diferentes componentes implicados:

Versiones de Java soportadas:

- JDK 1.5 y superiores.

Implementaciones de JavaServer Faces soportadas:

- ✓ Sun JSF 1.1 RI
- ✓ MyFaces 1.1.1 - 1.2
- ✓ Facelets JSF 1.1.1 - 1.2
- ✓ Seam 1.2. - 2.0

Servidores soportados

- ✓ Apache Tomcat 4.1 - 6.0
- ✓ IBM WebSphere 5.1 - 6.0
- ✓ BEA WebLogic 8.1 - 9.0
- ✓ Oracle AS/OC4J 10.1.3
- ✓ Resin 3.0
- ✓ Jetty 5.1.X
- ✓ SunApplication Server 8 (J2EE 1.4)
- ✓ Glassfish (J2EE 5)
- ✓ JBoss 3.2 - 4.2.x
- ✓ SybaseEAServer 6.0.1

Navegadores admitidos

- ✓ Internet Explorer 6.0 - 7.0
- ✓ Firefox 1.5 - 2.0

- ✓ Opera 8.5 - 9.0

4.5. Interacción con otros componentes

RichFaces se integra con diferentes componentes para poder realizar aplicaciones más potentes:

4.5.1. Sun JSF RI

RichFaces trabaja con cualquier implementación de JSF (1.1 y 1.2) y con la mayoría de las bibliotecas de componentes JSF sin ninguna configuración adicional.

4.5.2. Apache MyFaces

RichFaces trabaja con todos Apache MyFaces versiones (1.1.1 - 1.1.6), incluidas las bibliotecas específicas como Tomahawk, Sandbox y Trinidad (el anterior ADF Faces). Sin embargo, hay algunas consideraciones a tener en cuenta para la configuración de las aplicaciones para trabajar con MyFaces y RichFaces. Hay algunos problemas con diferentes filtros definidos en el archivo web.xml. Para evitar estos problemas, el filtro de RichFaces debe ser el primero entre otros filtros dentro del archivo de configuración web.xml. Aún más, existen algunos problemas si se opta por utilizar MyFaces + Seam. Si se usa esta combinación, se debe usar el tag **4j: page** dentro del f: view.

4.5.3. Facelets

RichFaces ofrece un alto nivel de soporte para Facelets. Cuando se trabaja con RichFaces, no hay diferencia entre las diferentes releases de Facelets que se utilizan.

4.5.4. JBoss Seam

Una de las novedades de RichFaces 3,1 es la integración con JBossSeam. Esto mejora aún más la experiencia del usuario mediante la simplificación de la configuración y el "plumbingcode", así como la prestación de estado y la concurrencia para la gestión de Ajax.

4.6. Recomendaciones de Uso

Con el fin de crear aplicaciones RichFaces correctamente, se debe tener en cuenta los siguientes puntos:

- ✓ Cualquier framework Ajax no debería añadir o suprimir, sólo sustituir elementos de la página. Para que las actualizaciones funcionen correctamente, en la página destino debe haber un elemento con el mismo ID que existe en la respuesta del servidor. Si se desea añadir cualquier código a la página, debería ponerse en un marcador de posición (cualquier lugar con algún elemento vacío). Por la misma razón, se recomienda colocar mensajes en el componente "AjaxOutput".
- ✓ No utilizar f:verbatim, ya que este componente es transitorio y no se guarda en el árbol.
- ✓ Las peticiones Ajax se realizan gracias a funciones de XMLHttpRequest en formato XML, pero ese XML no pasa por la mayoría de validaciones y correcciones que pueden realizarse en el navegador. Por ello, sólo debe crearse código compatible con los estándares de HTML y XHTML, sin saltarse ninguno de los elementos o atributos requeridos. Cualquier corrección necesaria del XML se realizará automáticamente por el filtro XML en el servidor, pero muchos de los efectos inesperados pueden ser producidos por un código HTML incorrecto.

4.7. Enterprise JavaBeans

Los **Enterprise JavaBeans** (también conocidos por sus siglas **EJB**) son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE (ahora JEE 5.0) de Oracle Corporation (inicialmente desarrollado por Sun Microsystems). Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, precisamente, los EJB:

- Comunicación remota utilizando CORBA
- Transacciones
- Control de la concurrencia
- Eventos utilizando JMS (Java messaging service)

- Servicios de nombres y de directorio
- Seguridad
- Ubicación de componentes en un servidor de aplicaciones.

La especificación de EJB define los papeles jugados por el contenedor de EJB y los EJB, además de disponer los EJB en un contenedor.

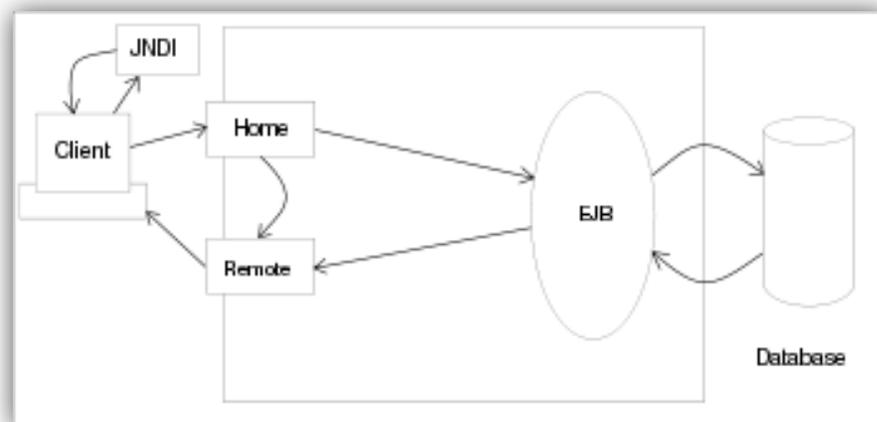


Ilustración 29. EJB

4.7.1 Definición

Los EJB proporcionan un modelo de componentes distribuido estándar del lado del servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad, etc.) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.

No hay que confundir los Enterprise JavaBeans con los JavaBeans. Los JavaBeans también son un modelo de componentes creado por Oracle - Sun Microsystems para la construcción de aplicaciones, pero no pueden utilizarse en entornos de objetos distribuidos al no soportar nativamente la invocación remota (RMI).

4.7.2 Tipos de Enterprise JavaBeans

Existen tres tipos de EJBs:

- **EJB de Entidad** (*EntityEJBs*): su objetivo es encapsular los objetos del lado del servidor que almacena los datos. Los EJB de entidad presentan la característica fundamental de la persistencia:
- **Persistencia gestionada por el contenedor** (CMP): el contenedor se encarga de almacenar y recuperar los datos del objeto de entidad mediante el mapeo o vinculación de las columnas de una tabla de la base de datos con los atributos del objeto.
- **Persistencia gestionada por el bean** (BMP): el propio objeto entidad se encarga, mediante una base de datos u otro mecanismo, de almacenar y recuperar los datos a los que se refiere, por lo cual, la responsabilidad de implementar los mecanismos de persistencia es del programador.

Nota: En la documentación de java para JEE 5.0, los entitybeans desaparecen ya que son remplazados por JPA (Java Persistence API).¹

- **EJB de Sesión** (*SessionEJBs*): gestionan el flujo de la información en el servidor. Generalmente sirven a los clientes como una fachada de los servicios proporcionados por otros componentes disponibles en el servidor. Puede haber dos tipos:
- **Con estado** (*stateful*). En un bean de sesión con estado, las variables de instancia del bean almacenan datos específicos obtenidos durante la conexión con el cliente. Cada bean de sesión con estado, por tanto, almacena el estado conversacional de un cliente que interactúa con el bean. Este estado conversacional se modifica conforme el cliente va realizando llamadas a los métodos de negocio del bean. El estado conversacional no se guarda cuando el cliente termina la sesión.
- **Sin estado** (*stateless*). Los beans de sesión sin estado son objetos distribuidos que carecen de estado asociado permitiendo por tanto que se los acceda concurrentemente. No se garantiza que los contenidos de las variables de instancia se conserven entre llamadas al método.
- **EJB dirigidos por mensajes** (*Message-drivenEJBs*): son los únicos beans con funcionamiento asíncrono. Usando el *Java MessagingSystem* (JMS), se suscriben a un

tema (*topic*) o a una cola (*queue*) y se activan al recibir un mensaje dirigido a dicho tema o cola. No requieren de su instanciación por parte del cliente.

4.7.3 Funcionamiento de un Enterprise JavaBean

Los EJB se disponen en un contenedor EJB dentro del servidor de aplicaciones. La especificación describe cómo el EJB interactúa con su contenedor y cómo el código cliente interactúa con la combinación del EJB y el contenedor.

Cada EJB debe facilitar una clase de implementación Java y dos interfaces Java. El contenedor EJB creará instancias de la clase de implementación Java para facilitar la implementación EJB. Las interfaces Java son utilizadas por el código cliente del EJB. Las dos interfaces, conocidas como interfaz "home" e interfaz remota, especifican las firmas de los métodos remotos del EJB. Los métodos remotos se dividen en dos grupos:

- Métodos que no están ligados a una instancia específica, por ejemplo aquellos utilizados para crear una instancia EJB o para encontrar una entidad EJB existente. Estos métodos se declaran en la interfaz "home".
- Métodos ligados a una instancia específica. Se ubican en la interfaz remota.

Dado que se trata simplemente de interfaces Java y no de clases concretas, el contenedor EJB genera clases para esas interfaces que actuarán como un proxy en el cliente. El cliente invoca un método en los proxies generados que a su vez sitúa los argumentos método en un mensaje y envía dicho mensaje al servidor EJB. Los proxies usan RMI-IIOP para comunicarse con el servidor EJB.

El servidor llamará a un método correspondiente a una instancia de la clase de implementación Java para manejar la llamada del método remoto.

4.7.4 Interfaz "Home"

La interfaz "Home" permite al código cliente manipular métodos de clase del EJB que no están asociados a ninguna instancia particular. La Interface "Home" permite crear las instancias de EJB de entidad o sesión a través del método create que puede ser sobrecargado.

La especificación EJB 1.1 establece el tipo de métodos de clase que se pueden definir como métodos que crean un EJB o para encontrar un EJB existente si es un "bean" de entidad.

La especificación EJB 2.0 permite a los desarrolladores de aplicaciones definir nuevos métodos de clase sin limitarse a su sola creación o borrado.

4.7.5 Interfaz remota

La interfaz remota especifica los métodos de instancia públicos encargados de realizar las operaciones.

Una sesión bean puede implementar múltiples interfaces, con cada interfaz apuntada por un tipo de cliente diferente. La interfaz local es para aquellos clientes que corren en la misma máquina virtual que el contenedor EJB. La interfaz remota es para clientes remotos. Frente a una consulta del cliente, el contenedor retorna un stub serializado del objeto que implementa la interfaz remota. El stub conoce cómo pasar llamadas a procedimientos remotos (RPCs) al servidor. Este tipo de interfaz es también un POJO.

4.7.6 Clase de implementación EJB

Las clases de implementación EJB las suministran los desarrolladores de aplicaciones, que facilitan la lógica de negocio ("businesslogic") o mantienen los datos ("business data") de la interfaz de objeto, esto es, implementan todos los métodos especificados por la interfaz remota y, posiblemente, algunos de los especificados por la interfaz "home".

4.7.7 Correspondencia entre métodos de interfaz y métodos de implementación

Las llamadas al método en la interfaz "home" se remiten al método correspondiente de la clase de implementación del bean con el prefijo 'ejb' añadido y con la primera letra de la interfaz "home" convertida en mayúscula y manteniendo exactamente el mismo tipo de argumentos¹¹. Por ejemplo:

```
create --->ejbCreate.
```

¹¹Developing EJB Applications

Las llamadas a métodos en la interfaz remota se remiten al método de implementación correspondiente del mismo nombre y argumentos en la clase del bean.

La complejidad ciclomática de las unidades semánticas de navegación (NavigationSemanticUnit (NSU)) no cumple el estándar UML 2.0 que recomienda el uso de screenshots sobre componentes programáticos.

6. METODOLOGÍA

1.1 Métodos

1.1.1 Método Científico

Este método nos ayudará para poder organizar y sistematizar el proceso investigativo del presente trabajo para la demostración de resultados; en donde se partirá de la revisión de teorías sobre Planeación Financiera, recopilación de la información de la institución hasta llegar a obtención de resultados reflejados en las conclusiones.

1.1.2 Método Inductivo

El método inductivo o inductivismo es un método científico que, partiendo de casos particulares, se eleva a conocimientos generales. Este método permite la formación de hipótesis

Nos permitirá alcanzar una solución general para cada uno de los problemas que se pueden presentar en los clientes, empleados, proveedores formulando consigo una serie de suposiciones a problemas específicos que se pueden presentar dentro del entorno empresarial.

1.1.3 Método Deductivo

Se basa en ir encadenando conocimientos que se suponen verdaderos de manera tal que se obtienen de nuevos conocimientos; es decir, es aquel que combina principios necesarios y simples (axiomas postulados, teoremas, conceptos no definidos, definiciones, etc.) para deducir nuevas proposiciones.

El mismo nos servirá para lograr optar por nuevas alternativas de solución que se puede presentar dentro del desarrollo de proyecto, mediante una solución general nos ayude a solventar problemas de menor prioridad.

1.1.4 Método Analítico

Es aquel método de investigación que consiste en la desmembración de un todo, descomponiéndolo en sus partes o elementos para observar las causas, la naturaleza y los efectos. El análisis es la observación y examen de un hecho en particular. Es necesario conocer la naturaleza del fenómeno y objeto que se estudia para comprender su esencia. Este método nos permite conocer más del objeto de estudio, con lo cual se puede: explicar, hacer analogías, comprender mejor su comportamiento y establecer nuevas teorías.

No facilitar el análisis para llegar a soluciones adecuadas previa separación de cada problema en subproblemas, que contribuirá a profundizarnos en el objeto de estudio en cuestión.

1.1.5 Metodología ICONIX

Consiste en un lenguaje de modelamiento y proceso que define que define quien debe hacer que y cuando para lograr cumplir los objetivos, puesto que presenta claramente las actividades a seguir en cada etapa, minimizando considerablemente la documentación de desarrollo.

Proporciona un proceso ágil para la obtención de especificaciones de requerimientos y modelar el comportamiento de sistemas, utilizando un lenguaje de modelado unificado (UML), que favorece la participación de usuarios finales y la documentación detallada de todo el proceso.

En el desarrollo del presente proyecto usar ICONIX nos permitirá llevar un orden adecuado de cada una de las actividades que se deben cumplir, alcanzando un esquema de desarrollo adecuado, en el que predomine el orden del modelado.

1.2 Técnicas

1.2.1 Entrevista

Una entrevista es un dialogo en el que la persona (entrevistador), generalmente un periodista hace una serie de preguntas a otra persona (entrevistado), con el fin de conocer mejor sus ideas, sus sentimientos su forma de actuar.

Está técnica la utilizaremos para obtener la mayor cantidad de información que nos provean todos las personas involucradas directa o indirectamente con la empresa.

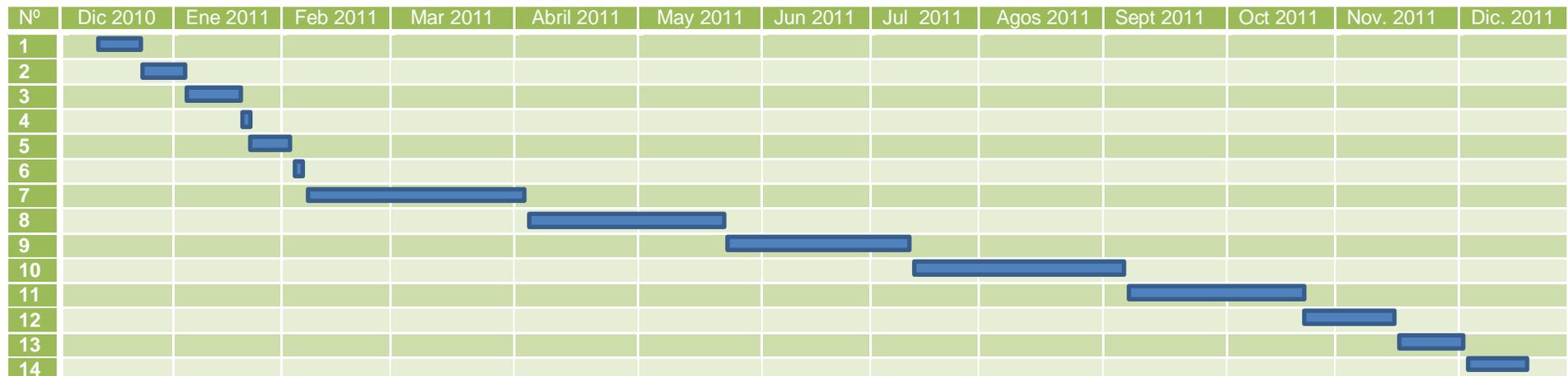
1.2.2 Observación

La observación es una actividad realizada por un ser vivo (como un ser humano), que detecta y asimila la información de un hecho, o el registro de los datos utilizando los sentidos como instrumentos principales. El término también puede referirse a cualquier dato recogido durante esta actividad.

Contribuirá en la determinación de todos los inconvenientes que se presentan durante el desarrollo de las actividades dentro y fuera de la empresa.

7. Cronograma

	NOMBRE	DURACIÓN	INICIO	TERMINADO
ANÁLISIS Y DISEÑO DEL SISTEMA OPENLOJA				
1	Establecer los Requerimientos del Sistema	8 días	13/12/2010	22/12/2010
2	Elaborar Modelo de Dominio y Diagrama de Clases	9 días	23/12/2010	04/01/2011
3	Elaborar Diagrama de Casos de Uso	10 días	05/01/2011	18/01/2011
4	Elaborar Descripción de Casos de Uso	3 días	19/01/2011	21/01/2011
5	Elaborar Prototipado de Pantallas	9 días	24/01/2011	03/02/2011
6	Elaborar Diagramas de Robustez y Secuencia	3 días	04/02/2011	08/02/2011
ELABORAR LOS MÓDULOS DEL SISTEMA OPENLOJA				
7	Construcción del Módulo de Inventario	40 días	09/02/2011	05/04/2011
8	Construcción del Módulo de Ventas	35 días	06/04/2011	24/05/2011
9	Construcción del Módulo de Compras	35 días	25/05/2011	12/07/2011
10	Construcción del Módulo de Recursos Humanos	40 días	13/07/2011	06/09/2011
11	Construcción del Módulo de Comunicación	35 días	07/09/2011	25/10/2011
12	Integración de Módulos	15 días	26/10/2011	15/11/2011
13	Implementar el Sistema	13 días	16/11/2011	02/12/2011
14	Ejecutar Plan de Validación	12 días	03/12/2011	20/12/2011



8. PRESUPUESTOS Y FINANCIAMIENTO

8.1. Recursos Humanos

Recursos Humanos	Cantidad	Horas	Costo por Hora	Costo Total
Tesistas	2	2456	5.00	12280
Director de Tesis	1	0	0	0
Subtotal				\$12280

Tabla 197. Recursos Humanos

8.2. Recursos Técnicos y Tecnológicos

Recursos Técnicos	Cantidad	Horas	Costo por Hora	Costo Total
Hardware				
Computadoras	2	2000	0.40	\$800
Impresora Canon MP250	1	150	0.50	\$95
Software				
Eclipse Helios 3.6	2		Libre	\$0
Java JDK 1.6	2		Libre	\$0
Licencia de Microsoft Project	2		\$170	\$340
Seam	2		Libre	\$0
JBoss	2		Libre	\$0
RichFaces	2		Libre	\$0
MySQL	2		Libre	\$0
Ant	2		Libre	\$0
EJB	2		Libre	\$0
Subtotal				\$1235

Tabla 198. Recursos Técnicos y Tecnológicos

8.3 Recursos Materiales

Recursos Materiales	Cantidad	Costo Unitario	Costo Total
Resma de Papel Bond	5	\$4	\$20
Bolígrafos	4	\$0.30	\$1.20



Cartuchos de Tinta	6	\$22	\$132
Internet/Horas	600	\$0.7	\$420
Subtotal			\$573.2

Tabla 199. Recursos Materiales

8.4 Total de Recursos

Resumen del Presupuestos	Costo Total
Recursos Humanos	\$12280
Recursos Técnicos y Tecnológicos	\$1215
Recursos Materiales	\$573.2
Total	\$ 14068.2

Tabla 200. Total Recursos

9. Bibliografía

Libros

- **Esteves, P. J.** (2001). Enterprise Resource Planning Systems. (2ª Ed.).
- **Gonzales, J. C.** CRITERIOS Y COSTES DE IMPLANTACIÓN DE UN ERP. (1ª Ed.).
- **Ortuño, M.** (2010). IMPLANTACIÓN DE UN SISTEMA ERP EN UNA ORGANIZACIÓN. (3ª Ed.)2008.

Recursos de internet

- Universidad Politécnica Salesiana Repositorio Digital. (s.f.). Recuperado el 24 de Septiembre del 2011 de http://www.dspace.ups.edu.ec/bitstream/123456789/1616/5/Capitulo_4.pdf
- SEAM. (s.f.). Recuperado el 26 de marzo del 2011 de: <http://seamframework.org/>

10. MATRICES

MATRIZ DE CONSISTENCIA GENERAL

PROBLEMA GENERAL DE INVESTIGACIÓN (ENUNCIADO):

“El software de gestión denominado OPENMARCO con que cuenta la empresa de servicios informáticos Lojanet Cía. Ltda.; no cubre adecuadamente con todas las necesidades que se desarrollan en su entorno laboral”.

TEMA	OBJETO DE INVESTIGACIÓN (el que)	OBJETIVO DE LA INVESTIGACIÓN (para que)	HIPÓTESIS DE INVESTIGACIÓN (propuesta de solución)
“Sistema Planificador de Recursos Empresariales (ERP) integrado a un módulo de comunicación con clientes para la empresa de servicios informáticos Lojanet++ Cía. Ltda.”	Construcción de un Sistema Planificador de Recursos Empresariales (ERP) integrado a un módulo de comunicación con clientes para la empresa de servicios informáticos Lojanet++.	Desarrollar un Sistema Planificador de Recursos Empresariales (ERP) integrado a un módulo de comunicación con clientes para la empresa de servicios informáticos Lojanet++.	Con el desarrollo de este Sistema la institución tendrá una eficiente y eficaz manera de laborar en sus funciones cotidianas, ayudará en la administración del inventario, control de ventas y compras, gestión de RRHH y envió y recepción de notificación de entes involucrados en los procesos de la institución.

Tabla 201. Matriz de consistencia general

MATRIZ DE CONSISTENCIA ESPECÍFICA (1)

OBJETIVO ESPECÍFICO: Analizar los Requerimientos del sistema OPENLOJA.		
PROBLEMAS	UNIDAD DE OBSERVACIÓN	SISTEMA CATEGORIAL
<ul style="list-style-type: none"> Falta de conocimiento de todos los requerimientos de para desarrollar el sistema. 	<ul style="list-style-type: none"> Gerente. Clientes. Empleados. Proveedores. 	<ul style="list-style-type: none"> Antecedentes Visión Misión Políticas Reglamento Interno Estructura Interna y Externa. <ul style="list-style-type: none"> a. Orgánico Estructural b. Orgánico Funcional

Tabla 202. Matriz de consistencia específica 1

MATRIZ DE OPERATIVIDAD DE OBJETIVOS (1)

OBJETIVO ESPECÍFICO: Analizar los Requerimientos del sistema OPENLOJA.

ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
<ul style="list-style-type: none"> Realizar una entrevista al gerente y los empleados para obtener la mayor cantidad de información, Observar las actividades que se desarrollan dentro y fuera de la empresa para determinar que inconvenientes suceden. 	<ul style="list-style-type: none"> Entrevista Observación 	13/12/2010	13/12/2010	Jorge Luis Tapia. Vladimir Romero.	\$ 5.00	<ul style="list-style-type: none"> Obtener la información requerida. Establecer los requerimientos del sistema.
		14/12/2010	15/12/2010		\$ 5.00	

Tabla 203. Matriz de operatividad de objetivos 1

MATRIZ DE CONSISTENCIA ESPECÍFICA (2)

OBJETIVO ESPECÍFICO: Diseñar los módulos de compras, ventas, inventario, RR HH, comunicación del sistema OPENLOJA.		
PROBLEMAS	UNIDAD DE OBSERVACIÓN	SISTEMA CATEGORIAL
<ul style="list-style-type: none"> ✓ Mal manejo, almacenamiento y respaldo de la toda la información. ✓ Inadecuada administración de los procesos tareas que se realizan en la empresa. ✓ Incorrecto manejo de inventario. ✓ Inexistencia de un módulo de comunicación entre clientes, empleados y proveedores con la empresa. 	<ul style="list-style-type: none"> • Clientes. • Empleados. • Proveedores. 	<ul style="list-style-type: none"> • JBoss Seam Framework. • Hibernate. • ReachFaces. ✓ S J F. ✓ Faceletes. • Enterprise JavaBeans.

Tabla 204. Matriz de consistencia específica 2

MATRIZ DE OPERATIVIDAD DE OBJETIVOS (2)

OBJETIVO ESPECÍFICO: Diseñar los módulos de compras, ventas, inventario, RR HH, comunicación del sistema OPENLOJA.

ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
<ul style="list-style-type: none"> • Determinar los requerimientos funcionales y no funcionales. • Establecer el modelo de dominio y Diagrama de clases. • Determinar los casos de uso y su descripción. • Diseñar el prototipado de pantallas. • Realizar los diagramas de secuencia y robustez. 	<ul style="list-style-type: none"> • Iconix 	6/12/2010	22/12/2010	Jorge Luis Tapia. Vladimir Romero.	\$ 244.50	<ul style="list-style-type: none"> • Requerimientos funcionales y no funcionales del sistema. • Diagrama de casos de uso y su descripción. • Modelo de dominio. • Diagramas de secuencia y robustez. • Diagrama de clases final.
		23/12/2010	04/01/2011		\$ 440.10	
		05/01/2011	24/01/2011		\$ 684.6	
		25/01/2011	09/02/2011		\$ 586.80	
		10/02/2011	14/02/2011		\$ 146.70	

Tabla 205. Matriz de operatividad de objetivos 2

MATRIZ DE CONSISTENCIA ESPECÍFICA (3)

OBJETIVO ESPECÍFICO: Construir módulos de compras, ventas, inventario, RR HH, comunicación del sistema OPENLOJA.		
PROBLEMAS	UNIDAD DE OBSERVACIÓN	SISTEMA CATEGORIAL
<ul style="list-style-type: none"> ✓ Mal manejo, almacenamiento y respaldo de la toda la información. ✓ Inadecuada administración de los procesos y tareas que se realizan en la empresa. ✓ Incorrecto manejo de inventario. ✓ Inexistencia de un módulo de comunicación entre clientes, empleados y proveedores con la empresa. 	<ul style="list-style-type: none"> • Sistema OPENLOJA. 	<ul style="list-style-type: none"> • JBoss Seam Framework. • Hibernate. • ReachFaces.. • Enterprise JavaBeans.

Tabla 206. Matriz de consistencia específica 3

MATRIZ DE OPERATIVIDAD DE OBJETIVOS (3)

OBJETIVO ESPECÍFICO: Construir módulos de compras, ventas, inventario, RR HH, comunicación del sistema OPENLOJA..						
ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
<ul style="list-style-type: none"> • Construir el módulo de inventario. • Construir el módulo de ventas. • Construir el módulo de compras. • Construir el módulo de RR HH. • Construir el módulo de comunicación. 	<ul style="list-style-type: none"> • Iconix 	15/02/2011	11/04/2011	Jorge Luis Tapia. Vladimir Romero.	\$ 1956.00	<ul style="list-style-type: none"> • Módulo de compras. • Módulo de ventas. • Módulo de inventario. • Módulo de RR HH. • Módulo de comunicación.
		12/04/2011	30/05/2011		\$ 1711.50	
		31/05/2011	18/07/2011		\$ 1711.50	
		19/07/2011	12/09/2011		\$ 1956.00	
		13/09/2011	07/11/2011		\$ 1956.00	

Tabla 207. Matriz de operatividad 3

MATRIZ DE CONSISTENCIA ESPECÍFICA (4)

OBJETIVO ESPECÍFICO: Integrar los módulos del sistema OPENLOJA previamente construidos.		
PROBLEMAS	UNIDAD DE OBSERVACIÓN	SISTEMA CATEGORIAL
<ul style="list-style-type: none"> ✓ Mal manejo, almacenamiento y respaldo de la toda la información. ✓ Inadecuada administración de los procesos y tareas que se realizan en la empresa. ✓ Módulos previamente construidos no concuerdan para integrarse. 	<ul style="list-style-type: none"> • Clientes. • Empleados. • Proveedores. • Sistema OPENLOJA. 	<ul style="list-style-type: none"> • JBoss Seam Framework. • Hibernate. • ReachFaces.. • Enterprise JavaBeans.

Tabla 208. Matriz de consistencia específica 4

MATRIZ DE OPERATIVIDAD DE OBJETIVOS (4)

OBJETIVO ESPECÍFICO: Integrar los módulos del sistema OPENLOJA previamente construidos.						
ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
<ul style="list-style-type: none"> Integrar los módulos de compras, ventas, inventario, RR HH y comunicación, previamente construidos. 	<ul style="list-style-type: none"> Iconix 	08/11/2011	21/11/2011	Jorge Luis Tapia. Vladimir Romero.	\$ 489.00	<ul style="list-style-type: none"> Sistema planificador de Recursos empresariales (ERP) OPENLOJA.

Tabla 209. Matriz de operatividad de objetivos 4

MATRIZ DE CONSISTENCIA ESPECÍFICA (5)

OBJETIVO ESPECÍFICO: Implementar el sistema OPENLOJA en la empresa Lojanet++.		
PROBLEMAS	UNIDAD DE OBSERVACIÓN	SISTEMA CATEGORIAL
<ul style="list-style-type: none"> ✓ El software de gestión denominado OPENMARCO con que cuenta la empresa de servicios informáticos Lojanet Cía. Ltda.; no cubre adecuadamente con todas las necesidades que se desarrollan en su entorno laboral. ✓ Desperdicio de tiempo al realizar las tareas. 	<ul style="list-style-type: none"> • Clientes. • Empleados. • Proveedores. • Sistema OENLOJA. 	<ul style="list-style-type: none"> • JBoss Seam Framework. • Hibernate. • ReachFaces.. • Enterprise JavaBeans.

Tabla 210. Matriz de consistencia específica 5

MATRIZ DE OPERATIVIDAD DE OBJETIVOS (5)

OBJETIVO ESPECÍFICO: : Implementar el sistema OPENLOJA en la empresa Lojanet++.						
ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
<ul style="list-style-type: none"> Implementar el sistema de recursos empresariales OPENLOJA en la empresa Lojanet Cia. Ltda. 	<ul style="list-style-type: none"> ICONIX. 	22/11/2010	28/11/2011	Jorge Luis Tapia. Vladimir Romero.	\$ 244.50	<ul style="list-style-type: none"> Sistema planificador de Recursos empresariales (ERP) OPENLOJA Implementado en la empresa Lojanet Cia. Ltda.

Tabla 211. Matriz de operatividad de objetivos 5

MATRIZ DE CONSISTENCIA ESPECÍFICA (6)

OBJETIVO ESPECÍFICO: Ejecutar el plan de validación al sistema OPENLOJA.		
PROBLEMAS	UNIDAD DE OBSERVACIÓN	SISTEMA CATEGORIAL
<p>✓ Sistema implementado puede tener errores que se han pasado por alto en la construcción.</p>	<ul style="list-style-type: none"> • Clientes. • Empleados. • Proveedores. • Sistema OPENLOJA. 	<ul style="list-style-type: none"> • Sistema planificador de recursos empresariales (ERP). • JBoss Seam Framework. • Testeabilidad.

Tabla 212. Matriz de consistencia específica 6

MATRIZ DE OPERATIVIDAD DE OBJETIVOS (6)

OBJETIVO ESPECÍFICO: Ejecutar el plan de validación al sistema OPENLOJA.						
ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
<ul style="list-style-type: none"> Realizar una prueba de manejo del sistema a los empleados y gerente de la empresa. Realizar pruebas al sistema ERP. 	<ul style="list-style-type: none"> Prueba mediante datos ficticios. 	29/11/2011	30/11/2011	Jorge Luis Tapia. Vladimir Romero.	\$ 0.00	<ul style="list-style-type: none"> Sistema planificador de Recursos empresariales (ERP) OPENLOJA evaluado y mejorado.
		01/12/2011	12/12/2011		\$ 489.00	

Tabla 213. Matriz de operatividad de objetivos 6



ENCUESTAS

2. ENCUESTAS

ENCUESTA DIRIGIDA AL ADMINISTRADOR DEL SISTEMA DE LOJANET++ SITEMA OPENLOJA

Departamento:..... Fecha:.....

1. ¿El sistema implementado en su empresa cumple con los requerimientos que ésta solicita?

Si () No ()

¿Por qué?

.....
.....

2. ¿El sistema diseñado para su empresa puede ser instalado tanto en Linux como en Windows?

Si () No ()

3. De la instalación del sistema, como considera los siguientes aspectos:

INSTALACIÓN	Malo	Regular	Normal	Bueno	Muy Bueno
Facilidad					
Confianza					
Requerimientos					
Estabilización					

4. En lo que respecta a la configuración del sistema que puede determinar en relación a los siguientes aspectos:

CONFIGURACIÓN	Malo	Regular	Normal	Bueno	Muy Bueno
Facilidad					
Confianza					
Requerimientos					
Estabilización					

5. ¿El sistema le permite ingresar al sistema de acuerdo al rol que se la ha asignado en la empresa?

Si () No ()

6. ¿El sistema indica qué campos son obligatorios de ingresar, así como también valida los datos que usted ingresa?

Si () No ()

7. ¿La interfaz gráfica del sistema esta adecuada en relación a la función o actividad que vaya a desarrollar?

Si () No ()

8. ¿El sistema se puede ejecutar y responde normalmente en cualquier navegador como mozilla firefox, internet explorer, google chrome, entre otros?

Si () No ()

9. ¿Cree que es apropiado el cierre de sesión luego de un tiempo de inactividad en su cuenta?

Si () No ()

¿Por qué?

.....
.....

10. ¿Considera adecuada la opción de recuperar su contraseña para ingresar al sistema vía mail en caso la haya olvidado?

Si () No ()

¿Por qué?

.....
.....

11. En lo que respecta a la información sobre el manejo del sistema (manuales) como considera los siguientes aspectos:

MANUALES	Malo	Regular	Normal	Bueno	Muy Bueno
Expectativas					
Comprensibles					
Disponibilidad					

12. Durante su interacción con el sistema en los siguientes aspectos como respondió éste ante sus peticiones:

CAPACIDAD DE RESPUESTA	Malo	Regular	Normal	Bueno	Muy Bueno
Persistencias					
Consultas					
Impresiones					
Validaciones					

.....

Firma

GRACIAS POR SU COLABORACIÓN



ENCUESTA DIRIGIDA AL PERSONAL DE LA EMPRESA LOJANET++
SISTEMA OPENLOJA

Departamento:..... Fecha:.....

1. ¿El sistema implementado en la empresa, en su departamento, cumple con los requerimientos que éste solicita?

Si () No ()

¿Por qué?

.....
.....

2. ¿El sistema le permite ingresar al sistema de acuerdo al rol que se le ha asignado en la empresa?

Si () No ()

3. ¿El sistema indica qué campos son obligatorios de ingresar, así como también valida los datos que usted ingresa?

Si () No ()

4. ¿La interfaz gráfica del sistema esta adecuada en relación a la función o actividad que vaya a desarrollar?

Si () No ()

5. ¿El sistema se puede ejecutar en cualquier navegador como mozilla, internet explorer, google chrome?

Si () No ()

6. ¿Cree conveniente el cierre de sesión luego de un tiempo de inactividad del sistema?

Si () No ()

¿Por qué?

.....
.....

7. ¿Considera adecuada la opción de recuperar su contraseña para ingresar al sistema vía mail en caso que no la recuerde?

Si () No ()

¿Por qué?

.....
.....

8. En lo que respecta a la información sobre el manejo del sistema (manuales) como considera los siguientes aspectos:

MANUALES	Malo	Regular	Normal	Bueno	Muy Bueno
Expectativas					
Comprensibles					
Disponibilidad					

9. Durante su interacción con el sistema en los siguientes aspectos como respondió éste ante sus peticiones:

CAPACIDAD DE RESPUESTA	Malo	Regular	Normal	Bueno	Muy Bueno
Persistencias					
Consultas					
Impresiones					
Validaciones					

.....

Firma

GRACIAS POR SU COLABORACIÓN



CERTIFICACIÓN
