

ESCUELA POLITÉCNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

IMPLEMENTACIÓN DE UN SOFTWARE ORIENTADO A LA
WEB QUE GESTIONE LA APLICACIÓN DE LA TÉCNICA DE
CALIDAD SEIS SIGMA AL PROCESO DE DESARROLLO DE
SOFTWARE, SOBRE LA PLATAFORMA “Java Enterprise
Edition 5.0” EMPLEANDO UN FRAMEWORK INTEGRADOR
“JBoss Seam 2.2.0”

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR:

SR. YACCHIREMA ESPÍN LUIS MIGUEL

SANGOLQUÍ, Agosto del 2011

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. YACCHIREMA ESPÍN LUIS MIGUEL, como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Sangolquí, Agosto del 2011

ING. CECILIA HINOJOSA R.
Directora

DEDICATORIA

Dedico el presente proyecto de tesis a quienes consideren de gran utilidad el contenido de este trabajo.

Yacchirema Espín Luis Miguel

AGRADECIMIENTOS

Al finalizar con éxito tan anhelado proyecto, quisiera brindar mis más sinceros agradecimientos a las siguientes personas:

A mis padres, Miguel Ángel Yacchirema Jiménez, y Luz María Espín Atiaja, por su apoyo desinteresado en todas las etapas de mi vida, y sobre todo, por haber forjando en mí valores que han influido positivamente en el alcance de todas mis metas propuestas.

A mis hermanas, María Soledad, y Ana Maribel, por ser mis más fieles amigas en todo momento.

Finalmente, quiero agradecer a mis profesores tutores, la Ing. Cecilia Hinojosa, Directora, y el Ing. Edison Lascano, Cordinador; por impartir sus conocimientos que fueron muy indispensables en la realización de mi proyecto de tesis.

Yacchirema Espín Luis Miguel

Índice de Contenidos

RESUMEN	1
CAPÍTULO I.....	2
INTRODUCCIÓN	2
1.1 TEMA	2
1.2 ANTECEDENTES	2
1.3 PLANTEAMIENTO DEL PROBLEMA	3
1.4 JUSTIFICACIÓN.....	5
1.5 ALCANCE.....	6
1.6 OBJETIVOS.....	9
1.6.1 <i>Objetivo General</i>	9
1.6.2 <i>Objetivos Específicos</i>	9
CAPÍTULO II.....	10
MARCO TEÓRICO	10
2.1 GUÍA BÁSICA DE APLICACIÓN DE LA TÉCNICA SEIS SIGMA AL PROCESO DE DESARROLLO DE SOFTWARE.....	10
2.2 METODOLOGÍA ICONIX	12
2.2.1 <i>Análisis de Requerimientos</i>	14
2.2.2 <i>Análisis y Diseño Preliminar</i>	25
2.2.3 <i>Diseño Detallado</i>	30
2.2.4 <i>Implementación</i>	33
2.2.5 <i>Resumen ICONIX</i>	34
2.3 PLATAFORMA JAVA EE 5.0	35
2.3.1 <i>Introducción</i>	35
2.3.2 <i>Que es Java EE</i>	36
2.3.3 <i>Arquitectura Java EE</i>	37
2.3.4 <i>Componentes de la arquitectura Java EE</i>	38
2.3.5 <i>Tecnologías y APIs más relevantes</i>	40
2.3.5.1 <i>Java Database Connectivity (JDBC)</i>	40
2.3.5.2 <i>Enterprise JavaBeans (EJB 3.0)</i>	44
2.3.5.3 <i>Java Persistence API (JPA)</i>	48
2.3.5.4 <i>JavaServer Faces (JSF)</i>	54
2.4 HERRAMIENTAS DE DISEÑO.....	55
2.4.1 <i>Introducción</i>	55
2.4.2 <i>Enterprise Architect</i>	55
2.5 HERRAMIENTAS DE DESARROLLO	57
2.5.1 <i>Introducción</i>	57
2.5.2 <i>Entorno de Desarrollo Integrado (IDE)</i>	57
2.5.3 <i>Control de Versiones</i>	58
2.5.4 <i>Sistema de Gestión de Base de Datos Relacionales (RDBMS)</i>	60
2.5.5 <i>Servidor de Aplicaciones</i>	62
2.6 FRAMEWORK DE IMPLEMENTACIÓN.....	63
2.6.1 <i>Introducción</i>	63
2.6.2 <i>JBoss Seam Framework</i>	64
2.6.2.1 <i>Que es JBoss Seam</i>	64
2.6.2.2 <i>Características principales de JBoss Seam</i>	65

2.6.2.3	Anotaciones Seam	68
2.6.2.4	Creación de un proyecto Seam	69
CAPÍTULO III		79
DESARROLLO DEL SOFTWARE		79
3.1	ANÁLISIS DE REQUERIMIENTOS	79
3.1.1	<i>Requerimientos del Software</i>	79
3.1.1.1	Introducción	79
3.1.1.2	Requerimientos Funcionales	80
3.1.2	<i>Modelado del dominio</i>	88
3.1.2.1	Listado de posibles objetos o clases de dominio.....	88
3.1.2.2	Modelo de Dominio Inicial	90
3.1.3	<i>Modelado de Casos de Uso</i>	90
3.1.3.1	Prototipos de Interfaz de usuario.....	90
3.1.3.2	Diagrama de Casos de Uso	103
3.1.3.3	Diagrama de Paquetes	105
3.2	ANÁLISIS Y DISEÑO PRELIMINAR.....	106
3.2.1	<i>Especificación de casos de uso</i>	106
3.2.1.1	Especificación de cada caso de uso	106
3.2.2	<i>Análisis de Robustez</i>	126
3.2.2.1	Diagrama de robustez por cada caso de uso.....	126
3.2.3	<i>Modelo de dominio actualizado</i>	150
3.3	DISEÑO DETALLADO	151
3.3.1	<i>Asignación de comportamiento</i>	151
3.3.1.1	Diagrama de secuencia por cada caso de uso.	151
3.3.2	<i>Diagrama de clases</i>	186
3.3.3	<i>Modelo de datos</i>	194
3.4	IMPLEMENTACIÓN.....	195
3.5	EVALUACIÓN DEL SOFTWARE.....	195
CAPÍTULO IV		200
4.1	CONCLUSIONES.....	200
4.2	RECOMENDACIONES	202
BIBLIOGRAFÍA		203
BIOGRAFÍA		205
HOJA DE LEGALIZACIÓN DE FIRMAS		206

Listado de Tablas

TABLA 2.1: (RELACIONES PARA ASOCIAR LOS CASOS DE USO)	21
TABLA 2.2: (RESUMEN DE ICONIX).....	34
TABLA 2.3: (ANOTACIONES EJB 3/JPA).....	51
TABLA 2.4: (ANOTACIONES DE SEAM).....	68
TABLA 3.1: (PLANTILLA PARA EL REPORTE “FICHA DEL PROYECTO”).....	85
TABLA 3.2: (PLANTILLA PARA EL REPORTE “INFORMACIÓN DEL EQUIPO DE DESARROLLO”).....	85
TABLA 3.3: (PLANTILLA PARA EL REPORTE “METAS ESTABLECIDAS PARA EL PROYECTO”).....	86
TABLA 3.4: (PLANTILLA PARA EL REPORTE “INFORMACIÓN DE LOS PROCESOS DEL NEGOCIO”).....	86
TABLA 3.5: (PLANTILLA PARA EL REPORTE “CARACTERIZACIÓN DE PROCESOS”).....	86
TABLA 3.6: (PLANTILLA PARA EL REPORTE “MATRIZ DE REQUERIMIENTOS DEL CLIENTE”).....	86
TABLA 3.7: (PLANTILLA PARA EL REPORTE “REGISTRO DE REQUERIMIENTOS”).....	87
TABLA 3.8: (PLANTILLA PARA EL REPORTE “CRITERIOS DE ACEPTACIÓN DE REQUERIMIENTOS”).....	87
TABLA 3.9: (PLANTILLA PARA EL REPORTE “MATRIZ DE PRIORIZACIÓN DE REQUERIMIENTOS DEL CLIENTE”)	87
TABLA 3.10: (PLANTILLA PARA EL REPORTE “EVALUACIÓN COSTO / TIEMPO DE MEJORAS”)	87
TABLA 3.11: (PLANTILLA PARA EL REPORTE “CUESTIONARIO DE REQUERIMIENTOS DEL CLIENTE”)	88
TABLA 3.12: (LISTADO DE CASOS DE USO)	107
TABLA 3.13: (ESPECIFICACIÓN DEL CASO DE USO “INICIAR SESIÓN”)	107
TABLA 3.14: (ESPECIFICACIÓN DEL CASO DE USO “CERRAR SESIÓN”)	107
TABLA 3.15: (ESPECIFICACIÓN DEL CASO DE USO “MANEJAR CUENTA DE USUARIO DEL SISTEMA”).....	108
TABLA 3.16: (ESPECIFICACIÓN DEL CASO DE USO “LISTAR DATOS ASOCIADOS A UN PROYECTO”)	109
TABLA 3.17: (ESPECIFICACIÓN DEL CASO DE USO “LISTAR DATOS INICIALES DE PROYECTOS”).....	109
TABLA 3.18: (ESPECIFICACIÓN DEL CASO DE USO “LISTAR DATOS ASOCIADOS A UN REQUERIMIENTO”) ..	110
TABLA 3.19: (ESPECIFICACIÓN DEL CASO DE USO “FILTRAR REGISTROS LISTADOS”)	110
TABLA 3.20: (ESPECIFICACIÓN DEL CASO DE USO “EDITAR REGISTRO SELECCIONADO DE UNA LISTA”)....	111
TABLA 3.21: (ESPECIFICACIÓN DEL CASO DE USO “ELIMINAR REGISTRO SELECCIONADO DE UNA LISTA”) 111	
TABLA 3.22: (ESPECIFICACIÓN DEL CASO DE USO “MANEJAR DATOS INICIALES DEL PROYECTO”).....	112
TABLA 3.23: (ESPECIFICACIÓN DEL CASO DE USO “MANEJAR DATOS ASOCIADOS A UN PROYECTO”).....	113
TABLA 3.24: (CASOS DE USO PARA MANEJAR DATOS ASOCIADOS A UN REQUERIMIENTO).....	113
TABLA 3.25: (ESPECIFICACIÓN DEL CASO DE USO “MANEJAR DATOS ASOCIADOS A UN REQUERIMIENTO”)	114
TABLA 3.26: (ESPECIFICACIÓN DEL CASO DE USO “QUITAR ÍTEM DE UNA LISTA DE DATOS”)	114
TABLA 3.27: (ESPECIFICACIÓN DEL CASO DE USO “MANEJAR ANEXO O ARCHIVO DEL PROYECTO”).....	115
TABLA 3.28: (ESPECIFICACIÓN DEL CASO DE USO “GENERAR REPORTE”).....	115
TABLA 3.29: (ESPECIFICACIÓN DEL CASO DE USO “LISTAR PARÁMETROS GENERALES DEL SISTEMA”)	116
TABLA 3.30: (ESPECIFICACIÓN DEL CASO DE USO “MANEJAR PARÁMETROS GENERALES DEL SISTEMA”).	117
TABLA 3.31: (ESPECIFICACIÓN DEL CASO DE USO “MANEJAR PERSONA DE DESARROLLO”).....	117
TABLA 3.32: (ESPECIFICACIÓN DEL CASO DE USO “ACTUALIZAR DATOS ADICIONALES AL PROYECTO”)....	118
TABLA 3.33: (ESPECIFICACIÓN DEL CASO DE USO “CONFORMAR EQUIPO DE DESARROLLO”).....	119
TABLA 3.34: (ESPECIFICACIÓN DEL CASO DE USO “ASIGNAR PERSONAS DE DESARROLLO AL PROYECTO”)	119
TABLA 3.35: (ESPECIFICACIÓN DEL CASO DE USO “REALIZAR ANÁLISIS DE METAS DEL PROYECTO”)	120
TABLA 3.36: (ESPECIFICACIÓN DEL CASO DE USO “MANEJAR ACTIVIDAD POR PROCESO”).....	121
TABLA 3.37: (ESPECIFICACIÓN DEL CASO DE USO “IDENTIFICAR RESPONSABLE POR ACTIVIDAD”).....	121
TABLA 3.38: (ESPECIFICACIÓN DEL CASO DE USO “ASIGNAR RESPONSABLES A UNA ACTIVIDAD”)	122

TABLA 3.39: (ESPECIFICACIÓN DEL CASO DE USO “REALIZAR IDENTIFICACIÓN MACRO DE PROCESOS”) ..	123
TABLA 3.40: (ESPECIFICACIÓN DEL CASO DE USO “MOSTRAR INFORMACIÓN ACTUAL DEL REQUERIMIENTO”)	123
TABLA 3.41: (ESPECIFICACIÓN DEL CASO DE USO “ACTUALIZAR DATOS ADICIONALES AL REQUERIMIENTO”)	124
TABLA 3.42: (ESPECIFICACIÓN DEL CASO DE USO “IDENTIFICAR POSIBLE MEJORA POTENCIAL”).....	125
TABLA 3.43: (ESPECIFICACIÓN DEL CASO DE USO “PRIORIZAR LOS REQUERIMIENTOS”).....	125
TABLA 3.44: (LISTADO DE CASOS DE USO)	126
TABLA 3.45: (LISTADO DE CASOS DE USO)	151
TABLA 3.46: (RESULTADOS DE LA EVALUACIÓN DEL SOFTWARE).....	199

Listado de Figuras

FIGURA 1.1: (INFORMACIÓN DE PROYECTOS EN LA INDUSTRIA TI)	4
FIGURA 2.1: (FLUJOS DE TRABAJO DE ICONIX).....	13
FIGURA 2.2: (MODELO DE DOMINIO, EN CONTEXTO DEL PROCESO ICONIX)	15
FIGURA 2.3: (NOTACIÓN PARA REPRESENTAR UNA CLASE)	16
FIGURA 2.4: (RELACIONES PARA ASOCIAR LOS OBJETOS DE DOMINIO)	17
FIGURA 2.5: (EJEMPLO DE UN MODELO DE DOMINIO)	17
FIGURA 2.6: (MODELO DE CASOS DE USO, EN CONTEXTO DEL PROCESO ICONIX)	18
FIGURA 2.7: (REPRESENTACIÓN GRÁFICA DE UN ACTOR)	18
FIGURA 2.8: (REPRESENTACIÓN GRÁFICA DE UN CASO DE USO).....	19
FIGURA 2.9: (EJEMPLO DE UN PROTOTIPO GUI).....	20
FIGURA 2.10: (EJEMPLO DE UN DIAGRAMA DE CASOS DE USO).....	21
FIGURA 2.11: (EJEMPLO DE UN DIAGRAMA DE PAQUETES).....	22
FIGURA 2.12: (ESCENARIO DE UN CASO DE USO).....	24
FIGURA 2.13: (ANÁLISIS DE ROBUSTEZ, EN CONTEXTO DEL PROCESO ICONIX).....	26
FIGURA 2.14: (SÍMBOLOS UTILIZADOS EN EL DIAGRAMA DE ROBUSTEZ).....	27
FIGURA 2.15: (EJEMPLO DE UN DIAGRAMA DE ROBUSTEZ, PARA EL CASO DE USO MOSTRAR LISTA DE LIBROS).....	29
FIGURA 2.16: (DISEÑO DETALLADO, EN CONTEXTO DEL PROCESO ICONIX)	30
FIGURA 2.17: (NOTACIÓN DE UN DIAGRAMA DE SECUENCIA)	31
FIGURA 2.18:(IMPLEMENTACIÓN, EN CONTEXTO DEL PROCESO ICONIX).....	33
FIGURA 2.19: (ARQUITECTURA JAVA EE Y SUS COMPONENTES).....	40
FIGURA 2.20: (ESQUEMA JDBC)	41
FIGURA 2.21: (FUNCIONAMIENTO DEL CONTROLADOR JDBC DE TIPO 1)	42
FIGURA 2.22: (FUNCIONAMIENTO DEL CONTROLADOR JDBC DE TIPO 2)	43
FIGURA 2.23: (FUNCIONAMIENTO DEL CONTROLADOR JDBC DE TIPO 3)	43
FIGURA 2.24: (FUNCIONAMIENTO DEL CONTROLADOR JDBC DE TIPO 4)	43
FIGURA 2.25: (INTERPRETACIÓN DE UN EJB).....	44
FIGURA 2.26: (CAPAS Y SERVICIOS SOPORTADOS POR LA TECNOLOGÍA EJB 3.0)	45
FIGURA 2.27: (ORGANIZACIÓN DEL API EJB3)	46
FIGURA 2.28: (EJEMPLO DE USO DE LAS ANOTACIONES JPA, PARTE 1)	52
FIGURA 2.29: (EJEMPLO DE USO DE LAS ANOTACIONES JPA, PARTE 2)	52
FIGURA 2.30: (EJEMPLO DE USO DE LAS ANOTACIONES JPA, PARTE 3)	53
FIGURA 2.31: (MODELO MVC).....	54
FIGURA 2.32: (ENTERPRISE ARCHITECT GUI).....	55
FIGURA 2.33: (NETBEANS 6.7.1 GUI).....	57
FIGURA 2.34: (VISUALSVN SERVER 2.0.8 GUI).....	59
FIGURA 2.35: (MYSQL ADMINISTRATOR GUI).....	60
FIGURA 2.36: (ESTRUCTURA DE DIRECTORIOS DE JBOSS)	62
FIGURA 2.37: (FRAMEWORKS PARA EL DESARROLLO DE APLICACIONES WEB).....	64
FIGURA 2.38: (EJEMPLO DE ALGUNAS TECNOLOGÍAS QUE INTEGRA SEAM)	65
FIGURA 2.39: (CONTEXTOS DE SEAM).....	66
FIGURA 2.40: (ARCHIVO DESCRIPTOR DE SEAM PARA EL CONTROL DE NAVEGACIÓN).....	67
FIGURA 2.41: (PROYECTO SEAM, EJECUCIÓN DEL COMANDO “SEAM SETUP”)	69
FIGURA 2.42: (PROYECTO SEAM, RESULTADO DEL COMANDO “SEAM SETUP”).....	72

FIGURA 2.43: (PROYECTO SEAM, EJECUCIÓN DEL COMANDO “SEAM NEW-PROJECT”)	72
FIGURA 2.44: (PROYECTO SEAM, RESULTADO DEL COMANDO “SEAM NEW-PROJECT”)	73
FIGURA 2.45: (PROYECTO SEAM, VISUALIZACIÓN DEL PROYECTO SEAM CON NETBEANS)	73
FIGURA 2.46: (PROYECTO SEAM, SCRIPT DE CREACIÓN DE LA BASE DE DATOS)	74
FIGURA 2.47: (PROYECTO SEAM, EJECUCIÓN DEL COMANDO “SEAM GENERATE-ENTITIES”)	75
FIGURA 2.48: (PROYECTO SEAM, RESULTADO DEL COMANDO “SEAM GENERATE-ENTITIES”)	75
FIGURA 2.49: (PROYECTO SEAM, EJECUCIÓN DE LA TAREA “DEPLOY”)	76
FIGURA 2.50: (PROYECTO SEAM, EJECUCIÓN DEL COMANDO “RUN.BAT”)	77
FIGURA 2.51: (PROYECTO SEAM, INTERFAZ DE LA PÁGINA PRINCIPAL Y LOGIN)	77
FIGURA 2.52: (PROYECTO SEAM, INTERFACES PARA MANTENER PAÍSES Y PROVINCIAS)	78
FIGURA 3.1: (MODELO DE DOMINIO INICIAL)	90
FIGURA 3.2: (PÁGINA PRINCIPAL)	91
FIGURA 3.3: (PÁGINA DE INICIO DE SESIÓN)	91
FIGURA 3.4: (PÁGINA DE MANEJO DE DATOS)	92
FIGURA 3.5: (PÁGINA DE CREACIÓN DE DATOS)	92
FIGURA 3.6: (PÁGINA DE EDICIÓN DE DATOS)	93
FIGURA 3.7: (VENTANA EMERGENTE DE ELIMINACIÓN DE DATOS)	93
FIGURA 3.8: (PÁGINA DE ADMINISTRACIÓN DE PROYECTOS)	93
FIGURA 3.9: (PÁGINA DE MANEJO DEL EQUIPO DE DESARROLLO)	94
FIGURA 3.10: (PÁGINA DE CONFORMACIÓN DEL EQUIPO DE DESARROLLO)	94
FIGURA 3.11: (PÁGINA DE SELECCIÓN DE PERSONAS DE DESARROLLO)	95
FIGURA 3.12: (PÁGINA DE CREACIÓN O EDICIÓN DE UNA PERSONA DE DESARROLLO)	95
FIGURA 3.13: (PÁGINA PARA MOSTRAR Y EDITAR INFORMACIÓN DEL PROYECTO)	96
FIGURA 3.14: (PÁGINA DE CREACIÓN O EDICIÓN DE META)	96
FIGURA 3.15: (PÁGINA DE MANEJO DE RESPONSABLES POR ACTIVIDAD)	97
FIGURA 3.16: (PÁGINA DE ASIGNACIÓN DE RESPONSABLES A UNA ACTIVIDAD)	97
FIGURA 3.17: (PÁGINA DE SELECCIÓN DE UN REQUERIMIENTO)	98
FIGURA 3.18: (PÁGINA DE ADMINISTRACIÓN DE REQUERIMIENTOS)	98
FIGURA 3.19: (PÁGINA DE ACTUALIZACIÓN DE DATOS ADICIONALES AL REQUERIMIENTO)	99
FIGURA 3.20: (PÁGINA PARA MOSTRAR INFORMACIÓN DE UN REQUERIMIENTO)	100
FIGURA 3.21: (PÁGINA DE MANEJO DE ANEXOS)	100
FIGURA 3.22: (PÁGINA DE CREACIÓN O EDICIÓN DE ANEXO)	101
FIGURA 3.23: (PÁGINA DE REPORTES)	101
FIGURA 3.24: (VENTANA EMERGENTE DE SELECCIÓN DE UN REPORTE – PARTE I)	102
FIGURA 3.25: (VENTANA EMERGENTE DE SELECCIÓN DE UN REPORTE – PARTE II)	102
FIGURA 3.26: (PÁGINA DE PRIORIZACIÓN DE REQUERIMIENTOS)	102
FIGURA 3.27: (MODELO DE CASOS DE USO – PARTE I)	103
FIGURA 3.28: (MODELO DE CASOS DE USO – PARTE II)	104
FIGURA 3.29: (DIAGRAMA DE PAQUETES)	105
FIGURA 3.30: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “INICIAR SESIÓN”)	127
FIGURA 3.31: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “CERRAR SESIÓN”)	127
FIGURA 3.32: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “MANEJAR CUENTA DE USUARIO DEL SISTEMA”)	128
FIGURA 3.33: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “LISTAR DATOS ASOCIADOS A UN PROYECTO”)	129
FIGURA 3.34: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “LISTAR DATOS INICIALES DE PROYECTOS”)	129

FIGURA 3.35: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “LISTAR DATOS ASOCIADOS A UN REQUERIMIENTO”)	130
FIGURA 3.36: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “FILTRAR REGISTROS LISTADOS”)	130
FIGURA 3.37: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “EDITAR REGISTRO SELECCIONADO DE UNA LISTA”)	131
FIGURA 3.38: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “ELIMINAR REGISTRO SELECCIONADO DE UNA LISTA”)	131
FIGURA 3.39: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “MANEJAR DATOS INICIALES DEL PROYECTO”)	132
FIGURA 3.40: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “MANEJAR DATOS ASOCIADOS A UN PROYECTO”)	133
FIGURA 3.41: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “MANEJAR DATOS ASOCIADOS A UN REQUERIMIENTO”)	134
FIGURA 3.42: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “QUITAR ÍTEM DE UNA LISTA DE DATOS”)	135
FIGURA 3.43: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “MANEJAR ANEXO O ARCHIVO DEL PROYECTO”)	135
FIGURA 3.44: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “GENERAR REPORTE”)	136
FIGURA 3.45: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “LISTAR PARÁMETROS GENERALES DEL SISTEMA”)	136
FIGURA 3.46: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “MANEJAR PARÁMETROS GENERALES DEL SISTEMA”)	137
FIGURA 3.47: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “MANEJAR PERSONA DE DESARROLLO”)	138
FIGURA 3.48: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “ACTUALIZAR DATOS ADICIONALES AL PROYECTO”)	139
FIGURA 3.49: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “CONFORMAR EQUIPO DE DESARROLLO”)	140
FIGURA 3.50: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “ASIGNAR PERSONAS DE DESARROLLO AL PROYECTO”)	141
FIGURA 3.51: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “REALIZAR ANÁLISIS DE METAS DEL PROYECTO”)	142
FIGURA 3.52: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “MANEJAR ACTIVIDAD POR PROCESO”)	143
FIGURA 3.53: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “IDENTIFICAR RESPONSABLE POR ACTIVIDAD”)	144
FIGURA 3.54: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “ASIGNAR RESPONSABLES A UNA ACTIVIDAD”)	145
FIGURA 3.55: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “REALIZAR IDENTIFICACIÓN MACRO DE PROCESOS”)	146
FIGURA 3.56: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “MOSTRAR INFORMACIÓN ACTUAL DEL REQUERIMIENTO”)	147
FIGURA 3.57: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “ACTUALIZAR DATOS ADICIONALES AL REQUERIMIENTO”)	147
FIGURA 3.58: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “IDENTIFICAR POSIBLE MEJORA POTENCIAL”)	148
FIGURA 3.59: (DIAGRAMA DE ROBUSTEZ DEL CASO DE USO “PRIORIZAR LOS REQUERIMIENTOS”)	149
FIGURA 3.60: (MODELO DE DOMINIO ACTUALIZADO)	150
FIGURA 3.61: (ARQUITECTURA JAVA EE CON SEAM)	152
FIGURA 3.62: (TRADUCCIÓN DE UNA SOLICITUD JSF EN UN ÁRBOL DE COMPONENTES DE INTERFAZ DE USUARIO)	153
FIGURA 3.63: (FRAMEWORK DE CLASES DE SEAM IMPLEMENTADAS PARA JPA)	154
FIGURA 3.64: (DIAGRAMA DE CLASES DE ENTITYHOME)	155
FIGURA 3.65: (DIAGRAMA DE CLASES DE ENTITYQUERY)	155

FIGURA 3.66: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “INICIAR SESIÓN”)	156
FIGURA 3.67: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “CERRAR SESIÓN”)	156
FIGURA 3.68: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “MANEJAR CUENTA DE USUARIO DEL SISTEMA”)	157
FIGURA 3.69: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “LISTAR DATOS ASOCIADOS A UN PROYECTO”)	158
FIGURA 3.70: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “LISTAR DATOS INICIALES DE PROYECTOS”)	159
FIGURA 3.71: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “LISTAR DATOS ASOCIADOS A UN REQUERIMIENTO”)	160
FIGURA 3.72: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “FILTRAR REGISTROS LISTADOS”)	161
FIGURA 3.73: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “EDITAR REGISTRO SELECCIONADO DE UNA LISTA”)	162
FIGURA 3.74: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “ELIMINAR REGISTRO SELECCIONADO DE UNA LISTA”)	163
FIGURA 3.75: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “MANEJAR DATOS INICIALES DEL PROYECTO”)	164
FIGURA 3.76: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “MANEJAR DATOS ASOCIADOS A UN PROYECTO”)	165
FIGURA 3.77: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “MANEJAR DATOS ASOCIADOS A UN REQUERIMIENTO”)	166
FIGURA 3.78: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “QUITAR ÍTEM DE UNA LISTA DE DATOS”)	167
FIGURA 3.79: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “MANEJAR ANEXO O ARCHIVO DEL PROYECTO”)	168
FIGURA 3.80: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “GENERAR REPORTE”)	169
FIGURA 3.81: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “LISTAR PARÁMETROS GENERALES DEL SISTEMA”)	170
FIGURA 3.82: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “MANEJAR PARÁMETROS GENERALES DEL SISTEMA”)	171
FIGURA 3.83: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “MANEJAR PERSONA DE DESARROLLO”)	172
FIGURA 3.84: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “ACTUALIZAR DATOS ADICIONALES AL PROYECTO”)	173
FIGURA 3.85: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “CONFORMAR EQUIPO DE DESARROLLO”)	174
FIGURA 3.86: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “ASIGNAR PERSONAS DE DESARROLLO AL PROYECTO”)	175
FIGURA 3.87: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “REALIZAR ANÁLISIS DE METAS DEL PROYECTO”)	176
FIGURA 3.88: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “MANEJAR ACTIVIDAD POR PROCESO”)	177
FIGURA 3.89: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “IDENTIFICAR RESPONSABLE POR ACTIVIDAD – PARTE I”)	178
FIGURA 3.90: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “IDENTIFICAR RESPONSABLE POR ACTIVIDAD – PARTE II”)	179
FIGURA 3.91: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “ASIGNAR RESPONSABLES A UNA ACTIVIDAD”)	180
FIGURA 3.92: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “REALIZAR IDENTIFICACIÓN MACRO DE PROCESOS”)	181
FIGURA 3.93: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “MOSTRAR INFORMACIÓN ACTUAL DEL REQUERIMIENTO”)	182
FIGURA 3.94: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “ACTUALIZAR DATOS ADICIONALES AL REQUERIMIENTO”)	183

FIGURA 3.95: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “IDENTIFICAR POSIBLE MEJORA POTENCIAL”)	184
FIGURA 3.96: (DIAGRAMA DE SECUENCIA DEL CASO DE USO “PRIORIZAR LOS REQUERIMIENTOS”)	185
FIGURA 3.97: (DIAGRAMA DE CLASES - ENTIDADES - PARTE I)	186
FIGURA 3.98: (DIAGRAMA DE CLASES - ENTIDADES - PARTE II)	187
FIGURA 3.99: (DIAGRAMA DE CLASES - ENTIDADES - PARTE III)	188
FIGURA 3.100: (DIAGRAMA DE CLASES – FRAMEWORK)	188
FIGURA 3.101: (DEFINICIÓN DE UN CONTROLADOR HOME)	189
FIGURA 3.102: (DEFINICIÓN DE UN CONTROLADOR LIST)	189
FIGURA 3.103: (DIAGRAMA DE CLASES – CONTROLADORES – HOME)	190
FIGURA 3.104: (DIAGRAMA DE CLASES – CONTROLADORES – LIST)	191
FIGURA 3.105: (DIAGRAMA DE CLASES – CONTROLADORES – ENUMERATIONS LIST)	192
FIGURA 3.106: (DIAGRAMA DE CLASES – CONTROLADORES – OTROS)	193
FIGURA 3.107: (MODELO DE DATOS)	194

RESUMEN

En el presente proyecto se desarrolló un software orientado a la web que sirva como herramienta de apoyo en el control del proceso de desarrollo de software, y complemente la “Guía Básica para la Aplicación de la Técnica de Calidad Seis Sigma al Proceso de Desarrollo de Software”. Esta Guía fue desarrollada por estudiantes de la Escuela Politécnica del Ejército como proyecto de graduación, y fue empleada como base para el desarrollo de este proyecto, teniendo en cuenta que la Guía busca mejorar el nivel de eficiencia en el control de calidad del proceso de desarrollo de software, a través de la aplicación de la técnica de calidad Seis Sigma.

Para alcanzar el objetivo del proyecto se trabajó sobre la plataforma **Java EE 5.0**, y se empleó un framework web denominado **JBoss Seam** en su versión 2.2.0, el cual facilitó la integración de los componentes de la capa web (JSF) y la capa de negocio (EJB 3.0/JPA) definidos por la arquitectura Java EE.

Además, todo el ciclo de vida de desarrollo del software fue cubierto por una metodología ágil denominada **ICONIX** que se fundamenta en cuatro fases que son: Análisis de Requerimientos, Análisis y Diseño Preliminar, Diseño Detallado, e Implementación. Cabe mencionar que la herramienta CASE Enterprise Architect facilitó la aplicación de esta metodología.

Tener presente que la mayoría de herramientas y tecnologías son open source, y fueron utilizadas con el fin de fomentar su aplicabilidad en proyectos de desarrollo de software basados en web.

CAPÍTULO I

Introducción

1.1 Tema

Implementación de un software orientado a la web que gestione la aplicación de la técnica de calidad SEIS SIGMA al proceso de desarrollo de software, sobre la plataforma “Java Enterprise Edition 5.0” empleando un framework integrador “JBoss Seam 2.2.0”.

1.2 Antecedentes

Estudiantes de la Escuela Politécnica del Ejército (ESPE), Carrera de Ingeniería en Sistemas e Informática desarrollaron una “Guía para la Aplicación de la Técnica de Calidad Seis Sigma al Proceso de Desarrollo de Software”¹, como proyecto de graduación, esta guía será empleada para el desarrollo de una herramienta que la complemente y sirva como apoyo en el control del proceso de desarrollo de software.

Al contar con una guía basada en los fundamentos de Seis Sigma, depende del equipo de desarrollo saber aplicarla adecuadamente para obtener resultados favorables, pero, ¿Cómo manejar de manera fácil y ágil la cantidad de datos que se generan en cada fase del proceso de desarrollo de software?, ¿Cómo obtener información confiable que permita evaluar y mejorar en cada proyecto?, ¿Cómo mantener disponible información que permita generar documentos entregables?, como respuesta a estas interrogantes surgió la idea de automatizar la guía

¹ Monge Daniel, A., y Mosquera Lourdes, V. (2009). Estudio de la Aplicación de Técnicas de Calidad al Proceso de Desarrollo de Software (Anexo E, código DCC- AIS - G1).

desarrollada por los estudiantes de la ESPE, tomando en cuenta que los requerimientos de los usuarios son cada vez más exigentes debido a los cambios tecnológicos, mientras los desarrolladores de software procuran utilizar varios modelos, técnicas y herramientas para satisfacer dichos requerimientos, y obtener sistemas fáciles de mantener, extender y modificar.

Es importante optimizar el proceso de desarrollo de software, porque los sistemas informáticos están sujetos a diferentes innovaciones en la industria de la tecnología, esto lleva a los desarrolladores a evolucionar para obtener aplicaciones en menor tiempo, más vistosas y de menor costo; entonces si se mantiene un proceso óptimo, que cumpla con estándares, será más fácil adaptarse al cambio y entregar un producto de calidad.

1.3 Planteamiento del Problema

El proceso de desarrollo de software es muy complejo y lleva consigo alto riesgo, existen diferentes fases que demandan organización que serán compartidas por un grupo grande de personas con una clara distribución de responsabilidades entre los diferentes roles que se dan en el proceso de producción. La inadecuada utilización de recursos, adopción de malas prácticas en sus procesos, presupuesto de tiempo y dinero insuficientes, etc., son factores que incrementan el fracaso de un proyecto.

Según “Standish Group” (organización que recoge información sobre los fracasos de proyectos en la Industria de la Tecnología), los últimos resultados que han sido recopilados en su reporte “CHAOS Report 2009” publicado en Abril, determinó que “Los proyectos de software tienen ahora una tasa de éxito del 32% frente al 35% de un estudio anterior en el 2006 y el 16% en 1994. Por otra parte,

el 44% de los proyectos están cuestionados (falencias, con respecto al presupuesto y / o con menos de las características requeridas y funciones) mientras que el 24% de fracasos (cancelado antes de su finalización o entregadas y no se utilizan)².

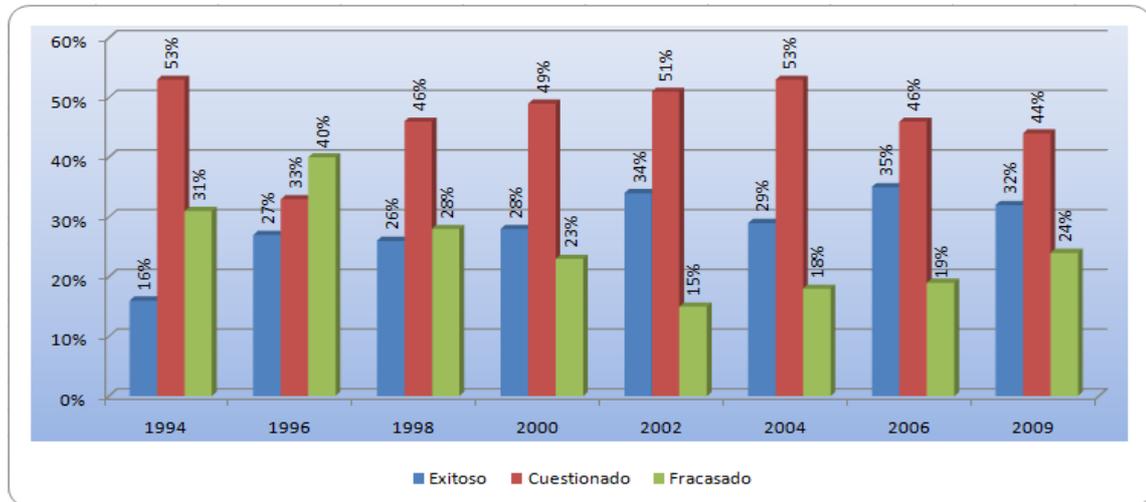


Figura 1.1: (Información de Proyectos en la Industria TI)

Las cifras que se muestran en la Figura 1.1, reflejan que existen problemas en la entrega de productos, principalmente porque no se cubren los requerimientos del cliente, procesos inmaduros, manejo inadecuado de los cambios, fallos en las estimaciones, complejidad de la tecnología; todo esto trae como consecuencia pérdidas económicas, desprestigio de la empresa, clientes insatisfechos, y lo que es peor, no se detecta claramente el motivo del fracaso. Entonces, si no se aplica un proceso de desarrollo claro y bien controlado, no se conseguirá reducir o eliminar los defectos o fallas en la entrega del producto final al cliente.

² Jorge Dominguez, (2009). The Curious Case of the CHAOS Report 2009. Obtenida el 5 de Noviembre del 2009, de <http://www.projectsmart.co.uk/pdf/the-curious-case-of-the-chaos-report-2009.pdf>

1.4 Justificación

Las organizaciones, tanto públicas como privadas, se ven abocadas cada vez más a un proceso competitivo, esta competencia se ve reflejada en estrategias de mejoramiento del servicio hacia el cliente y en reducción de los costos de elaboración de sus productos, esto ha conducido a la rápida adopción de sistemas informáticos capaces optimizar tareas mecánicas y rutinarias; esta realidad impone a los proveedores de software a mejorar sus procesos de desarrollo, avizorando el impacto que provocará la entrega de un producto de mala calidad.

Observando la importancia que adquiere el software en el desarrollo de las organizaciones y el esfuerzo que deben realizar las empresas que proveen software, el presente proyecto se enfocará en apoyar el proceso de desarrollo de software; para el efecto se desarrollará una herramienta que complemente la “Guía Básica para la Aplicación de la Técnica de Calidad Seis Sigma al Proceso de Desarrollo de Software”, guía que ha sido elaborada tomando en cuenta el modelo DMAIC (Definir, Medir, Analizar, Mejorar, Controlar) de Seis Sigma una metodología de referencia para llevar la calidad hasta niveles próximos a la perfección.

Al disponer de una metodología que permita la aplicación de buenas prácticas en los procesos, y una herramienta que permita llevar un control de su aplicabilidad, se conseguirá incrementar notablemente una mejora continua en los procesos de desarrollo de software, y con el transcurso del tiempo se podrá disponer de información muy importante que sirva para analizar posibles falencias

en proyectos anteriores, con el fin de tomar medidas preventivas y/o correctivas que fortalezcan dichos procesos.

1.5 Alcance

Empleando la guía mencionada en los antecedentes del presente proyecto, se implementará un software orientado a la web que gestione la aplicación de la técnica de calidad SEIS SIGMA al proceso de desarrollo de software.

La guía busca incrementar la eficiencia del proceso de desarrollo de software tradicional, y para facilitar su aplicabilidad se ha estructurado en diferentes fases:

- Iniciación del Proyecto.
- Análisis del Sistema.
- Diseño del Sistema.
- Construcción.
- Evaluación y Aseguramiento de Calidad.
- Implementación.

Se automatizará la aplicación de esta guía con el fin de apoyar en el control del proceso de desarrollo de software, pero sólo se tomará en cuenta las dos primeras fases:

- Iniciación del Proyecto.
- Análisis del Sistema.

En cada fase la guía considerara la siguiente estructura:

- Objetivo(s) de la fase.
- Listado de actividades de la fase.
- Contenido de cada actividad:
 - Tarea (s).

- Entregable(s).
- Consideraciones Seis Sigma.
- Formato(s).

Detalle de cada fase, en base a la estructura anterior:

➤ **Iniciación del Proyecto:**

• **Objetivos:**

- Definir con claridad el problema.
- Definir el alcance del proyecto.
- Proponer una solución.
- Determinar los costos y beneficios de la solución.
- Obtener la aprobación del proyecto.

• **Actividades:**

- Identificar el problema.
- Conformar el equipo de desarrollo.
- Identificar requerimientos preliminares.
- Validar los requerimientos.
- Desarrollar un estudio de fiabilidad.
- Obtener la aprobación del proyecto.

➤ **Análisis del Sistema:**

• **Objetivos:**

- Completar la identificación de los requerimientos que comenzó en la fase de Iniciación del Proyecto.
- Transformar los requerimientos de alto nivel en especificaciones detalladas del sistema.

- Crear un diseño conceptual del sistema a través del estudio de fiabilidad detallado.

- **Actividades:**

- Entender los procesos actuales.
- Identificar los requerimientos.
- Priorizar los requerimientos.
- Identificar mejoras potenciales en los procesos.
- Determinar las mejoras que tendrán el mayor impacto sobre los requerimientos prioritarios.
- Crear un mapa de estado detallado.
- Evaluar el impacto y riesgos de las propuestas de mejora de los procesos.
- Completar el desarrollo del diseño conceptual del sistema.
- Completar el documento de especificación de requerimientos.
- Obtener aprobación.

Se ha establecido por cada actividad diferentes tareas, consideraciones SEIS SIGMA y documentos entregables en base a diferentes formatos. Para automatizar estas dos fases se analizará los puntos más relevantes de la estructura de la guía, a fin de facilitar su aplicabilidad.

1.6 Objetivos

1.6.1 Objetivo General

Implementar un software orientado a la web que gestione la aplicación de la técnica de calidad SEIS SIGMA al proceso de desarrollo de software, sobre la plataforma “Java Enterprise Edition 5.0” empleando un framework integrador “JBoss Seam 2.2.0”.

1.6.2 Objetivos Específicos

- Definir los requerimientos que debe cumplir el software, empleando la guía mencionada en los antecedentes del presente proyecto.
- Analizar y Diseñar el software utilizando lenguaje UML.
- Desarrollar el software, sobre la plataforma Java EE 5.0, y empleando el framework integrador JBoss Seam 2.2.0.
- Ejecutar pruebas funcionales y correcciones.
- Implantar el software, instalando las herramientas necesarias para su correcto despliegue, a fin de probar el cumplimiento de todos sus requerimientos.

CAPÍTULO II

Marco Teórico

2.1 Guía básica de aplicación de la técnica Seis Sigma al proceso de desarrollo de software³

Con el fin de mejorar el nivel de eficiencia en el control de calidad del proceso de desarrollo de software, estudiantes de la ESPE realizaron una guía a través de la aplicación de la técnica de calidad Seis Sigma; esta guía será empleada como base para el desarrollo del presente proyecto.

Para sustentar el desarrollo de la guía se buscó información de los fundamentos de Seis Sigma y su relación con CMMI, se levantó información sobre las técnicas de calidad y metodologías de desarrollo más utilizadas en la industria del software en Ecuador, y se determinó el grado de conocimiento y aplicabilidad de la técnica Seis Sigma en el proceso de desarrollo de software a nivel nacional.

Siendo un factor limitante para su desarrollo un análisis estadístico, en el cual se obtuvo un bajo porcentaje de organizaciones (2%) que están familiarizadas con la técnica de calidad Seis Sigma, se observó un gran porcentaje de contradicciones, falencias y desconocimiento que poseen las empresas desarrolladoras sobre los principios de aseguramiento de calidad que se pueden aplicar en un proyecto de software. Además, se demostró que las organizaciones desarrolladoras presentan indicios de trabajo con herramientas de calidad (checklist), pero no aplican técnicas de calidad, se demostró también que la estructura del equipo de desarrollo que actualmente se maneja en las

³ Monge Daniel, A., y Mosquera Lourdes, V. (2009). Estudio de la Aplicación de Técnicas de Calidad al Proceso de Desarrollo de Software.

organizaciones desarrolladoras, no contemplan la presencia de personal especializado en áreas de auditoría, gestión de riesgos y mucho menos de aseguramiento de calidad.

La guía busca fomentar el manejo de documentos específicos utilizados en cada una de las seis fases que conforman el proceso de desarrollo de software, además, en cada fase se plantean diferentes objetivos y actividades que debe cumplir el equipo de desarrollo. Cada actividad está conformada por tareas, entregables, consideraciones Seis Sigma y formato sugerido.

Para facilitar la aplicabilidad de la guía mencionada, el objetivo del presente proyecto es automatizarla, a fin de implementar una herramienta que permita a los desarrolladores de software, manejar de una manera fácil e intuitiva la información que se generará en las diferentes fases del proceso de desarrollo de software.

En esta herramienta principalmente se automatizará las dos primeras fases: “Iniciación del Proyecto” y “Análisis del Sistema”, de las seis fases que contempla el modelo de la guía.

En la fase Iniciación del Proyecto, se gestionará datos muy importantes que permitan definir con claridad un problema, definir el alcance del proyecto, proponer una solución, determinar costos y beneficios de la solución, y obtener la aprobación del proyecto. En la fase Análisis del Sistema, de igual manera se gestionará datos que permitan la identificación de requisitos, transformar los requerimientos de alto nivel en especificaciones detalladas, y crear un diseño conceptual del sistema a través del estudio de fiabilidad detallado.

Para facilitar el análisis de los datos que se generarán en estas dos fases, se proporcionará opciones para la generación de reportes en base a formatos

establecidos en la guía y formatos propuestos, previo al análisis de los datos más relevantes.

Es importante recalcar que el objetivo del presente proyecto es desarrollar una herramienta que sirva como apoyo en el control del proceso de desarrollo de software, pero se debe tener en cuenta que para alcanzar la calidad en el producto software, no es suficiente con solo disponer de una herramienta que busca incrementar la eficiencia del proceso de desarrollo de software tradicional, sino, que los miembros que forman parte de un proyecto de desarrollo de software, asuman con responsabilidad sus roles identificados en cada fase de dicho proceso.

2.2 Metodología ICONIX

ICONIX es una metodología simplificada en comparación con otras metodologías más tradicionales, que unifica un conjunto de métodos de orientación a objetos con el objetivo de abarcar todo el ciclo de vida de un proyecto. Fue elaborado por Doug Rosenberg y Kendall Scott a partir de una síntesis del proceso unificado de los “Three Amigos”⁴ Booch, Rumbaugh y Jacobson y que ha dado soporte y conocimiento a la metodología ICONIX desde 1993. Además ICONIX está adaptado a los patrones de diseño y ofrece el soporte de UML, dirigido por casos de uso (como RUP) y es un proceso iterativo e incremental. Es un proceso pequeño y ligero, como XP, pero no descarta las fases del análisis y del diseño como lo hace XP.

ICONIX se divide en dos flujos de trabajo: Dinámico y Estático, los cuales son altamente interactivos (ver la Figura 2.1).

⁴ http://en.wikipedia.org/wiki/Unified_Modeling_Language

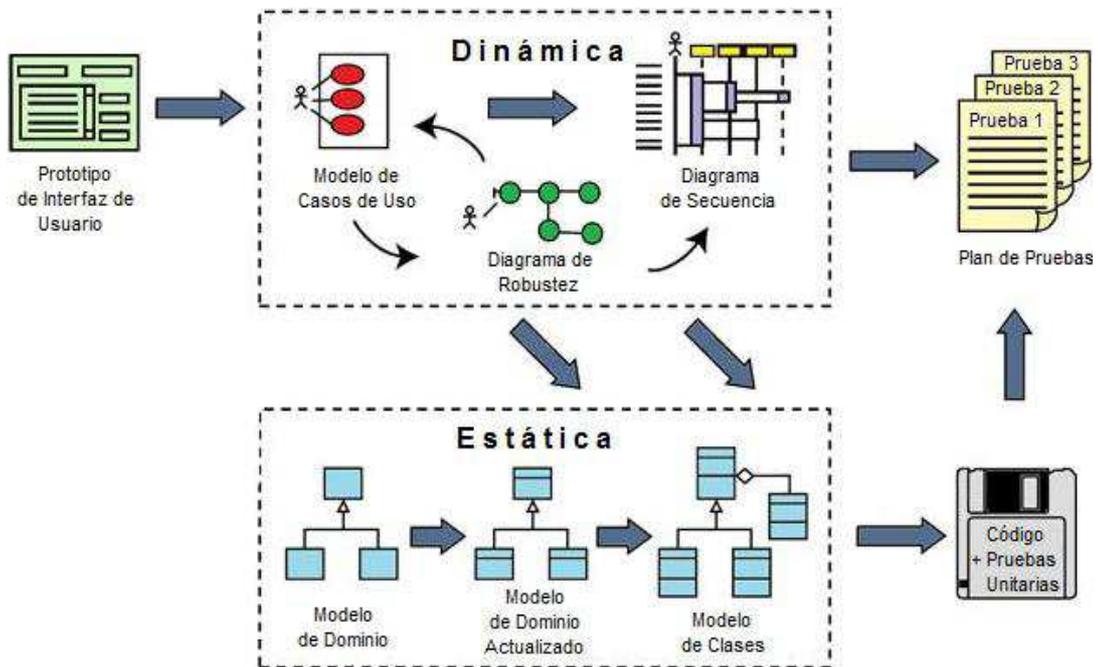


Figura 2.1: (Flujos de trabajo de ICONIX)
Traducción propia, del libro “Use Case Driven Object Modeling with UML”

Las tres características fundamentales de ICONIX son:

- **Iterativo e incremental.**- Varias iteraciones ocurren entre el desarrollo del modelo de dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos.
- **Trazabilidad.**- Cada paso está referenciado por algún requisito. Se define trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos producidos.
- **Dinámica del UML.**- La metodología ofrece un uso “dinámico del UML” como los diagramas de casos de uso, diagramas de secuencia, y diagramas de robustez.

Las principales fases de ICONIX son:

- Análisis de Requerimientos.
- Análisis y Diseño Preliminar.
- Diseño Detallado.
- Implementación.

2.2.1 Análisis de Requerimientos

El trabajo es iniciado con un relevamiento informal de todos los requisitos que en principio deberían ser parte del sistema.

Requerimientos Funcionales

Para empezar un proyecto adecuadamente, alguien (posiblemente un equipo de analistas del negocio) hablará con el cliente, usuarios finales, y stakeholders, a fin de obtener un documento con un listado de requerimientos funcionales. La definición de los requerimientos cae ligeramente fuera del alcance de ICONIX, pero se recomienda elaborarla.

Los requisitos funcionales de alto nivel, por lo general (aunque no siempre) empiezan con la frase “El sistema permitirá hacer..., o El sistema no permitirá hacer...”, y son una fuente importante para crear el **modelo de dominio**.

Modelado del Dominio

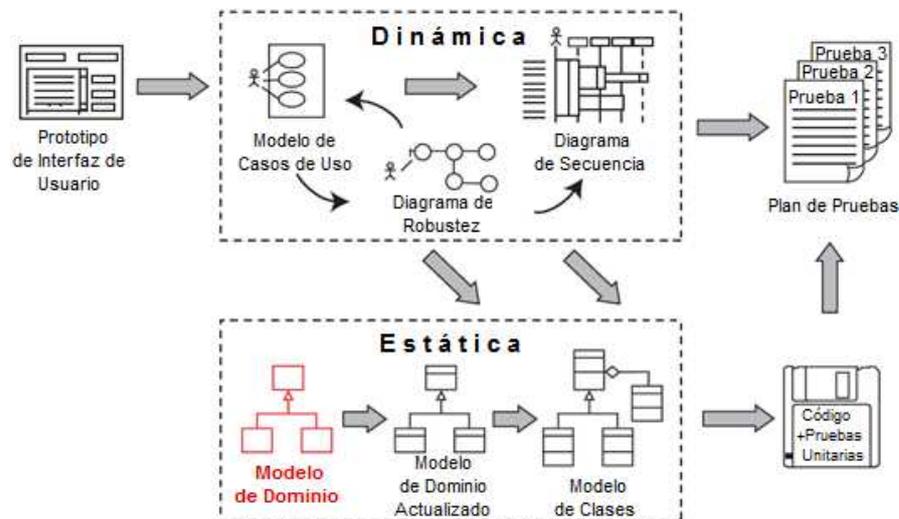


Figura 2.2: (Modelo de dominio, en contexto del proceso ICONIX)
Traducción propia, del libro “Use Case Driven Object Modeling with UML”

El modelado del dominio es la tarea de construir un glosario del proyecto, o un diccionario de términos (sustantivos, nombres simples o expresiones nominales) utilizados en un proyecto, y mostrar gráficamente cómo se relacionan entre sí todos esos términos, pero en la práctica se trata de una simplificación de diagrama de clases.

El modelo de dominio define el alcance de un proyecto y forma la base sobre la cual construir los casos de uso. Un modelo de dominio, también proporciona un vocabulario común para una comunicación clara entre los miembros de un equipo del proyecto.

El modelo de dominio constituye la base de la parte estática, mientras que los casos de uso son la base de la parte dinámica. La parte estática describe la estructura, y la parte dinámica describe el comportamiento.

Directrices para crear el modelo de dominio

- a. Iniciar con el modelo de dominio antes de escribir los casos de uso.
- b. En la redacción del listado de requisitos, se debe identificar (subrayar o resaltar) “objetos” que representan cosas y conceptos del mundo real. No confundir un objeto con una tabla de base de datos.
- c. Generar un listado ordenado alfabéticamente con todos los términos (nombres de objetos de dominio) identificados en los requerimientos, cambiando todos los términos en plural a singular.
- d. Realizar un análisis de todos los términos, a fin de eliminar términos ambiguos o duplicados.
- e. Con la lista de objetos de dominio o clases de dominio depurada, realizar un diagrama de clases simplificado, es decir, sin mostrar atributos, ni métodos.

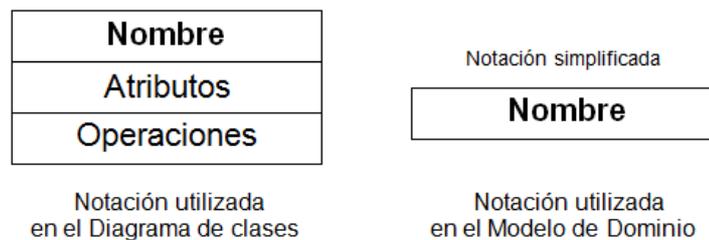


Figura 2.3: (Notación para representar una Clase)

A medida que se avance en las fases de ICONIX, se identificará los “atributos” y “operaciones”.

- f. Usar las relaciones de agregación y generalización para mostrar cómo se asocian los objetos (clases de dominio) entre sí. ICONIX resalta que casi

un 95% de las asociaciones se representan modelándolas como agregación o generalización.

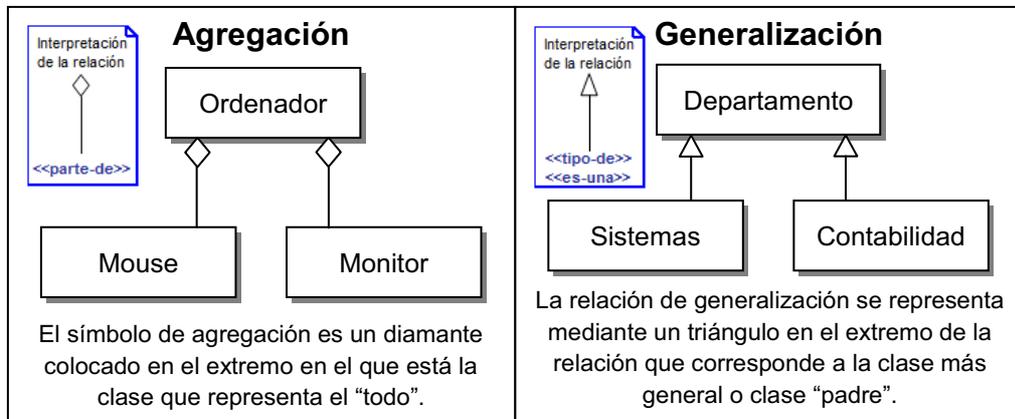


Figura 2.4: (Relaciones para asociar los objetos de dominio)

g. Para mostrar cómo se asocian las clases de dominio también se puede utilizar los otros tipos de relaciones (dependencia, asociación, composición).

A continuación se presenta un ejemplo de un modelo de dominio.

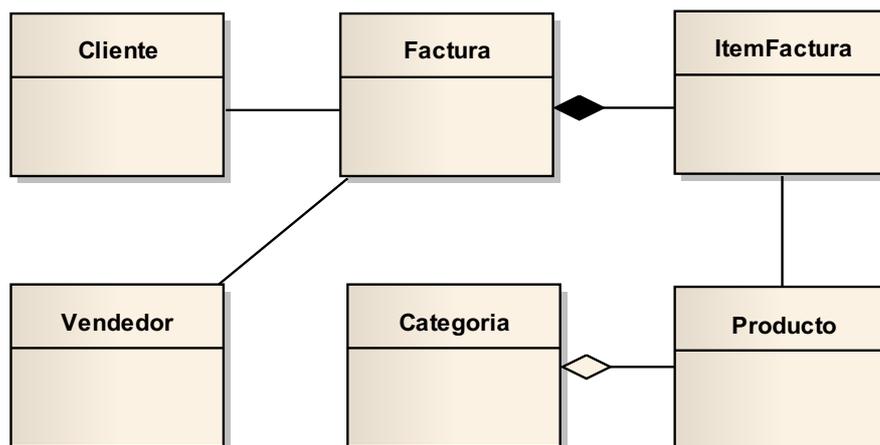


Figura 2.5: (Ejemplo de un Modelo de Dominio)

Modelado de Casos de Uso

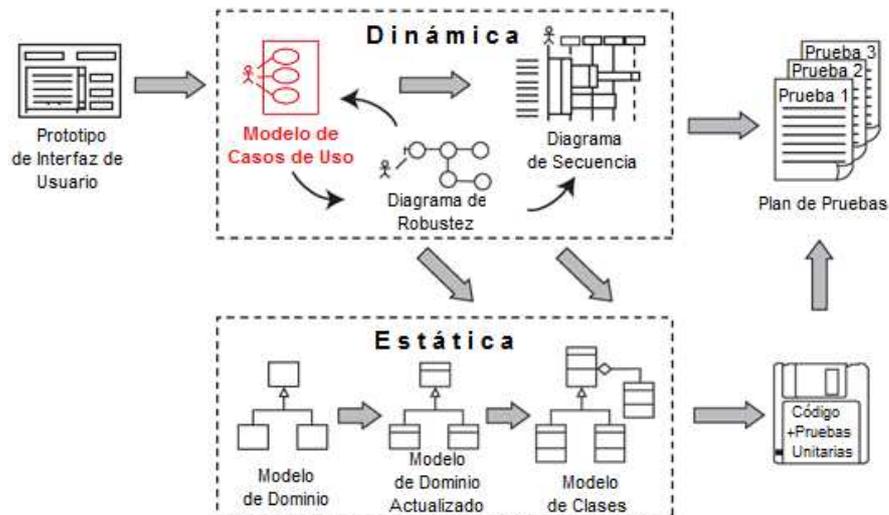


Figura 2.6: (Modelo de casos de uso, en contexto del proceso ICONIX)
Traducción propia, del libro “Use Case Driven Object Modeling with UML”

Con un modelo de dominio inicial se procede a escribir los **casos de uso** del sistema mostrando los **actores** involucrados. La propuesta ICONIX supone que el modelo de dominio inicial tiene errores, por tal motivo ICONIX propone una mejora incremental de este modelo a medida que se analice los casos de uso. También se actualiza el modelo de dominio cuando se dibuja los “diagramas de robustez” (que se verá en la sección, 2.2.2 Análisis y Diseño Preliminar) y secuencia.

Actores.- Un actor es una entidad externa (personas, otros sistemas, sensores, etc.) al sistema que realiza algún tipo de interacción con el mismo. Se representan mediante una figura humana (ver Figura 2.7).



Figura 2.7: (Representación gráfica de un actor)

Casos de Uso.- Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Se representa mediante una elipse con el nombre del caso de uso en su interior (ver Figura 2.8).

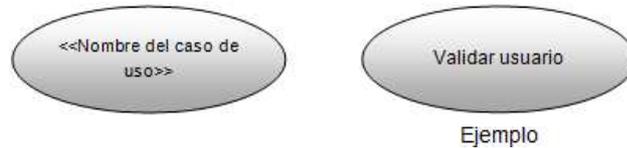


Figura 2.8: (Representación gráfica de un caso de uso)

Los casos de uso describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista de un usuario. Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.

Un caso de uso es una manera específica de utilizar un sistema. Es la imagen de una funcionalidad del sistema en respuesta a la estimulación de un actor externo. El conjunto de funcionalidades de un sistema se determina examinando las necesidades funcionales de cada actor. La asociación de un actor con un caso de uso significa responsabilidades.

Directrices para escribir los casos de uso

- a. El trabajo empieza identificando los casos de uso, a nivel de nombre (verbo o frase verbal corta).
- b. Para facilitar la identificación de los casos de uso se recomienda **realizar prototipos de interfaz gráfica de usuario**, maquetas de pantallas, estas son ayudas visuales muy útiles cuando se está escribiendo los casos de uso. Por ejemplo, si un caso de uso se basa en una interfaz gráfica de

usuario, es importante incluir todos los botones y menús que el usuario puede tocar para generar eventos dentro de un caso de uso. Estos prototipos deben ser sencillos pero claros (ver Figura 2.9). Se recomienda asignar nombres a estas interfaces, y utilizar esos nombres en las descripciones de los casos de uso.

Módulo de Administración >> Crear Usuarios

Usuario:

Clave:

Repetir Clave:

Dirección de correo electrónico:

Figura 2.9: (Ejemplo de un prototipo GUI)

- c. **Dibujar diagramas de casos de uso**, un diagrama de casos de uso muestra un conjunto de casos de uso, actores y sus relaciones. Son importantes en la captura de los requisitos funcionales del sistema.
- d. Para asociar los casos de uso existen relaciones como: generalización, <<includes>>, <<extends>>, <<precedes>>, <<invokes>>, pero ICONIX recomienda que las más utilizadas sean <<invokes>> y <<precedes>>, las otras relaciones en su mayoría no son necesarias.

Relación	Representación	Descripción	Ejemplo
Generalización		El caso de uso A es un tipo del caso de uso B.	
Inclusión o uso		El caso de uso B es llamado a través del caso de uso A. Cuando B finaliza, A continúa ejecutándose donde se detuvo. A depende a menudo del resultado de B.	

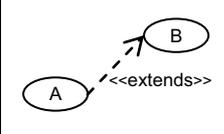
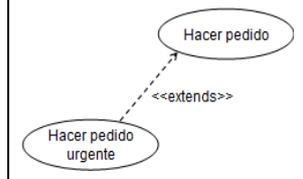
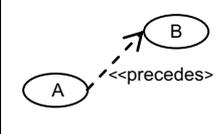
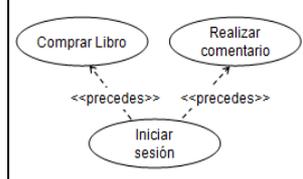
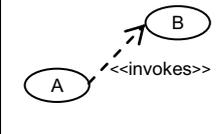
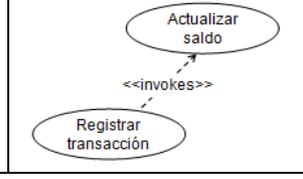
Extensión		Todos los pasos del caso de uso A son efectuados durante la ejecución del caso de uso B, en el punto de extensión especificado en B.	
Precede		El caso de uso A debe llevarse a cabo en su totalidad, antes que empiece el caso de uso B. Esto permitirá saber qué caso de uso debe programarse antes que otro.	
Invoca		El caso de uso B ocurre durante la vida útil del caso de uso A.	

Tabla 2.1: (Relaciones para asociar los casos de uso)

- e. En el modelado de casos de uso no se debe perder mucho tiempo en identificar las relaciones entre los casos de uso, esto se debe porque la parte más importante del trabajo es escribir el texto de los casos de uso.

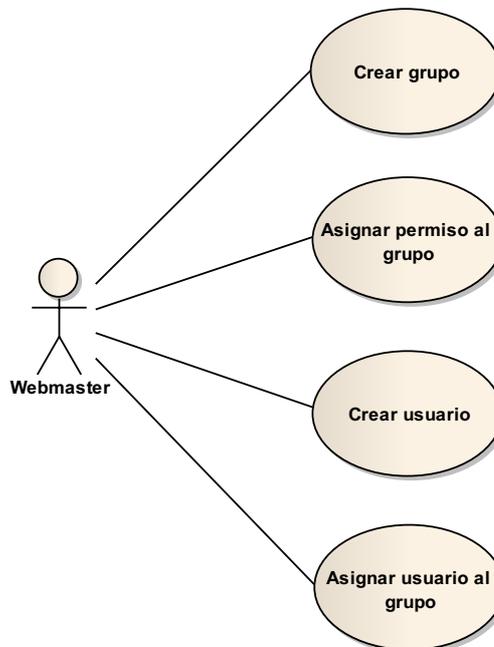


Figura 2.10: (Ejemplo de un diagrama de casos de uso)

- f. Antes de escribir los casos de uso, se debe organizar todos estos dentro de paquetes, es decir, utilizar los diagramas de paquetes, que son una forma de agrupar elementos fuertemente relacionados (por ejemplo, clases, diagramas, o casos de uso). Una buena práctica para organizar los casos de usos en paquetes, es identificando áreas funcionalmente relacionadas (subsistemas).



Figura 2.11: (Ejemplo de un diagrama de paquetes)

- g. **Escribir los casos de uso**, tomando en cuenta las siguientes recomendaciones:

- Escribir los casos de uso en contexto del modelo de dominio, esto es, asociar los requerimientos funcionales con los casos de uso y con los objetos de dominio (trazabilidad). Esto significa que se debe nombrar pantallas y objetos de dominio en el texto de los casos de uso. De lo contrario, los requisitos de comportamiento estarán completamente desconectados del modelo de objetos.
- Seguir la regla “dos párrafos”, es decir, cada caso de uso debe estar conformado máximo por dos párrafos, incluyendo los cursos básico y alternativo. Si algún caso de uso es demasiado largo resultará incomprensible en el diagrama de secuencia, entonces, si es necesario se debe dividir en dos o más casos de uso separados.

- Escribir los casos de uso en tiempo presente y voz activa, es probable que en la redacción de los requerimientos funcionales se encuentre texto en voz pasiva, entonces, se debe cambiar a voz activa.
- Redactar los casos de uso usando la siguiente estructura: sustantivo-verbo-sustantivo, esto facilitará el diseño de los diagramas de secuencia, ya que estos están orientados alrededor de los sustantivos y verbos:
 - **Los sustantivos.-** Son las instancias de objetos, estos por lo general vienen del modelo de dominio (entidades) o son objetos de la interfaz gráfica de usuario.
 - **Los verbos.-** Son los mensajes entre los objetos, estos representan las funciones del software (controladores) que deben ser construidos.
- Escribir los casos de uso usando flujos de evento/respuesta; cuando se escribe los escenarios de los casos de uso, se está describiendo la interacción del usuario con el sistema. Sin embargo, la iteración es en doble sentido, se debe describir el comportamiento del sistema, además del comportamiento del usuario. En general un caso de uso, constará de varias etapas. Cada paso involucra un evento y una respuesta: la acción del usuario y la reacción del sistema, o viceversa.

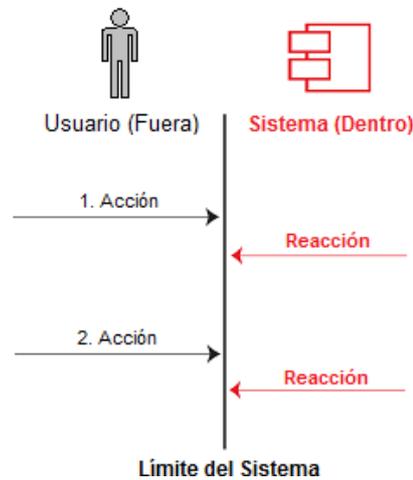


Figura 2.12: (Escenario de un caso de uso)
Traducción propia, del libro “Use Case Driven Object Modeling with UML”

- Si se dispone de manuales de usuario, utilizar estos para extraer los casos de uso.

Para presentar la especificación de cada caso de uso se recomienda utilizar la siguiente plantilla:

Especificación del caso de uso “<<nombre del caso uso>>”

- a. Caso de Uso:** [Clave y nombre del caso de uso.]
- b. Precondiciones:** [Una precondición (de un caso de uso) es el estado del sistema que debe existir antes de que inicie la ejecución del caso de uso, generalmente son aspectos que no van a ser validados durante el caso de uso, sino que se dan por cierto.]
 - b.1. Precondición 1...**
 - b.2. Precondición 2...**
- c. Flujo de Eventos:**

c.1. Flujo Básico: [Debe ser redactado en forma de diálogo entre el actor y el sistema o viceversa.]

c.2. Flujos Alternativos: [Cada flujo alterno representa un comportamiento diferente (muchas veces, debido a las excepciones que pueden ocurrir en el flujo principal).]

c.2.1. Flujo alternativo 1...

c.2.2. Flujo alternativo 2...

d. Pos condiciones: [Una postcondición (de un caso de uso) es una lista de posibles estados del sistema que pueden existir después de que termine la ejecución del caso de uso.]

d.1. Pos condición 1...

d.2. Pos condición 2...

Antes de empezar con la siguiente fase Análisis y Diseño Preliminar, se debe garantizar que el modelo de dominio y los casos de uso trabajen juntos, a fin de abordar las necesidades del cliente. Las personas involucradas en este punto son: el cliente, los usuarios finales, gerente de marketing, básicamente todos los interesados en el proyecto.

2.2.2 Análisis y Diseño Preliminar

Para ir de los casos de uso al diseño detallado (y luego al código), es necesario vincular los casos de uso a los objetos. Una técnica utilizada como puente para ir del análisis al diseño es el **Análisis de Robustez**.

Análisis de Robustez

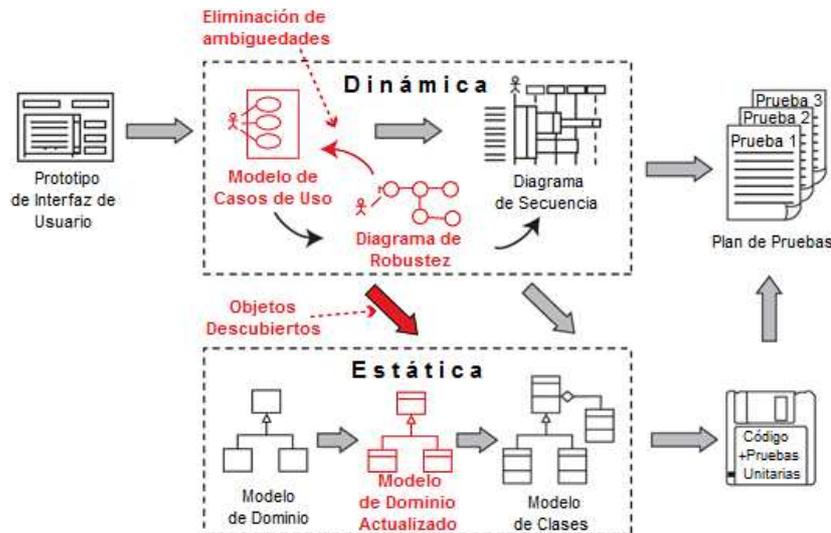


Figura 2.13: (Análisis de Robustez, en contexto del proceso ICONIX)
Traducción propia, del libro “Use Case Driven Object Modeling with UML”

El análisis de robustez facilita el reconocimiento de objetos, sirve para identificar más y mejores clases antes del desarrollo de los diagramas de secuencia. Esta técnica sirve como interfaz entre el “que” y el “como” de un análisis. Además el análisis de robustez provee una gran ayuda para saber si las especificaciones del sistema son razonables.

Realizar diagramas de robustez.- Se debe ilustrar gráficamente las interacciones entre los objetos participantes de un caso de uso. Estos diagramas permiten analizar el texto narrativo de cada caso de uso e identificar un conjunto inicial de objetos participantes de cada caso de uso.

Los símbolos utilizados en el diagrama de robustez son:

- **Objetos interfaz (Pantallas).-** Usados por los actores para interactuar con el sistema (es decir, la capa de presentación). Ejemplo: ventanas, diálogos y menús.

- **Objetos entidad (Almacenamientos).**- Son objetos del modelo de dominio.
- **Objetos controladores (Gestores).**- Es la unión entre la interfaz y los objetos entidad. Sirven como conexión entre los usuarios y los datos.

Es útil pensar en los objetos interfaz y los objetos entidad como sustantivos, y los objetos controladores como si fueran verbos.

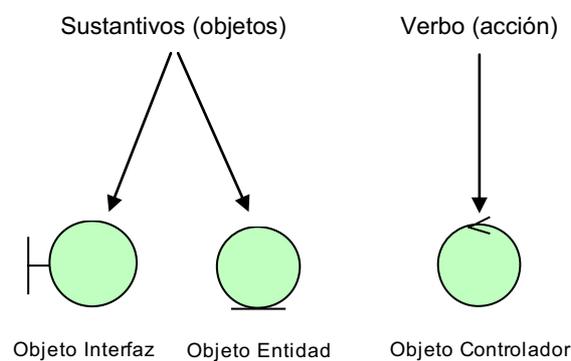


Figura 2.14: (Símbolos utilizados en el Diagrama de Robustez)
Traducción propia, del libro “Use Case Driven Object Modeling with UML”

Reglas básicas para realizar los diagramas de robustez

- Actores solo pueden comunicarse con objetos interfaz.
- Las interfaces solo pueden comunicarse con controles y actores.
- Los objetos entidad solo pueden comunicarse con controles.
- Los controles se comunican con interfaces, objetos entidad y con otros controles pero nunca con actores.

Tomando en cuenta que los objetos **entidad** y la **interfaces** son **sustantivos** y los **controles** son **verbos**. Se puede enunciar de manera sencilla que los

sustantivos nunca se comunican con otros sustantivos, pero los verbos, si se comunican con otros verbos y a su vez con sustantivos.

Estas reglas ayudan a cumplir el formato sustantivo-verbo-sustantivo sugerido para escribir el texto de los casos de uso.

Directrices para elaborar el análisis de robustez

- a. Se debe graficar un diagrama de robustez por cada caso de uso, el diagrama debe contemplar el flujo normal y todos los flujos alternativos.
- b. El texto de las etiquetas empleadas en los diagramas de robustez, debe ser tomada del texto de los casos de uso. Debido a que un diagrama de robustez es esencialmente una representación gráfica de un caso de uso.
- c. Para nombrar los objetos entidad, emplear el nombre de los objetos de dominio (ver el “Modelo de dominio”).
- d. Mientras se dibuja los diagramas de robustez, se debe **“actualizar el modelo de dominio”** de forma incremental, es decir, agregar las nuevas clases de dominio y atributos identificados.
- e. Es importante **“rescribir los casos de uso”** en paralelo con la elaboración de los diagramas de robustez. Esto se debe, a que al dibujar un diagrama de robustez, se obliga a trabajar a través del texto del caso de uso, una frase a la vez.
- f. Utilizar un objeto interfaz por cada pantalla, esto ayuda a identificar los nombres de pantallas.

g. Tener en cuenta que:

- Los controladores son funciones lógicas del software.
- Las interfaces y clases de entidad en un diagrama de robustez, se convertirán en instancias de objetos en un diagrama de secuencia, mientras que los controladores se convertirán en los mensajes.
- Un diagrama de robustez es un “objeto imagen” de un caso de uso.
- Los “atributos” se pueden descubrir en el texto de los casos de uso, o de los prototipos de interfaz de usuario, o incluso de los requerimientos funcionales.

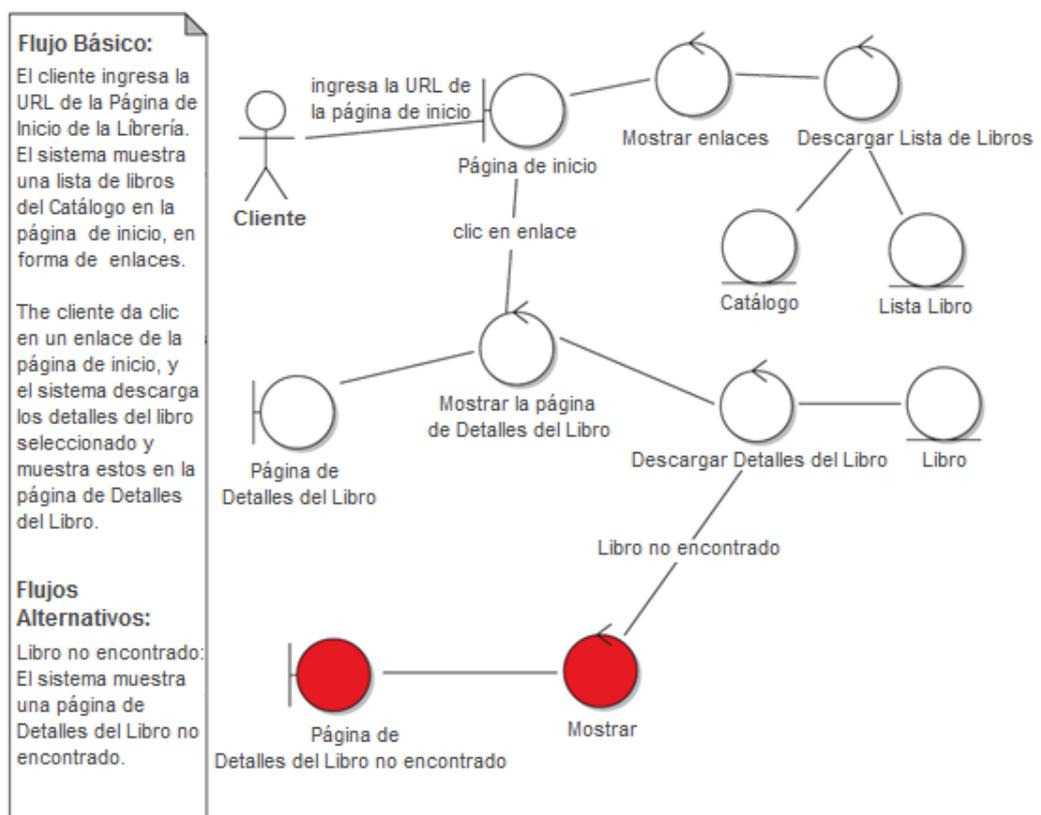


Figura 2.15: (Ejemplo de un diagrama de robustez, para el caso de uso Mostrar lista de Libros)

Traducción propia del libro “Use Case Driven Object Modeling with UML”

Antes de empezar con el Diseño Detallado, se debe asegurar que los diagramas de robustez, el Modelo de Dominio y el texto de los casos de uso coincidan entre sí.

Se recomienda que no se asigne las operaciones a las clases de dominio, ya que esta es una actividad del Diseño Detallado.

2.2.3 Diseño Detallado

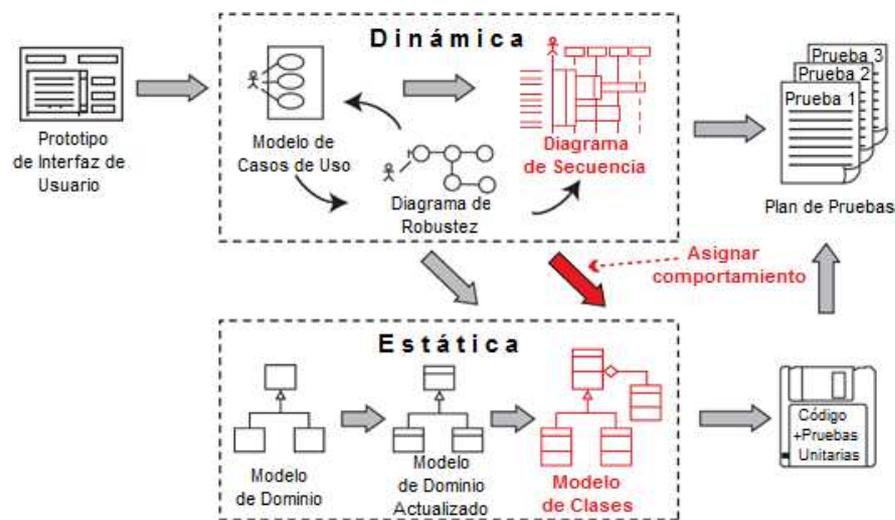


Figura 2.16: (Diseño detallado, en contexto del proceso ICONIX)
Traducción propia, del libro “Use Case Driven Object Modeling with UML”

En esta fase se creará un diseño detallado de los casos de uso, por tal motivo, el texto de los casos de uso debe estar completo, correcto y detallado. Además se debería disponer de casi todas las clases de dominio.

El diseño detallado es conocido como la asignación de comportamiento, es decir, la asignación de las funciones del software. Para conducir el diseño detallado se puede utilizar los **diagramas de secuencia**. Tener en cuenta que hay una relación directa entre cada caso de uso, su diagrama de robustez, y los diagramas de secuencia.

Diagrama de secuencia

El diagrama de secuencia es el núcleo del modelo dinámico y muestra todos los cursos alternos que pueden tomar todos los casos de uso, además, muestra interacciones entre objetos según un punto de vista temporal.

A pesar de que a partir de los diagramas de casos de uso y de los diagramas de robustez ya se dispone entre un 75 y 80 por ciento de atributos de las clases identificadas, es hasta los diagramas de secuencia donde se empiezan a ver qué métodos llevarán las clases del sistema.

Notación de un diagrama de secuencia

En el siguiente ejemplo se muestra como está compuesto un diagrama de secuencia.

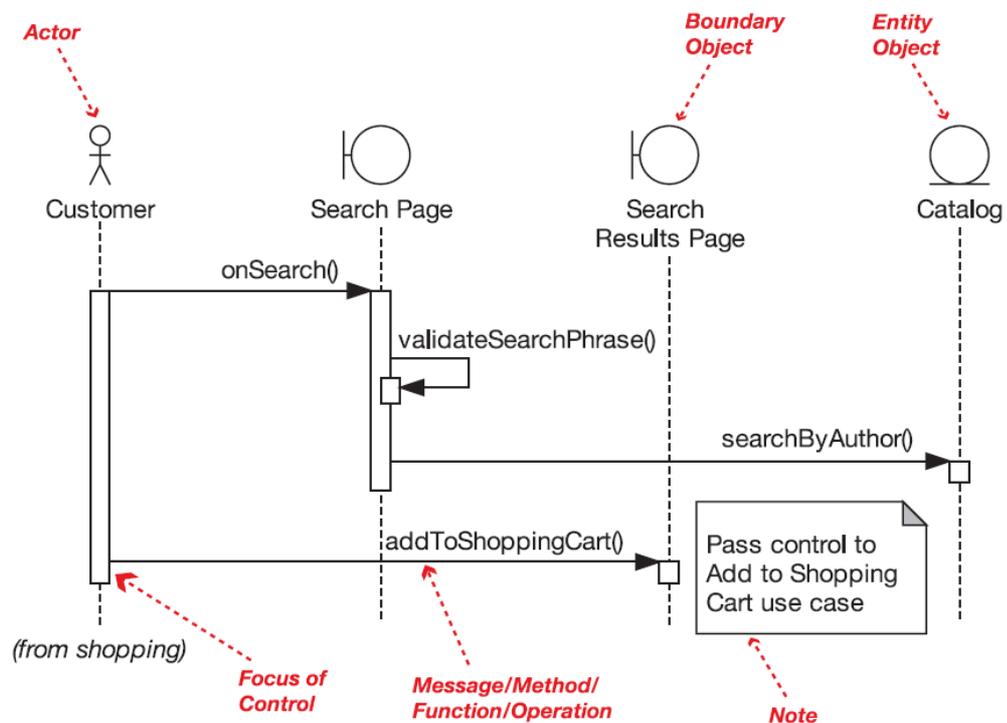


Figura 2.17: (Notación de un diagrama de secuencia)
Gráfico del libro "Use Case Driven Object Modeling with UML"

Directrices para dibujar los diagramas de secuencia

- a. Dibujar un diagrama de secuencia por cada caso de uso.
- b. Se recomienda pegar el texto (flujo básico y alternativo) del caso de uso en una nota en el margen izquierdo del diagrama de secuencia.
- c. En la parte superior del diagrama de secuencia copiar: actores, objetos de tipo interfaz y entidad, identificados en el diagrama de robustez. Los objetos controladores (verbos) del diagrama de robustez se convierten en mensajes.
- d. Mostrar cómo el comportamiento del caso de uso es llevada a cabo por los objetos controladores del diagrama de robustez, es decir, mostrar cómo los objetos interactúan mediante el envío de mensajes unos a otros. A menudo un controlador se convierte en dos o más operaciones.
- e. **Asignar las operaciones a las clases** mientras se dibuja los mensajes. Es posible detectar nuevos atributos, estos deben incluirse en el diagrama de clases.
- f. Tener en cuenta que, mensajes, métodos, funciones, operaciones, verbos y controladores, todos estos son básicamente diferentes versiones de lo mismo: el comportamiento que se puede asignar a una clase (a través de diagramas de secuencia) y, finalmente implementar y probar.

Antes de empezar a codificar se debe finalizar el modelo estático, adicionando los detalles del diseño (tipos de datos, multiplicidad, etc.) en el **diagrama de**

clases. Por último verificar si el diseño satisface todos los requerimientos identificados.

2.2.4 Implementación

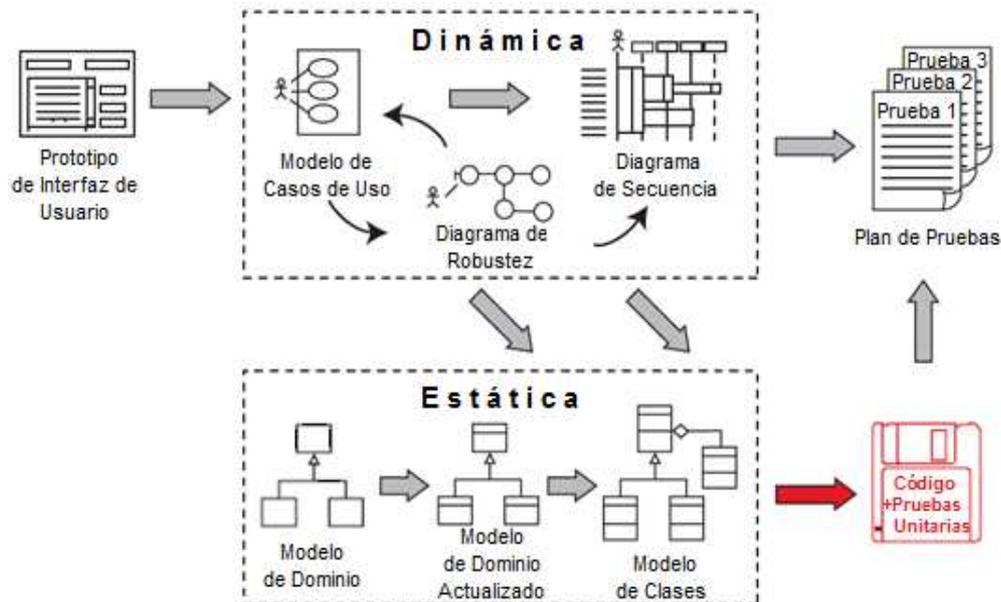


Figura 2.18:(Implementación, en contexto del proceso ICONIX)
Traducción propia, del libro “Use Case Driven Object Modeling with UML”

Después de realizar un buen análisis y diseño, se procede a **generar el código**, tomando en cuenta algunos factores como:

- **La reusabilidad.-** Que es la posibilidad de hacer uso de los componentes en diferentes aplicaciones.
- **La extensibilidad.-** Que consiste en modificar con facilidad el software.
- **La confiabilidad.-** Realización de sistemas descartando las posibilidades de error.

Realizar pruebas. Test de unidades, de casos, datos y resultados. Test de integración con los usuarios para verificar la aceptación de los resultados.

Si al codificar se detecta algún error en el diseño, se debe corregir inmediatamente.

2.2.5 Resumen ICONIX

Con el fin de facilitar la aplicación de la Metodología ICONIX, se presenta un resumen de las fases de ICONIX.

METODOLOGÍA ICONIX			
Fase	Actividad	Técnica / Notación	Entregable/Artefacto
Análisis de Requerimientos	Definir requerimientos funcionales	Entrevistas, encuestas, con las personas interesadas en el Proyecto.	<ul style="list-style-type: none"> • Documento: Listado de Requerimientos.
	Modelar el dominio	<ul style="list-style-type: none"> • Modelo de Dominio. • Lenguaje UML. 	<ul style="list-style-type: none"> • Modelo de Domino Inicial.
	Modelar los casos de uso	<ul style="list-style-type: none"> • Prototipado. • Diagrama de casos de uso. • Diagrama de paquetes. • Lenguaje UML. 	<ul style="list-style-type: none"> • Prototipo GUI. • Diagramas de Casos de Uso. • Diagrama de Paquetes. • Especificación de casos de uso (Borrador).
Análisis y Diseño Preliminar	Realizar análisis de robustez	<ul style="list-style-type: none"> • Diagrama de robustez. • Lenguaje UML. 	<ul style="list-style-type: none"> • Diagrama de Robustez por cada caso de uso. • Especificación de casos de uso. • Modelo de dominio actualizado.
Diseño Detallado	Asignar comportamiento	<ul style="list-style-type: none"> • Diagrama de secuencia. • Lenguaje UML. 	<ul style="list-style-type: none"> • Diagrama de secuencia por cada caso de uso. • Diagrama de Clases. • Modelo de Datos.
Implementación	Generar código	Utilizar seam-gen 2.2.0.GA.	<ul style="list-style-type: none"> • Código fuente.
	Realizar pruebas funcionales	Interacción con usuario final.	<ul style="list-style-type: none"> • Software funcionando correctamente.

Tabla 2.2: (Resumen de ICONIX)

En la fase de Implementación se utilizará como técnica para generar el código inicial (estructura del proyecto) la herramienta **seam-gen** (que se verá en la sección, 2.5 Herramientas de Desarrollo).

2.3 Plataforma Java EE 5.0

2.3.1 Introducción

La **Plataforma Java** (originaria de la ex empresa “Sun Microsystems”) es un entorno, orientado a ejecutar aplicaciones desarrolladas en lenguaje de programación java. Con el tiempo ha evolucionado y se han creado tres ediciones (plataformas) distintas con funciones específicas, a fin de abarcar el desarrollo de diferentes tipos de aplicaciones.

A continuación se nombran las tres plataformas definidas por Sun:

- **Plataforma Java, Edición Micro (Java Platform, Micro Edition), o Java ME (antes J2ME).**- Aplicaciones para dispositivos móviles, orientadas a entornos de recursos limitados, como teléfonos móviles, PDAs (Personal Digital Assistant), etc.
- **Plataforma Java, Edición Estándar (Java Platform, Standard Edition), o Java SE (antes J2SE).**- Aplicaciones de escritorio y applets.
- **Plataforma Java, Edición Empresa (Java Platform, Enterprise Edition), o Java EE (antes J2EE).**- Aplicaciones distribuidas multicapa sobre Web.

En esta sección se describirá con más detalle la plataforma Java EE, ya que el objetivo del presente proyecto es implementar un software sobre esta plataforma, empleando un framework integrador **JBoss Seam** (que se verá en la sección, 2.6

Framework de Implementación) el cual reúne algunas tecnologías de Java EE, como: JSF, EJB 3.0, JPA, entre otras.

2.3.2 Que es Java EE

Java EE es un entorno de programación, que define un conjunto de **especificaciones** de APIs Java, para el desarrollo de aplicaciones distribuidas multicapa (3-capas, n-capas) basadas en web, empleando el lenguaje de programación Java.

En la plataforma Java EE 5.0 se incorporaron algunos cambios profundos de modelo conceptual, arquitectura y estrategia de desarrollo. Cambios tales como:

- Simplificación del desarrollo de componentes de la capa de negocio (idea principal de esta versión).
- Los ficheros descriptores de despliegue XML son opcionales, en su lugar se puede utilizar “**annotations**” (anotaciones, también conocidos como metadatos) que serán interpretadas por el servidor de aplicaciones Java EE en tiempo de ejecución. Estas anotaciones se las reconoce dentro del código fuente java ya que empiezan con el símbolo “@”, por ejemplo **@EJB**, **@Id**, entre otras.
- Java EE 5 proporciona **anotaciones** que cumplen diferentes tareas tales como: definir y usar Web Services, desarrollar aplicaciones EJB, mapeo de clases Java a XML o a Bases de datos, mapeo de métodos a operaciones, especificación de dependencias externas, especificación de información de despliegue incluyendo seguridad.

- Se programa en términos de objetos (entidades) y no en términos de tablas de base de datos.

2.3.3 Arquitectura Java EE

Java EE permite definir una arquitectura multicapa en la que cada una de las capas está orientada a proporcionar un tipo específico de funcionalidad a una aplicación.

La arquitectura contempla cuatro capas, en función al tipo de servicio y contendor:

- **Capa Cliente.-** También conocida como capa de presentación o de aplicación, en ella se ubican los **clientes** finales. Generalmente está compuesta por los programas que interactúan con el usuario de la aplicación.
- **Capa Web.-** Intermediario entre el cliente y otras capas, aunque componentes de la capa cliente (applets o aplicaciones) pueden acceder directamente a la capa EJB. Los componentes de esta capa utilizan el protocolo HTTP para recibir peticiones de los clientes y enviar respuestas a los mismos.
- **Capa Enterprise JavaBeans (EJB).-** También conocida como capa de negocio, en ella se ubican las reglas de negocio de una aplicación que son implementadas usando componentes EJB. Principalmente los EJB van a permitir que múltiples componentes de la aplicación tengan acceso a la lógica y datos de negocio de forma concurrente.

- **Capa Enterprise Information Systems (EIS).**- Llamada la capa de sistemas de información empresarial, es la encargada de gestionar la información permanente del sistema. En esta capa la aplicación Java EE se conecta con tecnologías como RDBMS.

2.3.4 Componentes de la arquitectura Java EE

Las aplicaciones Java EE están hechas a base de componentes. Siendo un componente una unidad de software auto contenida que es ensamblada dentro de una aplicación Java EE, con sus clases relacionadas y ficheros que se comunican con otros componentes.

La especificación de Java define los siguientes componentes Java EE⁵:

- Las aplicaciones clientes y Applets, son componentes que se ejecutan en el cliente.
- Java Servlets, JavaServer Faces (JSF), y JavaServer Pages (JSP), son componentes web que se ejecutan en el lado del servidor.
- Enterprise JavaBeans (EJB), son componentes (**beans empresariales**) de negocio y persistencia que se ejecutan en el lado del servidor.

Tener en cuenta que los componentes se comunican entre sí mediante métodos estándar como: HTTP, SSL, XML, RMI e IIOP.

A continuación se detalla cada uno de los componentes Java EE antes mencionados:

⁵ Sun Microsystems, Inc. The Java EE 5 Tutorial. Santa Clara, U.S.A

Clientes Java EE

Puede ser un **cliente web** o una **aplicación cliente**.

- **Cliente web**, consiste en una **página web dinámica** (generada por componentes web), y un **navegador web** que despliega las páginas. Además, si es necesario una página web puede alojar un **Applet** incrustado. Un **Applet** es una pequeña aplicación escrita en lenguaje java que se ejecuta en la máquina virtual de java (JVM) en el navegador web.
- **Aplicación cliente**, se ejecuta en la máquina cliente y proporciona una GUI creada con la API de Swing o AWT.

Componentes web

Pueden ser: Servlets o páginas creadas utilizando tecnología JavaServer Faces (JSF) y/o tecnología JavaServer Pages (JSP). Estos componentes se ejecutan en un contenedor (servidor) Web y son responsables de componer la presentación de datos en formato HTML.

Componentes de negocio

Pueden ser: Enterprise JavaBeans (EJB).

- Se ejecutan en un contenedor (servidor) de EJBs.
- Responsables de implementar la lógica de negocio y acceso a datos.
- Los EJBs reciben datos de un programa cliente, los procesan (si es necesario) y los envían a la capa EIS para almacenarlos.

- Un EJB puede conectarse a cualquier capa, aunque su misión esencial es conectarse con los sistemas de información empresarial (como un gestor de base de datos, un ERP, etc.).

La capa del sistema de información empresarial

- Capa de datos.
- Maneja el software EIS e incluye sistemas que son infraestructura como planeadores de recursos empresariales (ERPs), procesamiento de transacciones mainframe, sistemas de bases de datos y otros sistemas legados.

En el siguiente gráfico se ilustra una visión general de la arquitectura Java EE.

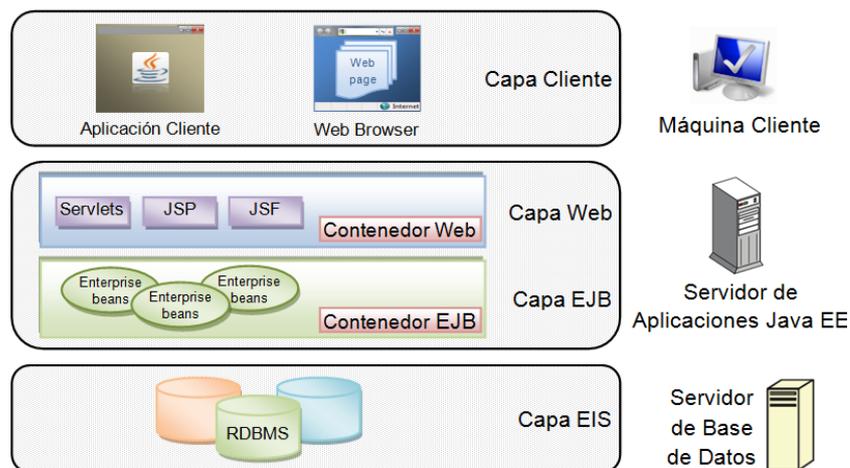


Figura 2.19: (Arquitectura Java EE y sus componentes)

2.3.5 Tecnologías y APIs más relevantes.

2.3.5.1 Java Database Connectivity (JDBC)

JDBC es un API estándar de Java para ejecutar sentencias SQL. Está formado por un conjunto de clases e interfaces escritas en el lenguaje de

programación Java. Provee acceso a una gran variedad de base de datos relacionales (RDBMS).

JDBC está diseñado para ser independiente de la plataforma, además provee una interface de programación única, que independiza a las aplicaciones del motor de base de datos usado.

Esta API está formada por dos componentes:

- Una interfaz a nivel de aplicación que usan los componentes para poder acceder a la base de datos.
- Una interfaz del proveedor de servicios (base de datos) que hace de puente entre el controlador JDBC y la plataforma Java EE.

La idea de utilizar esta API JDBC es que cualquier aplicación (java) que requiera acceder a una base de datos, solo deba comunicarse con la interfaz JDBC sin importar la base de datos. Esta interfaz será la encargada de comunicarse con los RDBMS. Esto hace que no se necesite conocer el funcionamiento de un RDBMS en particular, al contrario solo hay que conocer el funcionamiento de JDBC (ver Figura 2.20).

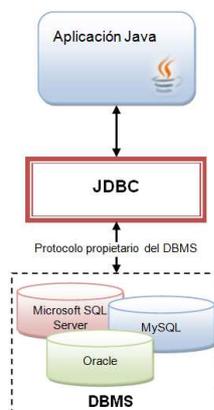


Figura 2.20: (Esquema JDBC)

Para acceder a una base de datos (desde programas java) se requiere de un **driver JDBC**, en el cual se encuentran definidas un conjunto de interfaces que el proveedor de base de datos implementa. Este driver es usado por la **JVM** para traducir las invocaciones **JDBC** genéricas en invocaciones que la base de datos propietaria entiende. Los drivers (controladores) son clases Java que se cargan en ejecución.

Existen cuatro tipos de controladores:

- a. **Puente JDBC-ODBC más driver ODBC.-** Suministra acceso vía drivers ODBC. No es muy productiva ya que necesita ser configurada para un controlador ODBC concreto, y en muchos casos el código cliente de base de datos, debe cargarse en cada máquina cliente que use este driver.

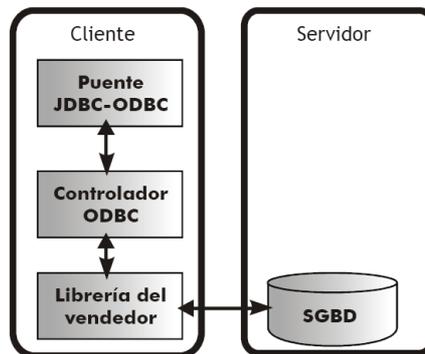


Figura 2.21: (Funcionamiento del controlador JDBC de tipo 1)

- b. **Driver Java parcialmente Nativo.-** Este driver se encuentra escrito parcialmente en Java y parcialmente escrito en el código nativo que comunica con el API de la base de datos. Este driver requiere que cierto código binario sea cargado en cada máquina cliente.

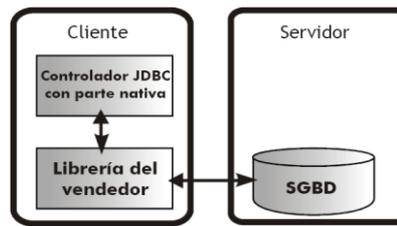


Figura 2.22: (Funcionamiento del controlador JDBC de tipo 2)

- c. **Driver Java nativo JDBC-Net.-** Este driver está escrito en java puro y se encarga de traducir llamadas JDBC al protocolo de red independiente del DBMS que después es traducido en el protocolo del DBMS mediante un servidor genérico de base de datos.

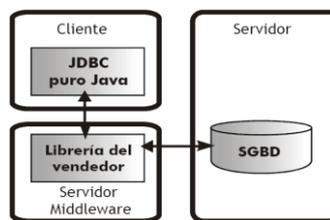


Figura 2.23: (Funcionamiento del controlador JDBC de tipo 3)

- d. **Driver Java puro y protocolo-nativo.-** Este driver está escrito en java puro y se comunica directamente a la base de datos con el protocolo de red del DBMS. No requiere de intermediarios entre el software JDBC y la base de datos.

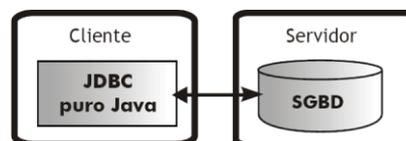


Figura 2.24: (Funcionamiento del controlador JDBC de tipo 4)

Por lo general los proveedores de DBMS proporcionan drivers del tipo 4, para la comunicación con sus bases de datos.

Debido a que en el presente proyecto se va utilizar **MySQL** como **RDBMS** se empleará el driver nativo (tipo 4) **Connector/J** (que se verá en la sección, 2.5.4 Sistema de Gestión de Base de Datos Relacionales (RDBMS)), el cual permite la comunicación directa de JDBC con MySQL.

2.3.5.2 Enterprise JavaBeans (EJB 3.0)

También conocidos como **Enterprise Beans**, estos son componentes del lado del servidor que encapsulan la lógica de negocio de una aplicación y son administrados por el contenedor de EJBs. Básicamente los EJBs son empleados para construir aplicaciones de negocio portables, reusables y escalables usando el lenguaje de programación Java; para esto la especificación EJB 3.0 define una arquitectura de componentes que facilita la implementación de este tipo de aplicaciones.

En la especificación 3.0, los Enterprise Beans no son más que simples **POJOs** (Plain Old Java Objects), en otras palabras son clases planas comunes y corrientes de java. Estas clases pueden ser configuradas por los programadores usando **annotations** (también llamados **metadata**), con el fin de agregar diferentes servicios que deberá proveer el **servidor de aplicaciones**. En la siguiente figura se ilustra como una **anotación** transforma un simple **POJO** en un **EJB**.



Figura 2.25: (Interpretación de un EJB)

Con el uso de **matadata** el **servidor de aplicaciones** (a través del contenedor de EJBs y el proveedor de persistencia) proveerá **servicios de nivel de sistema** a los Enterprise Beans, servicios tales como la concurrencia, la persistencia, la seguridad, las transacciones, entre otras; esto ha simplificado considerablemente el desarrollo de aplicaciones empresariales, debiendo el programador enfocarse más en la lógica de negocio y de presentación.

En la Figura 2.26 se muestra una arquitectura tradicional de cuatro capas, en la cual se resalta las capas de **lógica de negocio** y de **persistencia** que se implementan empleando la tecnología EJB 3, además se puede observar con más detalle los diferentes servicios que se proveen a los componentes EJB 3 de estas dos capas.

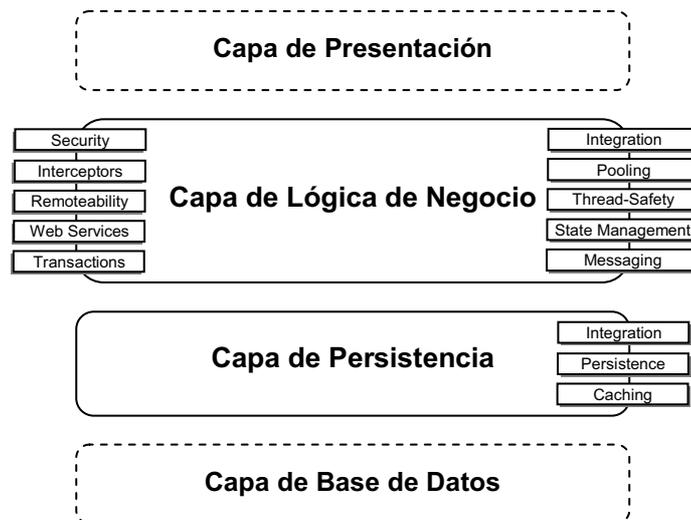


Figura 2.26: (Capas y servicios soportados por la tecnología EJB 3.0)
 Generado a partir de dos gráficos, del libro “EJB 3 in Action”, pág. 10.

Tipos de Enterprise Beans

La especificación 3.0 define tres tipos de componentes EJBs:

- a. Beans de sesión (session beans).

b. Beans dirigidos por mensajes (message-driven beans).

c. Entidades (entities).

Antes de describir cada uno de estos componentes, se presenta una figura en la cual se puede observar cómo está organizado el API EJB 3.

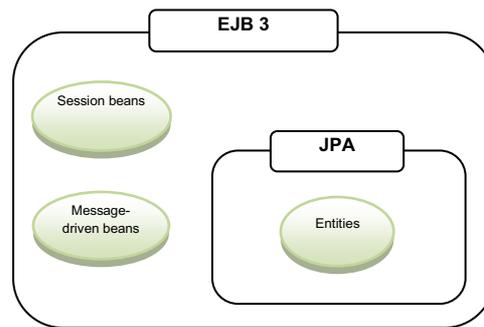


Figura 2.27: (Organización del API EJB3)

Los **session beans** y los **message-driven beans** son los responsables del procesamiento de la lógica de negocio, y están administrados por el **contenedor EJB**. Hasta J2EE 1.4 existían los **entity beans**, pero en Java EE 5.0 han sido reemplazados por las **entidades JPA** (objetos persistentes), las cuales son administradas por el **proveedor de persistencia** a través de la interface **EntityManager**.

Beans de sesión (session beans)

Este tipo de componentes **residen** en la capa de **lógica de negocio**, y son invocados por el cliente con el propósito de ejecutar operaciones (métodos de los session beans) de negocio específicas. Un session beans representa una conversación temporal con un cliente.

Hay dos tipos de session beans:

- a. **Beans de sesión con estado (stateful session beans).**- Tener presente que el estado de un objeto consiste en los valores de las variables de su instancia. Este tipo de beans **conserva** el estado de sus atributos a lo largo de toda una sesión. Un ejemplo típico de uso de estos beans, es en aplicaciones tipo wizard (programa intuitivo que guía paso a paso al usuario para realizar una tarea).
- b. **Beans de sesión sin estado (stateless session beans).**- Este tipo de beans **no conserva** el estado de sus atributos. Son los únicos que pueden exponerse como **web services**.

Beans dirigidos por mensajes (message-driven beans)

También **residen** en la capa de **lógica de negocio** y los servicios que proveen son parecidos a los session beans, con la diferencia de que los MDBs (message-driven beans) son usados para invocar métodos de forma asincrónica.

Un MDB es accionado por mensajes enviados a un servidor de mensajería (que permite el envío asíncrono de mensajes entre componentes del sistema), es decir, los clientes nunca invocan directamente métodos de los MDBs. Los mensajes pueden ser enviados por cualquier componente de Java EE (una aplicación cliente, otro enterprise bean, o un componente web) o por una aplicación JMS o sistema que no utiliza la tecnología Java EE.

Cuando se invoca un método de un MDB desde un cliente, la llamada no bloquea el código cliente y el mismo puede seguir ejecutándose, sin tener que

esperar indefinidamente por la respuesta del servidor. Los MDBs encapsulan el popular servicio de mensajería de Java (JMS).

Entidades (entities)

Al contrario de los otros componentes este **reside** en la capa de **persistencia**. Las entidades JPA son clases POJOs, no extienden de ninguna clase y no implementan ninguna interface. Normalmente una **entidad** representa una **tabla** en el modelo de datos relacional y cada **instancia** de esta entidad corresponde a un **registro** de esa tabla.

En EJB 3 la persistencia es administrada por la Java Persistence API (JPA), la cual persiste objetos java (entidades) usando la técnica ORM (Object Relational Mapping - Mapeo objeto relacional).

2.3.5.3 Java Persistence API (JPA)

JPA es una API de persistencia desarrollada para la **plataforma Java EE** e incluida en el estándar **EJB 3**. Pero también puede funcionar independientemente del resto de componentes EJB 3, hasta puede ser usado por una aplicación Java SE (aplicación desktop). JPA proporciona un modelo de persistencia basado en **POJOs** para mapear bases de datos relacionales.

El estándar JPA define⁶:

- El mapeo objeto relacional (ORM), mediante **metadata** que mapea **entidades** (objetos java) a **tablas relacionales**. Esto hace que no sea necesario el uso de ficheros descriptores XML.

⁶ Debu Panda, Reza Rahman, Derek Lane, (2007). EJB 3 in Action. United States of America: Manning Publications Co.

- La interface **EntityManager**, que define una API estándar para realizar las operaciones de persistencia (**CRUD**) de las entidades.
- El **Java Persistence Query Language (JPQL)**, para consultas y lecturas de datos de aplicación persistidos (algo así como un SQL orientado a objetos).

Fue desarrollado por el grupo de expertos de EJB 3 combinando ideas y conceptos de los principales **frameworks de persistencia** como Hibernate, Toplink, JDO y de las versiones anteriores de EJB, y también JDBC con DAO. Tener en cuenta que los frameworks que permiten aplicar la técnica ORM se los conoce como **frameworks ORM**, es así que en términos EJB 3 un **proveedor de persistencia** es básicamente un framework ORM que soporte JPA.

En este contexto, al codificar el software que es el objetivo del presente proyecto se utilizará como **proveedor de persistencia** JBoss Hibernate el cual es distribuido bajo los términos de la licencia GNU LGPL (software libre), además es el proveedor de persistencia por defecto de **JBoss Application Server** (que se verá en la sección, 2.5.5 Servidor de Aplicaciones) que es el servidor de aplicaciones Java EE que también se empleará en el presente proyecto.

Mapeo Objeto Relacional (ORM – Object Relational Mapping)

ORM es una técnica de programación muy útil para mapear (o persistir) **objetos/atributos** (entidades JPA) a **tablas/campos** con sus correspondientes relaciones, es decir es una técnica que permite manejar datos en Base de Datos Relaciones mediante objetos.

Como se mencionó anteriormente el mapeo se realiza mediante anotaciones (metadata), aunque esta tarea también se puede realizar a través de ficheros XML, pero en este proyecto se mostrará como efectuar el mapeo utilizando anotaciones. En este caso se empleará anotaciones JPA, conocidas también como anotaciones EJB 3.0 que se encuentran en el paquete “javax.persistence.*”.

Las anotaciones JPA se clasifican en dos categorías:

- **Anotaciones de mapeo lógico**, que permiten describir el modelo de objetos, asociaciones de clases, etc.
- **Anotaciones de mapeo físico**, que describen esquemas físicos de base de datos, tablas, columnas, índices, etc.

JPA reconoce dos tipos de clases persistentes: las clases entidad y las clases inmersas o embebidas.

En la siguiente tabla se muestra un conjunto de anotaciones JPA que serán las más utilizadas en este proyecto.

Anotaciones JPA		
Anotación	Valores posibles	Descripción
@Entity	<ul style="list-style-type: none"> ▪ name: por defecto el nombre de la clase, pero se puede especificar otra diferente. 	Define una clase como un bean de entidad, es decir la clase va a ser persistente.
@Table	<ul style="list-style-type: none"> ▪ name: nombre de la tabla, por defecto el nombre de la entidad. ▪ catalog: nombre del catálogo. ▪ schema: nombre del esquema. ▪ uniqueConstraints: constraints entre tablas relacionadas. 	Especifica la tabla relacionada con la entidad.
@Column	<ul style="list-style-type: none"> ▪ name: nombre de la columna, por defecto el nombre del atributo de la clase entidad. ▪ unique: si el campo tiene un único valor. ▪ nullable: si el campo permite valores nulos. ▪ insertable: si la columna se incluirá en la sentencia INSERT generada. ▪ updatable: si la columna se incluirá en la sentencia UPDATE generada. ▪ table: nombre de la tabla que contiene la columna. 	Define una columna de la tabla a mapear con un atributo de la entidad.

	<ul style="list-style-type: none"> ▪ length: longitud de la columna. ▪ precisión: número de dígitos decimales. ▪ scale: escala decimal. 	
@JoinColumn	<ul style="list-style-type: none"> ▪ name: nombre de la columna de la foreign key. ▪ referenced: nombre la columna referencia. ▪ unique: si el campo tiene un único valor. ▪ insertable: si la columna se incluirá en la sentencia INSERT generada. ▪ updatable: si el campo se incluirá en la sentencia UPDATE generada. ▪ table: nombre de la tabla que contiene la columna. 	Define un campo de la tabla que es foreign key de otra tabla definiendo la relación del lado propietario.
@Id	-	Indica la clave primaria de la tabla.
@Transient	-	Indica que el campo no debe persistir.
@OneToOne	<ul style="list-style-type: none"> ▪ cascade: indica la forma en que se deben actualizar los campos: ALL, PERSIST, MERGE, REMOVE y REFRESH. ▪ fetch: determina la forma en que se cargan los datos: FetchType.LAZY (carga de la entidad únicamente cuando se utiliza), FetchType.EAGER (carga de todas las entidades relacionadas con ella). ▪ optional: si la asociación es opcional. ▪ mappedBy: el campo que posee la relación, únicamente se especifica en un lado de la relación. 	(1:1) Cada instancia de una entidad está relacionada a una sola instancia de otra entidad.
@ManyToOne	<ul style="list-style-type: none"> ▪ cascade, fetch y optional: igual que la anterior anotación. 	(N:1) Muchas instancias de una entidad se relacionan a una sola instancia de otra entidad.
@OneToMany	<ul style="list-style-type: none"> ▪ cascade, fetch y optional: igual que la anterior anotación. ▪ mappedBy: el campo que posee la relación. Es obligatorio que la relación sea unidireccional. 	(1:N) Una entidad puede relacionarse a múltiples instancias de otra entidad.
@ManyToMany	<ul style="list-style-type: none"> ▪ cascade, fetch y mappedBy: igual que la anotación @OneToMany. 	(N:M) Las instancias de una entidad se pueden relacionar con múltiples entidades de otra clase y viceversa.
@Enumerated	-	Indica que el campo es un tipo enumerado (STRING), por defecto ORDINAL.

Tabla 2.3: (Anotaciones EJB 3/JPA)

Para entender mejor el uso de las anotaciones antes mencionadas se describe el siguiente ejemplo:

- ❖ Se desea registrar empresas con los siguientes datos: identificación, nombre, dirección y fono; además registrar los empleados que forman parte de la misma con los siguientes datos: identificación, número de cédula, nombre, apellido, género, correo electrónico, y número telefónico.

Tener en cuenta que los valores permitidos para género son: Masculino o Femenino.

Primero se diseñó un **diagrama de clases** que soporte los requerimientos anteriores.

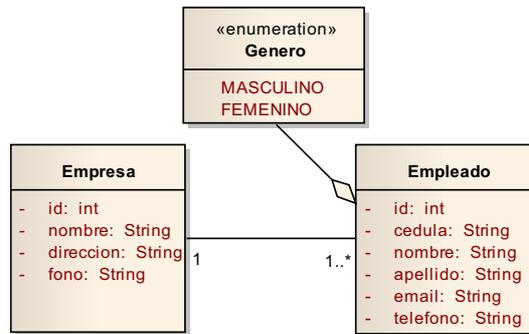


Figura 2.28: (Ejemplo de uso de las anotaciones JPA, Parte 1)

Como se puede observar en la figura anterior se han creado tres clases que son “Empresa”, “Empleado” y “Genero”; la clase Genero es de tipo Enumerado creada con el fin de restringir los valores posibles de género. Luego se creó el **modelo entidad relación** como se puede observar en la siguiente figura.

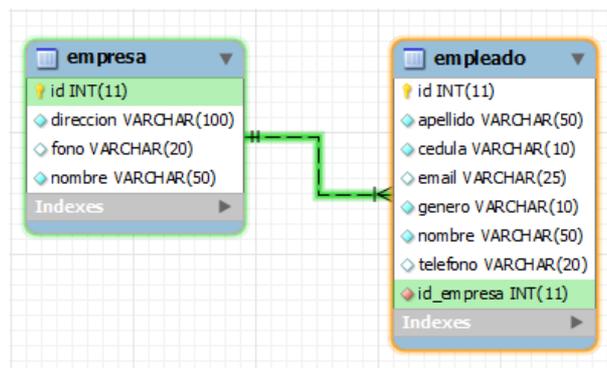


Figura 2.29: (Ejemplo de uso de las anotaciones JPA, Parte 2)

Finalmente, en base al modelo entidad relación se realiza el mapeo sobre las clases planas de java (POJOs), cuyo código fuente se puede observar a continuación:

```

@Entity
@Table(name = "EMPRESA")
public class Empresa {

    private int id;
    private String nombre;
    private String direccion;
    private String fono;
    private List<Empleado> empleados;

    @Column(name = "direccion", length = 100, nullable = false)
    public String getDireccion() {
        return direccion;
    }

    public void setDireccion(String direccion) {
        this.direccion = direccion;
    }

    @Column(name = "fono", length = 20)
    public String getFono() {
        return fono;
    }

    public void setFono(String fono) {
        this.fono = fono;
    }

    @Id
    @Column(name = "id")
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    @Column(name = "nombre", length = 50, nullable = false)
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    @OneToMany(mappedBy = "empresa")
    public List<Empleado> getEmpleados() {
        return empleados;
    }

    public void setEmpleados(List<Empleado> empleados) {
        this.empleados = empleados;
    }
}

@Entity
@Table(name = "EMPLEADO")
public class Empleado {

    private int id;
    private String cedula;
    private String nombre;
    private String apellido;
    private Genero genero;
    private String email;
    private String telefono;
    private Empresa empresa;

    @Column(name = "apellido", length = 50, nullable = false)
    public String getApellido() {
        return apellido;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }

    @Column(name = "cedula", length = 10, nullable = false, unique = true)
    public String getCedula() {
        return cedula;
    }

    public void setCedula(String cedula) {
        this.cedula = cedula;
    }

    @Column(name = "email", length = 25)
    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    @Enumerated(EnumType.STRING)
    @Column(name = "genero", length = 10, nullable = false)
    public Genero getGenero() {
        return genero;
    }

    public void setGenero(Genero genero) {
        this.genero = genero;
    }

    @Id
    @Column(name = "id")
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    @Column(name = "nombre", length = 50, nullable = false)
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    @Column(name = "telefono", length = 20)
    public String getTelefono() {
        return telefono;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "id_empresa", nullable = false)
    public Empresa getEmpresa() {
        return empresa;
    }

    public void setEmpresa(Empresa empresa) {
        this.empresa = empresa;
    }
}

```

Figura 2.30: (Ejemplo de uso de las anotaciones JPA, Parte 3)

Este ejemplo fue realizado empleando las herramientas de diseño y desarrollo descritas más adelante.

2.3.5.4 JavaServer Faces (JSF)

La tecnología JSF es un marco de trabajo (framework para capa web) que simplifica el desarrollo de **interfaces de usuario** en aplicaciones **Java EE**. Este framework implementa el patrón **MVC** (Modelo-Vista-Controlador), el cual separa las aplicaciones en tres capas diferenciadas (ver figura 2.31): datos (Modelo), interfaz de usuario (Vista) y lógica de control (Controlador). Esto permite que el mantenimiento de las aplicaciones JSF sea sencillo.

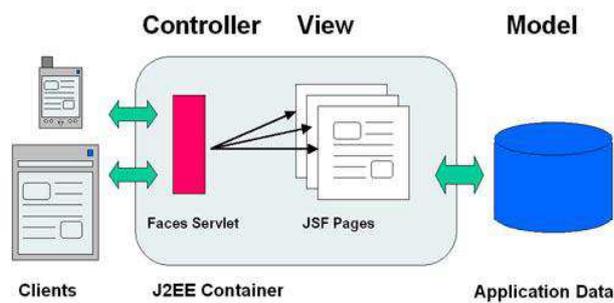


Figura 2.31: (Modelo MVC)

JavaServer Faces es un conjunto de componentes de usuario (UI) para construir la capa de vista de las aplicaciones web.

JSF define un conjunto de APIs para representar componentes de interfaz de usuario (UI-User Interface), y gestionar su funcionamiento mediante el tratamiento de eventos, las validaciones de entrada, la definición de un esquema de navegación entre páginas y el soporte para internacionalización y accesibilidad.

2.4 Herramientas de Diseño

2.4.1 Introducción

En el mercado existen numerosas herramientas CASE para llevar a cabo el Análisis y Diseño de cualquier sistema. Se ha seleccionado la herramienta **Enterprise Architect versión 7.5**, fundamentalmente por dos razones: dispone de soporte UML y facilita la aplicación de la metodología ICONIX. Esta herramienta es de tipo comercial, pero en este proyecto se empleará una versión de evaluación.

2.4.2 Enterprise Architect

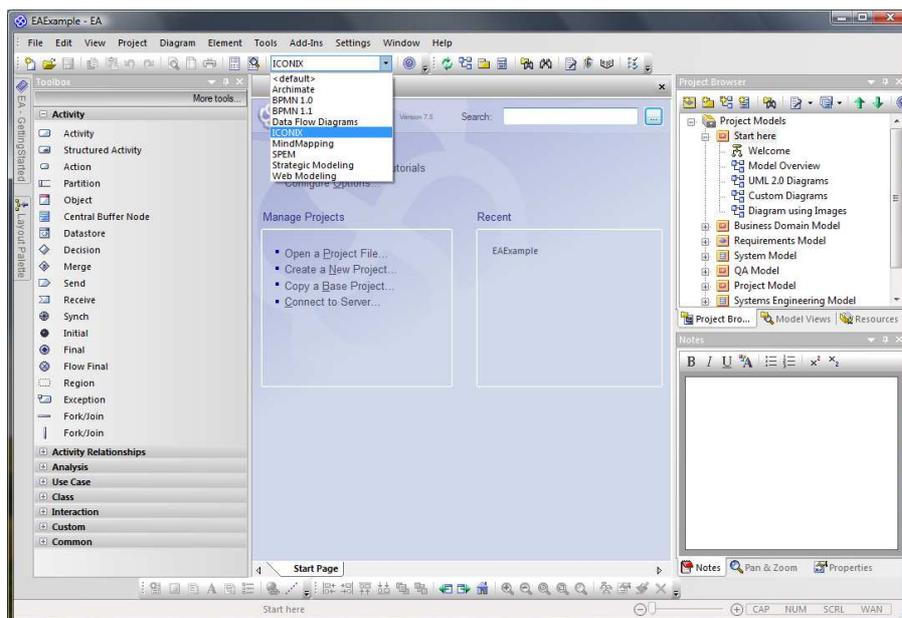


Figura 2.32: (Enterprise Architect GUI)

Enterprise Architect (EA) es una herramienta CASE, de Sparx Systems. Utilizada para la construcción de sistemas de software, para el modelado de procesos de negocios, y para objetivos de modelado más generalizados. Esta herramienta está basada en la especificación UML 2.1, que define un lenguaje

visual que usa para modelar un dominio o sistema en particular (existente o propuesto).

EA abarca todos los aspectos del ciclo de desarrollo de software, desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. Además proporciona una trazabilidad completa desde la fase inicial del diseño a través del despliegue y mantenimiento.

Características principales:

- Permite modelar sistemas de hardware y software complejos en notación UML.
- Ofrece soporte para diferentes tipos de diagramas de UML como: diagramas estructurales (clase, objeto, compuesto, paquete, componente, y despliegue.), diagramas de comportamiento (casos de uso, comunicación, secuencia, descripción de la interacción, actividad, estado, y tiempo). Además extendidos como: análisis (actividad simple), personalizado (para requisitos, cambios, y UI).
- Generación e ingeniería inversa de código fuente, para distintos lenguajes de programación como: C++, C#, Java, VB.Net, entre otros.
- Soporte para casos de prueba, JUnit y NUnit.
- Permite control de versiones, empleando CVS o SCC.
- Generación de documentación en formatos HTML y RTF.

2.5 Herramientas de Desarrollo

2.5.1 Introducción

En esta sección se hace referencia a los recursos de software necesarios para el desarrollo del software objetivo del presente proyecto. La mayoría de las herramientas son de tipo “software libre”; a continuación se detalla cada una de estas.

2.5.2 Entorno de Desarrollo Integrado (IDE)

El entorno de programación que se empleará es **Netbeans 6.7.1**. Tener presente que para crear aplicaciones java con Netbeans se debe disponer de la plataforma de desarrollo **Java Development Kit (JDK)**.

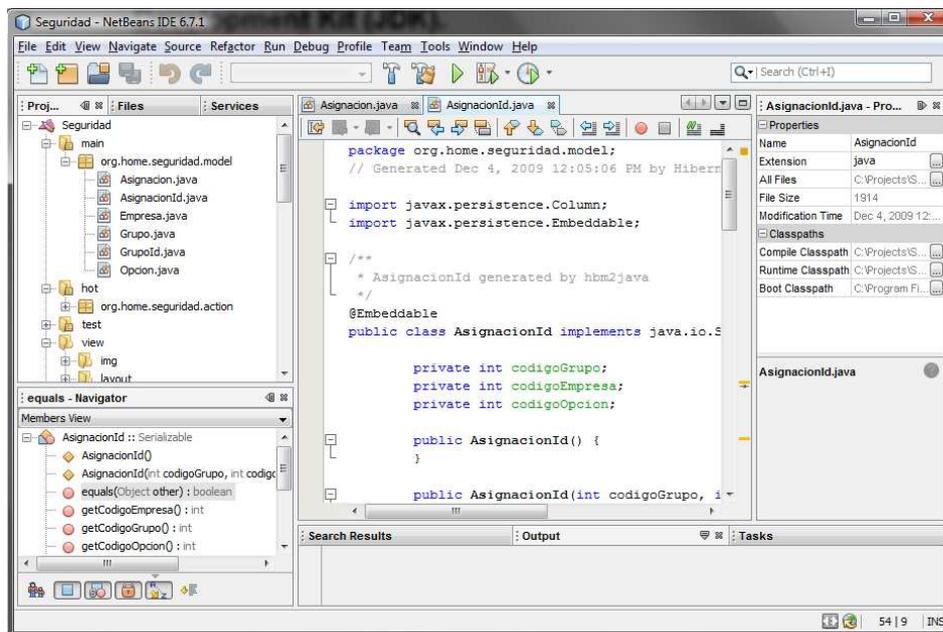


Figura 2.33: (Netbeans 6.7.1 GUI)

Características principales de Netbeans:

- Es un **producto libre y gratuito** sin restricciones de uso.

- Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación.
- Soporta el desarrollo de todos los tipos de aplicaciones Java.
- Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. Existe además un número importante de módulos para extender el IDE Netbeans.

Tener en cuenta que, para generar el esqueleto del proyecto seam (incluyendo configuración y librerías) se empleará **Seam Generator (seam-gen)**. Seam Generator es una aplicación de línea de comandos para crear y mantener aplicaciones Seam.

2.5.3 Control de Versiones

Para controlar las distintas versiones del código fuente, se empleará **Subversion 1.6.6**.

Subversion es un sistema de control de versiones, utilizado para el mantenimiento de código fuente, documentación técnica y páginas web. Es decir, Subversion maneja ficheros y directorios a través del tiempo. Hay un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios.

Para facilitar la creación de repositorios, usuarios y otras configuraciones se utilizará el paquete **VisualSVN Server 2.0.8**, que incluye **Subversion 1.6.6** y el servidor web **Apache 2.2.13** ya configurados y listos para funcionar.

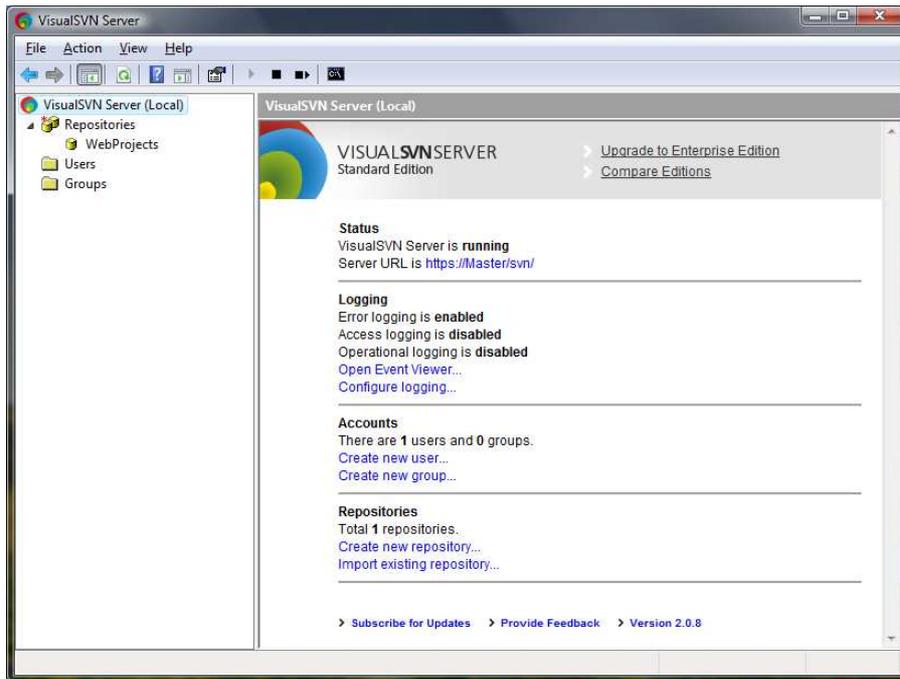


Figura 2.34: (VisualSVN Server 2.0.8 GUI)

Características principales de Subversión:

- Es un **producto libre** y de código fuente abierto.
- Se utiliza en cualquier lenguaje de Programación y para cualquier tipo de proyecto.
- Implementa un sistema de ficheros versionado “virtual” que sigue los cambios sobre árboles de directorios completos a través del tiempo.
- Se puede añadir, borrar, copiar, y renombrar ficheros y directorios.
- Una colección cualquiera de modificaciones o bien entra por completo al repositorio, o bien no lo hace en absoluto.
- Subversion expresa las diferencias del fichero usando un algoritmo de diferenciación binario.

- Subversion puede acceder al repositorio a través de redes.

Tener en cuenta que se puede instalar el soporte de Subversion en Netbeans.

2.5.4 Sistema de Gestión de Base de Datos Relacionales (RDBMS)

Un sistema de gestión de bases de datos es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se empleará **MySQL Server 5.1** como RDBMS. Además se instalará **MySQL Tools for 5.0**, el cual proporciona interfaz gráfica de usuario para utilizar con el servidor de bases de datos.

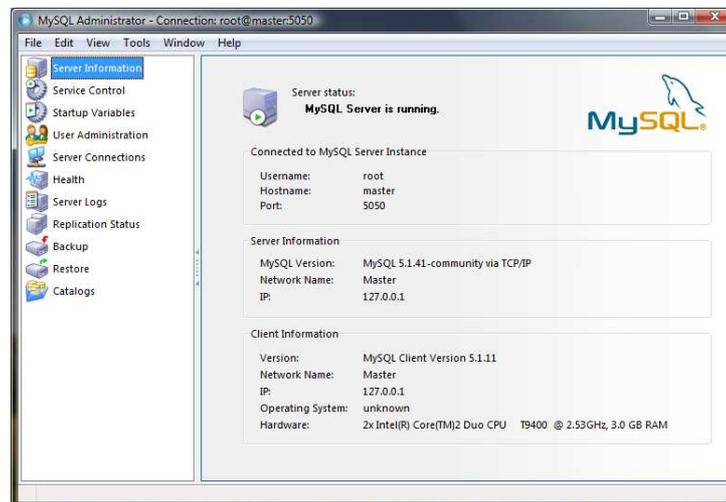


Figura 2.35: (MySQL Administrator GUI)

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. De licencia dual depende del uso, existen 2 versiones: una gratuita y otra más completa de pago. Se utilizará la **versión gratuita**.

Características principales de MySQL Server:

- Es un producto de licencia dual (gratuita y comercial).
- El servidor de base de datos MySQL es muy rápido, fiable y fácil de usar.

- MySQL Server trabaja en entornos cliente/servidor o incrustados.
- Funciona en diferentes plataformas.
- Proporciona sistemas de almacenamiento transaccional y no transaccional.
- MySQL server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas.
- Seguridad: Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host.
- Escalabilidad: Soporte a grandes bases de datos.

Conectores MySQL

Son controladores (drivers) que proporcionan a los programas cliente conectividad con el servidor MySQL. Se empleará el **Connector/J (mysql-connector-java-5.1.10)**, que proporciona soporte de controlador para conectar con MySQL desde una aplicación Java usando la API de Conectividad con Bases de Datos Java estándar (JDBC). Conector/J es un driver nativo que convierte llamadas JDBC al protocolo de red utilizado por la base de datos MySQL.

Es importante mencionar que la tecnología utilizada en el desarrollo del software objetivo del presente proyecto, permitirá con mínimas configuraciones, usar otro RDBMS como: Oracle, Microsoft SQL Server, etc.

2.5.5 Servidor de Aplicaciones

Como servidor de aplicaciones se utilizará JBoss Application Server 5.1.0.

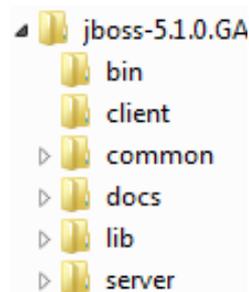


Figura 2.36: (Estructura de directorios de JBoss)

JBoss es un servidor de aplicaciones Java EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte. JBoss implementa todo el paquete de servicios de Java EE (EJB, JMS, JTS/JTA, Servlets/JSP, JNDI).

Características principales de JBoss:

- Producto de **licencia de código abierto** sin costo adicional.
- Cumple los estándares.
- Confiable a nivel de empresa.
- Incrustable, orientado a arquitectura de servicios.
- Flexibilidad consistente.
- Servicios del middleware para cualquier objeto de Java.
- Soporte completo para JMX.

Finalmente, cabe mencionar que se trabajará sobre el sistema operativo **Windows Vista Home Premium**, y se empleará **Mozilla Firefox 3.6.6** como browser para visualizar las páginas web del proyecto.

2.6 Framework de Implementación

2.6.1 Introducción

Con el fin de facilitar el proceso de desarrollo, reutilizar código, y promover buenas prácticas de desarrollo como el uso de patrones, se aplicará como apoyo un framework web.

Antes de mencionar el **framework web** que se empleará en el desarrollo del software orientado a la web; se debe tener presente que un **framework** es una estructura conceptual y tecnológica de soporte definida, normalmente compuesta de componentes personalizables e intercambiables, en base a los cuales un proyecto de software puede ser organizado y desarrollado. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable (puede incluir soporte de programas, bibliotecas y un lenguaje interpretado, entre otros programas) a la que se puede añadir las últimas piezas para construir una aplicación concreta.

Existen muchos frameworks para apoyar el desarrollo de aplicaciones web basadas en java (ver Figura 2.37). Se ha seleccionado **JBoss Seam 2.2.0.GA**, ya que es un framework desarrollado exclusivamente para facilitar la creación de aplicaciones Java EE.

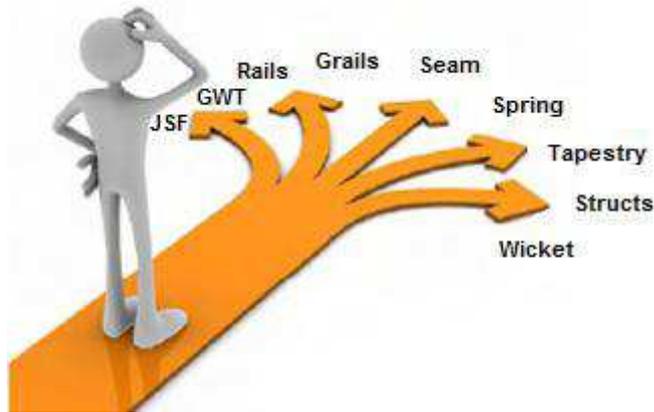


Figura 2.37: (Frameworks para el desarrollo de aplicaciones web)
Gráfico del libro “Seam in Action”.

2.6.2 JBoss Seam Framework

2.6.2.1 Que es JBoss Seam

Seam es un framework que reúne los estándares de Java EE para que puedan trabajar como una solución integrada. “Seam integra tecnologías como JavaScript asíncrono y XML (AJAX), JavaServer Faces (JSF), Java Persistence API (JPA)/Hibernate, Enterprise Java Beans (EJB 3.0) y Business Process Management (BPM). Seam ha sido diseñado desde cero para eliminar la complejidad, tanto en la arquitectura y los niveles de la API. Permite a los desarrolladores ensamblar aplicaciones web complejas con simples clases Java anotadas, un rico conjunto de componentes de interfaz de usuario, y muy poco XML”⁷. En la siguiente figura se muestra las tecnologías que Seam es capaz de reunir en una aplicación típica.

⁷ <http://www.seamframework.org/Home>

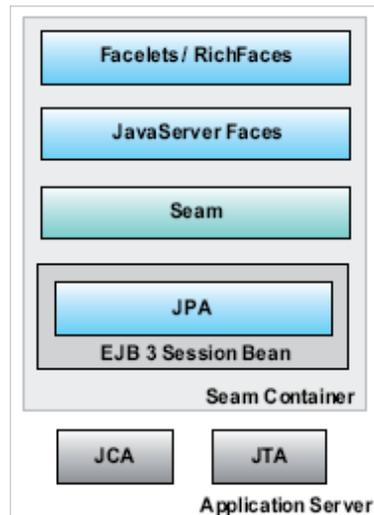


Figura 2.38: (Ejemplo de algunas tecnologías que integra Seam)
Gráfico del libro “Seam in Action”.

Seam no sólo es considerado como un tradicional **web framework**, sino como un **application stack** que incluye, el framework, las librerías, y el generador del proyecto (Seam Generator).

2.6.2.2 Características principales de JBoss Seam

- Básicamente Seam facilita que EJB 3.0 y JSF (componentes principales de Java EE) trabajen juntos, es decir integra la capa de presentación (**JSF**) con la capa de negocio y persistencia (**EJB**). Para lo cual Seam provee un **modelo de componentes** que permiten el uso directo de componentes EJB como backing beans en páginas JSF.
- Integración con **AJAX**, mediante el soporte de tecnologías como **ICEFaces** o **RichFaces**. Esto hace posible la creación de aplicaciones interactivas o RIA (Rich Internet Application). En el presente proyecto se utilizará la tecnología RichFaces.

- Seam introduce el concepto de **contextos** (manejo de estados), es así que cada **componente seam** existe dentro de un contexto. Los contextos que maneja seam se muestran en la siguiente figura.

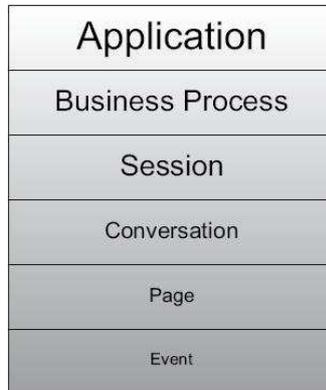


Figura 2.39: (Contextos de Seam)

Los componentes seam pueden ser:

- ❖ Java Beans (POJO)
 - JavaBean
 - Groovy class (Groovy Bean)
 - Spring bean
- ❖ EJB Component
 - Stateless session bean
 - Stateful session bean
 - Message-driven vean
- ❖ JPA entity class.

- Elimina el exceso de uso de archivos descriptores XML. Para esto, seam ha reducido algunas configuraciones de estos archivos a simples anotaciones.
- Seam define un archivo descriptor denominado **pages.xml** que ofrece una gama mucho amplia de controles de navegación de lo que el archivo **faces-config.xml** de JSF puede soportar. Este archivo se utiliza para configurar una infinita serie de páginas, aunque también puede ser configurada individualmente por cada página JSF (ver Figura 2.40), con el fin de definir reglas de navegación contextual, generar mensajes, pasar parámetros en una redirección, invocar acciones (pueden ser métodos de un session beans) antes de representar una vista JSF, hacer cumplir restricciones de seguridad, manejar excepciones, entre otras configuraciones.

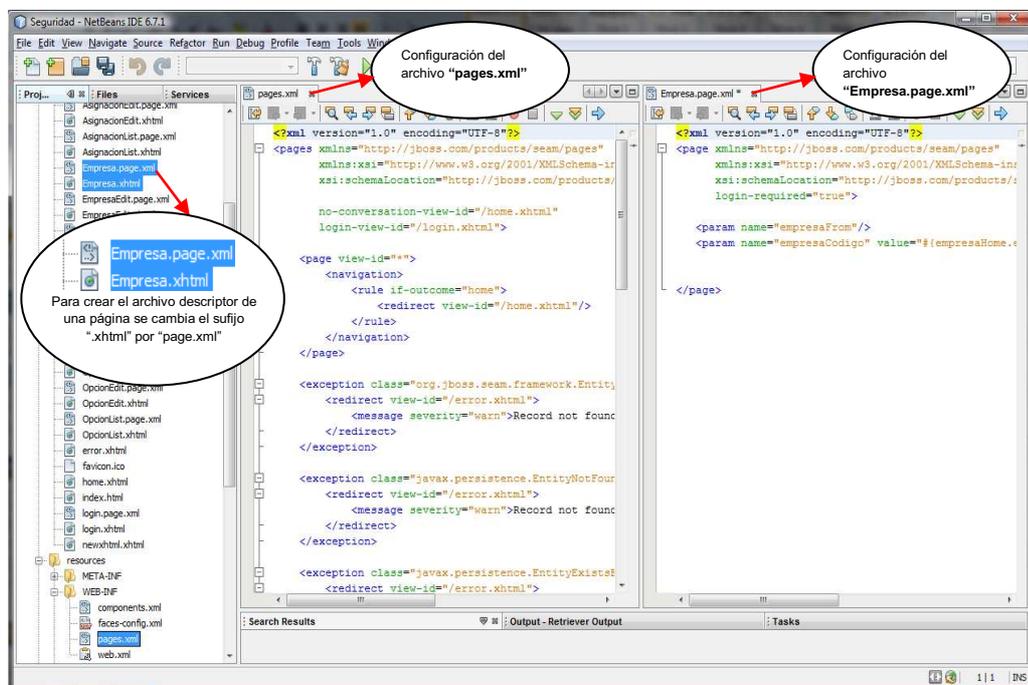


Figura 2.40: (Archivo descriptor de Seam para el control de navegación)

2.6.2.3 Anotaciones Seam

Al desarrollar una aplicación Java EE 5.0 empleando el framework seam se dispondrá de diferentes anotaciones que facilitarán al programador la creación de aplicaciones empresariales. La mayoría de las anotaciones son suministradas por el API EJB 3, y las anotaciones para las validaciones de datos se encuentran en el paquete Hibernate Validator (org.hibernate.validator.). En la siguiente tabla se muestra un resumen de las anotaciones más relevantes de seam que se encuentran definidas en el paquete org.jboss.seam.annotations.

Anotaciones Seam		
Anotación	Ejemplo	Descripción
@Name	@Name("nombreComponente")	Define el nombre del componente seam.
@Scope	@Scope(ScopeType.CONVERSATION)	Define el contexto por defecto del componente seam. Los valores posibles son: EVENT, PAGE, CONVERSATION, SESSION, BUSINESS_PROCESS, APPLICATION, STATELESS.
@Role	@Role(name="roleName", scope=" ScopeType.SESSION")	Permite a un componente seam estar relacionado a multiples variables de contexto. El parámetro name define el nombre de la variable de contexto, y scope define el alcance de la variable de contexto.
@JndiName	@JndiName("my/jndi/name")	Especifica el nombre JNDI que seam usará para buscar el componente EJB.
@In	@In @In(required=false) @In(create=true)	Especifica que el atributo de un componente será inyectado desde una variable de contexto al comienzo de cada llamada al componente.
@Out	@Out @Out(required=false) @Out(scope=ScopeType.SESSION)	Especifica que el atributo de un componente será colocado a una variable de contexto al final de la invocación del componente.
@RequestParameter	@RequestParameter("requestParameterName")	Especifica que un atributo de un componente debe ser inyectado con el valor de un parámetro solicitado (request parameter).

Tabla 2.4: (Anotaciones de Seam)

2.6.2.4 Creación de un proyecto Seam

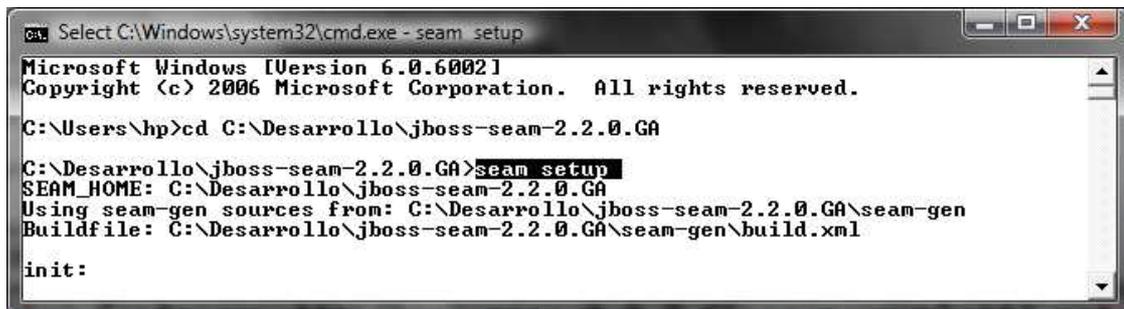
En esta sección se explicará paso a paso la creación de un proyecto web empleando JBoss Seam Framework. Tener en cuenta que para realizar este ejemplo se emplearán las herramientas de desarrollo antes mencionadas.

Prerrequisitos

Instalar el Framework Seam, el Servidor de aplicaciones, la plataforma de desarrollo de java (JDK 1.6) y Apache Ant.

Configuración del proyecto

En la consola de Windows ingresar al directorio donde previamente se instaló (descomprimió) el framework seam, y ejecutar el comando **seam setup**.



```
GA: Select C:\Windows\system32\cmd.exe - seam_setup
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Desarrollo\jboss-seam-2.2.0.GA
C:\Desarrollo\jboss-seam-2.2.0.GA>seam setup
SEAM_HOME: C:\Desarrollo\jboss-seam-2.2.0.GA
Using seam-gen sources from: C:\Desarrollo\jboss-seam-2.2.0.GA\seam-gen
Buildfile: C:\Desarrollo\jboss-seam-2.2.0.GA\seam-gen\build.xml

init:
```

Figura 2.41: (Proyecto Seam, ejecución del comando “seam setup”)

A continuación se debe ingresar un conjunto de propiedades (o preguntas) que solicita seam con el fin de configurar la estructura del proyecto:

Directorio donde se ubicará el proyecto: C:\Projects\Test

Directorio donde se encuentra el servidor JBoss:

C:\Desarrollo\jboss-5.1.0.GA

Dominio del servidor JBoss: default

Directorio del servidor GlassFish:

Ignorar esta pregunta ya que se está empleando el servidor JBoss.

Dominio del servidor GlassFish:

Ignorar esta pregunta ya que se está empleando el servidor JBoss.

Nombre del proyecto: myproject

Usar ICEfaces en lugar de RichFaces? (s/n): n

Skin que será utilizado por RichFaces (blueSky, ruby, wine...): wine

Empaquetamiento del proyecto (ear/war): ear

Nombre del paquete base para las clases java: com.tesis.myproject

Nombre del paquete que contendrá los beans de sesión:

com.tesis.myproject.action

Nombre del paquete que contendrá los beans de entidad:

com.tesis.myproject.entity

Nombre del paquete que contendrá las clases de prueba:

com.tesis.myproject.test

Base de datos que se empleará (mysql, derby, oracle, db2...): mysql

Directorio del driver JDBC:

C:\Desarrollo\jboss-5.1.0.GA\server\default\lib\mysql-connector-java-5.1.10-

bin.jar (colocar el driver JDBC en el directorio server/default/lib de JBoss)

Dialecto de Hibernate para la base de datos:

org.hibernate.dialect.MySQLDialect

Clase del driver JDBC para MySQL: com.mysql.jdbc.Driver

Clase del Datasource JDBC para MySQL:

com.mysql.jdbc.jdbc2.optional.MysqlDataSource

URL de conexión a la base de datos:

jdbc:mysql://localhost:5050/mydatabase

Usuario de la base de datos: root

Password de la base de datos: root

Nombre de la base de datos: mydatabase

¿Usar todas las tablas que están en la base de datos?: y

¿Borrar la base de datos en cada despliegue?: n

Finalmente si todo se ejecuta sin problemas se mostrará lo siguiente:



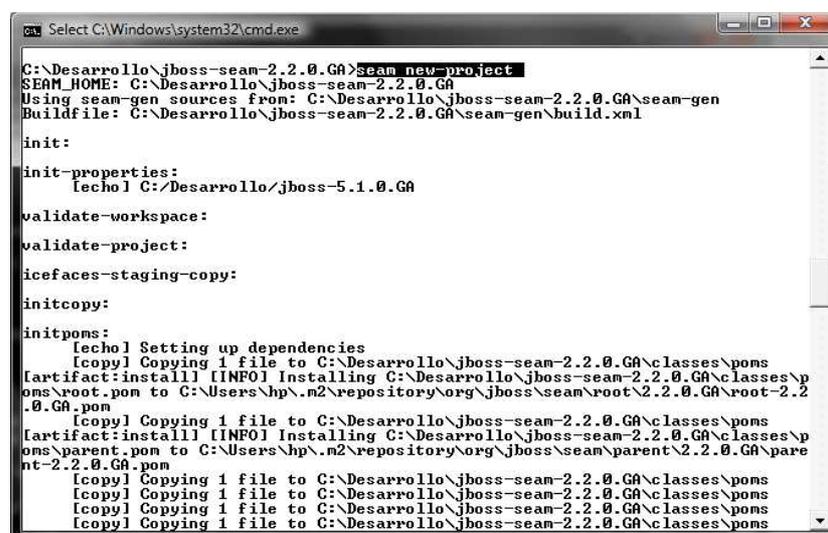
```
C:\Windows\system32\cmd.exe
[propertyfile] Creating new property file: C:\Desarrollo\jboss-seam-2.2.0.GA\seam-gen\build.properties
[echo] Installing JDBC driver jar to JBoss AS
[copy] Copying 1 file to C:\Desarrollo\jboss-5.1.0.GA\server\default\lib
init:
init-properties:
  [echo] C:/Desarrollo/jboss-5.1.0.GA
validate-workspace:
validate-project:
settings:
  [echo] JBoss AS home: C:/Desarrollo/jboss-5.1.0.GA
  [echo] GlassFish home: -
  [echo] Project name: myproject
  [echo] Project location: C:/Projects/Test/myproject
  [echo] Project type: ear
  [echo] IceFaces: n
  [echo] Action package: com.tesis.myproject.action
  [echo] Model package: com.tesis.myproject.entity
  [echo] Test package: com.tesis.myproject.test
  [echo] JDBC driver class: com.mysql.jdbc.Driver
  [echo] JDBC DataSource class: com.mysql.jdbc.jdbc2.optional.MysqlDataSource

  [echo] Hibernate dialect: org.hibernate.dialect.MySQLDialect
  [echo] JDBC URL: jdbc:mysql://localhost:5050/mydatabase
  [echo] Database username: root
  [echo] Database password: root
  [echo]
  [echo] Type 'seam create-project' to create the new project
BUILD SUCCESSFUL
Total time: 6 minutes 54 seconds
C:\Desarrollo\jboss-seam-2.2.0.GA>
```

Figura 2.42: (Proyecto seam, resultado del comando “seam setup”)

Generación de la estructura del proyecto

Para generar el proyecto en base a las propiedades antes configuradas se debe ejecutar el comando **seam new-project**.



```
C:\Windows\system32\cmd.exe
C:\Desarrollo\jboss-seam-2.2.0.GA>seam new-project
SEAM_HOME: C:\Desarrollo\jboss-seam-2.2.0.GA
Using seam-gen sources from: C:\Desarrollo\jboss-seam-2.2.0.GA\seam-gen
Buildfile: C:\Desarrollo\jboss-seam-2.2.0.GA\seam-gen\build.xml
init:
init-properties:
  [echo] C:/Desarrollo/jboss-5.1.0.GA
validate-workspace:
validate-project:
icefaces-staging-copy:
initcopy:
initpons:
  [echo] Setting up dependencies
  [copy] Copying 1 file to C:\Desarrollo\jboss-seam-2.2.0.GA\classes\pons
[artifact:install] INFO! Installing C:\Desarrollo\jboss-seam-2.2.0.GA\classes\pons\root.pon to C:\Users\hp\.m2\repository\org\jboss\seam\root\2.2.0.GA\root-2.2.0.GA.pon
  [copy] Copying 1 file to C:\Desarrollo\jboss-seam-2.2.0.GA\classes\pons
[artifact:install] INFO! Installing C:\Desarrollo\jboss-seam-2.2.0.GA\classes\pons\parent.pon to C:\Users\hp\.m2\repository\org\jboss\seam\parent\2.2.0.GA\parent-2.2.0.GA.pon
  [copy] Copying 1 file to C:\Desarrollo\jboss-seam-2.2.0.GA\classes\pons
  [copy] Copying 1 file to C:\Desarrollo\jboss-seam-2.2.0.GA\classes\pons
  [copy] Copying 1 file to C:\Desarrollo\jboss-seam-2.2.0.GA\classes\pons
  [copy] Copying 1 file to C:\Desarrollo\jboss-seam-2.2.0.GA\classes\pons
```

Figura 2.43: (Proyecto seam, ejecución del comando “seam new-project”)

Al finalizar la ejecución del comando **seam new-project** se crearán los siguientes archivos en el directorio “C:\Projects\Test” previamente configurado:

Name	Date modified	Type	Size
.settings	7/20/2010 4:50 PM	File Folder	
bootstrap	7/20/2010 4:50 PM	File Folder	
lib	7/20/2010 4:50 PM	File Folder	
nbproject	7/20/2010 4:50 PM	File Folder	
resources	7/20/2010 4:50 PM	File Folder	
src	7/20/2010 4:50 PM	File Folder	
view	7/20/2010 4:50 PM	File Folder	
.classpath	7/20/2010 4:50 PM	CLASSPATH File	3 KB
.project	7/20/2010 4:50 PM	PROJECT File	1 KB
build.properties	7/20/2010 4:50 PM	PROPERTIES File	1 KB
build	7/20/2010 4:50 PM	XML Document	21 KB
build-dev.properties	7/20/2010 4:50 PM	PROPERTIES File	1 KB
build-prod.properties	7/20/2010 4:50 PM	PROPERTIES File	1 KB
debug-glassfish-mypr...	7/20/2010 4:50 PM	LAUNCH File	1 KB
debug-jboss-myprojec...	7/20/2010 4:50 PM	LAUNCH File	1 KB
deployed-jars-ear.list	7/20/2010 4:50 PM	LIST File	1 KB
deployed-jars-war.list	7/20/2010 4:50 PM	LIST File	1 KB
explode.launch	7/20/2010 4:50 PM	LAUNCH File	2 KB
glassfish-build	7/20/2010 4:50 PM	XML Document	11 KB
glassfish-readme	7/20/2010 4:50 PM	Text Document	3 KB
hibernate-console.pro...	7/20/2010 4:50 PM	PROPERTIES File	1 KB
myproject.iml	7/20/2010 4:50 PM	IML File	12 KB
myproject.ipr	7/20/2010 4:50 PM	IPR File	15 KB
myproject.launch	7/20/2010 4:50 PM	LAUNCH File	1 KB
seam-gen.properties	7/20/2010 4:50 PM	PROPERTIES File	2 KB
validate	7/20/2010 4:50 PM	XML Document	5 KB

Figura 2.44: (Proyecto seam, resultado del comando “seam new-project”)

Este proyecto se puede abrir en Eclipse o Netbeans, en este caso se abrirá en Netbeans.

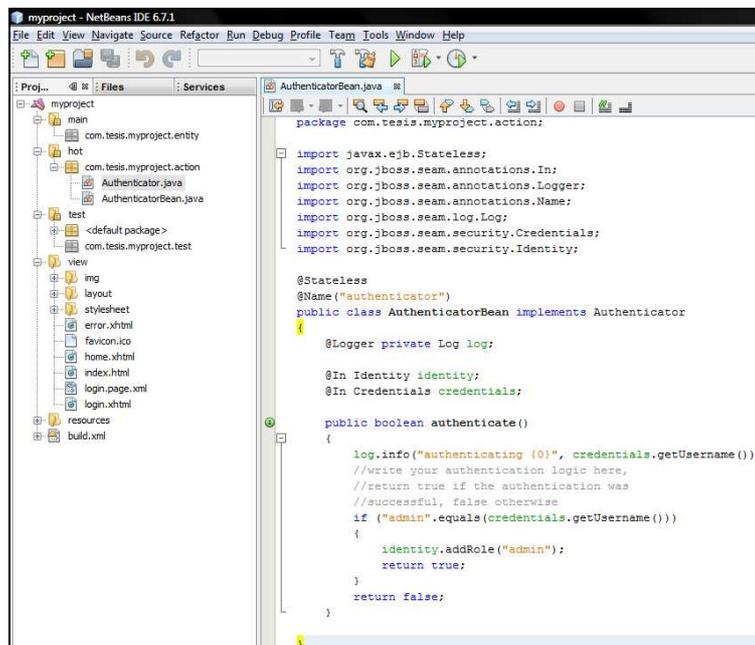


Figura 2.45: (Proyecto seam, Visualización del Proyecto seam con NetBeans)

En la figura anterior se puede observar la estructura del proyecto generado en base a las propiedades previamente configuradas. En este punto se puede desplegar y probar la aplicación la cual mostrará una página principal y una página para iniciar sesión (con el usuario “admin”). El inicio de sesión se logra gracias a las clases “Authenticator” y “AuthenticatorBean”.

Generación de la funcionalidad de la aplicación

Seam permite a partir de una base de datos crear los beans de entidad y la lógica de negocio (operaciones CRUD de las entidades) de la aplicación. Para el ejemplo se creó una base de datos que permite almacenar **países** con sus respectivas **provincias**, cuyo script de creación se muestra en la siguiente figura.

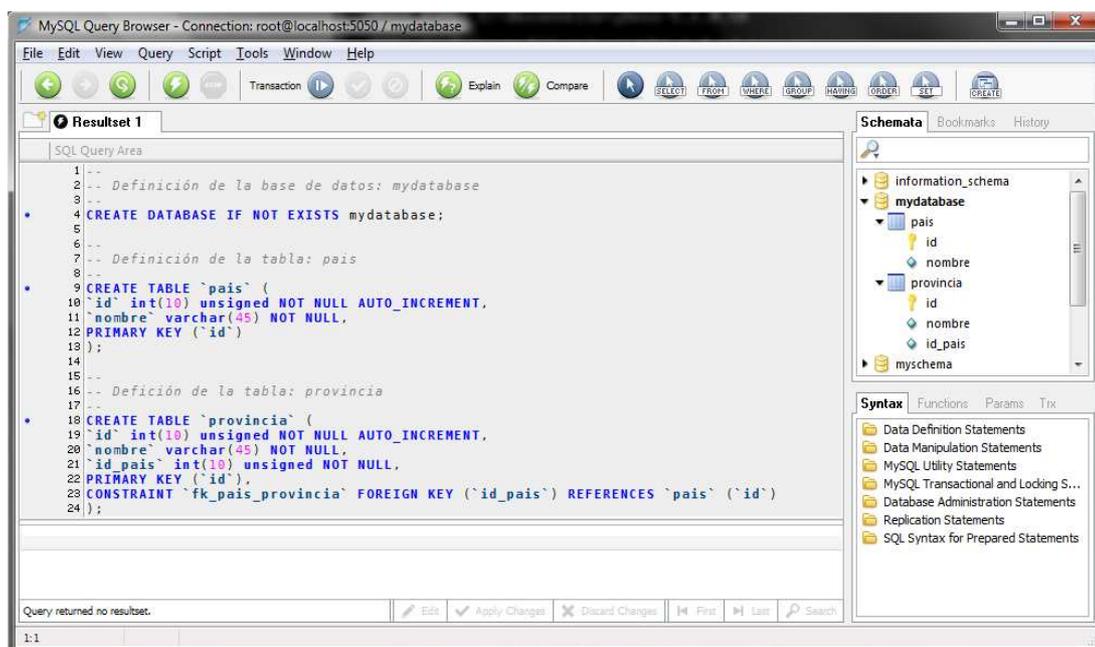
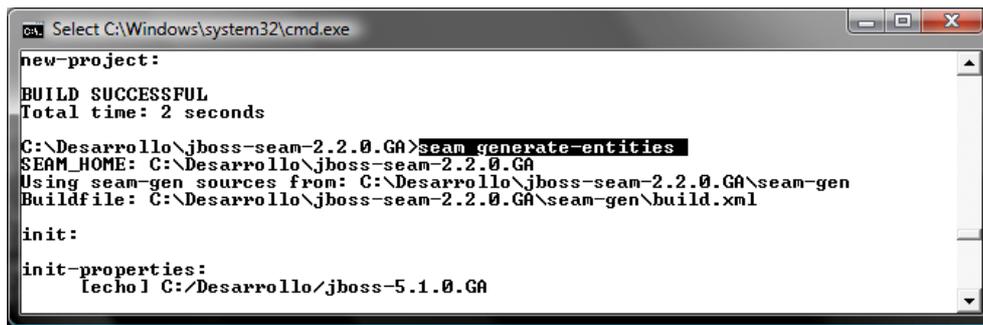


Figura 2.46: (Proyecto seam, Script de creación de la Base de Datos)

Luego de crear la base de datos ejecutar el comando **seam generate-entities**.



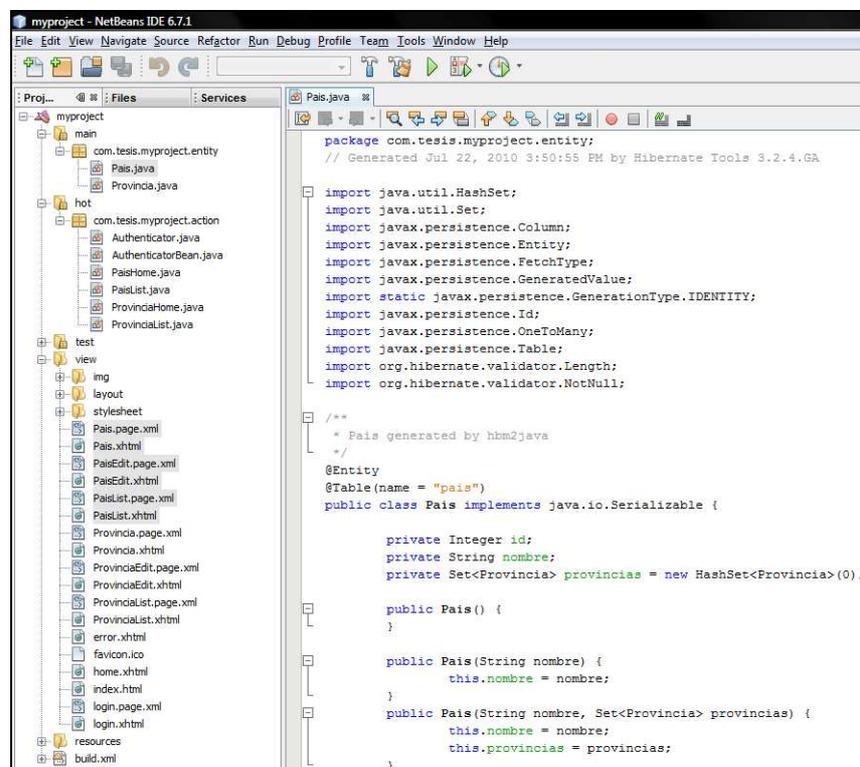
```
new-project :
BUILD SUCCESSFUL
Total time: 2 seconds

C:\Desarrollo\jboss-seam-2.2.0.GA>seam generate-entities
SEAM_HOME: C:\Desarrollo\jboss-seam-2.2.0.GA
Using seam-gen sources from: C:\Desarrollo\jboss-seam-2.2.0.GA\seam-gen
Buildfile: C:\Desarrollo\jboss-seam-2.2.0.GA\seam-gen\build.xml

init:
init-properties:
  [echo] C:/Desarrollo/jboss-5.1.0.GA
```

Figura 2.47: (Proyecto seam, ejecución del comando “seam generate-entities”)

Como se puede observar en la siguiente figura se ha creado el código de las entidades (Pais, Provincia), y los componentes Seam (PaisHome, PaisList, ProvinciaHome, ProvinciaList) que se encargan de la lógica de negocio, además se ha creado la parte web como las paginas para gestionar (crear, editar, eliminar, listar) país y provincia, reglas de navegación, página de login, menú de navegación, página de error.



```
package com.tesis.myproject.entity;
// Generated Jul 22, 2010 3:50:55 PM by Hibernate Tools 3.2.4.GA

import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
import org.hibernate.validator.Length;
import org.hibernate.validator.NotNull;

/**
 * Pais generated by hbm2java
 */
@Entity
@Table(name = "pais")
public class Pais implements java.io.Serializable {

    private Integer id;
    private String nombre;
    private Set<Provincia> provincias = new HashSet<Provincia>(0);

    public Pais() {
    }

    public Pais(String nombre) {
        this.nombre = nombre;
    }

    public Pais(String nombre, Set<Provincia> provincias) {
        this.nombre = nombre;
        this.provincias = provincias;
    }
}
```

Figura 2.48: (Proyecto seam, resultado del comando “seam generate-entities”)

Despliegue del proyecto

En el archivo “build.xml” que se encuentra en la raíz del proyecto ejecutar la tarea “deploy” con ayuda de Netbeans. Esta tarea se encargará de compilar el proyecto, empaquetarlo como un “.ear” y echarlo al directorio deploy del servidor JBoss. También copiará el datasource necesario para la conexión a la base de datos.

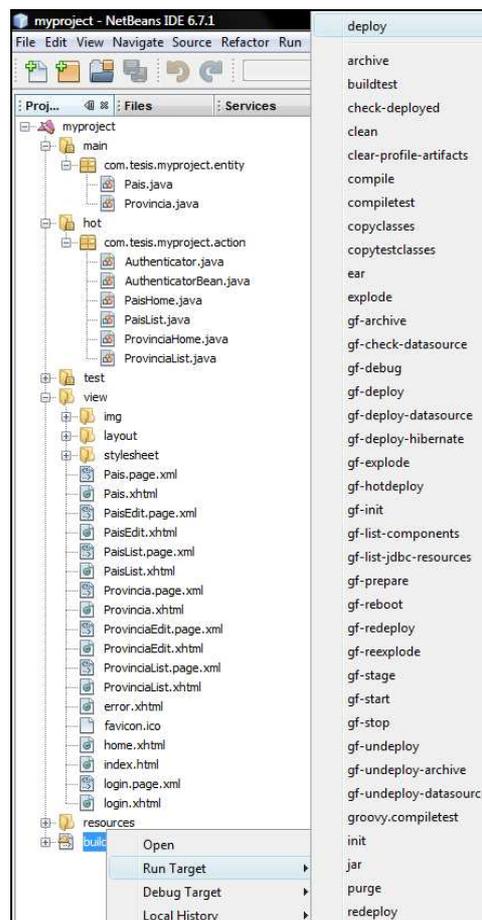
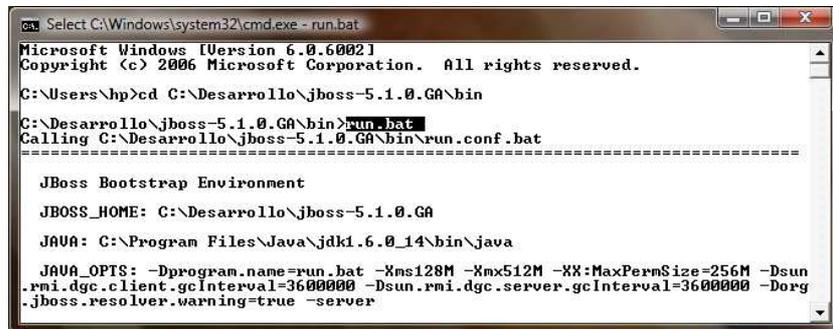


Figura 2.49: (Proyecto seam, ejecución de la tarea “deploy”)

Probar la aplicación

Se debe iniciar el servidor de aplicaciones JBoss, para esto en la consola de Windows ingresar al directorio donde se instaló (descomprimió) JBoss y ejecutar

el comando “run.bat”. Tener en cuenta que antes de iniciar el servidor se debe copiar el driver JDBC de MySQL en el directorio “server/default/lib” de JBoss.



```
cs. Select C:\Windows\system32\cmd.exe - run.bat
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Desarrollo\jboss-5.1.0.GA\bin
C:\Desarrollo\jboss-5.1.0.GA\bin>run.bat
Calling C:\Desarrollo\jboss-5.1.0.GA\bin\run.conf.bat
-----
JBoss Bootstrap Environment
JBOSS_HOME: C:\Desarrollo\jboss-5.1.0.GA
JAVA: C:\Program Files\Java\jdk1.6.0_14\bin\java
JAVA_OPTS: -Dprogram.name=run.bat -Xms128M -Xmx512M -XX:MaxPermSize=256M -Dsun
.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Dorg
.jboss.resolver.warning=true -server
```

Figura 2.50: (Proyecto seam, ejecución del comando “run.bat”)

Ingresar a la aplicación mediante la siguiente URL:
<http://localhost:8080/myproject/home.seam>.

Pantalla principal con menú de navegación, Login y Home.

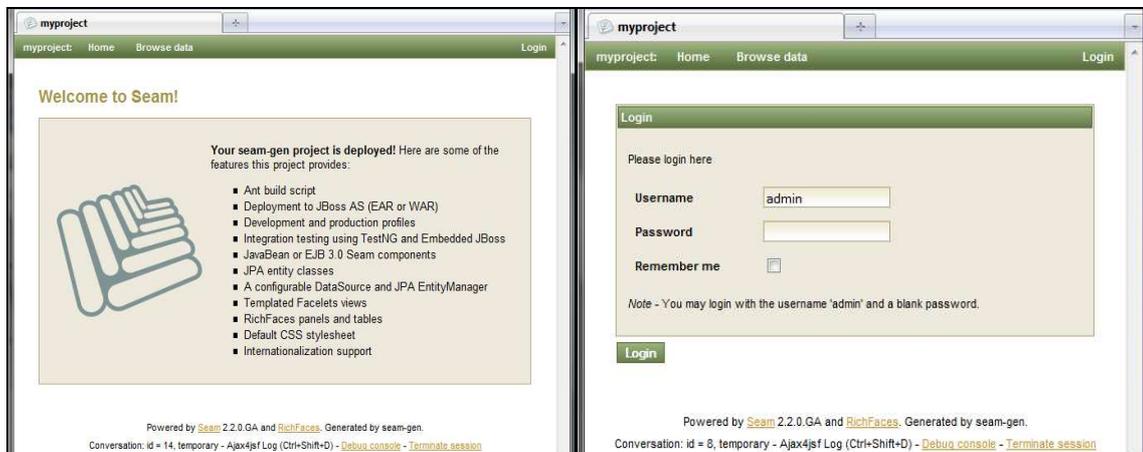


Figura 2.51: (Proyecto seam, Interfaz de la Página Principal y Login)

Para probar la aplicación se creará un país con una provincia mediante la opción “Pais List”.

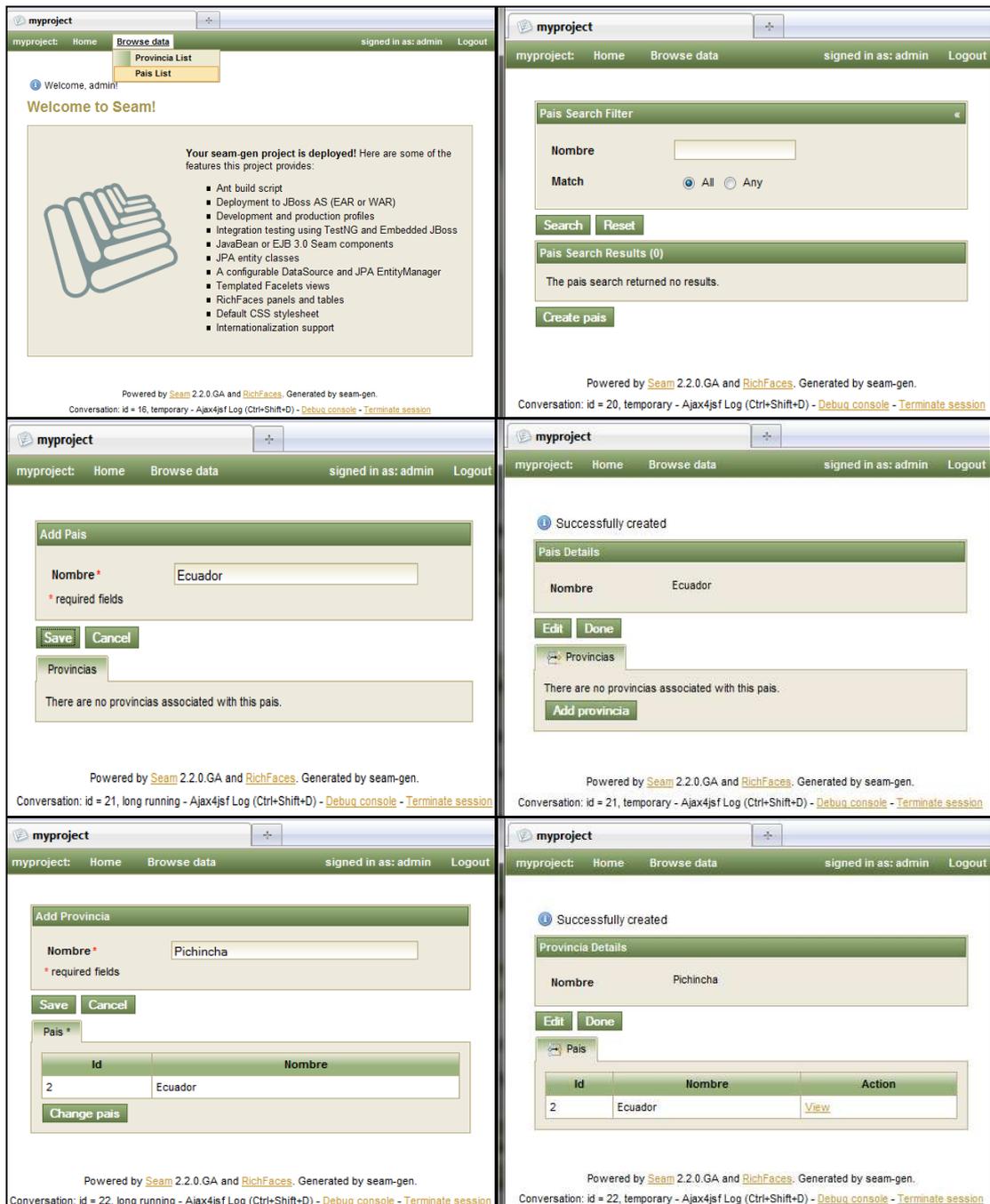


Figura 2.52: (Proyecto seam, Interfaces para Mantener Países y Provincias)

CAPÍTULO III

Desarrollo del Software

3.1 Análisis de Requerimientos

De acuerdo a la metodología la primera tarea es crear un “**modelo de dominio**”, pero antes se debe realizar un listado de requerimientos que son la fuente principal para elaborar este modelo. Recordar que para empezar con la primera tarea de ICONIX solo es necesario realizar un relevamiento informal de requerimientos funcionales.

3.1.1 Requerimientos del Software

3.1.1.1 Introducción

Empleando la “guía” (nombre que será empleado de aquí en adelante) mencionada en los antecedentes del presente proyecto se identificarán los diferentes requerimientos a implementarse. Además, se agregará si es necesario alguna funcionalidad propia del sistema (ingreso al sistema, validaciones de datos, etc.) u otra funcionalidad en particular que facilite la automatización de esta guía, y permita un uso intuitivo de la aplicación por parte del usuario; todo esto encaminado a desarrollar una aplicación que controle adecuadamente el proceso de desarrollo de software.

3.1.1.2 Requerimientos Funcionales

➤ Opciones de Administración del Sistema

- **R001:** Para utilizar el sistema el usuario debe iniciar sesión ingresando su nombre de usuario y contraseña.
- **R002:** El sistema permitirá al usuario cerrar sesión, esta acción permitirá al usuario salir del sistema.
- **R003:** El sistema permitirá manejar datos de cuentas de usuarios del sistema, datos como: nombre de usuario, clave, y rol (valores posibles de rol: Administrador, Líder del Proyecto, Analista, Cliente).
- **R004:** Al ingresar al sistema se mostrará una página principal con texto informativo del software, y las diferentes opciones disponibles para el usuario dependiendo del tipo de rol.

➤ Administración de Parámetros

Para la administración de parámetros u opciones generales, el sistema permitirá al usuario:

- **R005:** Manejar datos de cargos o roles, datos tales como: código y nombre.
- **R006:** Manejar datos de clientes, datos tales como: código, nombre y número telefónico.
- **R007:** Manejar datos de las personas que formarán parte del equipo de desarrollo de un proyecto, datos tales como: código, nombre, apellido,

cargo (rol), nivel sigma (puede ser una de las siguientes opciones: campeón, maestro cinturón negro, cinturón negro, cinturón verde, ninguna), y características (pueden ser: compromiso, sesgo de acción, flexibilidad, innovación, influencia sobre el personal, trabajo en equipo, tiempo disponible).

➤ **Administración de Proyectos**

Para la administración de proyectos el sistema permitirá al usuario:

- **R008:** Manejar datos iniciales de proyectos, datos tales como: código, nombre, cliente quien solicita el proyecto, y estado (Pendiente, En proceso, Finalizado, Cancelado). El estado inicial del proyecto es “Pendiente”.

Tener presente que la guía considera seis fases que forman parte del proceso de desarrollo de software, pero el presente software permitirá administrar las dos primeras fases que son: “Iniciación del proyecto” y “Análisis del Sistema”.

- **Administración de la Fase “Iniciación del Proyecto”**

Para administrar esta fase dentro de un proyecto, el sistema permitirá al usuario:

- **R009:** Mostrar información detallada del proyecto.
- **R010:** Actualizar datos adicionales al proyecto como: director, líder, fecha de recepción de la solicitud del nuevo sistema, fecha de inicio

planificada, fecha de inicio real, fecha de finalización planificada, y fecha de finalización real.

- **R011:** Manejar datos de las diferentes peticiones del cliente para el nuevo sistema.
- **R012:** Manejar datos de las diferentes razones por las cuales el cliente solicita el nuevo sistema.
- **R013:** Conformar el equipo de desarrollo del proyecto, agregando o quitando personas de desarrollo.
- **R014:** Manejar datos de los factores críticos del proyecto ya sean de éxito o riesgo.
- **R015:** Manejar datos de las diferentes metas identificadas en un proyecto con su respectivo análisis. El análisis de las metas consiste en describir una justificación por cada una de las siguientes características: Específica, Medurable, Alcanzable y Relevante.
- **R016:** Manejar datos de los diferentes procesos que se automatizarán en el proyecto de desarrollo, datos tales como: código, nombre, alcance o propósito, y una breve descripción del proceso.
- **R017:** Por cada proceso manejar datos de sus respectivas actividades como: código y descripción.
- **R018:** Manejar datos de los actores o participantes (internos o externos) que serán los responsables de ejecutar las diferentes

actividades, datos tales como: código, nombre y tipo (interno o externo).

- **R019:** Asignar responsables a las actividades con el fin de identificar las personas encargadas de cada actividad.
- **R020:** Definir por cada proceso sus entradas, sus salidas, los proveedores de las entradas, y los clientes de las salidas. Esto proveerá un nivel de comprensión macro sobre los procesos que requiere el nuevo sistema.
- **R021:** Mostrar un cuestionario como ayuda para abstraer los requerimientos del cliente.
- **R022:** Manejar datos de los diferentes requerimientos del cliente, datos tales como: código, descripción, rango de importancia, rango de satisfacción actual, y estado (puede ser: Pendiente, Aceptado, o Negado). El estado inicial de un requerimiento será "Pendiente".

- **Administración de la Fase "Análisis del Sistema"**

Para administrar esta fase dentro de un proyecto el sistema permitirá al usuario:

- **R023:** Mostrar información actual de cada requerimiento.
- **R024:** Identificar los requerimientos del cliente que fueron creados en la fase anterior. La identificación consiste en agregar datos adicionales a cada requerimiento, datos tales como: tipo (funcional, no funcional), fuente de obtención, descripción de alguna

justificación, observación, y métodos de obtención aplicados para obtener el requerimiento (pueden ser: Encuesta, Focus Group, Entrevistas Individuales, Sitios visitados, Quejas de los Clientes, Retornos). Tener presente que en esta parte el estado del requerimiento puede cambiar a “Aceptado” o “Negado”.

- **R025:** Por cada requerimiento registrar los diferentes conflictos que se pretenden resolver.
- **R026:** Por cada requerimiento registrar criterios de aceptación. Esta parte permite describir las razones por las cuales se “Aceptó” o “Negó” la implementación del requerimiento.
- **R027:** Registrar las posibles mejoras a los requerimientos.
- **R028:** Priorizar los requerimientos aceptados.
- **R029:** Por cada fase del proceso de desarrollo de software, el sistema permitirá el manejo de anexos o archivos, los cuales contendrán un código, nombre, descripción, y el archivo en cualquier formato (Ejemplo: .pdf, .xls, .docx, entre otros).

➤ **Generación de Reportes**

➤ **R030:** El sistema mostrará diferentes reportes en base a los siguientes formatos:

Ficha del Proyecto
<p>Proyecto</p> <p>Código: Nombre: Director: Líder: Estado: Fecha de recepción: Fecha de inicio planificada: Fecha de inicio efectivo: Fecha fin planificado: Fecha fin efectiva:</p>
<p>Cliente</p> <p>Código: Nombre: Número telefónico: Listado de Cosas que le gustaría hacer Descripción:</p> <ul style="list-style-type: none"> • <descripción 1> • <descripción 2> <p>Razones por las cuales solicita la realización del sistema Descripción:</p> <ul style="list-style-type: none"> • <descripción 1> • <descripción 2>
<p>Factores críticos de éxito y riesgo</p> <p>De éxito</p> <ul style="list-style-type: none"> • <descripción 1> • <descripción 2> <p>De riesgo</p> <ul style="list-style-type: none"> • <descripción 1> • <descripción 2>

Tabla 3.1: (Plantilla para el reporte “Ficha del Proyecto”)

Información del Equipo de Desarrollo
<p>Proyecto: <nombre del proyecto></p>
<p><apellidos y nombres de la persona de desarrollo 1></p> <p>Código: Rol: Nivel Sigma: Características:</p> <ul style="list-style-type: none"> ➤ <característica 1.1> ➤ <característica 1.2>
<p><apellidos y nombres de la persona de desarrollo 2></p> <p>Código: Rol: Nivel Sigma: Características:</p> <ul style="list-style-type: none"> ➤ <característica 2.1> ➤ <característica 2.2>

Tabla 3.2: (Plantilla para el reporte “Información del Equipo de Desarrollo”)

Metas establecidas para el Proyecto	
Proyecto: <nombre del proyecto>	
Meta: <descripción de la meta>	
Características	Justificación
Específica	<descripción justificación>
Mesurable	<descripción justificación>
Alcanzable	<descripción justificación>
Relevante	<descripción justificación>

Tabla 3.3: (Plantilla para el reporte “Metas establecidas para el Proyecto”)

Información de los Procesos del Negocio	
Proyecto: <nombre del proyecto>	
Proceso: <nombre del proceso1>	
Actividad: <descripción de la actividad 1.1>	
Responsables de la Actividad	
Nombre	Tipo
<nombres completos del responsables>	<tipo de responsable>
Actividad: <descripción de la actividad 1.2>	
Responsables de la Actividad	
Nombre	Tipo
<nombres completos del responsables>	<tipo de responsable>
Proceso: <nombre del proceso 2>	
Actividad: <descripción de la actividad 2.1>	
Responsables de la Actividad	
Nombre	Tipo
<nombres completos del responsables>	<tipo de responsable>

Tabla 3.4: (Plantilla para el reporte “Información de los Procesos del Negocio”)

Caracterización de Procesos	
Proyecto: <nombre del proyecto>	
Proceso: <<nombre del proceso>>	
Propósito / Alcance: <descripción alcance del proceso>	
Descripción: <descripción del proceso>	
Proveedores	
>	<proveedor 1>
>	<proveedor 2>
Entradas	
>	<entrada 1>
>	<entrada 2>
Salidas	
>	<salida 1>
>	<salida 2>
Clientes	
>	<cliente 1>
>	<cliente 2>

Tabla 3.5: (Plantilla para el reporte “Caracterización de Procesos”)

Matriz de Requerimientos del Cliente			
Proyecto: <nombre del proyecto>			
Cliente: <nombre del cliente>			
Código	Requerimiento	Rango de importancia	Rango de satisfacción actual
<código 1>	<descripción 1>	<rango 1>	<rango 1>
<código 2>	<descripción 2>	<rango 2>	<rango 2>
<código 3>	<descripción 3>	<rango 3>	<rango 3>

Tabla 3.6: (Plantilla para el reporte “Matriz de Requerimientos del Cliente”)

Registro de Requerimientos	
Proyecto: <nombre del proyecto>	
Requerimiento: <descripción del requerimiento>	
Código:	
Estado:	
Tipo:	
Fuente:	
Justificación:	
Observación:	
Métodos de Obtención:	
➤ <método 1>	
➤ <método 2>	
Conflictos por resolver:	
➤ <conflicto 1>	
➤ <conflicto 2>	

Tabla 3.7: (Plantilla para el reporte “Registro de Requerimientos”)

Criterios de Aceptación de Requerimientos	
Proyecto: <nombre del proyecto>	
Cliente: <nombre del cliente>	
Requerimiento: <descripción del requerimiento 1>	
Código:	
Estado:	
Criterios de Aceptación:	
➤ <criterio 1.1>	
➤ <criterio 1.2>	
Requerimiento: <descripción del requerimiento 2>	
Código:	
Estado:	
Criterios de Aceptación:	
➤ <criterio 2.1>	
➤ <criterio 2.2>	

Tabla 3.8: (Plantilla para el reporte “Criterios de Aceptación de Requerimientos”)

Matriz de Priorización de Requerimientos del Cliente				
Proyecto: <nombre del proyecto>				
Cliente: <nombre del cliente>				
Prioridad	Código	Requerimiento	Rango de importancia	Rango de satisfacción actual
1	<código 1>	<descripción 1>	<rango 1>	<rango 1>
2	<código 2>	<descripción 2>	<rango 2>	<rango 2>
3	<código 3>	<descripción 3>	<rango 3>	<rango 3>

Tabla 3.9: (Plantilla para el reporte “Matriz de Priorización de Requerimientos del Cliente”)

Evaluación Costo / Tiempo de Mejoras		
Proyecto: <nombre del proyecto>		
Cliente: <nombre del cliente>		
Requerimiento: <descripción del requerimiento>		
Mejora	Costo	Tiempo
<mejora 1>	<costo 1>	<tiempo 1>
<mejora 2>	<costo 2>	<tiempo 2>
<mejora 3>	<costo 3>	<tiempo 3>

Tabla 3.10: (Plantilla para el reporte “Evaluación Costo / Tiempo de Mejoras”)

Cuestionario de Requerimientos del Cliente	
Estructura organizacional y objetivos	
1.	¿Cuál es su rol o responsabilidad?
2.	¿Cómo es la estructura organizacional donde se aplicará el proyecto?
3.	¿Cuáles son los objetivos principales de la organización que requiere el sistema?
4.	¿Cómo usted sabe que los han alcanzado?
5.	¿Qué problemas ah tenido en el alcance de los objetivos durante el año pasado?
6.	¿Qué nuevos obstáculos cree usted que impedirían la consecución de los objetivos?
7.	¿Cómo es la medición del rendimiento de producción en la organización?
8.	¿Qué otras mediciones realiza y reporta?
Factores críticos de éxito	
1.	¿Cuáles son las decisiones que usted espera tomar?
2.	Si ocurrieran errores, ¿Cuál podría afectar gravemente a sus procesos, y a la organización?
3.	Si usted pudiera tener solamente tres piezas de información, ¿Con cuál juzgaría usted el rendimiento diario de producción del proceso?
Proceso actual	
1.	Describa el flujo actual de trabajo del proceso
2.	¿Dónde están los retrasos?
3.	¿Usted mide el costo por transacción? En caso afirmativo, ¿Qué es?
Sistema actual	
1.	En una escala del 1 al 10, ¿Cuán crítico es el sistema actual para el éxito del proceso?
2.	¿Cuál podría ser el impacto en su organización, si el sistema no funciona?
3.	En una escala del 1 al 10, ¿Qué tan bien el actual sistema satisface las necesidades de su organización?
4.	¿Cuál es la información más usual que usted recibe ahora?
5.	¿Cuál es la menos usual?
6.	¿Qué información necesita usted que actualmente no es provista por el actual sistema?
7.	En una escala del 1 al 10, ¿Cómo podría tasar la información que usted recibe en cada una de las siguientes categorías?: <ul style="list-style-type: none"> ➤ Puntualidad. ➤ Exactitud. ➤ Exhaustividad. ➤ Facilidad de uso/acceso. ➤ Costo.
8.	Si usted pudiera cambiar tres cosas sobre el sistema actual. ¿Qué sería?

Tabla 3.11: (Plantilla para el reporte “Cuestionario de Requerimientos del Cliente”)

3.1.2 Modelado del dominio

3.1.2.1 Listado de posibles objetos o clases de dominio

- **Cargo:** rol que desempeña una persona de desarrollo.
- **Cliente:** persona (natural o jurídica) que solicita el desarrollo del software.
- **Persona Desarrollo:** integrante de un equipo de desarrollo de un proyecto.
- **Característica Persona:** criterio de valoración de una persona de desarrollo.

- **Proyecto:** identificación de un proceso de desarrollo de software.
- **Petición:** lo que solicita el cliente.
- **Razón:** motivo por el cual el cliente solicita el nuevo sistema.
- **Equipo de Desarrollo:** o asignación de persona de desarrollo, conjunto de personas que forman parte de un proyecto.
- **Factor Crítico:** motivo de éxito o fracaso de un proyecto.
- **Meta:** lo que se desea alcanzar en un proyecto.
- **Análisis Meta:** justificación de una meta.
- **Proceso:** o proceso de negocio, es lo que se automatizará en el proyecto de desarrollo.
- **Actividad:** forma parte de un proceso.
- **Responsable de Actividad:** persona encargada de ejecutar una o varias actividades.
- **Requerimiento:** petición del cliente a ser desarrollada, previo a un análisis.
- **Método de Obtención:** técnica que fue empleada para obtener un requerimiento.
- **Criterio de Aceptación:** justificación por la cual se acepta o niega la realización de un requerimiento.
- **Posible Mejora:** posible perfeccionamiento a un requerimiento.

3.1.2.2 Modelo de Dominio Inicial

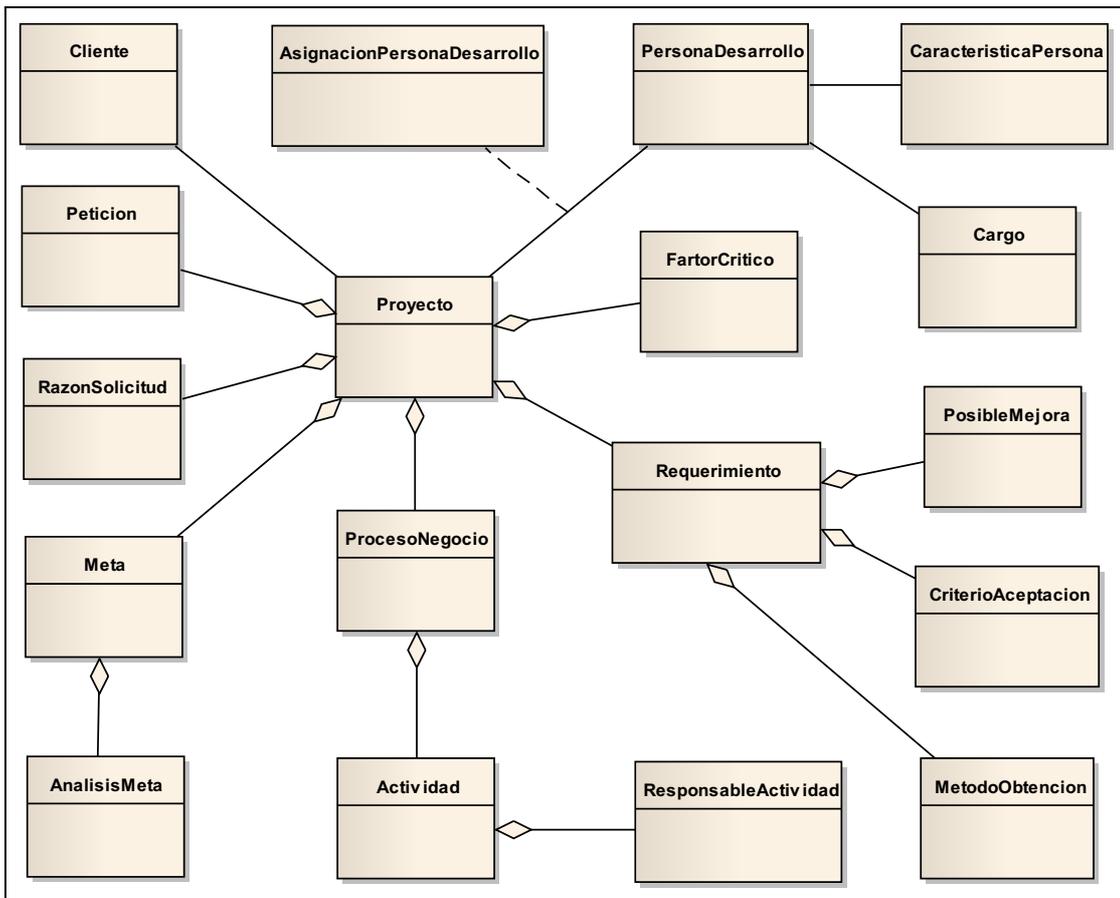


Figura 3.1: (Modelo de dominio inicial)

3.1.3 Modelado de Casos de Uso

3.1.3.1 Prototipos de Interfaz de usuario

Con el fin de facilitar la identificación de los casos de uso se realizaron diferentes prototipos de interfaz de usuario, como se pueden observar en las siguientes figuras:

<<Contenido de la cabecera del sistema>>						
Home	Proyectos	Reportes	Parámetros Globales	Bienvenido: <<Usuario>>		Administración
<<Ruta de la opción seleccionada>>		<<Nombre opción>>		<<Nombre opción>>		
		<<Nombre opción>>		<<Nombre opción>>		
		<<Nombre opción>>		<<Nombre opción>>		
<<Contenido de la opción seleccionada>>						
<<Contenido del pie del sistema>>						

Figura 3.2: (Página Principal)

<<Contenido de la cabecera del sistema>>							
Home	Proyectos	Reportes	Parámetros Globales	Bienvenido: <<Usuario>>		Administración	
Iniciar sesión							
<table border="1" style="margin: auto;"> <tr> <td style="padding: 5px;"> Usuario: <input style="width: 150px;" type="text"/> Contraseña: <input style="width: 150px;" type="password"/> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 5px;"> <input type="button" value="Ingresar"/> <input type="button" value="Cancelar"/> </div> </td> </tr> </table>							Usuario: <input style="width: 150px;" type="text"/> Contraseña: <input style="width: 150px;" type="password"/> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 5px;"> <input type="button" value="Ingresar"/> <input type="button" value="Cancelar"/> </div>
Usuario: <input style="width: 150px;" type="text"/> Contraseña: <input style="width: 150px;" type="password"/> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 5px;"> <input type="button" value="Ingresar"/> <input type="button" value="Cancelar"/> </div>							
<<Contenido del pie del sistema>>							

Figura 3.3: (Página de Inicio de Sesión)

Páginas de Manejo de Datos

Estas interfaces se mostrarán en la sección “Contenido de la opción seleccionada” de la Página Principal, y pueden ser utilizadas por otras opciones similares.

Al ingresar a la respectiva opción se mostrará la siguiente interfaz:

Ruta opción >> Nombre de la opción seleccionada

Filtros de Búsqueda

Dato 1:

Dato 2:

Opciones

Dato 1	Dato 2	Editar	Eliminar
<<valor>>	<<valor>>	link	link
<<valor>>	<<valor>>	link	link
<<valor>>	<<valor>>	link	link
<<valor>>	<<valor>>	link	link
<<valor>>	<<valor>>	link	link

<< < Paginador > >>

Figura 3.4: (Página de Manejo de Datos)

La opción “Buscar” permitirá filtrar la lista de datos, en base a los filtros de búsqueda previamente ingresados; la opción “Resetear Búsqueda” permitirá quitar los filtros de búsqueda; la opción “Crear” permitirá ingresar nuevos datos; las opciones por cada registro permitirán “Editar” y “Eliminar” los respectivos datos; además, al final de la lista de datos se mostrará un paginador (10 registros). A continuación se muestra la interfaz de las diferentes opciones.

Ruta opción >> Nombre de la opción seleccionada >> Crear

Crear <<nombre que identifique los datos>>

Dato 1:

Dato 2:

Opciones

Figura 3.5: (Página de Creación de Datos)

Ruta opción >> Nombre de la opción seleccionada >> Editar

Editar <<nombre que identifique los datos>>	
Dato 1:	<<valor>>
Dato 2:	<<valor>>
Opciones	
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	

Figura 3.6: (Página de Edición de Datos)

Advertencia	
¿Está seguro que desea eliminar: <<dato>>?	
<input type="button" value="Confirmar"/>	<input type="button" value="Cancelar"/>

Figura 3.7: (Ventana Emergente de Eliminación de Datos)

Página de Administración de Proyectos

Para ingresar a esta página el usuario debe ingresar al enlace “Proyectos” de la Página Principal, y luego ingresar a la opción “Administrar” de la lista de Proyectos.

<<Contenido de la cabecera del sistema>>						
Home	Proyectos	Reportes	Parámetros Globales	Bienvenido: <<Usuario>>	Administración	Iniciar sesión / Cerrar sesión
Usted está administrando el Proyecto :: <<Nombre del Proyecto Seleccionado>>						
Fases de Desarrollo:		Iniciación del Proyecto		Análisis del Sistema		
		<<Opción 1>>	<<Opción 1>>			
		<<Opción 2>>	<<Opción 2>>			
		<<Opción n>>	<<Opción n>>			
<<Contenido de la opción seleccionada>>						
<<Contenido del pie del sistema>>						

Figura 3.8: (Página de Administración de Proyectos)

Páginas de Conformación del Equipo de Desarrollo

Al ingresar a la opción “Iniciación del Proyecto/Equipo de Desarrollo” de la Página de Administración de Proyectos se mostrará la siguiente interfaz:

Iniciación del Proyecto >> Equipo de Desarrollo

Filtros de Búsqueda

Código:

Nombres:

Apellidos:

Opciones

Código	Nombres	Apellidos	Rol	Eliminar
1	Soledad	Espín	Analista	link
9	Juan	Delgado	Programador	link
3	Pedro	Zapata	Programador	link

<< < Paginador > >>

Figura 3.9: (Página de Manejo del Equipo de Desarrollo)

Mediante la opción “Conformar equipo” (ver figura anterior) se accederá a la página para agregar o quitar integrantes al Equipo de Desarrollo, como se muestra en la siguiente figura.

Iniciación del Proyecto >> Equipo de Desarrollo >> Conformar Equipo

Personas de Desarrollo

Personas de Desarrollo asignadas

Código	Nombres	Apellidos	Rol	Quitar
1	Javier	Espín	Analista	opción
9	Juan	Delgado	Programador	opción
3	Pedro	Zapata	Programador	opción

<< < Paginador > >>

Opciones

Figura 3.10: (Página de Conformación del Equipo de Desarrollo)

La opción “Quitar” permitirá eliminar una persona de la lista de integrantes del Equipo de Desarrollo; la opción “Agregar” permitirá acceder a la página de búsqueda de una persona de desarrollo que formará parte del Equipo de Desarrollo de un Proyecto, como se observa en la siguiente figura.

Iniciación del Proyecto >> Equipo de Desarrollo >> Buscar Persona de Desarrollo

Filtros de Búsqueda

Código:

Nombres:

Apellidos:

Opciones

Código	Nombres	Apellidos	Rol	Seleccionar
1	Ana	Sandoval	Ing. Procesos	link
9	Santiago	Castro	Arq. Software	link
3	Geovanny	Maila	Analista	link

<< < Paginador > >>

Figura 3.11: (Página de Selección de Personas de Desarrollo)

Página para Crear o Editar una Persona de Desarrollo

Parámetros Generales >> Personas de Desarrollo >> Crear o Editar

<<Crear o Editar>> Persona de Desarrollo

Código:

Nombres:

Apellidos:

Rol: ▾

Nivel Sigma:

- Ninguna
- Campeón
- Maestro Cinturón Negro
- Cinturón Negro
- Cinturón Verde

Características

- Compromiso
- Sesgo de Acción
- Flexibilidad
- Innovación
- Influencia sobre el personal
- Trabajo en Equipo
- Tiempo Disponible

Opciones

Figura 3.12: (Página de Creación o Edición de una Persona de Desarrollo)

Página para Mostrar y Editar Información del Proyecto

Iniciación del Proyecto >> Información del Proyecto

Proyecto	
Código:	<<Código del Proyecto>>
Proyecto:	<<Nombre del Proyecto>>
Cliente:	<<Nombre del Cliente>>
Estado:	<<Estado del Proyecto>>
Director:	<input type="text"/>
Líder:	<input type="text"/>
Fecha de recepción:	<input type="text" value="Dd/mm/aaaa"/> <input type="checkbox"/>
Fecha de inicio planificada:	<input type="text" value="Dd/mm/aaaa"/> <input type="checkbox"/>
Fecha efectiva:	<input type="text" value="Dd/mm/aaaa"/> <input type="checkbox"/>
Fecha de finalización planificada:	<input type="text" value="Dd/mm/aaaa"/> <input type="checkbox"/>
Fecha de finalización real:	<input type="text" value="Dd/mm/aaaa"/> <input type="checkbox"/>

Opciones

Figura 3.13: (Página para Mostrar y Editar Información del Proyecto)

Página para Crear o Editar Metas con su respectivo Análisis

Iniciación del Proyecto >> Análisis de Metas >> Crear o Editar

<<Crear o Editar>> Meta	
Código:	<input type="text"/>
Descripción:	<input type="text"/>

Análisis de Meta	
Característica	Justificación
Específica	Justificación: <input type="text"/>
Mesurable	Justificación: <input type="text"/>
Alcanzable	Justificación: <input type="text"/>
Relevante	Justificación: <input type="text"/>

Opciones

Figura 3.14: (Página de Creación o Edición de Meta)

Páginas de Asignación de Responsables por Actividad

Mediante la opción “Identificación de Responsables por Actividad” se accederá a la página que muestra la cantidad de Responsables asignados a cada actividad. La página con sus respectivas opciones se ilustra en la siguiente figura.

Iniciación del Proyecto >> Identificación de Responsables por Actividad

Filtros de Búsqueda

Proceso del Negocio: ▾

Opciones

Proceso del Negocio	Actividad	Cant. Responsables Asignados	Responsables
Proceso 1	Actividad 1	2	link
Proceso 1	Actividad 2	0	link
Proceso 2	Actividad 3	1	link

<< < Paginador > >>

Figura 3.15: (Página de Manejo de Responsables por Actividad)

La opción “Responsables” (ver figura anterior) permitirá acceder a la página para asignar responsables a una Actividad seleccionada, como se muestra en la siguiente figura.

Iniciación del Proyecto >> Identificación de Responsables por Actividad >> Asignar Responsables

Actividad Seleccionada

Proceso del Negocio: <<nombre>>
Actividad: <<descripción>>

Responsables

Responsables Asignados

Código	Nombres Completos	Quitar
1	Luz María Espín Atiaja	opción
9	José Miguel Yacchirema Jiménez	opción
3	Juan Carlos Jácome Moreno	opción

Opciones

Figura 3.16: (Página de Asignación de Responsables a una Actividad)

Páginas de Administración de Requerimientos

Para manejar (listar, buscar, guardar, editar, eliminar) los datos asociados a un Requerimiento, el usuario debe seleccionar un Requerimiento mediante la opción “Administrar” como se muestra en la Figura 3.20. Posteriormente el sistema mostrará el requerimiento seleccionado con sus respectivas opciones en la Página de Administración de Requerimientos (ver Figura 3.21).

Análisis del Sistema >> Análisis de Requerimientos

Filtros de Búsqueda

Código:

Descripción:

Rango de importancia:

Rango de satisfacción actual:

Estado:

Opciones

Código	Descripción	Rango de importancia	Rango de satisfacción Actual	Estado	Administrar
<<código>>	<<descripción>>	<<valor>>	<<valor>>	<estado>	link
<código>	<<descripción>>	<<valor>>	<<valor>>	<estado>	link

<< < Paginador > >>

Figura 3.17: (Página de Selección de un Requerimiento)

<<Contenido de la cabecera del sistema>>						
Home	Proyectos	Reportes	Parámetros Globales	Bienvenido: <<Usuario>>	Administración	Iniciar sesión / Cerrar sesión
<<Nombre del Proyecto seleccionado>>						
Fases de Desarrollo:		Iniciación del Proyecto	Análisis del Sistema			
<<Nombre del Requerimiento seleccionado>>						
Información del Requerimiento		<<Ruta de la opción seleccionada>> <<Contenido de la opción seleccionada>>				
Datos Adicionales						
Conflictos por Resolver						
Criterios de Aceptación						
Posibles Mejoras						
<<Contenido del pie del sistema>>						

Figura 3.18: (Página de Administración de Requerimientos)

Página de Actualización de Datos Adicionales al Requerimiento

Al ingresar a la opción “Datos Adicionales” de la Página de Administración de Requerimientos, el sistema mostrará una página (ver Figura 3.22) que permitirá al usuario adicionar datos al requerimiento previamente seleccionado.

Análisis del Sistema >> Análisis de Requerimientos >> Datos Adicionales

Datos Adicionales

Tipo de Requerimiento: ▾

Fuente:

Justificación:

Observación:

Implementar Requerimiento: SI NO

Métodos de obtención aplicados

- Encuesta
- Focus Group
- Entrevistas Individuales
- Sitios visitados
- Quejas de los Clientes
- Retornos

Opciones

Figura 3.19: (Página de Actualización de datos adicionales al Requerimiento)

Página para Mostrar Información de un Requerimiento

Dentro de las opciones de la Página de Administración de Requerimientos se dispondrá de un enlace (Información del Requerimiento) que permitirá al usuario observar con detalle el estado actual de un determinado requerimiento. Los datos que se observarán se ilustran en la siguiente figura.

Análisis del Sistema >> Análisis de Requerimientos >> Información del Requerimiento

Requerimiento	
Código:	<<valor>>
Descripción:	<<valor>>
Rango de Importancia:	<<valor>>
Rango de satisfacción actual:	<<valor>>
Prioridad:	<<valor>>
Estado:	<<valor>>
Tipo:	<<valor>>
Fuente:	<<valor>>
Justificación:	<<valor>>
Observación:	<<valor>>

Figura 3.20: (Página para Mostrar Información de un Requerimiento)

Páginas de Manejo de Anexos

Mediante la opción “Iniciación del Proyecto/Anexos” o “Análisis del Sistema/Anexos” de la Página de Administración de Proyectos se accederá a la siguiente interfaz.

<<Fase de Desarrollo >> >> Anexos

Filtros de Búsqueda					
Código:	<input type="text"/>				
Nombre del Archivo:	<input type="text"/>				
Opciones					
<input type="button" value="Buscar"/>		<input type="button" value="Resetear Búsqueda"/>		<input type="button" value="Crear"/>	
Código	Nombre del Archivo	Descripción	Descargar	Editar	Eliminar
A00RI	Requerimientos_Iniciales.xlsx	Lista de Requerimientos	link	link	link
A00RD	Requerimientos_Diseño.docx	Diseño propuesto por el Cliente	link	link	link
A00MU	Manual_Usuario.pdf	Manual del Sistema anterior	link	link	link

<< < Paginador >>

Figura 3.21: (Página de Manejo de Anexos)

La opción “Descargar” permitirá obtener el respectivo archivo, y la opción “Crear” permitirá acceder a la siguiente interfaz.

<<Fase de Desarrollo>> >> Anexos >> Crear o Editar

<<Crear o Editar>> Anexo

Código:

Descripción:

Archivo:

<<Archivo cargado>>

Opciones

Figura 3.22: (Página de Creación o Edición de Anexo)

Páginas de Generación de Reportes

Mediante la opción “Reportes” de la Página Principal se accederá a la página que permitirá seleccionar un Proyecto con el fin de obtener sus respectivos reportes.

Reportes

Filtros de Búsqueda

Código:

Proyecto:

Opciones

Código	Proyecto	Cliente	Estado	Reportes
POSRI	Sistema de RR-HH	SRI	Pendiente	link
POSPX	Migración del Sistema de Ventas	Supermaxi	En Proceso	link

<<< Paginador >>>

Figura 3.23: (Página de Reportes)

La opción “Reportes” mostrará una ventana emergente que permitirá descargar un determinado reporte en formato de documento portátil (PDF). Los reportes disponibles por Proyecto se muestran en las siguientes interfaces.

Seleccionar Reporte	
Proyecto: <<nombre del proyecto>>	
Reporte	Descargar
Ficha del Proyecto	link
Equipo de Desarrollo	link
Metas establecidas para el Proyecto	link
Procesos del Negocio	link
Caracterización de Procesos	link
Matriz de Requerimientos del Cliente	link
<< < 1 - 2 > >>	
<input type="button" value="Aceptar"/>	

Figura 3.24: (Ventana Emergente de Selección de un Reporte – Parte I)

Seleccionar Reporte	
Proyecto: <<nombre del proyecto>>	
Reporte	Descargar
Registro de Requerimientos	link
Criterios de Aceptación de Requerimientos	link
Matriz de Priorización de Requerimientos del Cliente	link
Evaluación Costo / Tiempo de Mejoras	link
Cuestionario de Requerimientos del Cliente	link
<< < 1 - 2 > >>	
<input type="button" value="Aceptar"/>	

Figura 3.25: (Ventana Emergente de Selección de un Reporte – Parte II)

Página de Priorización de Requerimientos

Mediante la opción “Análisis del Sistema/Priorización de Requerimientos” de la Página de Administración de Proyectos se accederá a la página (ver Figura 3.26) que permite priorizar los requerimientos aceptados.

Análisis del Sistema >> Priorización de Requerimientos						
Opciones						
<input type="button" value="Guardar"/>						
Prioridad	Código	Descripción	Rango de importancia	Rango de satisfacción actual	Ordenar Prioridad	
					Subir	Bajar
1	<código>	<requerimiento>	<rango>	<rango>		opción
2	<código>	<requerimiento>	<rango>	<rango>	opción	opción
3	<código>	<requerimiento>	<rango>	<rango>	opción	opción
4	<código>	<requerimiento>	<rango>	<rango>	opción	
<< < Paginador > >>						

Figura 3.26: (Página de Priorización de Requerimientos)

3.1.3.2 Diagrama de Casos de Uso

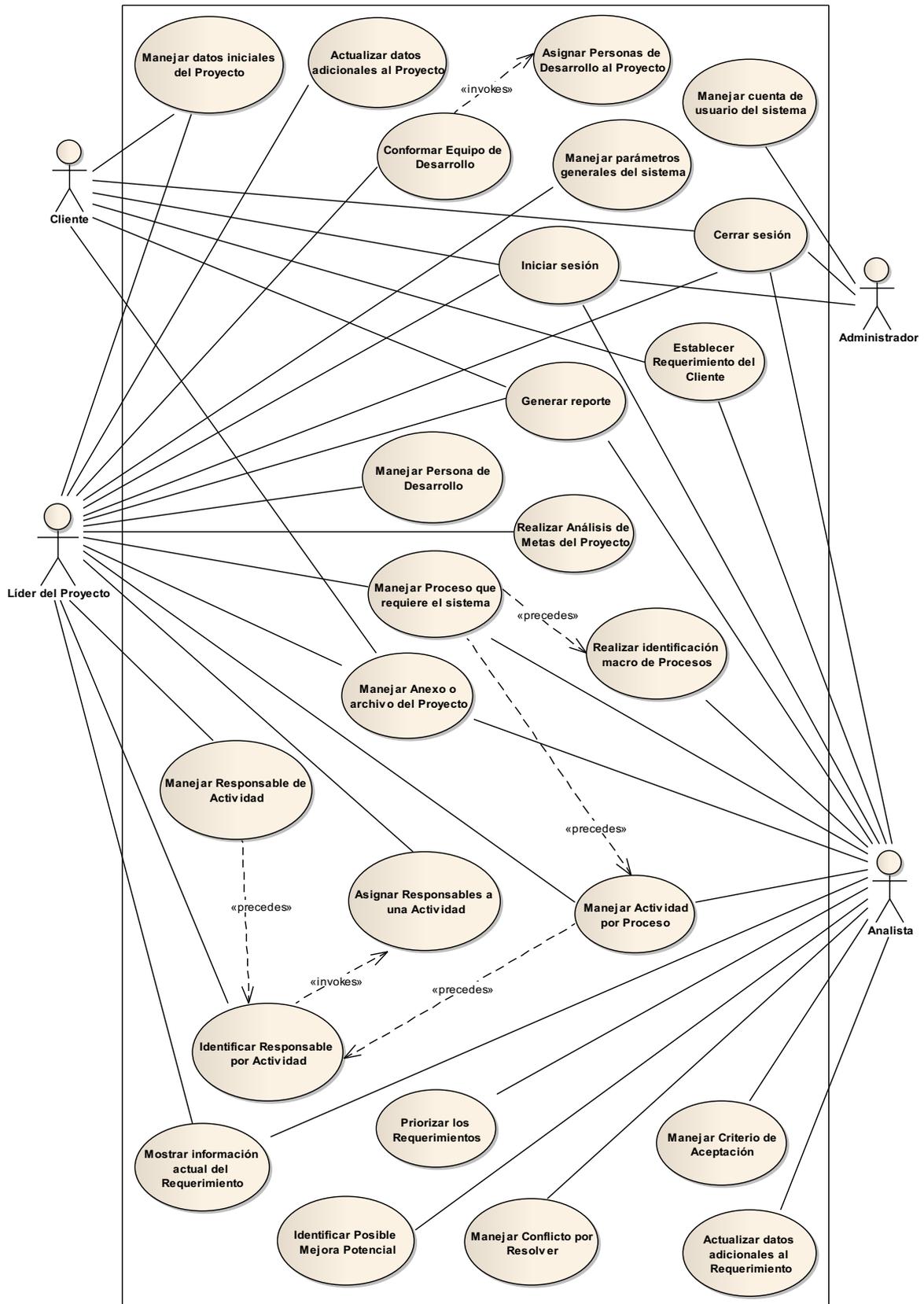


Figura 3.27: (Modelo de Casos de Uso – Parte I)



Figura 3.28: (Modelo de Casos de Uso – Parte II)

3.1.3.3 Diagrama de Paquetes

Este diagrama fue realizado con el fin de agrupar y organizar los diferentes casos de uso, los cuales fueron identificados de acuerdo a los requerimientos establecidos por el usuario, además la mayoría de los paquetes han sido nombrados en base a las opciones u sub-módulos que dispondrá el software a desarrollarse.

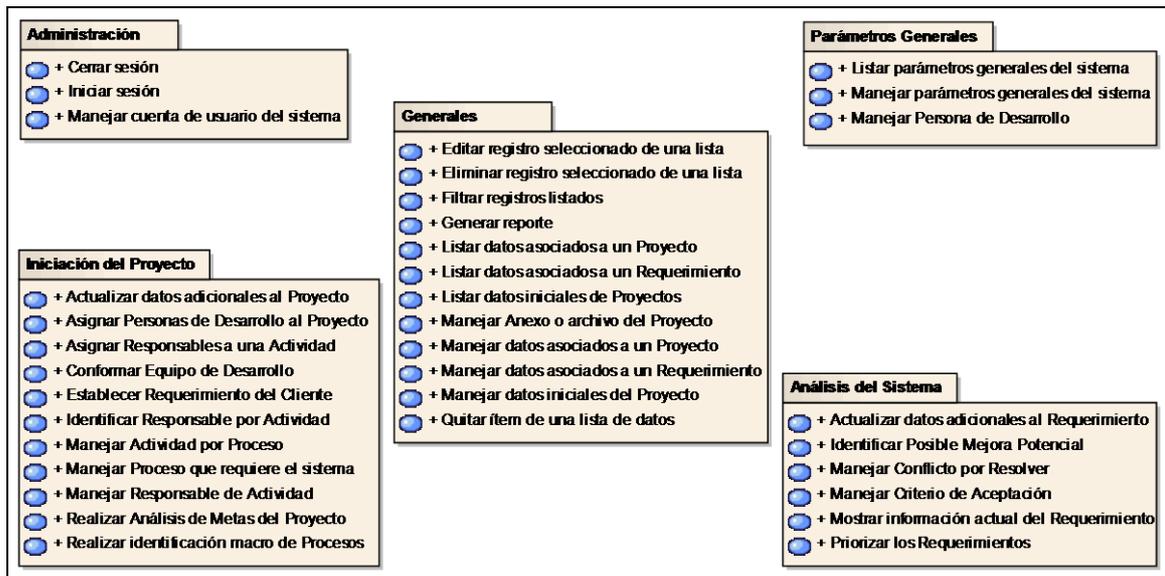


Figura 3.29: (Diagrama de Paquetes)

Esta fase (Análisis de Requisitos) finaliza definiendo la interacción del usuario con el sistema, es decir, escribir el primer proyecto de casos de uso (o especificación de casos de uso) que será utilizado y refinado en la siguiente fase (Análisis y Diseño Preliminar).

3.2 Análisis y Diseño Preliminar

3.2.1 Especificación de casos de uso

Tener en cuenta que la presente especificación de casos de uso fue iniciada en la fase de “Análisis de Requisitos” y refinada en esta fase mediante el análisis de robustez (ver sección 3.2.2 Análisis de Robustez).

3.2.1.1 Especificación de cada caso de uso

En la siguiente tabla se muestra un listado de todos los casos de uso identificados en el **Modelado de Casos de uso**, de los cuales se realizará su respectiva especificación.

Listado de Casos de uso		
Paquete	Id	Caso de uso
Administración	01	Iniciar sesión
	02	Cerrar sesión
	03	Manejar cuenta de usuario del sistema
Generales	04	Listar datos asociados a un Proyecto
	05	Listar datos iniciales de Proyectos
	06	Listar datos asociados a un Requerimiento
	07	Filtrar registros listados
	08	Editar registro seleccionado de una lista
	09	Eliminar registro seleccionado de una lista
	10	Manejar datos iniciales del Proyecto
	11	Manejar datos asociados a un Proyecto
	12	Manejar datos asociados a un Requerimiento
	13	Quitar ítem de una lista de datos
	14	Manejar Anexo o archivo del Proyecto
	15	Generar reporte
Parámetros Generales	16	Listar parámetros generales del sistema
	17	Manejar parámetros generales del sistema
	18	Manejar Persona de Desarrollo
Iniciación del Proyecto	19	Actualizar datos adicionales al Proyecto
	20	Conformar Equipo de Desarrollo
	21	Asignar Personas de Desarrollo al Proyecto
	22	Realizar Análisis de Metas del Proyecto
	23	Manejar Proceso que requiere el sistema
	24	Manejar Actividad por Proceso
	25	Manejar Responsable de Actividad
	26	Identificar Responsable por Actividad
	27	Asignar Responsables a una Actividad
	28	Realizar identificación macro de Procesos
	29	Establecer Requerimiento del Cliente
Análisis del Sistema	30	Mostrar información actual del Requerimiento

	31	Actualizar datos adicionales al Requerimiento
	32	Manejar Conflicto por Resolver
	33	Manejar Criterio de Aceptación
	34	Identificar Posible Mejora Potencial
	35	Priorizar los Requerimientos

Tabla 3.12: (Listado de Casos de Uso)

Especificación de Caso de Uso		
Código	01	
Caso de Uso	Iniciar sesión	
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar conectado al intranet mediante el browser. El usuario debe estar viendo la Página Principal del sistema en el browser. 	
Flujo de Eventos	Básico	<p>El usuario da clic en el enlace "Iniciar sesión" de la Página Principal (ver Figura 3.2), y el sistema muestra la Página de Inicio de sesión (ver Figura 3.3). El usuario ingresa su nombre de usuario y contraseña y da clic en la opción "Ingresar".</p> <p>El sistema verifica si existen los datos ingresados en cuenta de usuario. Si estos datos existen, el sistema autentica al usuario en sesión y se direcciona a la Página Principal mostrando un mensaje de bienvenida, y el nombre del usuario autenticado. Además el sistema cambia el enlace "Iniciar sesión" por el enlace "Cerrar sesión".</p>
	Alternativos	<ul style="list-style-type: none"> Datos inválidos: El sistema se direcciona a la misma Página de Inicio de sesión mostrando un mensaje que dice "Nombre de usuario y/o contraseña son incorrectos, inicio de sesión falló". Usuario cancela inicio de sesión: El usuario da clic en la opción "Cancelar", y el sistema muestra la Página Principal.
Pos condiciones	<ul style="list-style-type: none"> Se observa el nombre de usuario en sesión en la Página Principal. 	

Tabla 3.13: (Especificación del caso de uso "Iniciar sesión")

Especificación de Caso de Uso		
Código	02	
Caso de Uso	Cerrar sesión	
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar en sesión, para esta acción se incluye el caso de uso "Iniciar sesión". 	
Flujo de Eventos	Básico	<p>El usuario da clic en el enlace "Cerrar sesión" de la Página Principal (ver Figura 3.2), y el sistema muestra una ventana pop-up (ventana emergente) con un mensaje que dice "¿Está seguro que desea salir de la Aplicación?", y dos opciones (Confirmar, Cancelar).</p> <p>El usuario da clic en la opción "Confirmar", y el sistema descarga los datos del usuario en sesión y vuelve a mostrar la Página Principal cambiando el enlace "Cerrar sesión" por el enlace "Iniciar sesión".</p>
	Alternativos	<ul style="list-style-type: none"> Usuario cancela cerrar sesión: El usuario da clic en la opción "Cancelar", y el sistema oculta la ventana pop-up.
Pos condiciones	<ul style="list-style-type: none"> El usuario esta fuera de sesión. 	

Tabla 3.14: (Especificación del caso de uso "Cerrar sesión")

Especificación de Caso de Uso	
Código	03
Caso de Uso	Manejar cuenta de usuario del sistema
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar viendo la lista de cuentas de usuario en la Página de Manejo de Cuentas de Usuario, para esta acción se incluye el caso de uso "Listar parámetros generales del sistema".
Flujo de Eventos	<p>Básico</p> <p>El usuario da clic en la opción "Crear" de la Página de Manejo de Cuentas de Usuario del sistema (ver Figura 3.4), y el sistema genera una lista que indica el rol del usuario (ítems: Administrador, Líder del Proyecto, Analista, Cliente), y muestra esta lista en forma desplegable junto a los campos de "usuario", y "contraseña" en la Página de Creación de Cuenta de Usuario (ver Figura 3.5). El usuario asigna un rol seleccionado un ítem de la lista e ingresa cada dato u campo, y el sistema valida que los datos sean correctos (no excedan el límite de caracteres permitidos, obligatoriedad, tipos de datos, reglas de negocio).</p> <p>El usuario da clic en la opción "Guardar". Si cada dato es correcto el sistema persiste la Cuenta de Usuario en la respectiva tabla de la base de datos y se direcciona a la Página de Manejo de Cuentas de Usuario mostrando los nuevos datos en la lista de cuentas de usuario, y un mensaje que dice "Grabación exitosa". El usuario puede buscar, editar y eliminar las Cuentas de usuario creadas, para lo cual se incluye los casos de uso "Filtrar registros listados", "Editar registro seleccionado de una lista" y "Eliminar registro seleccionado de una lista" respectivamente.</p>
	<p>Alternativos</p> <ul style="list-style-type: none"> Dato incorrecto: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. El usuario cancela la creación de la cuenta de usuario: El sistema se direcciona a la Página de Manejo de Cuentas de Usuario mostrando la lista de cuentas de usuario. Excepción al guardar la cuenta de usuario: El sistema captura el tipo de error y se direcciona a la misma Página de Creación de Cuenta de Usuario mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> Se visualiza los nuevos datos en la lista de cuentas de usuario de la Página de Manejo de Cuentas de Usuario.

Tabla 3.15: (Especificación del caso de uso "Manejar cuenta de usuario del sistema")

Especificación de Caso de Uso	
Código	04
Caso de Uso	Listar datos asociados a un Proyecto
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar viendo la lista de Proyectos en la Página de Manejo de Proyectos, para esta acción se incluye el caso de uso "Listar datos iniciales de Proyectos".
Flujo de Eventos	<p>Básico</p> <p>El usuario da clic en la opción "Administrar" de la lista de Proyectos en la Página de Manejo de Proyectos, y el sistema se direcciona a la Página de Administración de Proyectos mostrando las respectivas opciones como se observa en la Figura 3.8.</p> <p>El usuario ingresa a la opción respectiva de la Página de Administración de Proyectos, y el sistema obtiene una lista de datos de la Entidad asociada al Proyecto seleccionado y verifica si la lista no está vacía. Si no está vacía el sistema muestra la lista de datos en su respectiva Página de Manejo de Datos relacionados a un Proyecto.</p>
	<p>Alternativos</p> <ul style="list-style-type: none"> Lista de datos vacía: El sistema muestra en la Página de Manejo de Datos relacionados a un Proyecto un mensaje que dice "No se encontraron registros". Excepción al listar datos: El sistema captura el tipo de error y se

		direcciona a la Página de Manejo de Datos relacionados a un Proyecto mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> Se visualiza la lista de datos asociada al Proyecto seleccionado en la Página de Manejo de Datos relacionados a un Proyecto. 	

Tabla 3.16: (Especificación del caso de uso “Listar datos asociados a un Proyecto”)

Especificación de Caso de Uso		
Código	05	
Caso de Uso	Listar datos iniciales de Proyectos	
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar en sesión, para esta acción se incluye el caso de uso “Iniciar sesión”. 	
Flujo de Eventos	Básico	<p>El usuario ingresa a la opción "Proyectos" de la Página Principal (ver Figura 3.2), y el sistema obtiene una lista de Proyectos, y verifica si la lista no está vacía.</p> <p>Si la lista no está vacía el sistema muestra los campos de código, nombre, estado, y nombre del cliente el cual se obtiene mediante el identificador del Cliente asociado al Proyecto; esta lista se muestra en la Página de Manejo de Proyectos con sus respectivas opciones.</p>
	Alternativos	<ul style="list-style-type: none"> Lista de Proyectos vacía: El sistema muestra en la Página de Manejo de Proyectos un mensaje que dice “No se encontraron registros”. Excepción al listar Proyectos: El sistema captura el tipo de error y se direcciona a la Página de Manejo de Proyectos mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> Se visualiza la lista de Proyectos en la Página de Manejo de Proyectos. 	

Tabla 3.17: (Especificación del caso de uso “Listar datos iniciales de Proyectos”)

Especificación de Caso de Uso		
Código	06	
Caso de Uso	Listar datos asociados a un Requerimiento	
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar viendo la lista de Requerimientos asociada al Proyecto seleccionado en la Página de Selección de un Requerimiento, para esta acción se incluye el caso de uso “Listar datos asociados a un Proyecto”. 	
Flujo de Eventos	Básico	<p>El usuario puede buscar un requerimiento antes de seleccionarlo, para lo cual se incluye el caso de uso “Filtrar registros listados”. El usuario da clic en la opción “Administrar” de la lista de Requerimientos asociada al Proyecto seleccionado en la Página de Selección de un Requerimiento (ver Figura 3.17), y el sistema se direcciona a la Página de Administración de Requerimientos mostrando las respectivas opciones como se observa en la Figura 3.18.</p> <p>El usuario ingresa a la opción respectiva de la Página de Administración de Requerimientos, y el sistema obtiene una lista de datos de la Entidad asociada al Requerimiento seleccionado y verifica si la lista no está vacía. Si no está vacía el sistema muestra la lista de datos en su respectiva Página de Manejo de Datos relacionados a un Requerimiento seleccionado.</p>
	Alternativos	<ul style="list-style-type: none"> Lista de datos vacía: El sistema muestra en la Página de Manejo de Datos relacionados a un Requerimiento seleccionado un mensaje que dice “No se encontraron registros”.

		<ul style="list-style-type: none"> • Excepción al listar datos: El sistema captura el tipo de error y se direcciona a la Página de Manejo de Datos relacionados a un Requerimiento seleccionado mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> • Se visualiza la lista de datos asociada al Requerimiento seleccionado en la Página de Manejo de Datos relacionados a un Requerimiento. 	

Tabla 3.18: (Especificación del caso de uso “Listar datos asociados a un Requerimiento”)

Especificación de Caso de Uso		
Código	07	
Caso de Uso	Filtrar registros listados	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de datos en la respectiva Página de Manejo de Datos, para esta acción se incluye el caso de uso “Listar parámetros generales del sistema”. 	
Flujo de Eventos	Básico	<p>El usuario ingresa los filtros (o campos) de búsqueda y da clic en la opción “Buscar” de la Página de Manejo de Datos (ver Figura 3.4), el sistema obtiene una lista de datos de la respectiva Entidad filtrada por los campos de búsqueda y verifica si la lista no está vacía.</p> <p>Si no está vacía el sistema muestra la lista filtrada en la respectiva Página de Manejo de Datos.</p>
	Alternativos	<ul style="list-style-type: none"> • Lista filtrada está vacía: El sistema muestra un mensaje en la Página de Manejo de Datos que dice “No se encontraron registros”. • El usuario resetea la búsqueda: El sistema quita todos los filtros de búsqueda y muestra la lista de datos sin filtrarla en la Página de Manejo de Datos. • Excepción al buscar datos: El sistema captura el tipo de error y se direcciona a la Página de Manejo de Datos mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> • Se visualiza la lista de datos filtrada por los campos de búsqueda en la Página de Manejo de Datos con sus respectivas opciones. 	

Tabla 3.19: (Especificación del caso de uso “Filtrar registros listados”)

Especificación de Caso de Uso		
Código	08	
Caso de Uso	Editar registro seleccionado de una lista	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de datos en la respectiva Página de Manejo de Datos, para esta acción se incluye el caso de uso “Listar parámetros generales del sistema”. 	
Flujo de Eventos	Básico	<p>El usuario da clic en la opción “Editar” de la lista de datos en la Página de Manejo de Datos (ver Figura 3.4), y el sistema obtiene los datos del registro seleccionado de la respectiva Entidad y muestra estos en la Página de Edición de Datos (ver Figura 3.6). El usuario edita cada dato, y el sistema valida que el dato sea correcto (no exceda el límite de caracteres permitidos, obligatoriedad, tipo de datos, reglas de negocio).</p> <p>El usuario da clic en la opción “Guardar”. Si cada dato es correcto el sistema actualiza los datos u campos en la respectiva tabla de la base de datos y se direcciona a la Página de Manejo de Datos mostrando la lista de datos con sus respectivos campos editados, y un mensaje que dice “Actualización exitosa”.</p>

	Alternativos	<ul style="list-style-type: none"> • Dato incorrecto: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. • El usuario cancela la edición de datos: El sistema se direcciona a la Página de Manejo de Datos mostrando la lista de datos. • Excepción al editar datos: El sistema captura el tipo de error y se direcciona a la misma Página de Edición de Datos mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> • Se visualiza la lista de datos con sus respectivos campos editados en la Página de Manejo de Datos. 	

Tabla 3.20: (Especificación del caso de uso “Editar registro seleccionado de una lista”)

Especificación de Caso de Uso		
Código	09	
Caso de Uso	Eliminar registro seleccionado de una lista	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de datos en la respectiva Página de Manejo de Datos, para esta acción se incluye el caso e uso “Listar parámetros generales del sistema”. 	
Flujo de Eventos	Básico	<p>El usuario da clic en la opción “Eliminar” de la lista de datos en la Página de Manejo de Datos (ver Figura 3.4), y el sistema obtiene los datos del registro seleccionado de la respectiva Entidad y muestra estos en una Ventana Emergente (tipo pop-up, ver Figura 3.7).</p> <p>El usuario da clic en la opción “Confirmar”, y el sistema elimina los datos de la respectiva tabla de la base de datos y se direcciona a la Página de Manejo de Datos quitando los datos eliminados de la lista de datos, y mostrando un mensaje “Eliminación exitosa”.</p>
	Alternativos	<ul style="list-style-type: none"> • El usuario cancela la eliminación de datos: El sistema se direcciona a la Página de Manejo de Datos mostrando la lista de datos. • Excepción al eliminar datos: El sistema captura el tipo de error y se direcciona a la Página de Manejo de Datos mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> • Se visualiza la lista de datos sin el registro eliminado en la Página de Manejo de Datos. 	

Tabla 3.21: (Especificación del caso de uso “Eliminar registro seleccionado de una lista”)

Especificación de Caso de Uso		
Código	10	
Caso de Uso	Manejar datos iniciales del Proyecto	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de Proyectos en la Página de Manejo de Proyectos, para esta acción se incluye el caso de uso “Listar datos iniciales de Proyectos”. 	
Flujo de Eventos	Básico	<p>El usuario da clic en la opción “Crear” de la Página de Manejo de Proyectos (ver Figura 3.4), y el sistema obtiene una lista de Clientes, y muestra esta lista de forma desplegable junto a los campos de “código” y “nombre” en la Página de Creación de Proyecto (ver Figura 3.5). El usuario asocia un Cliente al Proyecto seleccionado un Cliente de la lista e ingresa cada dato u campo, y el sistema valida que los datos sean correctos (no excedan el límite de caracteres permitidos, obligatoriedad, tipos de datos, reglas de negocio).</p>

		El usuario da clic en la opción "Guardar". Si cada dato es correcto el sistema persiste el Proyecto en la respectiva tabla de la base de datos y se direcciona a la Página de Manejo de Proyectos mostrando los nuevos datos en la lista de Proyectos, y un mensaje que dice "Grabación exitosa". El usuario puede buscar, editar y eliminar el Proyecto creado, para lo cual se incluye los casos de uso "Filtrar registros listados", "Editar registro seleccionado de una lista" y "Eliminar registro seleccionado de una lista" respectivamente.
	Alternativos	<ul style="list-style-type: none"> • Dato incorrecto: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. • El usuario cancela la creación del Proyecto: El sistema se direcciona a la Página de Manejo de Proyectos mostrando la lista de Proyectos. • Excepción al guardar Proyecto: El sistema captura el tipo de error y se direcciona a la misma Página de Creación de Proyecto mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones		<ul style="list-style-type: none"> • Se visualiza los nuevos datos en la lista de Proyectos de la Página de Manejo de Proyectos.

Tabla 3.22: (Especificación del caso de uso "Manejar datos iniciales del Proyecto")

Tener en cuenta que la especificación del caso de uso que se ilustra en la Tabla 3.12, hace referencia al manejo de datos asociados a un Proyecto como: datos de las peticiones del Cliente, y datos de las razones por las cuales el Cliente solicita el nuevo Sistema. Además esta especificación es también para los casos de uso "Manejar Proceso que requiere el Sistema", "Manejar Responsable de Actividad", y "Establecer Requerimiento del Cliente" debido a su comportamiento similar.

Especificación de Caso de Uso	
Código	11
Caso de Uso	Manejar datos asociados a un Proyecto
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de datos asociada a un Proyecto en la respectiva Página de Manejo de Datos relacionados a un Proyecto, para esta acción se incluye el caso de uso "Listar datos asociados a un Proyecto".
Flujo de Eventos	<p>Básico</p> <p>El usuario da clic en la opción "Crear" de la Página de Manejo de Datos relacionados a un Proyecto (ver Figura 3.4), y el sistema muestra la respectiva Página de Creación de Datos relacionados a un Proyecto (ver Figura 3.5). El usuario ingresa cada dato u campo, y el sistema valida que el dato sea correcto (no exceda el límite de caracteres permitidos, obligatoriedad, tipos de datos, reglas de negocio).</p> <p>El usuario da clic en la opción "Guardar". Si cada dato es correcto el sistema persiste los datos o campos en la respectiva tabla (asociada a un Proyecto) de la base de datos y se direcciona a la Página de Manejo de Datos relacionados a un Proyecto mostrando los nuevos datos en la lista de datos asociada a un Proyecto seleccionado, y un mensaje que dice "Grabación exitosa". El usuario puede buscar, editar y eliminar los datos creados, para lo cual se incluye los casos de uso "Filtrar registros listados",</p>

		"Editar registro seleccionado de una lista" y "Eliminar registro seleccionado de una lista" respectivamente.
	Alternativos	<ul style="list-style-type: none"> • Dato incorrecto: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. • El usuario cancela la creación de un nuevo dato: El sistema se direcciona a la Página de Manejo de Datos relacionados a un Proyecto mostrando la lista de datos asociada a un Proyecto. • Excepción al guardar datos: El sistema captura el tipo de error y se direcciona a la misma Página de Creación de Datos relacionados a un Proyecto mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones		<ul style="list-style-type: none"> • Se visualiza los nuevos datos en la lista de datos asociada a un Proyecto de la Página de Manejo de Datos relacionados a un Proyecto.

Tabla 3.23: (Especificación del caso de uso "Manejar datos asociados a un Proyecto")

Quando se realizó el modelado de casos de uso se detectó que los casos de uso agrupados en la Tabla 3.24 tienen comportamiento similar, por esta razón se realizará la especificación de estos mediante un caso de uso genérico identificado como "Manejar datos asociados a un Requerimiento".

Id	Caso de Uso
32	Manejar Conflicto por Resolver
33	Manejar Criterio de Aceptación

Tabla 3.24: (Casos de uso para Manejar datos asociados a un Requerimiento)

Especificación de Caso de Uso	
Código	12
Caso de Uso	Manejar datos asociados a un Requerimiento
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de datos asociada a un Requerimiento en la respectiva Página de Manejo de Datos relacionados a un Requerimiento, para esta acción se incluye el caso de uso "Listar datos asociados a un Requerimiento".
Flujo de Eventos	<p>Básico</p> <p>El usuario da clic en la opción "Crear" de la Página de Manejo de Datos relacionados a un Requerimiento seleccionado (ver Figura 3.4), y el sistema muestra la respectiva Página de Creación de Datos relacionados a un Requerimiento (ver Figura 3.5). El usuario ingresa cada dato u campo, y el sistema valida que el dato sea correcto (no exceda el límite de caracteres permitidos, obligatoriedad, tipos de datos, reglas de negocio).</p> <p>El usuario da clic en la opción "Guardar". Si cada dato es correcto el sistema persiste los datos u campos en la respectiva tabla (asociada a un Requerimiento) de la base de datos y se direcciona a la Página de Manejo de Datos relacionados a un Requerimiento mostrando los nuevos datos en la lista de datos asociada a un Requerimiento, y un mensaje que dice "Grabación exitosa". El usuario puede buscar, editar y eliminar los datos creados, para lo cual se incluye los casos de uso "Filtrar registros listados", "Editar registro seleccionado de una lista" y "Eliminar registro seleccionado</p>

		de una lista" respectivamente.
	Alternativos	<ul style="list-style-type: none"> • Dato incorrecto: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. • El usuario cancela la creación de un nuevo dato: El sistema se direcciona a la Página de Manejo de Datos relacionados a un Requerimiento mostrando la lista de datos asociada a un Requerimiento. • Excepción al guardar datos: El sistema captura el tipo de error y se direcciona a la misma Página de Creación de Datos relacionados a un Requerimiento mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones		<ul style="list-style-type: none"> • Se visualiza los nuevos datos en la lista de datos asociada a un Requerimiento de la Página de Manejo de Datos relacionados a un Requerimiento.

Tabla 3.25: (Especificación del caso de uso “Manejar datos asociados a un Requerimiento”)

Especificación de Caso de Uso		
Código	13	
Caso de Uso	Quitar ítem de una lista de datos	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo los ítems de la lista de datos en su respectiva Página de ítems asociados a un registro seleccionado. 	
Flujo de Eventos	Básico	El usuario da clic en la opción “Quitar” del ítem de la lista de datos en su respectiva Página de ítems asociados a un registro seleccionado, y el sistema remueve el ítem seleccionado de la lista de datos de la respectiva Entidad, además actualiza la lista de datos quitando el ítem seleccionado.
	Alternativos	<ul style="list-style-type: none"> • Excepción al remover un ítem de una lista de datos: El sistema captura el tipo de error y se direcciona a su respectiva Página mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> • No se observa el ítem removido en la lista de datos de la respectiva Página. 	

Tabla 3.26: (Especificación del caso de uso “Quitar ítem de una lista de datos”)

Especificación de Caso de Uso		
Código	14	
Caso de Uso	Manejar Anexo o archivo del Proyecto	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de Anexos asociada a un Proyecto en la Página de Manejo de Anexos, para esta acción se incluye el caso de uso “Listar datos asociados a un Proyecto”. 	
Flujo de Eventos	Básico	El usuario da clic en la opción “Crear” de la Página de Manejo de Anexos (ver Figura 3.21), y el sistema muestra la Página de Creación de Anexo (ver Figura 3.22). El usuario ingresa los campos “código”, “descripción”, y el sistema valida que estos datos sean correctos (no excedan el límite de caracteres permitidos, obligatoriedad, tipos de datos, reglas de negocio); el usuario da clic en la opción “Agregar”, y el sistema muestra una ventana emergente para navegar entre directorios. El usuario selecciona un archivo (de cualquier extensión) y cierra la ventana emergente, y el sistema muestra el nombre del archivo cargado, bloquea la opción “Agregar”, y muestra una opción “Quitar”.
		El usuario da clic en la opción “Guardar”. Si cada dato es correcto el sistema persiste el Anexo en la respectiva tabla (asociada al Proyecto) de la base de datos y se direcciona a la Página de Manejo de Anexos

		mostrando el nuevo anexo en la lista de anexos asociada a un Proyecto seleccionado, y un mensaje que dice "Grabación exitosa". El usuario da clic en la opción "Descargar" de la lista de anexos, y el sistema muestra una ventana emergente para descargar (abrir o guardar) el respectivo archivo. El usuario puede buscar, editar y eliminar los anexos creados, para lo cual se incluye los casos de uso "Filtrar registros listados", "Editar registro seleccionado de una lista" y "Eliminar registro seleccionado de una lista" respectivamente.
	Alternativos	<ul style="list-style-type: none"> • Dato incorrecto: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. • El usuario cancela la creación de un nuevo anexo: El sistema se direcciona a la Página de Manejo de Anexos mostrando la lista de anexos asociada a un Proyecto. • El usuario quita un archivo cargado: El sistema oculta el nombre del archivo cargado, y la opción "Quitar", y activa la opción "Agregar". • Excepción al guardar datos: El sistema captura el tipo de error y se direcciona a la misma Página de Creación de un Anexo mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones		<ul style="list-style-type: none"> • Se obtiene el respectivo archivo, el cual podrá ser visualizado en su respectivo Software dependiendo de su extensión.

Tabla 3.27: (Especificación del caso de uso "Manejar Anexo o archivo del Proyecto")

Especificación de Caso de Uso		
Código	15	
Caso de Uso	Generar reporte	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar en sesión, para esta acción se incluye el caso de uso "Iniciar sesión". 	
Flujo de Eventos	Básico	<p>El usuario ingresa a la opción "Reportes" de la Página Principal (ver Figura 3.2), y el sistema obtiene una lista de Proyectos y verifica si la lista no está vacía. Si la lista no está vacía el sistema muestra esta lista (mostrar los campos: código, nombre del proyecto, cliente asociado al proyecto, y estado del proyecto) en la Página de Reportes con sus respectivas opciones como se puede observar en la Figura 3.23.</p> <p>El usuario da clic en la opción "Reportes" de la lista de Proyectos en la Página de Reportes, y el sistema obtiene una lista de tipos de reportes, y muestra esta lista en una ventana emergente con sus respectivas opciones (ver Figura 3.24 y 3.25). El usuario da clic en la opción "Descargar", y el sistema obtiene el respectivo reporte del Proyecto seleccionado mostrando este reporte en formato PDF.</p>
	Alternativos	<ul style="list-style-type: none"> • Lista de datos vacía: El sistema muestra en la Página de Reportes un mensaje que dice "No se encontraron registros". • El usuario cierra la ventana emergente: El sistema oculta la ventana emergente, y muestra la Página de Reportes. • Excepción al listar datos: El sistema captura el tipo de error y se direcciona a la Página de Reportes mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> • Se visualiza el reporte del Proyecto seleccionado en formato PDF. 	

Tabla 3.28: (Especificación del caso de uso "Generar reporte")

Especificación de Caso de Uso	
Código	16
Caso de Uso	Listar parámetros generales del sistema
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar en sesión, para esta acción se incluye el caso de uso "Iniciar sesión".
Flujo de Eventos	<p>Básico</p> <p>El usuario ingresa a la opción respectiva de la Página Principal (ver Figura 3.2), y el sistema obtiene una lista de datos de la respectiva Entidad y verifica si la lista no está vacía.</p> <p>Si la lista no está vacía el sistema muestra la lista de datos en su respectiva Página de Manejo de Datos con sus respectivas opciones como se puede observar en la Figura 3.4.</p>
	<p>Alternativos</p> <ul style="list-style-type: none"> Lista de datos vacía: El sistema muestra en la Página de Manejo de Datos un mensaje que dice "No se encontraron registros". Excepción al listar datos: El sistema captura el tipo de error y se direcciona a la Página de Manejo de Datos mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> La lista de datos se visualiza en la Página de Manejo de Datos con sus respectivas opciones.

Tabla 3.29: (Especificación del caso de uso "Listar parámetros generales del sistema")

La especificación del caso de uso que se ilustra en la Tabla 3.30 hace referencia al manejo de datos de las opciones generales del sistema como: datos de Clientes, y datos de Cargos (roles).

Especificación de Caso de Uso	
Código	17
Caso de Uso	Manejar parámetros generales del sistema
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar viendo la lista de datos en la respectiva Página de Manejo de Datos, para esta acción se incluye el caso de uso "Listar parámetros generales del sistema".
Flujo de Eventos	<p>Básico</p> <p>El usuario da clic en la opción "Crear" de la Página de Manejo de Datos (ver Figura 3.4), y el sistema muestra la Página de Creación de Datos (ver Figura 3.5). El usuario ingresa cada dato u campo, y el sistema valida que los datos sean correctos (no excedan el límite de caracteres permitidos, obligatoriedad, tipos de datos, reglas de negocio).</p> <p>El usuario da clic en la opción "Guardar". Si cada dato es correcto el sistema persiste los datos u campos en la respectiva tabla de la base de datos y se direcciona a la Página de Manejo de Datos mostrando los nuevos datos en la lista de datos, y un mensaje que dice "Grabación exitosa". El usuario puede buscar, editar y eliminar los datos creados, para lo cual se incluye los casos de uso "Filtrar registros listados", "Editar registro seleccionado de una lista" y "Eliminar registro seleccionado de una lista" respectivamente.</p>
	<p>Alternativos</p> <ul style="list-style-type: none"> Dato incorrecto: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. El usuario cancela la creación de un nuevo dato: El sistema se direcciona a la Página de Manejo de Datos mostrando la lista de datos. Excepción al guardar datos: El sistema captura el tipo de error y se direcciona a la misma Página de Creación de Datos mostrando un

		mensaje que informa el tipo de error capturado.
Pos condiciones		<ul style="list-style-type: none"> Se visualiza los nuevos datos en la lista de datos de la Página de Manejo de Datos.

Tabla 3.30: (Especificación del caso de uso “Manejar parámetros generales del sistema”)

Especificación de Caso de Uso		
Código	18	
Caso de Uso	Manejar Persona de Desarrollo	
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar viendo la lista de Personas de Desarrollo en la Página de Manejo de Personas de Desarrollo, para esta acción se incluye el caso de uso “Listar parámetros generales del sistema”. 	
Flujo de Eventos	Básico	<p>El usuario da clic en la opción “Crear” de la Página de Manejo de Personas de Desarrollo, y el sistema se direcciona a la Página de Creación de Persona de Desarrollo mostrando una lista desplegable de Cargos, una lista checklist de Características, y el resto de campos como se muestra en la Figura 3.12. El usuario ingresa los campos solicitados y el sistema valida que los datos sean correctos (no excedan el límite de caracteres permitidos, obligatoriedad, tipos de datos, reglas de negocio).</p> <p>El usuario da clic en la opción “Guardar”. Si los campos son correctos el sistema persiste los datos u campos en las respectivas tablas de la base de datos y se direcciona a la Página de Manejo de Personas de Desarrollo mostrando los nuevos datos en la lista de Personas de Desarrollo, y un mensaje que dice “Grabación exitosa”. El usuario puede buscar, editar y eliminar las Personas de Desarrollo creadas, para lo cual se incluye los casos de uso “Filtrar registros listados”, “Editar registro seleccionado de una lista” y “Eliminar registro seleccionado de una lista” respectivamente.</p>
	Alternativos	<ul style="list-style-type: none"> Datos incorrectos: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. El usuario cancela la creación de una Nueva Persona de Desarrollo: El sistema se direcciona a la Página de Manejo de Personas de Desarrollo mostrando la lista de Personas de Desarrollo. Excepción al guardar datos: El sistema captura el tipo de error y se direcciona a la misma Página de Creación de Persona de Desarrollo mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> Se visualiza la nueva Persona de Desarrollo en la lista de Personas de Desarrollo de la Página de Manejo de Personas de Desarrollo. 	

Tabla 3.31: (Especificación del caso de uso “Manejar Persona de Desarrollo”)

Especificación de Caso de Uso		
Código	19	
Caso de Uso	Actualizar datos adicionales al Proyecto	
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar viendo la lista de Proyectos en la Página de Manejo de Proyectos para lo cual se incluye el caso de uso “Listar datos iniciales de Proyectos”. 	
Flujo de Eventos	Básico	<p>El usuario da clic en la opción “Administrar” de la lista de Proyectos en la Página de Manejo de Proyectos, y el sistema se direcciona a la Página de Administración de Proyectos mostrando las respectivas opciones como se observa en la Figura 3.8. El usuario ingresa al enlace “Información del Proyecto”, y el sistema obtiene los datos del Proyecto seleccionado y muestra estos en la Página de Información del Proyecto como se observa en la Figura 3.13.</p>

		El usuario edita los datos u campos solicitados en la Página de Información del Proyecto, y el sistema valida que los datos sean correctos (no excedan el límite de caracteres permitidos, obligatoriedad, tipo de datos, reglas de negocio). El usuario da clic en la opción "Guardar". Si los datos son correctos el sistema actualiza los datos u campos en la respectiva tabla de la base de datos y se direcciona a la misma Página de Información del Proyecto mostrando los datos actualizados, y un mensaje que dice "Actualización exitosa".
	Alternativos	<ul style="list-style-type: none"> • Datos incorrectos: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. • Excepción al mostrar información del Proyecto: El sistema captura el tipo de error y muestra este en la Página de Información del Proyecto. • Excepción al actualizar los datos adicionales al Proyecto: El sistema captura el tipo de error y se direcciona a la Página de Información del Proyecto mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones		<ul style="list-style-type: none"> • Se visualizan los datos actualizados del Proyecto en la Página de Información del Proyecto.

Tabla 3.32: (Especificación del caso de uso "Actualizar datos adicionales al Proyecto")

Especificación de Caso de Uso		
Código	20	
Caso de Uso	Conformar Equipo de Desarrollo	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de Personas de Desarrollo asignadas al Proyecto seleccionado, para esta acción se incluye el caso de uso "Listar datos asociados a un Proyecto". 	
Flujo de Eventos	Básico	<p>El usuario da clic en la opción "Conformar Equipo" de la Página de Manejo del Equipo de Desarrollo (ver Figura 3.9), y el sistema obtiene la lista de Personas de Desarrollo asignadas al Equipo de Desarrollo del Proyecto seleccionado, y verifica si la lista está vacía. Si la lista no está vacía el sistema muestra la lista en la Página de Conformación del Equipo de Desarrollo con sus respectivas opciones como se observa en la Figura 3.10. El usuario puede agregar o quitar integrantes (Personas de Desarrollo) al Equipo de Desarrollo para lo cual se incluye los casos de uso "Asignar Personas de Desarrollo al Proyecto" y "Quitar ítem de una lista de datos" respectivamente.</p> <p>El usuario da clic en la opción "Guardar" y el sistema persiste el Proyecto seleccionado con su respectiva lista de integrantes del Equipo de Desarrollo en las respectivas tablas de la base de datos, y se direcciona a la Página de Manejo del Equipo de Desarrollo mostrando la lista de integrantes del Equipo de Desarrollo asociada al Proyecto seleccionado, y un mensaje que dice "Grabación exitosa". El usuario puede buscar y eliminar Personas de Desarrollo del Equipo de Desarrollo conformado, para lo cual se incluye los casos de uso "Filtrar registros listados", y "Eliminar registro seleccionado de una lista" respectivamente.</p>
	Alternativos	<ul style="list-style-type: none"> • La lista de integrantes del Equipo de Desarrollo está vacía: El sistema muestra en la Página de Conformación del Equipo de Desarrollo un mensaje que dice "No se encontraron registros". • El usuario cancela la conformación del Equipo de Desarrollo: El sistema se direcciona a la Página de Manejo del Equipo de Desarrollo mostrando la lista de integrantes del Equipo de Desarrollo asociada al Proyecto seleccionado. • Excepción al guardar el Equipo de Desarrollo: El sistema captura el

		tipo de error y se direcciona a la misma Página de Conformación del Equipo de Desarrollo mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones		<ul style="list-style-type: none"> Se visualiza la lista actual de Integrantes del Equipo de Desarrollo asociada al Proyecto seleccionado en la Página de Manejo del Equipo de Desarrollo.

Tabla 3.33: (Especificación del caso de uso “Conformar Equipo de Desarrollo”)

Especificación de Caso de Uso		
Código	21	
Caso de Uso	Asignar Personas de Desarrollo al Proyecto	
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar viendo la lista de Asignaciones de Persona de Desarrollo asociada al Proyecto seleccionado en la Página de Conformación del Equipo de Desarrollo, para esta acción se incluye el caso de uso “Conformar Equipo de Desarrollo”. 	
Flujo de Eventos	Básico	<p>El usuario da clic en la opción “Agregar” de la Página de Conformación del Equipo de Desarrollo (ver Figura 3.10), y el sistema obtiene una lista de Personas de Desarrollo que no están en la lista de Asignaciones de Personas de Desarrollo, luego el sistema verifica si esta lista obtenida contiene al menos un ítem. Si esta lista obtenida contiene al menos un ítem el sistema muestra estos ítems con sus respectivas opciones en la Página de Selección de Personas de Desarrollo como se observa en la Figura 3.11. El usuario puede buscar una Persona de Desarrollo antes de seleccionarla, para lo cual se incluye el caso de uso “Filtrar registros listados”.</p> <p>El usuario da clic en la opción “Seleccionar” de la lista de Personas de Desarrollo y el sistema agrega la respectiva Persona de Desarrollo seleccionada a la lista de Asignaciones de Personas de Desarrollo asociada al Proyecto seleccionado, además el sistema actualiza la lista de Personas de Desarrollo quitando (ocultando) la Persona de Desarrollo seleccionada. El usuario da clic en la opción “Listo” de la Página de Selección de Personas de Desarrollo y el sistema obtiene la lista actual de Asignaciones de Personas de Desarrollo asociada al Proyecto seleccionado y muestra esta lista en la Página de Conformación del Equipo de Desarrollo.</p>
	Alternativos	<ul style="list-style-type: none"> La lista de Personas de Desarrollo no asignadas está vacía: El sistema muestra en la Página de Selección de Personas de Desarrollo un mensaje que dice “No se encontraron registros”. Excepción al agregar una Persona de Desarrollo: El sistema captura el tipo de error e informa este error con un mensaje en la Página de Selección de Personas de Desarrollo.
Pos condiciones	<ul style="list-style-type: none"> Se observa la lista actual de Asignaciones de Personas de Desarrollo asociada al Proyecto seleccionado en la Página de Conformación del Equipo de Desarrollo. 	

Tabla 3.34: (Especificación del caso de uso “Asignar Personas de Desarrollo al Proyecto”)

Especificación de Caso de Uso		
Código	22	
Caso de Uso	Realizar Análisis de Metas del Proyecto	
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar viendo la lista de Metas asociadas al Proyecto seleccionado, para esta acción se incluye el caso de uso “Listar datos asociados a un Proyecto”. 	
Flujo de	Básico	El usuario da clic en opción “Crear” de la Página de Manejo de Metas, y el sistema se direcciona a la Página de Creación de Meta mostrando una

Eventos		<p>lista de Análisis de Meta inicializada con una lista de Características (ítems de características: Específica, Medurable, Alcanzable, Relevante), como se observa en la figura 3.14. El usuario ingresa cada campo, y el sistema valida si este sea correcto (no exceda el límite de caracteres permitidos, obligatoriedad, tipos de datos, reglas de negocio).</p> <p>El usuario da clic en la opción “Guardar”. Si cada campo es correcto, el sistema persiste la Meta con su lista de Análisis en las respectivas tablas de la base de datos, y se direcciona a la Página de Manejo de Metas mostrando la nueva Meta en la lista de Metas asociada a un Proyecto seleccionado, y un mensaje que dice “Grabación exitosa”. El usuario puede buscar, editar y eliminar las Metas creadas con su respectivo Análisis, para lo cual se incluye los casos de uso “Filtrar registros listados”, “Editar registro seleccionado de una lista”, y “Eliminar registro seleccionado de una lista” respectivamente.</p>
	Alternativos	<ul style="list-style-type: none"> • Dato incorrecto: El sistema muestra al lado del dato u campo incorrecto un mensaje que informa el tipo de error. • El usuario cancela la creación de una nueva Meta: El sistema se direcciona a la Página de Manejo de Metas mostrando la lista de Metas asociada al Proyecto seleccionado. • Excepción al guardar datos: El sistema captura el tipo de error y se direcciona a la misma Página de Creación de Meta mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> • Se visualiza la lista actual de Metas asociada al Proyecto seleccionado en la Página de Manejo de Metas. 	

Tabla 3.35: (Especificación del caso de uso “Realizar Análisis de Metas del Proyecto”)

Especificación de Caso de Uso		
Código	24	
Caso de Uso	Manejar Actividad por Proceso	
Precondiciones	<ul style="list-style-type: none"> • El usuario lista las Actividades asociadas al Proyecto seleccionado, para esta acción se incluye el caso de uso “Listar datos asociados a un Proyecto”. 	
Flujo de Eventos	Básico	<p>El usuario da clic en la opción “Crear” de la Página de Manejo de Actividades, y el sistema se direcciona a la Página de Creación de Actividad mostrando una lista desplegable de Procesos asociados al Proyecto seleccionado, junto a sus respectivos campos. El usuario ingresa cada dato u campo y asocia la actividad a un Proceso seleccionando un Proceso de la lista desplegable, y el sistema valida que los datos sean correctos (no excedan el límite de caracteres permitidos, obligatoriedad, tipos de datos, reglas de negocio).</p> <p>El usuario da clic en la opción “Guardar”. Si cada dato es correcto el sistema persiste la Actividad en la respectiva tabla de la base de datos y se direcciona a la Página de Manejo de Actividades mostrando la nueva Actividad en la lista de Actividades, y un mensaje que dice “Grabación exitosa”. El usuario puede buscar, editar y eliminar las Actividades creadas, para lo cual se incluye los casos de uso “Filtrar registros listados”, “Editar registro seleccionado de una lista” y “Eliminar registro seleccionado de una lista” respectivamente.</p>
	Alternativos	<ul style="list-style-type: none"> • Dato incorrecto: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. • El usuario cancela la creación de una nueva Actividad: El sistema se direcciona a la Página de Manejo de Actividades mostrando la lista de Actividades. • Excepción al guardar una Actividad: El sistema captura el tipo de error y se direcciona a la misma Página de Creación de Actividad

		mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> Se visualiza la lista actual de Actividades asociada a diferentes Procesos de un Proyecto seleccionado en la Página de Manejo de Actividades. 	

Tabla 3.36: (Especificación del caso de uso “Manejar Actividad por Proceso”)

Especificación de Caso de Uso		
Código	26	
Caso de Uso	Identificar Responsable por Actividad	
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar viendo la Página de Administración de Proyectos. 	
Flujo de Eventos	Básico	<p>El usuario ingresa a la opción “Iniciación del Proyecto/Identificación de Responsables por Actividad” de la Página de Administración de Proyectos (ver Figura 3.8), y el sistema obtiene una lista de Actividades con su respectivo Proceso asociado, y la cantidad de Responsables asignados, y verifica que la lista no esté vacía. Si la lista no está vacía el sistema muestra esta lista en la Página de Manejo de Responsables por Actividad como se puede observar en la Figura 3.15. El usuario da clic en la opción “Responsables” de la lista de Actividades, y el sistema muestra el nombre de la Actividad seleccionada con su Proceso asociado, y obtiene su respectiva lista de Responsables asignados, y verifica que la lista no esté vacía. Si la lista no está vacía el sistema muestra esta lista en la Página de Asignación de Responsables a una Actividad con sus respectivas opciones como se observa en la Figura 3.16.</p> <p>El usuario puede agregar y quitar Responsables de la Actividad seleccionada, para lo cual se incluye los casos de uso “Asignar Responsables a una Actividad”, y “Quitar ítem de una lista de datos” respectivamente. El usuario da clic en la opción “Guardar”, y el sistema persiste la Actividad con su respectiva lista de Responsables asignados en las respectivas tablas de la base de datos, y se direcciona a la Página de Manejo de Responsables por Actividad, actualizando la cantidad de responsables asignados a la Actividad seleccionada, y un mensaje que dice “Grabación exitosa”. El usuario puede buscar las Actividades, para lo cual se incluye el caso de uso “Filtrar registros listados”.</p>
	Alternativos	<ul style="list-style-type: none"> La lista de actividades está vacía: El sistema muestra en la Página de Manejo de Responsables por Actividad un mensaje que dice “No se encontraron registros”. La lista de Responsables asignados a una Actividad está vacía: El sistema muestra en la Página de Asignación de Responsables a una Actividad un mensaje que dice “No se encontraron registros”. El usuario cancela la asignación de Responsables a la Actividad: El sistema se direcciona a la Página de Manejo de Responsables por Actividad mostrando la lista de Actividades. Excepción al guardar datos: El sistema captura el tipo de error y se direcciona a la misma Página de Asignación de Responsables a una Actividad mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> Se visualiza la lista actual de Actividades con su respectiva cantidad de Responsables asignados, en la Página de Manejo de Responsables por Actividad. 	

Tabla 3.37: (Especificación del caso de uso “Identificar Responsable por Actividad”)

Especificación de Caso de Uso	
Código	27
Caso de Uso	Asignar Responsables a una Actividad
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar viendo la lista de Responsables asignados a una Actividad seleccionada, en la Página de Asignación de Responsables a una Actividad, para esta acción se incluye el caso de uso "Identificar Responsable por Actividad".
Flujo de Eventos	<p>Básico</p> <p>El usuario da clic en la opción "Agregar" de la Página de Asignación de Responsables a una Actividad (ver Figura 3.16), y el sistema obtiene una lista de Responsables que no están en la lista de Asignaciones de Responsables, luego el sistema verifica si esta lista obtenida contiene al menos un ítem. Si esta lista obtenida contiene al menos un ítem el sistema muestra estos ítems con sus respectivas opciones en la Página de Selección de Responsables. El usuario puede buscar un Responsable antes de seleccionarlo, para lo cual se incluye el caso de uso "Filtrar registros listados".</p> <p>El usuario da clic en la opción "Seleccionar" de la lista de Responsables, y el sistema agrega el Responsable seleccionado a la lista de Asignaciones de Responsables asociada a la Actividad seleccionada, además el sistema actualiza la lista de Responsables quitando el Responsable seleccionado. El usuario da clic en la opción "Listo" de la Página de Selección de Responsables, y el sistema obtiene la lista actual de Asignaciones de Responsables asociada a la Actividad seleccionada, y muestra esta lista en la Página de Asignación de Responsables a una Actividad.</p>
	<p>Alternativos</p> <ul style="list-style-type: none"> La lista de Responsables no asignados está vacía: El sistema muestra en la Página de Selección de Responsables un mensaje que dice "No se encontraron registros". Excepción al agregar un Responsable: El sistema captura el tipo de error e informa este error con un mensaje en la Página de Selección de Responsables.
Pos condiciones	<ul style="list-style-type: none"> Se observa la lista actual de Asignaciones de Responsables asociada a la Actividad seleccionada en la Página de Asignación de Responsables a una Actividad.

Tabla 3.38: (Especificación del caso de uso "Asignar Responsables a una Actividad")

Especificación de Caso de Uso	
Código	28
Caso de Uso	Realizar identificación macro de Procesos
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar viendo la lista de Detalles de Procesos asociados al Proyecto seleccionado, en la Página de Manejo de Detalles de Procesos, para esta acción se incluye el caso de uso "Listar datos asociados a un Proyecto".
Flujo de Eventos	<p>Básico</p> <p>El usuario da clic en la opción "Crear" de la Página de Manejo de Detalles de Procesos (ver Figura 3.4), y el sistema obtiene una lista de Procesos asociados al Proyecto seleccionado, obtiene una lista que indica el tipo de detalle del Proceso (ítems: Proveedor, Entrada, Salida, Cliente), y muestra estas listas desplegables junto a los campos de código y descripción en la Página de Creación de Detalle del Proceso relacionado al Proyecto seleccionado (ver Figura 3.5). El usuario ingresa cada dato u campo y asocia el Detalle a un Proceso seleccionando un Proceso de la lista desplegable, y el sistema valida que los datos sean correctos (no excedan el límite de caracteres permitidos, obligatoriedad, tipos de datos, reglas de negocio).</p> <p>El usuario da clic en la opción "Guardar". Si cada dato es correcto el sistema persiste el Detalle asociado al Proceso seleccionado en la respectiva tabla de la base de datos, y se direcciona a la Página de Manejo de Detalles de Procesos mostrando el nuevo Detalle en la lista de</p>

		Detalles, y un mensaje que dice "Grabación exitosa". El usuario puede buscar, editar y eliminar los Detalles creados, para lo cual se incluye los casos de uso "Filtrar registros listados", "Editar registro seleccionado de una lista" y "Eliminar registro seleccionado de una lista" respectivamente.
	Alternativos	<ul style="list-style-type: none"> • Dato incorrecto: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. • El usuario cancela la creación de un Detalle asociado a un Proceso: El sistema se direcciona a la Página de Manejo de Detalles de Procesos mostrando la lista de Detalles. • Excepción al guardar un Detalle: El sistema captura el tipo de error y se direcciona a la misma Página de Creación de Detalle del Proceso mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones		<ul style="list-style-type: none"> • Se visualiza la lista actual de Detalles asociada a diferentes Procesos de un Proyecto seleccionado en la Página de Manejo de Detalles de Procesos.

Tabla 3.39: (Especificación del caso de uso "Realizar identificación macro de Procesos")

Especificación de Caso de Uso		
Código	30	
Caso de Uso	Mostrar información actual del Requerimiento	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de Requerimientos asociada a un Proyecto en la Página de Selección de un Requerimiento, para lo cual se incluye el caso de uso "Listar datos asociados a un Proyecto". 	
Flujo de Eventos	Básico	<p>El usuario puede buscar un requerimiento antes de seleccionarlo, para lo cual se incluye el caso de uso "Filtrar registros listados". El usuario da clic en la opción "Administrar" de la lista de Requerimientos asociada a un Proyecto en la Página de Selección de un Requerimiento (ver Figura 3.17), y el sistema se direcciona a la Página de Administración de Requerimientos con sus respectivas opciones como se observa en la Figura 3.18.</p> <p>El usuario ingresa al enlace "Información del Requerimiento", y el sistema obtiene los datos del Requerimiento seleccionado y muestra estos en la Página de Información de un Requerimiento como se observa en la Figura 3.20.</p>
	Alternativos	<ul style="list-style-type: none"> • Excepción al mostrar información del Requerimiento: El sistema captura el tipo de error y muestra este en la Página de Información de un Requerimiento.
Pos condiciones	<ul style="list-style-type: none"> • Se visualiza los datos del Requerimiento en la Página de Información de un Requerimiento. 	

Tabla 3.40: (Especificación del caso de uso "Mostrar Información actual del Requerimiento")

Especificación de Caso de Uso		
Código	31	
Caso de Uso	Actualizar datos adicionales al Requerimiento	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de Requerimientos asociada a un Proyecto en la Página de Selección de un Requerimiento, para lo cual se incluye el caso de uso "Listar datos asociados a un Proyecto" 	
Flujo de	Básico	El usuario puede buscar un requerimiento antes de seleccionarlo, para lo cual se incluye el caso de uso "Filtrar registros listados". El usuario da clic

Eventos		<p>en la opción "Administrar" de la lista de Requerimientos asociada al Proyecto seleccionado en la Página de Selección de un Requerimiento (ver Figura 3.17), y el sistema se direcciona a la Página de Administración de Requerimientos mostrando las respectivas opciones como se observa en la Figura 3.18. El usuario da clic en la opción "Datos Adicionales", y el sistema obtiene una lista desplegable que indica el tipo de requerimiento (ítems: "Funcional", "No Funcional"), y obtiene los Métodos de obtención asociados al requerimiento mostrando estos como una lista tipo checklist (posibles métodos de obtención: "Encuesta", "Focus Group", "Entrevistas Individuales", "Sitios visitados", "Quejas de los Clientes", "Retornos"), además, si la prioridad del Requerimiento seleccionado es diferente de cero el sistema bloquea el campo "Implementar Requerimiento", finalmente el sistema muestra las listas junto a otros campos en la Página de Actualización de datos adicionales al Requerimiento (ver Figura 3.19).</p> <p>El usuario ingresa los campos solicitados y el sistema valida que los datos sean correctos (no excedan el límite de caracteres permitidos, tipo de datos, reglas de negocio). El usuario da clic en la opción "Guardar". Si los datos son correctos el sistema actualiza los datos u campos en la respectiva tabla de la base de datos y se direcciona a la misma Página de Actualización de datos adicionales al Requerimiento mostrando los datos actualizados, y un mensaje que dice "Actualización exitosa".</p>
	Alternativos	<ul style="list-style-type: none"> • Datos incorrectos: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. • Excepción al actualizar los datos adicionales al Requerimiento: El sistema captura el tipo de error y se direcciona a la misma Página de Actualización de datos adicionales al Requerimiento mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> • Se visualizan los datos actualizados del Requerimiento en la Página de Actualización de datos adicionales al Requerimiento. 	

Tabla 3.41: (Especificación del caso de uso "Actualizar datos adicionales al Requerimiento")

Especificación de Caso de Uso	
Código	34
Caso de Uso	Identificar Posible Mejora Potencial
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de Posibles Mejoras asociada a un Requerimiento en la Página de Manejo de Posibles Mejoras, para esta acción se incluye el caso de uso "Listar datos asociados a un Requerimiento".
Flujo de Eventos	<p>Básico</p> <p>El usuario da clic en la opción "Crear" de la Página de Manejo de Posibles Mejoras, y el sistema genera dos listas desplegables, una que indica el tipo de tiempo (ítems: Bajo, Medio, Alto), y la otra que indica el tipo de costo (ítems: Bajo Medio, Alto, Desconocido), y muestra estas listas junto a los campos de código y descripción en la Página de Creación de Posible Mejora relacionada al Requerimiento seleccionado. El usuario ingresa cada dato u campo, y el sistema valida que el dato sea correcto (no exceda el límite de caracteres permitidos, obligatoriedad, tipos de datos, reglas de negocio).</p> <p>El usuario da clic en la opción "Guardar". Si cada dato es correcto el sistema persiste la Posible Mejora en la respectiva tabla de la base de datos y se direcciona a la Página de Manejo de Posibles Mejoras mostrando los nuevos datos en la lista de Posibles Mejoras asociada al Requerimiento seleccionado, y un mensaje que dice "Grabación exitosa". El usuario puede buscar, editar y eliminar las Posibles Mejoras creadas, para lo cual se incluye los casos de uso "Buscar datos", "Editar datos" y "Eliminar datos" respectivamente.</p>

	Alternativos	<ul style="list-style-type: none"> • Dato incorrecto: El sistema muestra al lado del dato incorrecto un mensaje que informa el tipo de error. • El usuario cancela la creación de una Posible Mejora: El sistema se direcciona a la Página de Manejo de Posibles Mejoras mostrando la lista de Posibles Mejoras asociada al Requerimiento seleccionado. • Excepción al guardar Posible Mejora: El sistema captura el tipo de error y se direcciona a la misma Página de Creación de Posible Mejora mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones		<ul style="list-style-type: none"> • Se visualiza los nuevos datos en la lista de Posible Mejoras de la Página de Manejo de Posibles Mejoras.

Tabla 3.42: (Especificación del caso de uso “Identificar Posible Mejora Potencial”)

Especificación de Caso de Uso		
Código	35	
Caso de Uso	Priorizar los Requerimientos	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar viendo la lista de Proyectos en la Página de Manejo de Proyectos, para esta acción se incluye el caso de uso “Listar datos iniciales de Proyectos”. 	
Flujo de Eventos	Básico	<p>El usuario da clic en la opción “Administrar” de la lista de Proyectos en la Página de Manejo de Proyectos, y el sistema se direcciona a la Página de Administración de Proyectos mostrando las respectivas opciones como se observa en la Figura 3.8. El usuario ingresa a la opción “Análisis del Sistema / Priorización de Requerimientos”, y el sistema genera una lista de requerimientos con estado “Aceptado” y ordenada por prioridad, y verifica si la lista está vacía. Si no está vacía el sistema muestra esta lista en una tabla con los columnas “Prioridad”, “Código”, “Descripción”, “Rango de importancia”, “Rango de satisfacción actual”, y dos opciones en cada fila (Subir, Bajar) para ordenar la prioridad; la opción “Subir” no se debe visualizar en la primera fila, y la opción “Bajar” no se debe visualizar en la última fila; esta tabla con sus respectivas opciones se muestra en la Página de Priorización de Requerimientos como se observa en la Figura 3.26.</p> <p>El usuario da clic en la opción “Subir” de una fila, y el sistema intercambia los datos de la fila actual con los datos de la fila anterior; el usuario da clic en la opción “Bajar”, y el sistema intercambia los datos de la fila actual con los datos de la siguiente fila. El usuario da clic en la opción “Guardar”, y el sistema actualiza cada Requerimiento en la base de datos.</p>
	Alternativos	<ul style="list-style-type: none"> • La lista de Requerimientos está vacía: El sistema muestra en la Página de Priorización de Requerimientos un mensaje que dice “No se encontraron registros”. • Excepción al subir o bajar un Requerimiento: El sistema captura el tipo de error y se direcciona a la misma Página de Priorización de Requerimientos mostrando un mensaje que informa el tipo de error capturado. • Excepción al actualizar los Requerimientos: El sistema captura el tipo de error y se direcciona a la misma Página de Priorización de Requerimientos mostrando un mensaje que informa el tipo de error capturado.
Pos condiciones	<ul style="list-style-type: none"> • Se visualiza la lista de Requerimientos con estado “Aceptado” ordenados por prioridad (de menor a mayor), en la Página de Priorización de Requerimientos. 	

Tabla 3.43: (Especificación del caso de uso “Priorizar los Requerimientos”)

3.2.2 Análisis de Robustez

Para iniciar esta fase se debe disponer del primer proyecto de casos de uso (especificación de cada caso de uso a nivel de borrador) que será utilizado en cada diagrama de robustez.

3.2.2.1 Diagrama de robustez por cada caso de uso

En la siguiente tabla se muestra un listado de todos los casos de uso de los cuales se realizó su respectiva especificación, y ahora por cada caso de uso se realizará un diagrama de robustez.

Id	Caso de uso
01	Iniciar sesión
02	Cerrar sesión
03	Manejar cuenta de usuario del sistema
04	Listar datos asociados a un Proyecto
05	Listar datos iniciales de Proyectos
06	Listar datos asociados a un Requerimiento
07	Filtrar registros listados
08	Editar registro seleccionado de una lista
09	Eliminar registro seleccionado de una lista
10	Manejar datos iniciales del Proyecto
11	Manejar datos asociados a un Proyecto
12	Manejar datos asociados a un Requerimiento
13	Quitar ítem de una lista de datos
14	Manejar Anexo o archivo del Proyecto
15	Generar reporte
16	Listar parámetros generales del sistema
17	Manejar parámetros generales del sistema
18	Manejar Persona de Desarrollo
19	Actualizar datos adicionales al Proyecto
20	Conformar Equipo de Desarrollo
21	Asignar Personas de Desarrollo al Proyecto
22	Realizar Análisis de Metas del Proyecto
24	Manejar Actividad por Proceso
26	Identificar Responsable por Actividad
27	Asignar Responsables a una Actividad
28	Realizar identificación macro de Procesos
30	Mostrar información actual del Requerimiento
31	Actualizar datos adicionales al Requerimiento
34	Identificar Posible Mejora Potencial
35	Priorizar los Requerimientos

Tabla 3.44: (Listado de Casos de Uso)

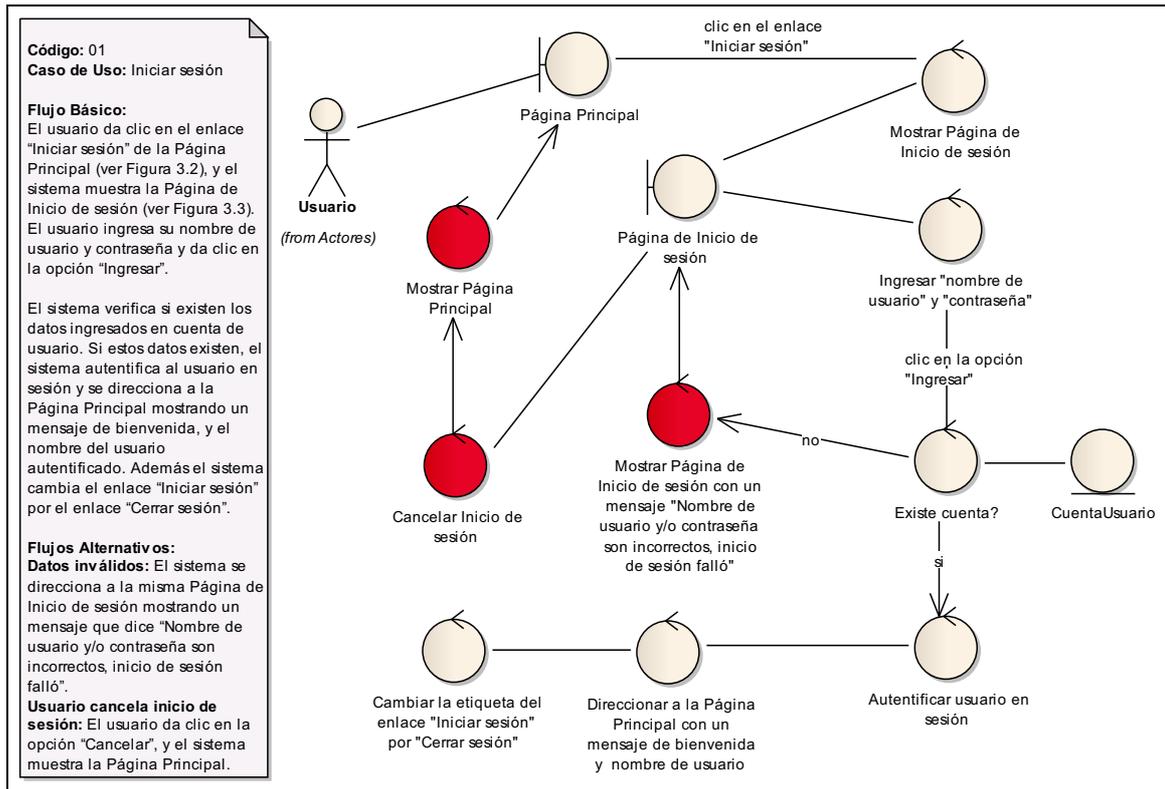


Figura 3.30: (Diagrama de Robustez del caso de uso "Iniciar sesión")

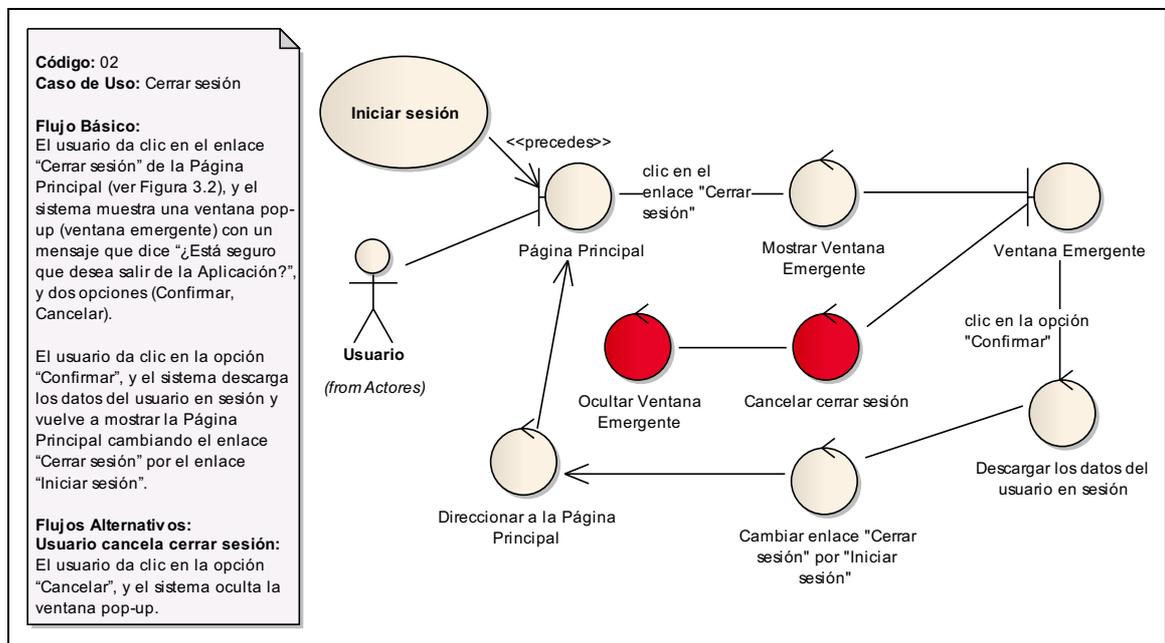


Figura 3.31: (Diagrama de Robustez del caso de uso "Cerrar sesión")

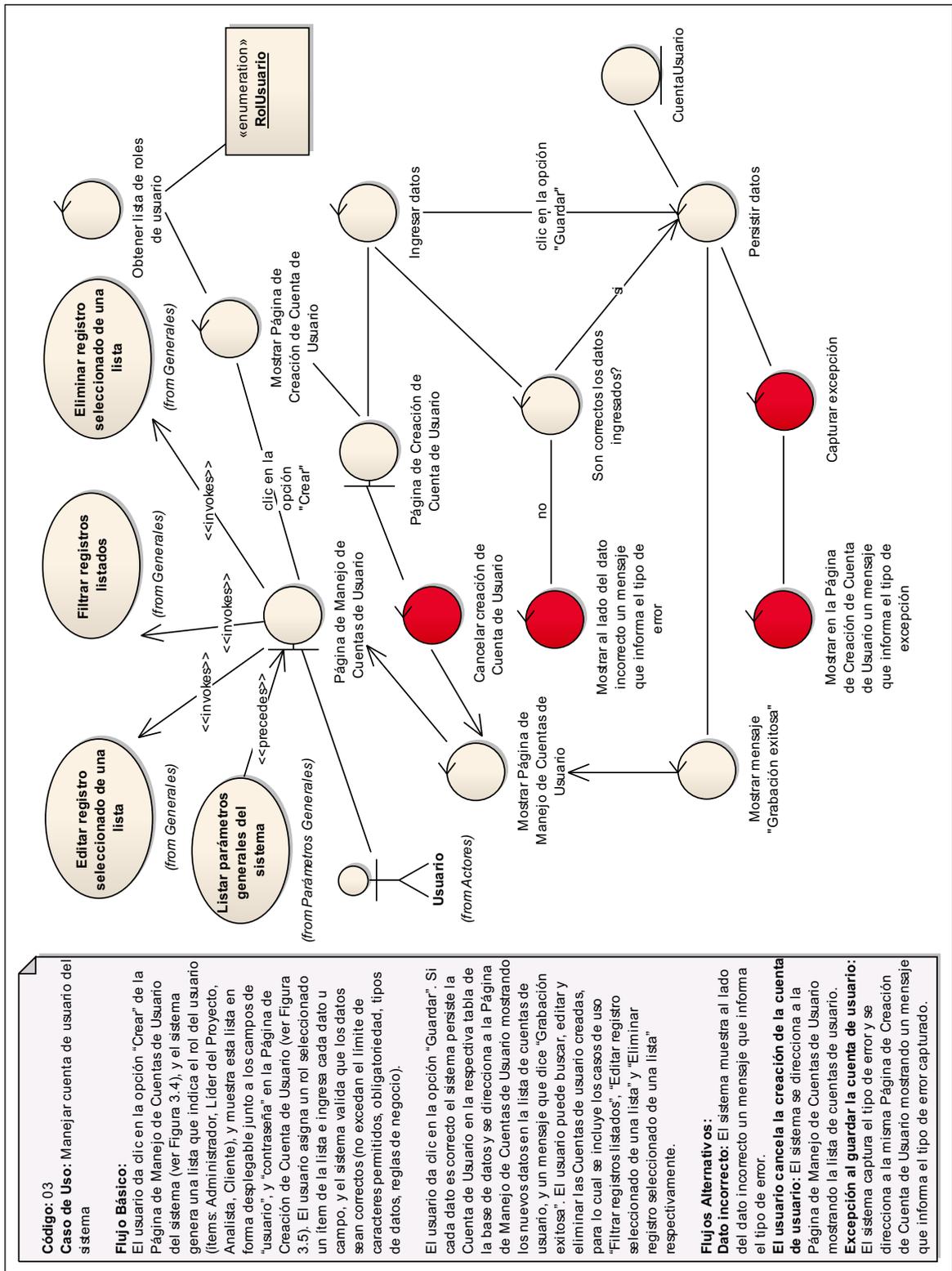


Figura 3.32: (Diagrama de Robustez del caso de uso “Manejar cuenta de usuario del sistema”)

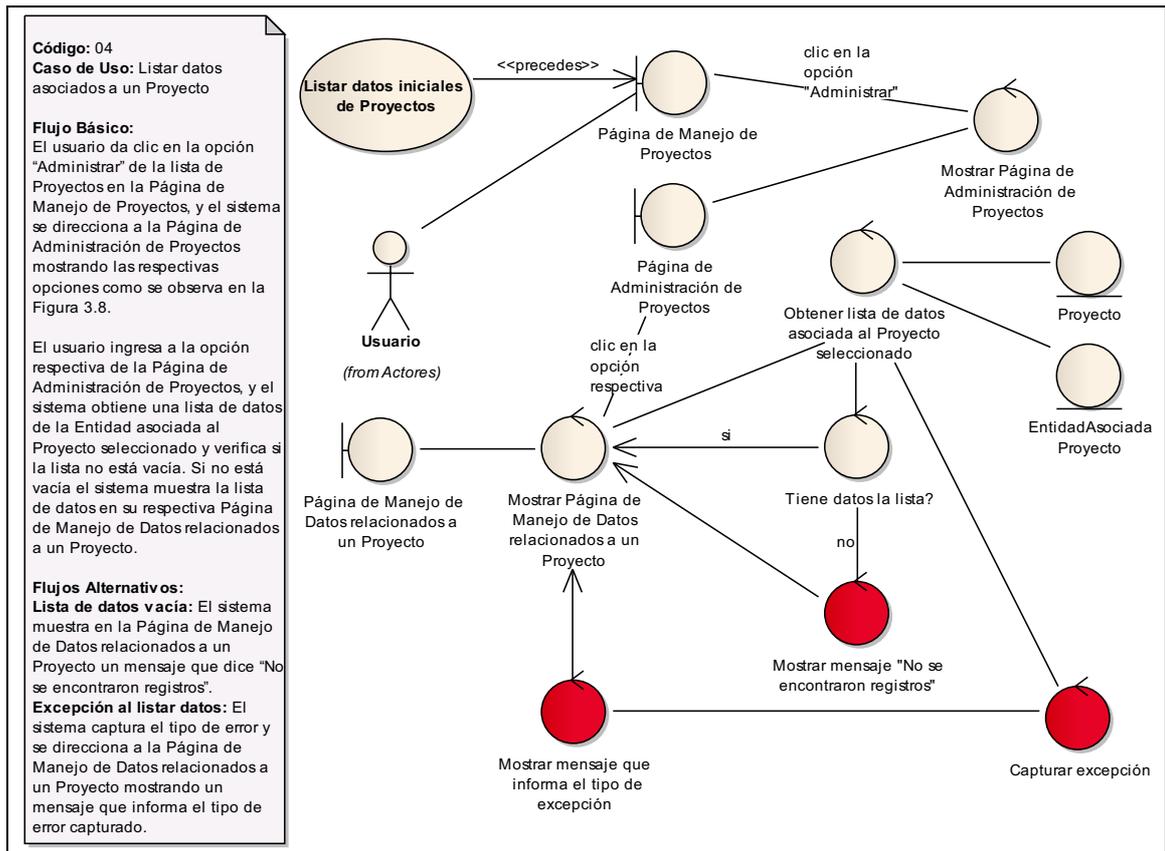


Figura 3.33: (Diagrama de Robustez del caso de uso "Listar datos asociados a un Proyecto")

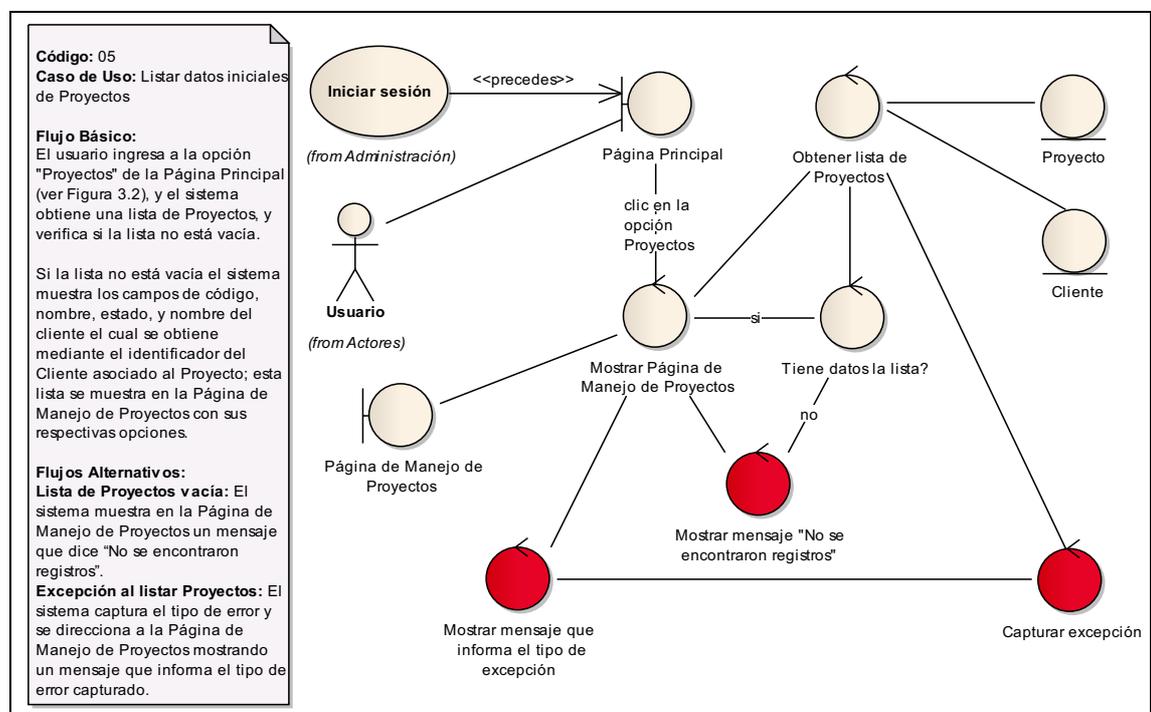


Figura 3.34: (Diagrama de Robustez del caso de uso "Listar datos iniciales de Proyectos")

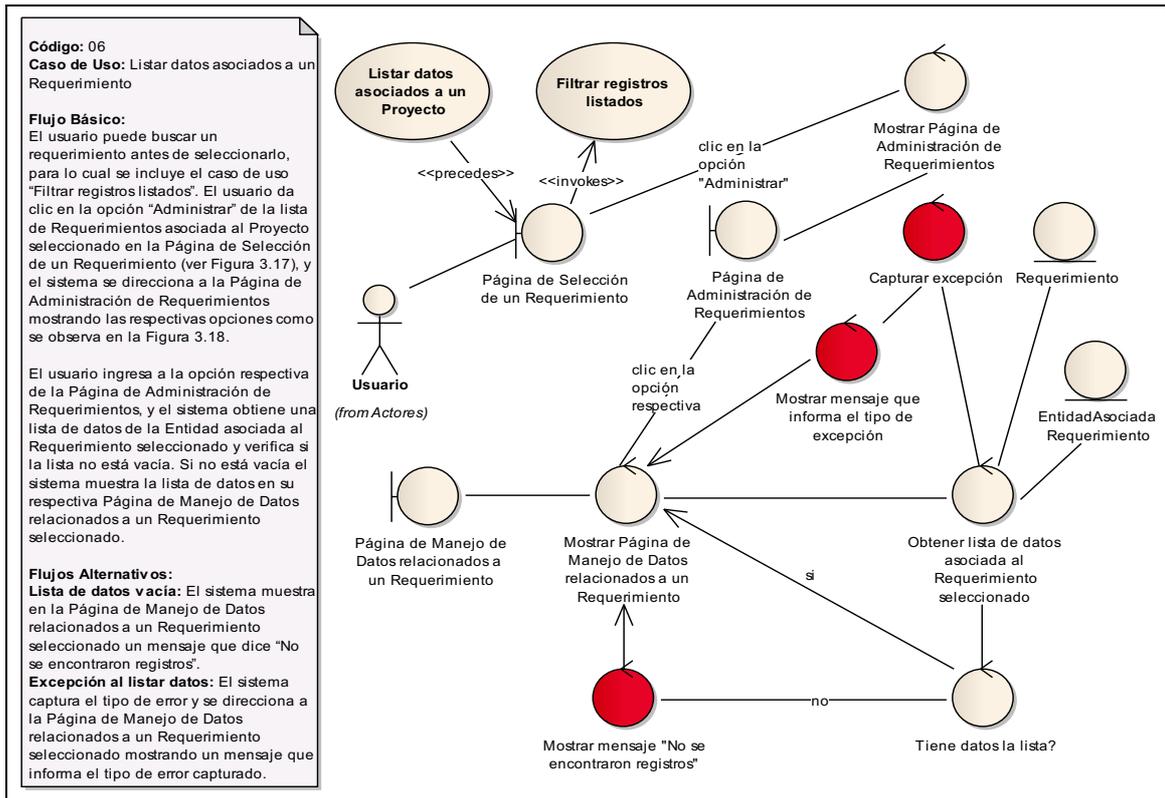


Figura 3.35: (Diagrama de Robustez del caso de uso "Listar datos asociados a un Requerimiento")

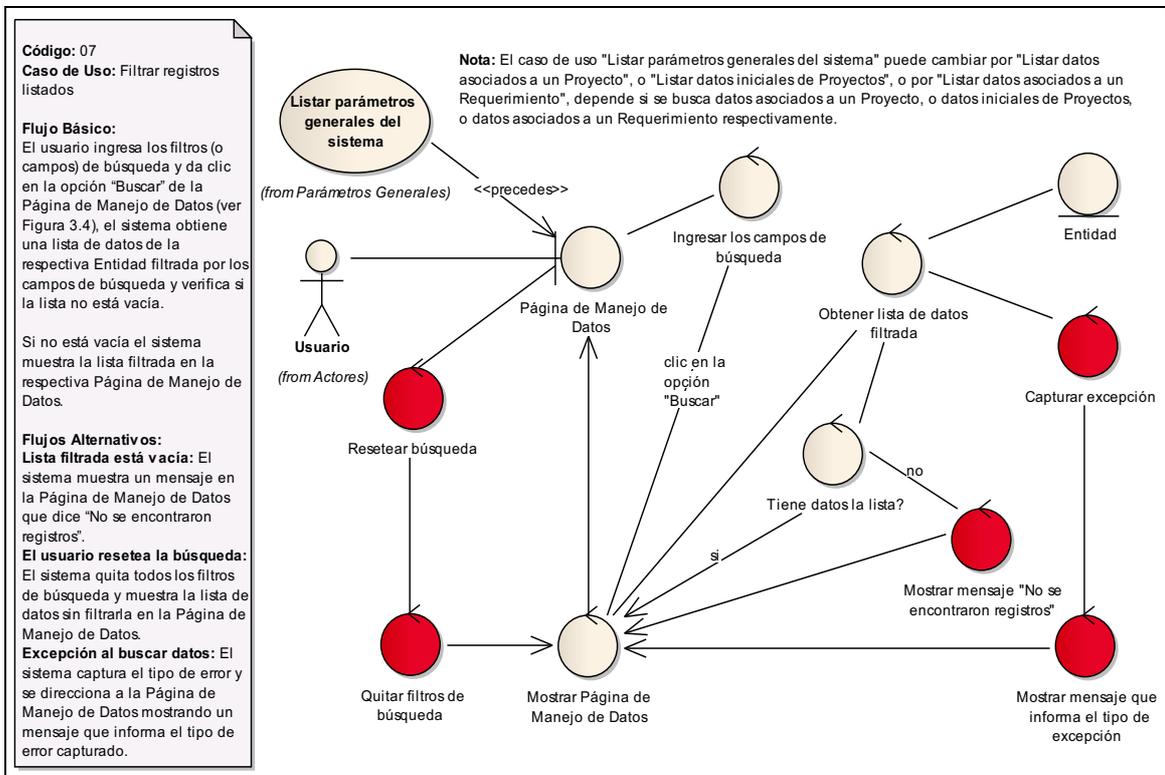


Figura 3.36: (Diagrama de Robustez del caso de uso "Filtrar registros listados")

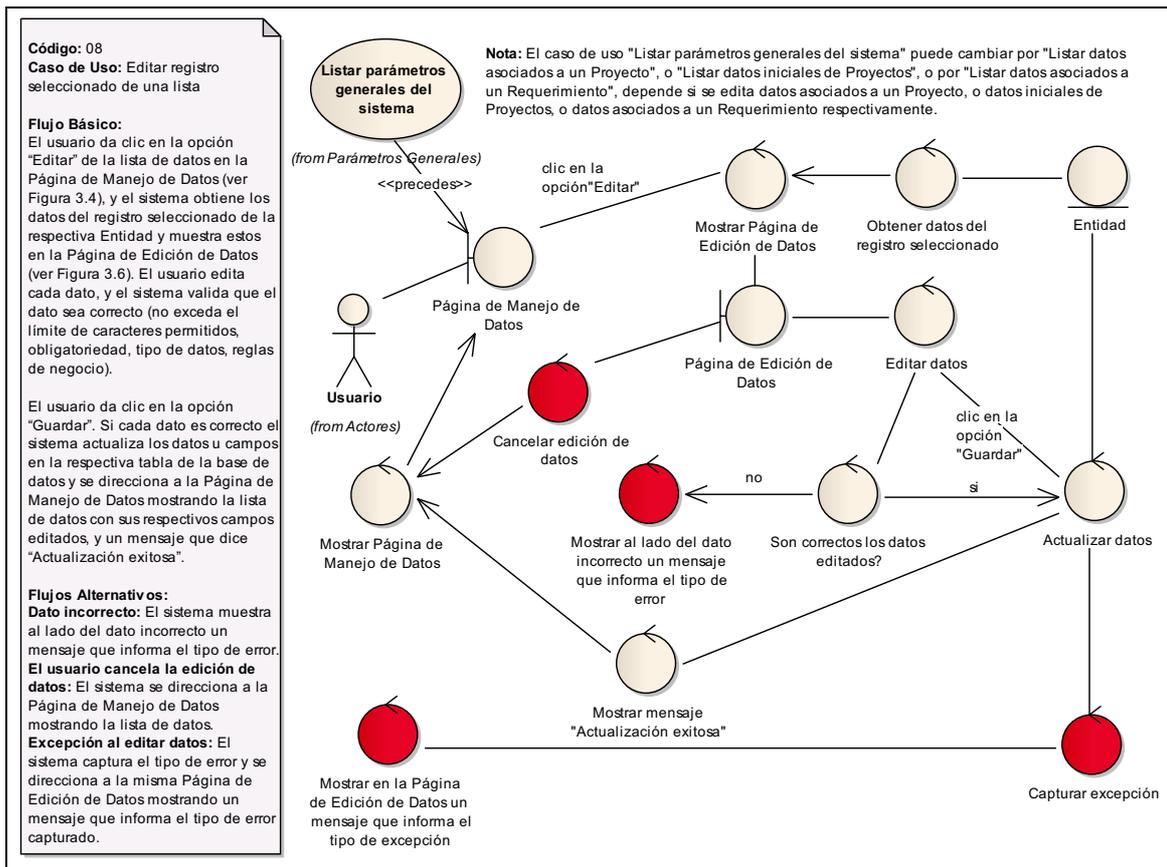


Figura 3.37: (Diagrama de Robustez del caso de uso "Editar registro seleccionado de una lista")

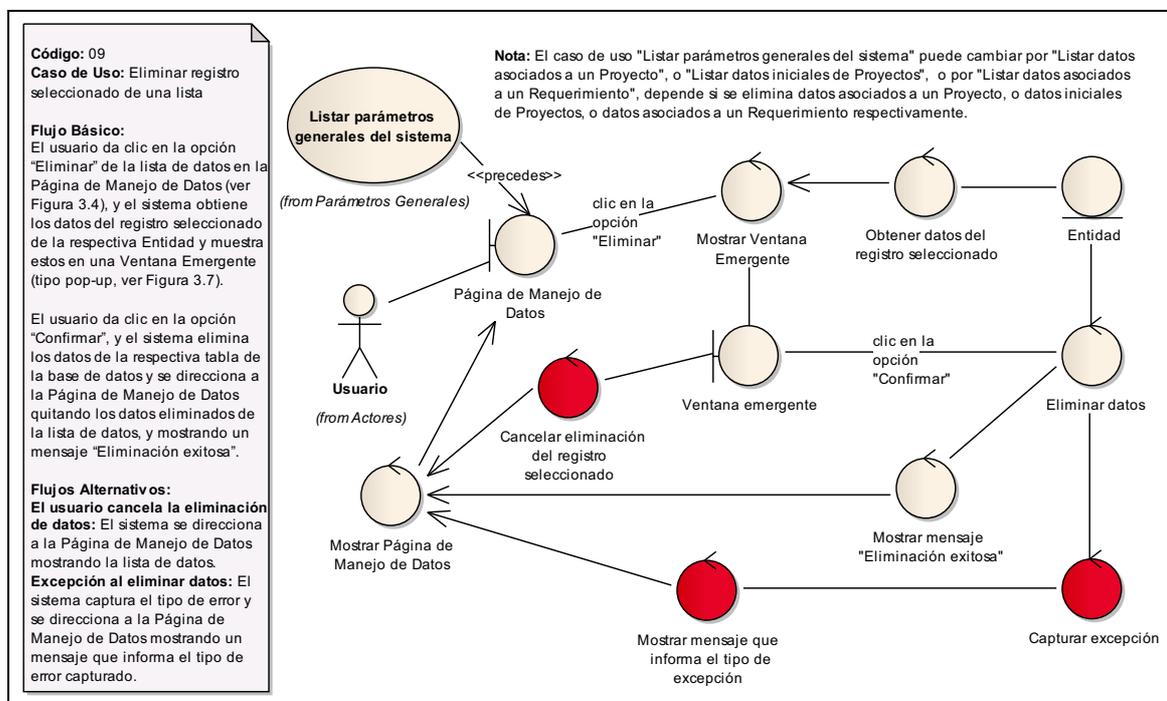


Figura 3.38: (Diagrama de Robustez del caso de uso "Eliminar registro seleccionado de una lista")

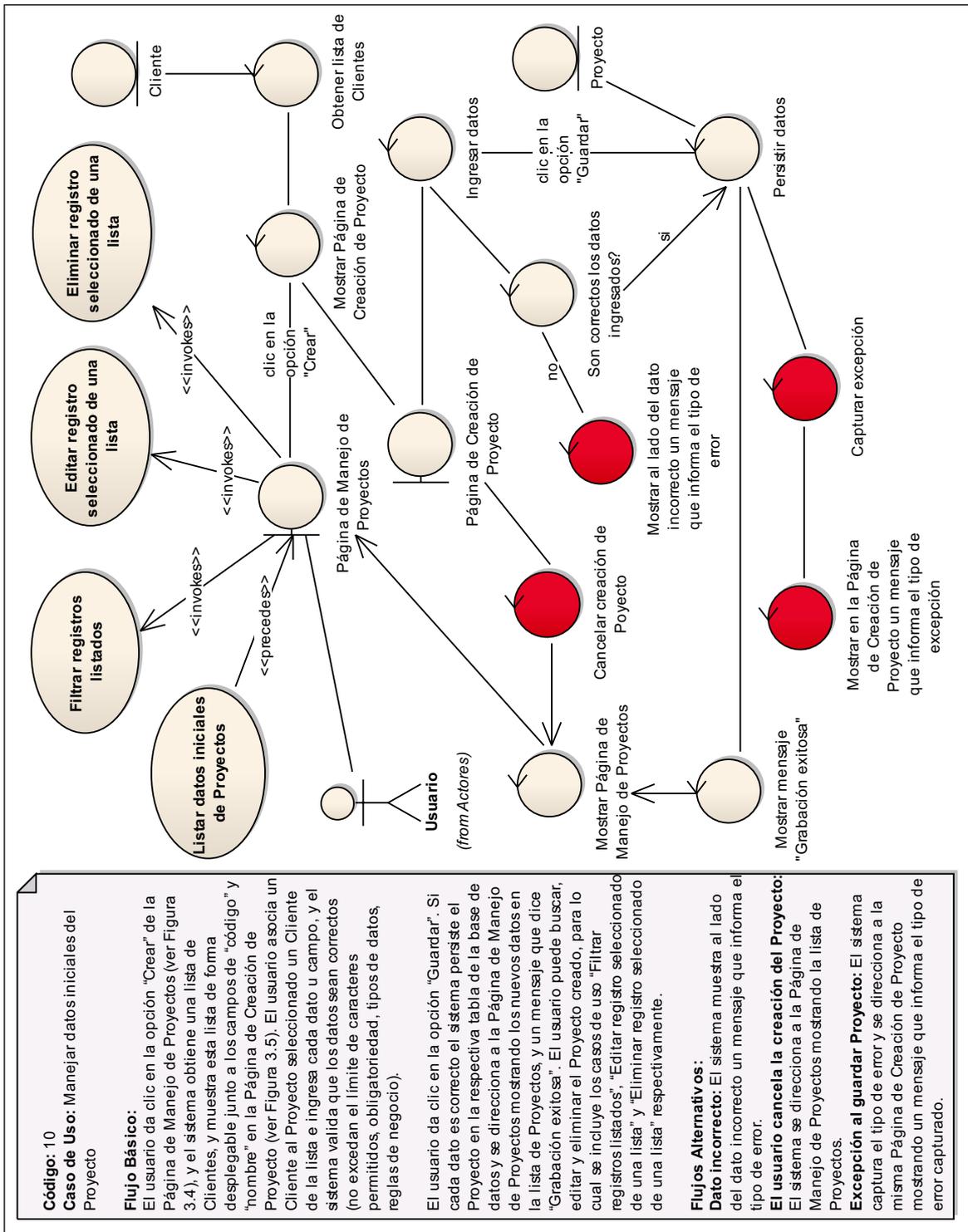


Figura 3.39: (Diagrama de Robustez del caso de uso “Manejar datos iniciales del Proyecto”)

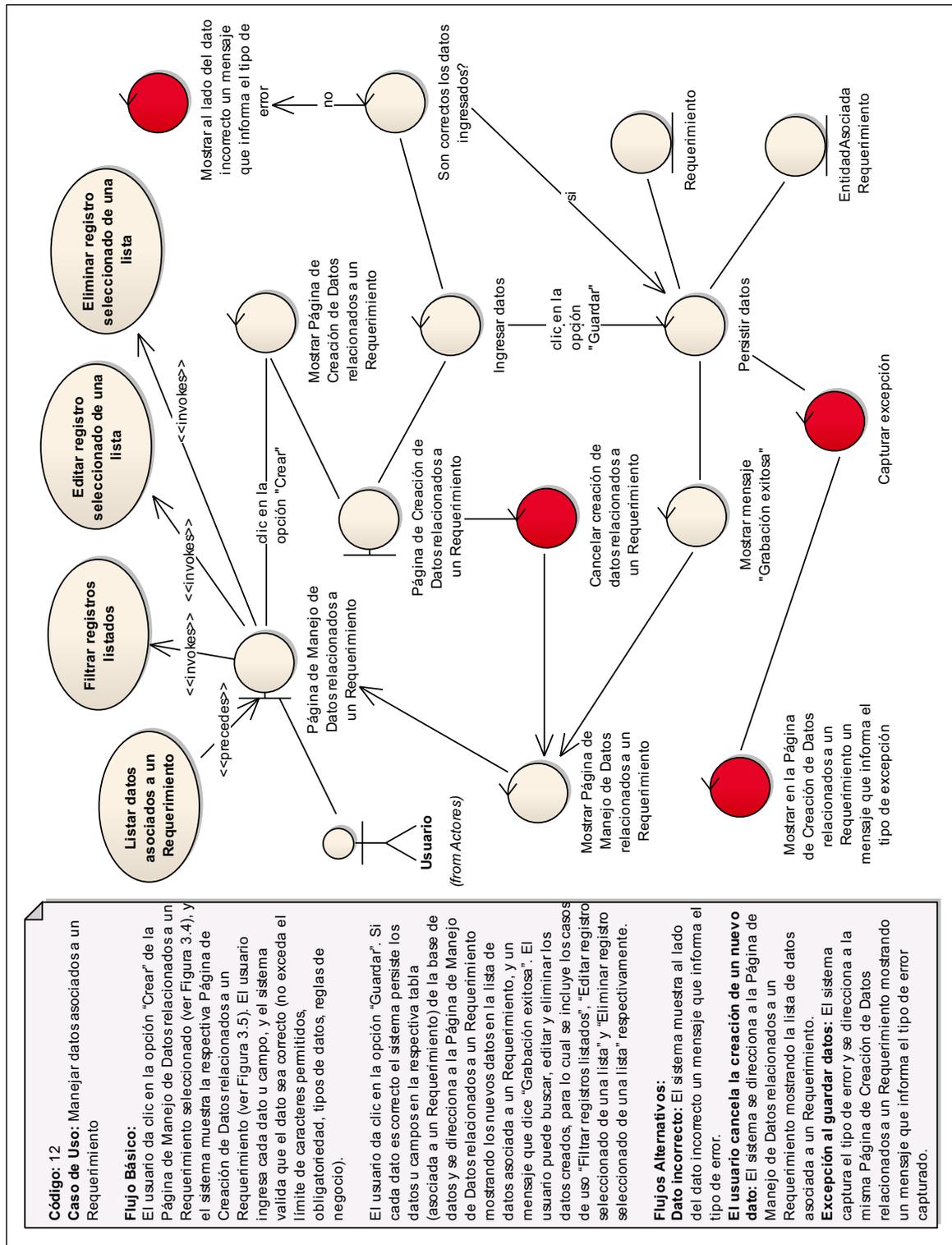


Figura 3.41: (Diagrama de Robustez del caso de uso “Manejar datos asociados a un Requerimiento”)

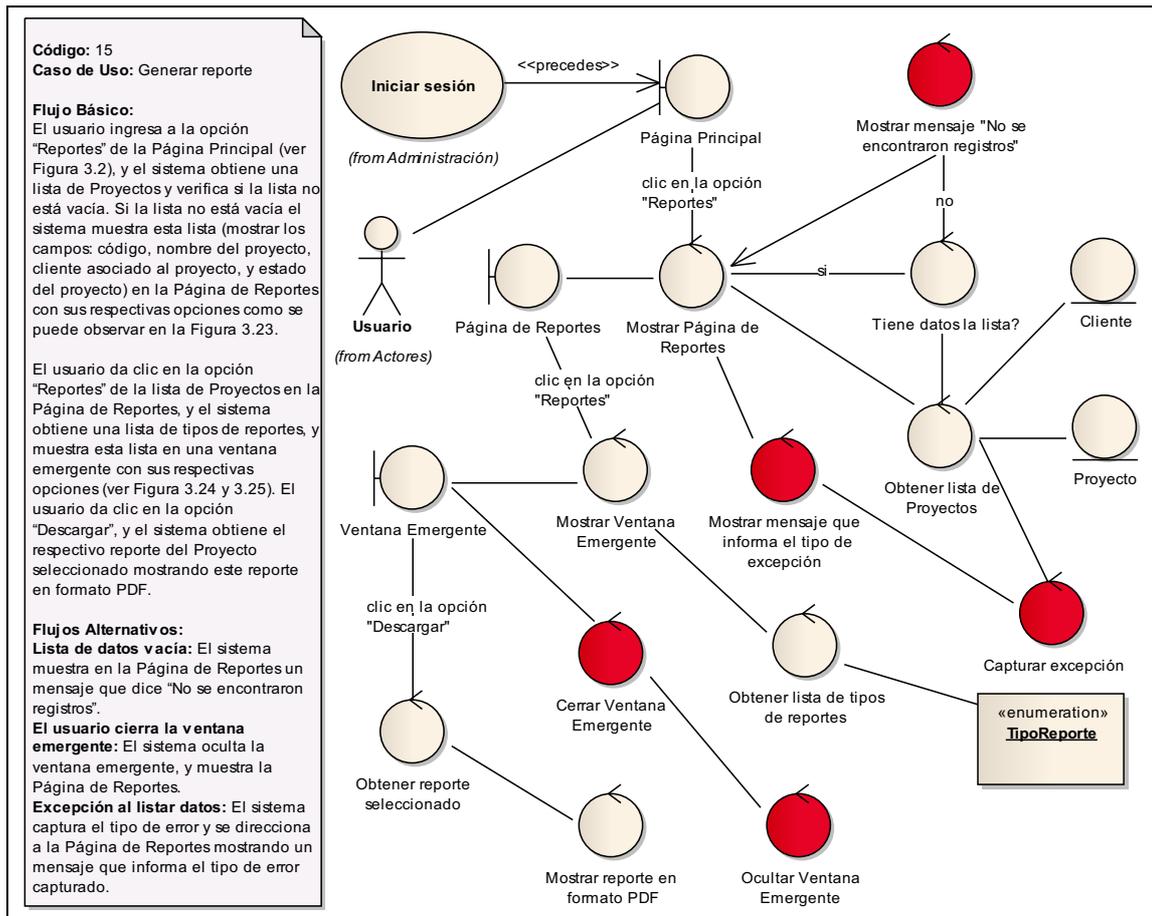


Figura 3.44: (Diagrama de Robustez del caso de uso "Generar reporte")

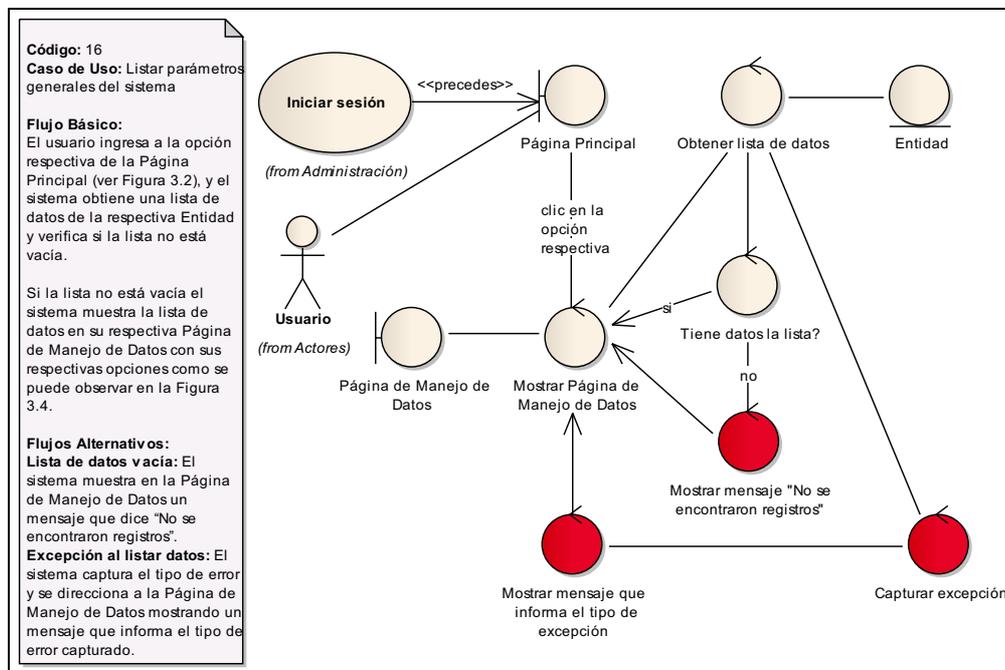


Figura 3.45: (Diagrama de Robustez del caso de uso "Listar parámetros generales del sistema")

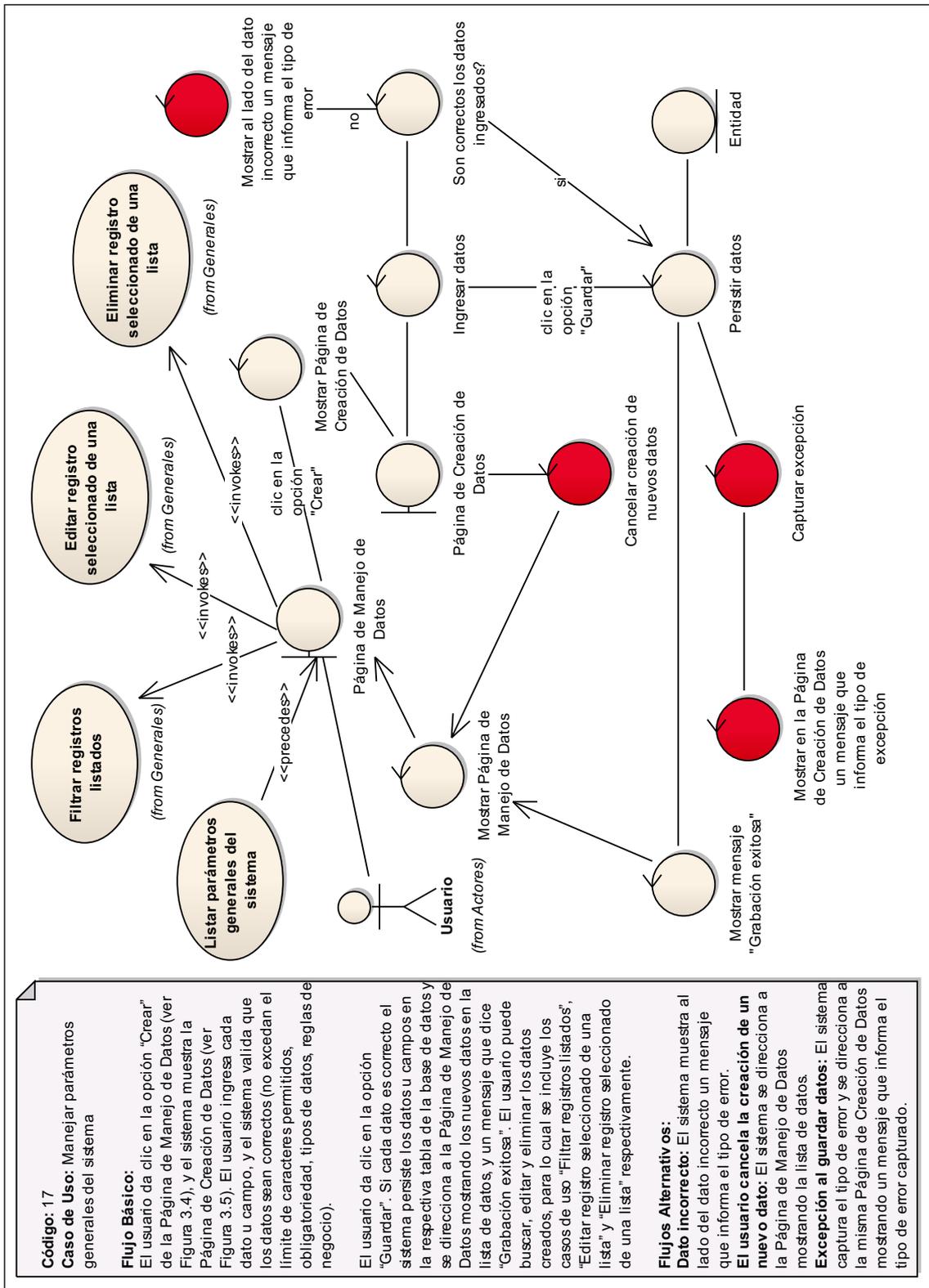


Figura 3.46: (Diagrama de Robustez del caso de uso “Manejar parámetros generales del sistema”)

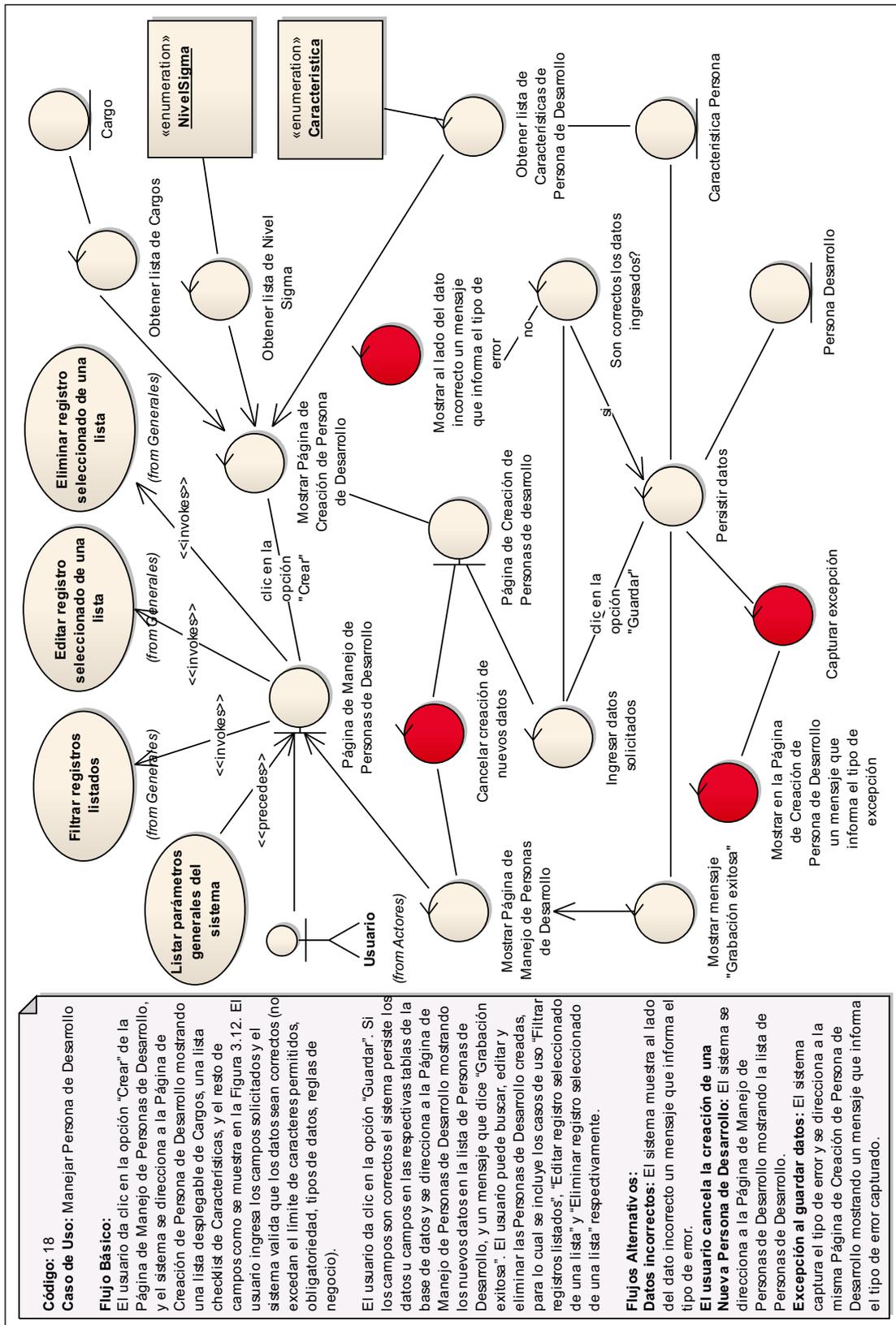


Figura 3.47: (Diagrama de Robustez del caso de uso “Manejar Persona de Desarrollo”)

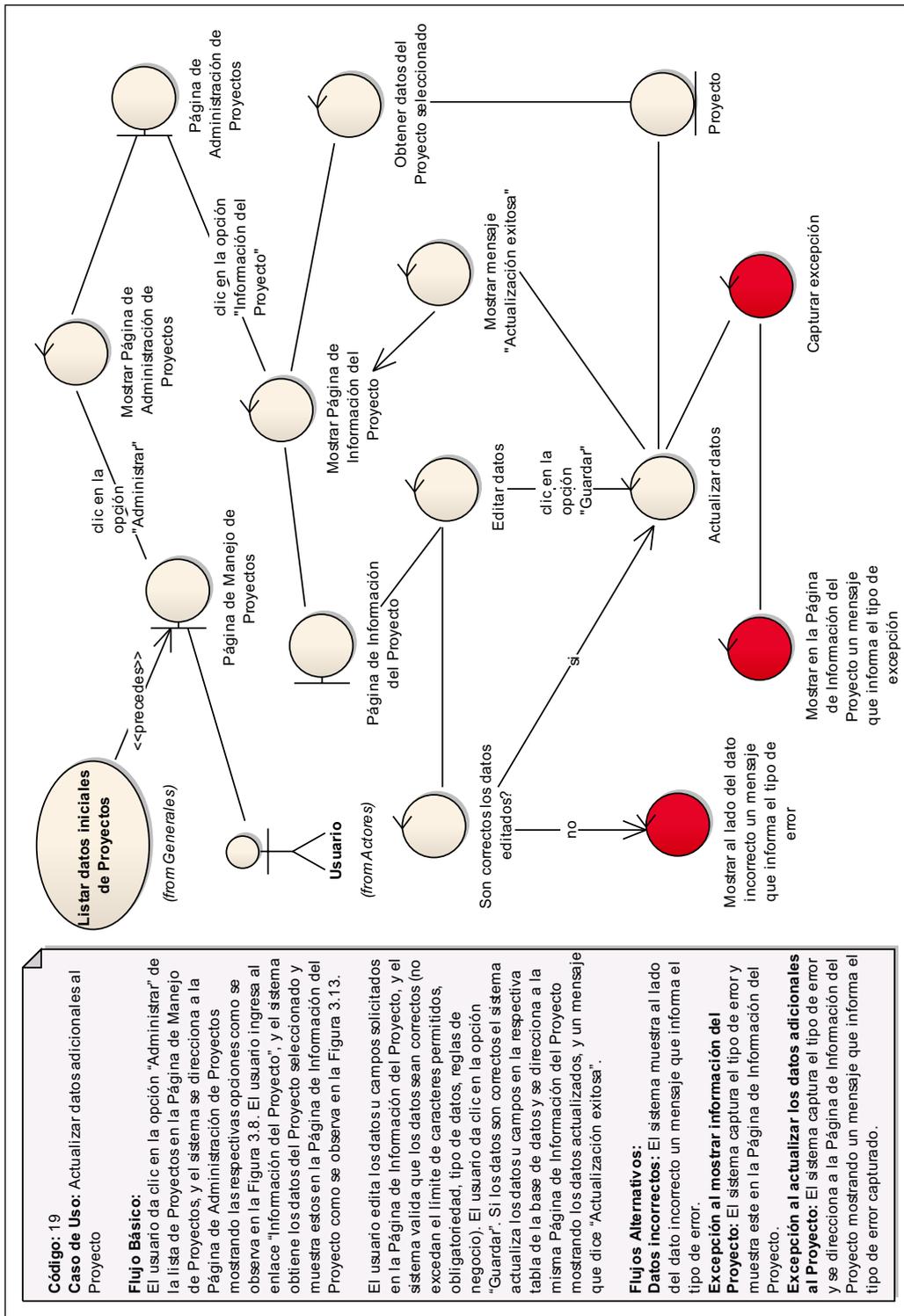


Figura 3.48: (Diagrama de Robustez del caso de uso “Actualizar datos adicionales al Proyecto”)

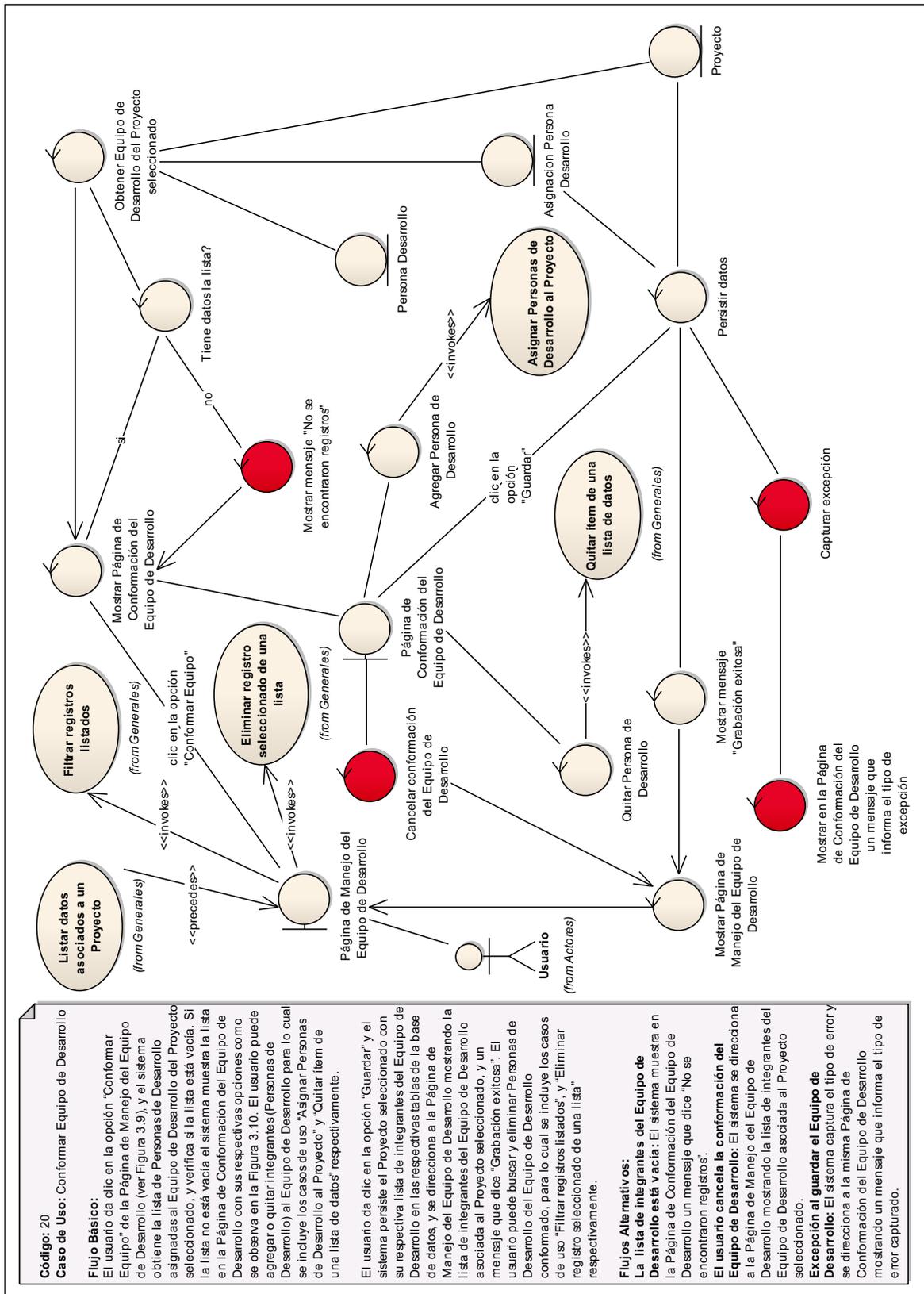


Figura 3.49: (Diagrama de Robustez del caso de uso “Conformar Equipo de Desarrollo”)

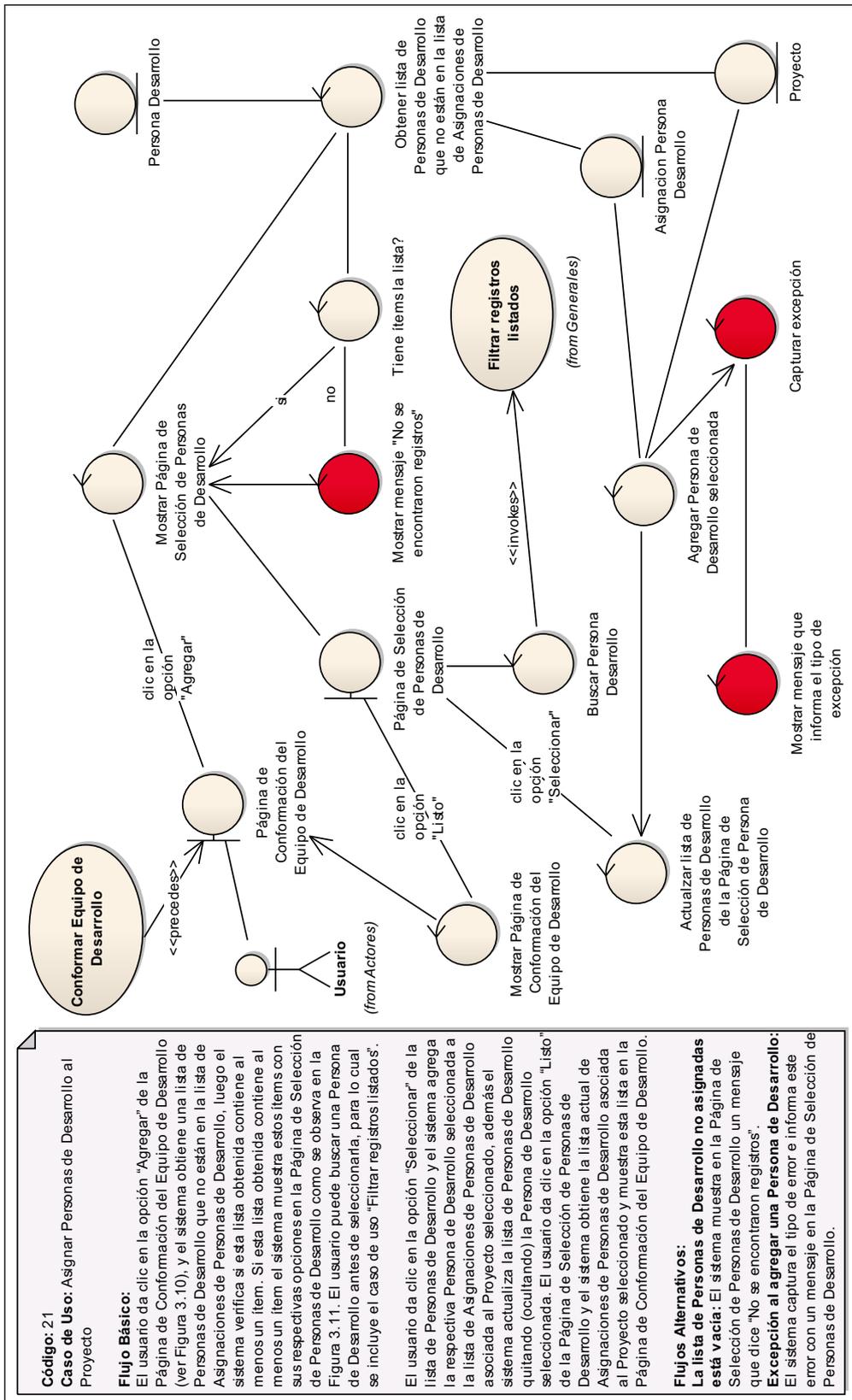


Figura 3.50: (Diagrama de Robustez del caso de uso “Asignar Personas de Desarrollo al Proyecto”)

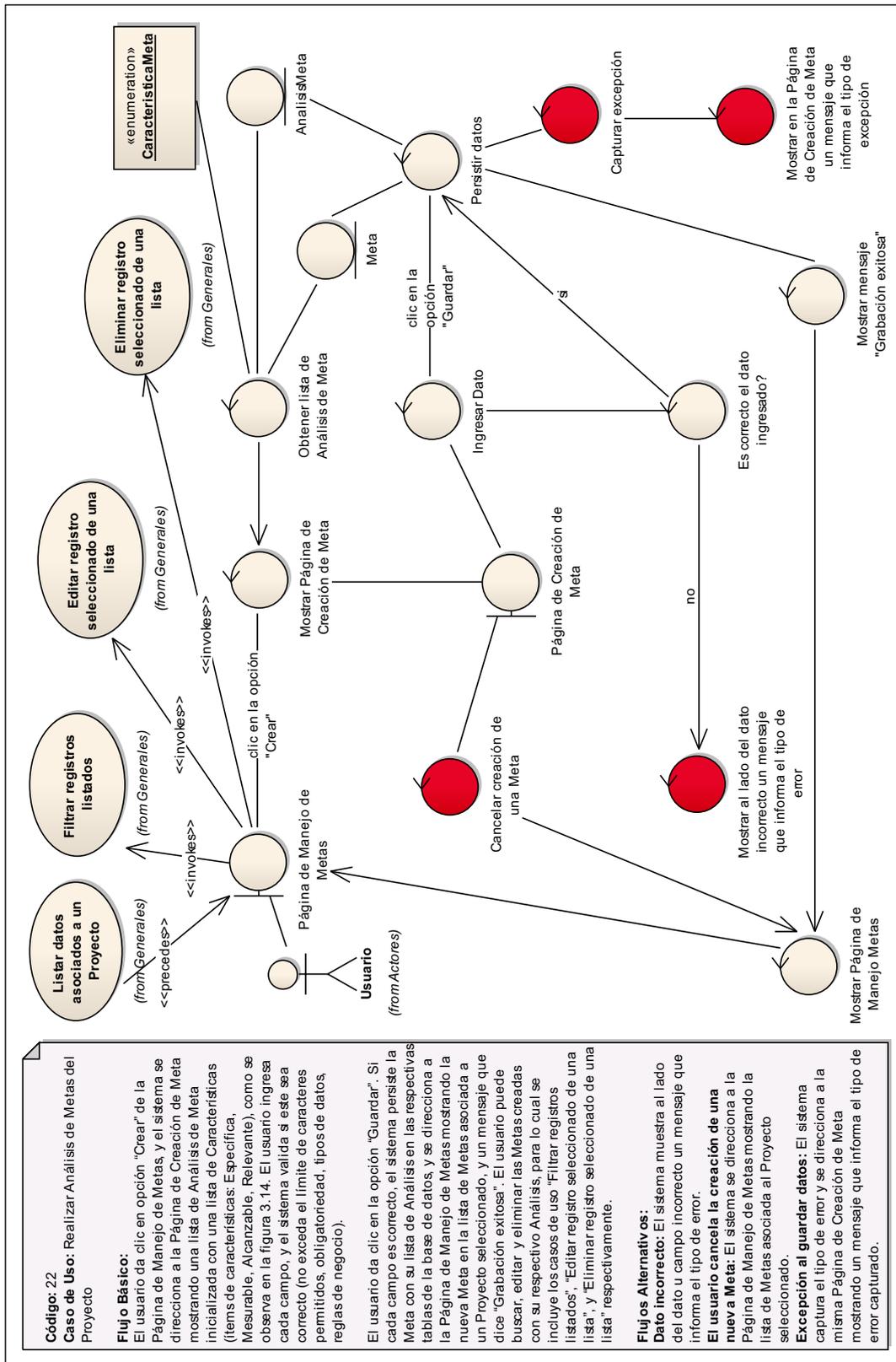


Figura 3.51: (Diagrama de Robustez del caso de uso “Realizar Análisis de Metas del Proyecto”)

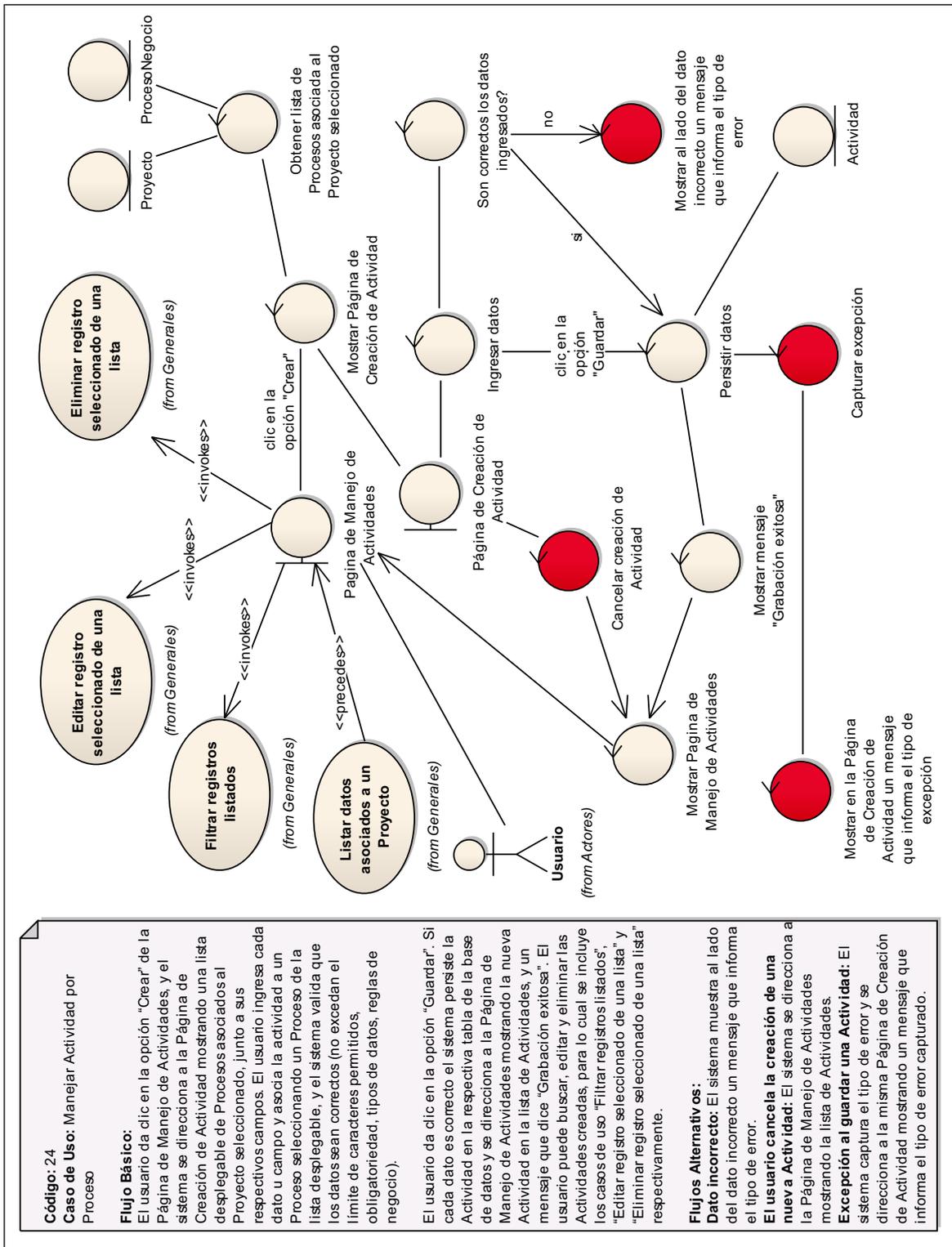


Figura 3.52: (Diagrama de Robustez del caso de uso “Manejar Actividad por Proceso”)

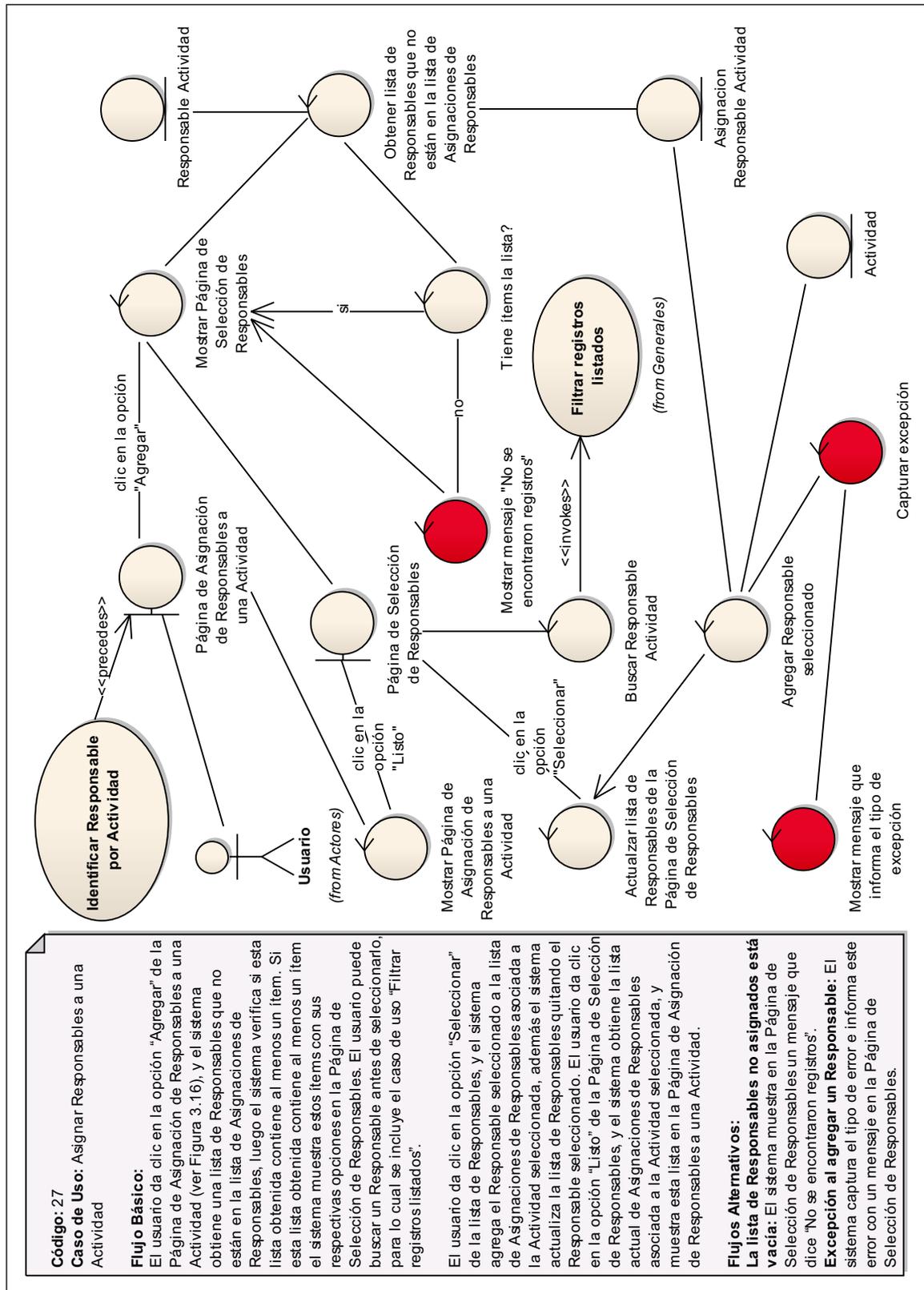


Figura 3.54: (Diagrama de Robustez del caso de uso "Asignar Responsables a una Actividad")

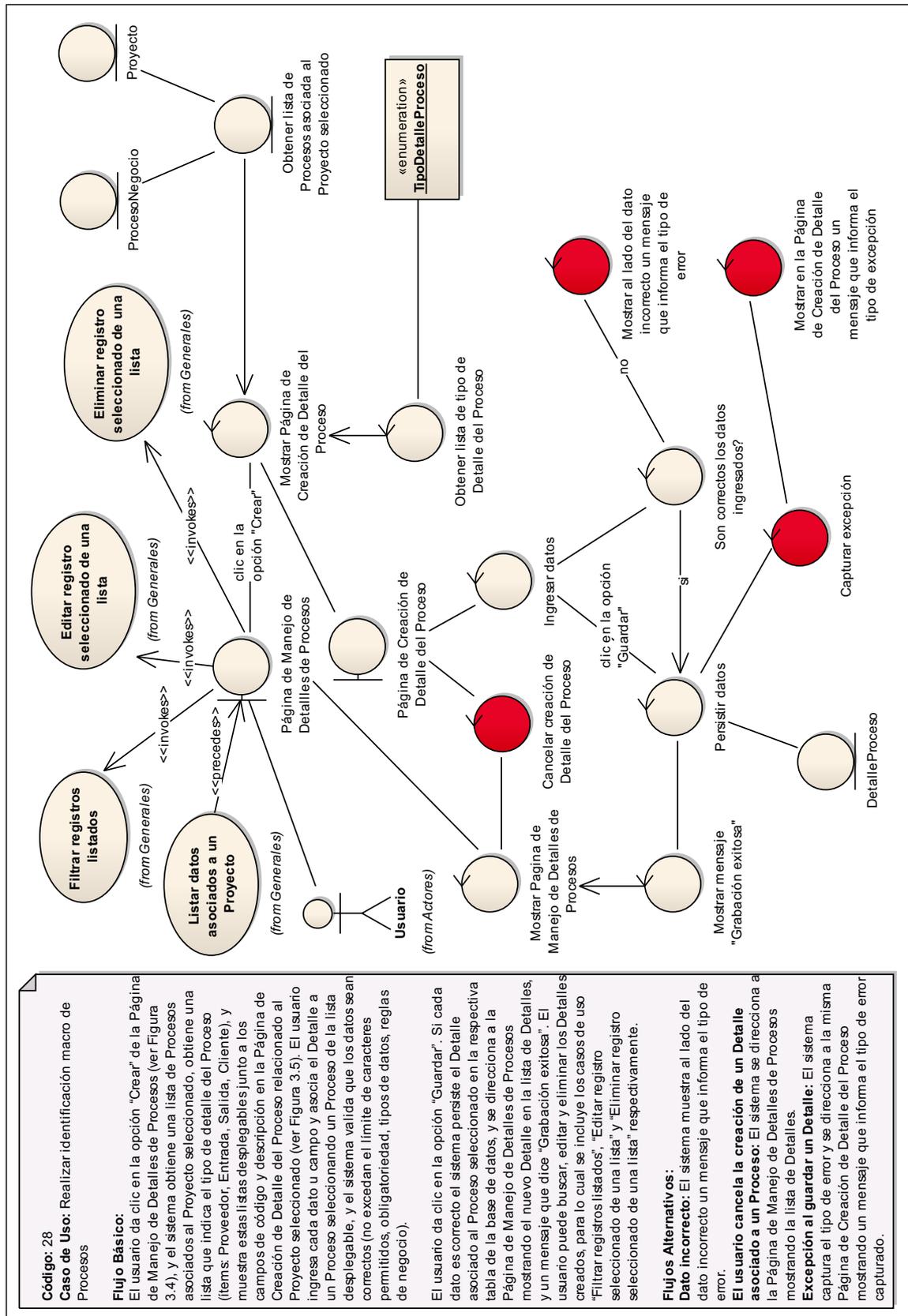


Figura 3.55: (Diagrama de Robustez del caso de uso “Realizar identificación macro de Procesos”)

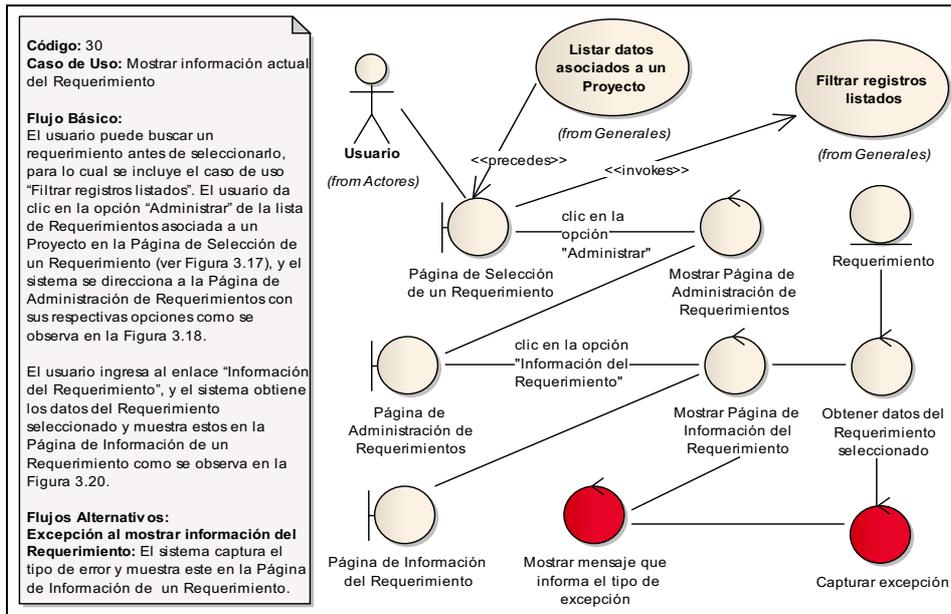


Figura 3.56: (Diagrama de Robustez del caso de uso "Mostrar información actual del Requerimiento")

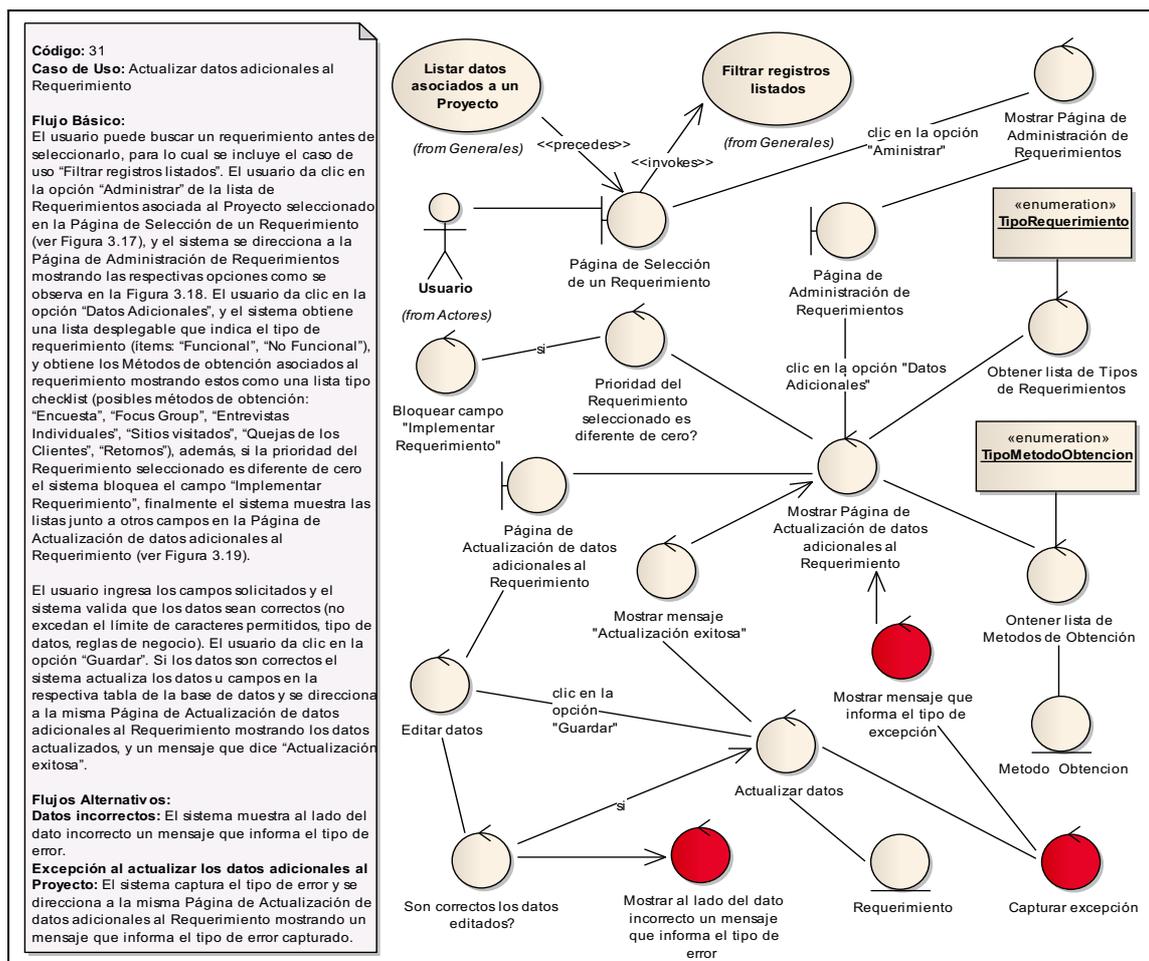


Figura 3.57: (Diagrama de Robustez del caso de uso "Actualizar datos adicionales al Requerimiento")

3.2.3 Modelo de dominio actualizado

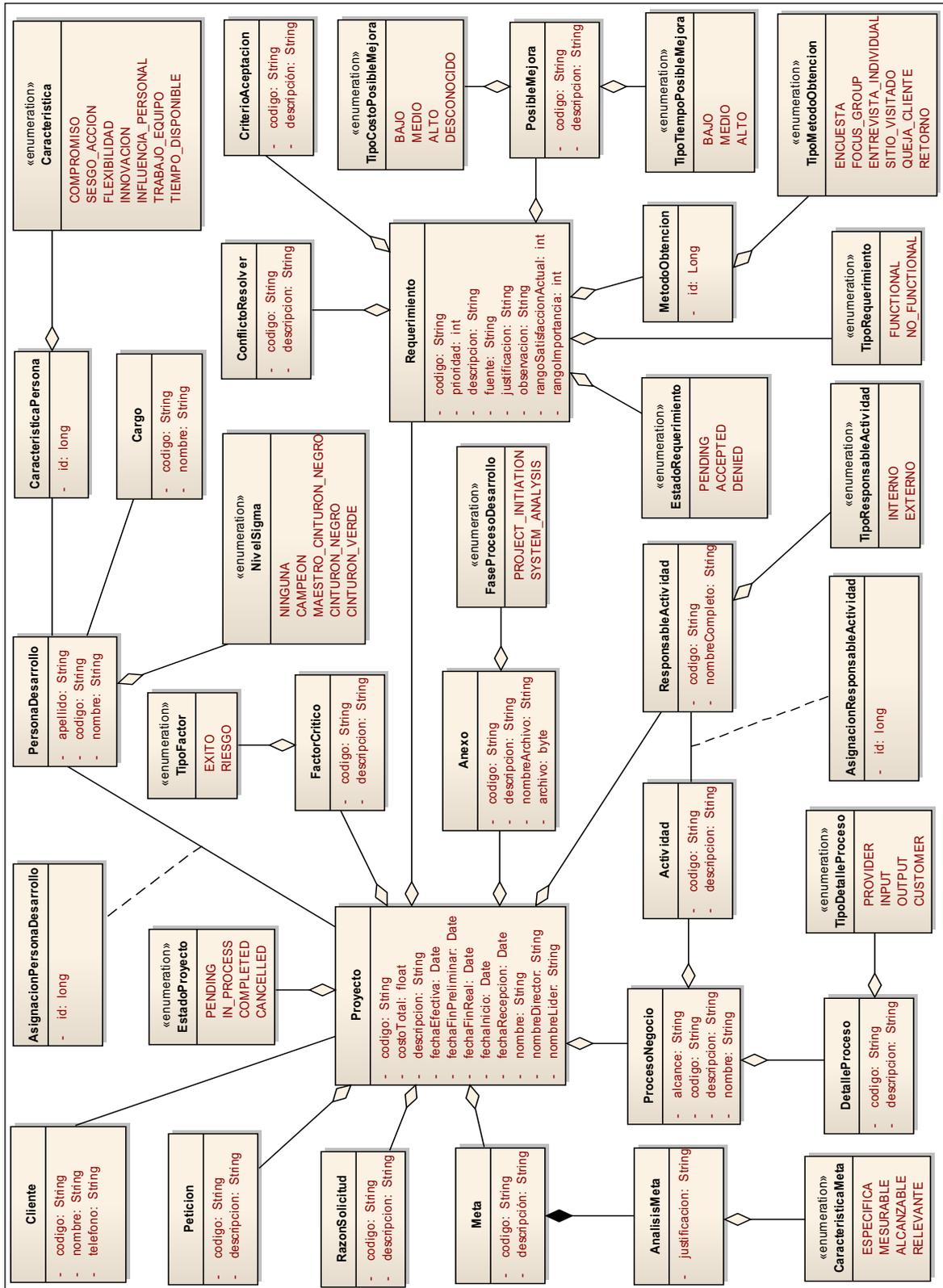


Figura 3.60: (Modelo de Dominio actualizado)

3.3 Diseño Detallado

3.3.1 Asignación de comportamiento

3.3.1.1 Diagrama de secuencia por cada caso de uso.

En la siguiente tabla se muestra un listado de todos los casos de uso de los cuales se realizó su respectiva especificación, y ahora por cada caso de uso se realizará un diagrama de secuencia.

Id	Caso de uso
01	Iniciar sesión
02	Cerrar sesión
03	Manejar cuenta de usuario del sistema
04	Listar datos asociados a un Proyecto
05	Listar datos iniciales de Proyectos
06	Listar datos asociados a un Requerimiento
07	Filtrar registros listados
08	Editar registro seleccionado de una lista
09	Eliminar registro seleccionado de una lista
10	Manejar datos iniciales del Proyecto
11	Manejar datos asociados a un Proyecto
12	Manejar datos asociados a un Requerimiento
13	Quitar ítem de una lista de datos
14	Manejar Anexo o archivo del Proyecto
15	Generar reporte
16	Listar parámetros generales del sistema
17	Manejar parámetros generales del sistema
18	Manejar Persona de Desarrollo
19	Actualizar datos adicionales al Proyecto
20	Conformar Equipo de Desarrollo
21	Asignar Personas de Desarrollo al Proyecto
22	Realizar Análisis de Metas del Proyecto
24	Manejar Actividad por Proceso
26	Identificar Responsable por Actividad
27	Asignar Responsables a una Actividad
28	Realizar identificación macro de Procesos
30	Mostrar información actual del Requerimiento
31	Actualizar datos adicionales al Requerimiento
34	Identificar Posible Mejora Potencial
35	Priorizar los Requerimientos

Tabla 3.45: (Listado de Casos de Uso)

Antes de realizar los diagramas de secuencia se detallará la arquitectura técnica del sistema, teniendo en cuenta que el software a desarrollar está basado en web, y se trabajará sobre la plataforma Java EE empleando un framework denominado JBoss Seam.

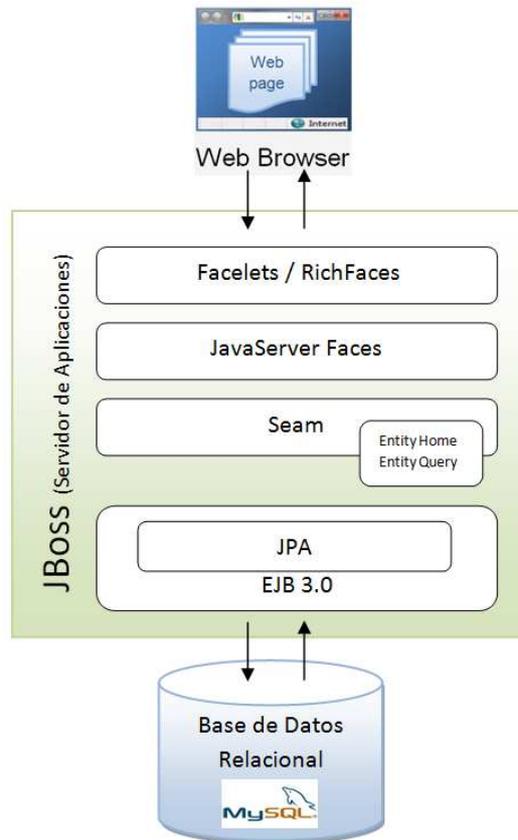


Figura 3.61: (Arquitectura Java EE con Seam)

Aspectos relevantes de la arquitectura:

- Al trabajar con **JavaServer Faces (JSF)** es recomendable usar **Facelets** como controlador de vista en lugar de **JavaServer Pages (JSP)**. Facelets es un framework open source para plantillas (templates) centrado en JSF, motivo por el cual se integran de manera muy fácil. Facelets no depende del contenedor web, el proceso de compilación es más rápido que JSP, soporta EL (Expression Language), además, provee templating, lo cual

implica reutilización de código, simplificación de desarrollo y facilidad en el mantenimiento de grandes aplicaciones.

- Se empleará **RichFaces** para añadir la capacidad de Ajax (Asynchronous JavaScript And XML - JavaScript asíncrono y XML) dentro de las aplicaciones JSF. RichFaces es un framework open source que provee una librería de componentes visuales para JSF, y permite la integración de funcionalidades Ajax en dichos componentes mediante el soporte de la librería Ajax4JSF.
- En un proyecto Seam las vistas de JSF terminan con el sufijo .xhtml, siendo esta extensión el sufijo por defecto utilizado por Facelets. En la Figura 3.62 se ilustra como una solicitud JSF entrante es procesada y traducida en un árbol de componentes de interfaz de usuario.

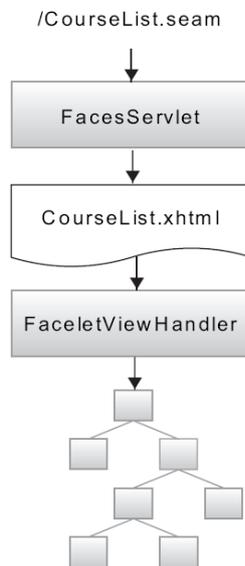


Figura 3.62: (Traducción de una solicitud JSF en un árbol de componentes de interfaz de usuario)
Gráfico del libro “Seam in Action”

- Seam proporciona un “framework de clases” (colección especializada de componentes plantillas) para construir controladores de página que realizan operaciones CRUD y consultas en instancias de entidades. El conjunto jerárquico de estas clases se ilustra en la siguiente figura.

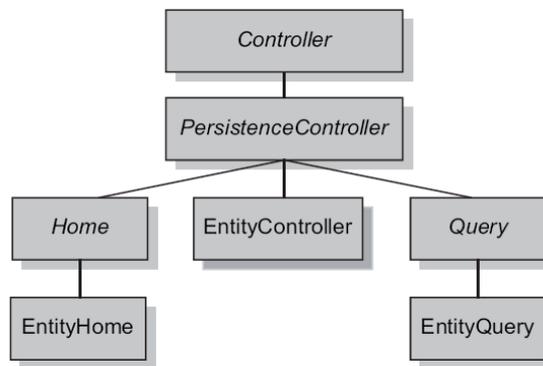


Figura 3.63: (Framework de Clases de Seam implementadas para JPA)

- **Controller**, clase base que contiene métodos de convivencia para el acceso a los contextos e instancias de componentes Seam que interactúan con el API Servlet y el ciclo de vida de JSF. **PersistenceController**, clase padre para el desarrollo de controladores para las páginas JSF, y diseñada para ser utilizada como un gestor de persistencia administrada por Seam.
- Las tres ramas de clases que extienden de PersistenceController (ver Figura 3.63) facilitan la interacción con el gestor de persistencia. Cada rama tiene una implementación para JPA (EntityHome, EntityQuery, y EntityController) y para Hibernate (HibernateEntityHome, HibernateEntityQuery, y HibernateEntityController). **Home**, administra una instancia de entidad única, y apoya las operaciones CRUD. **Query**, administra un conjunto de resultados de una consulta JPQL/HQL, y apoya las restricciones, ordenamiento, y paginación. Tener presente que estas

clases están destinadas a ser utilizadas como componentes **JavaBeans**.

En la Figura 3.64 y 3.65 se ilustra el diagrama de clases de EntityHome y EntityQuery respectivamente.

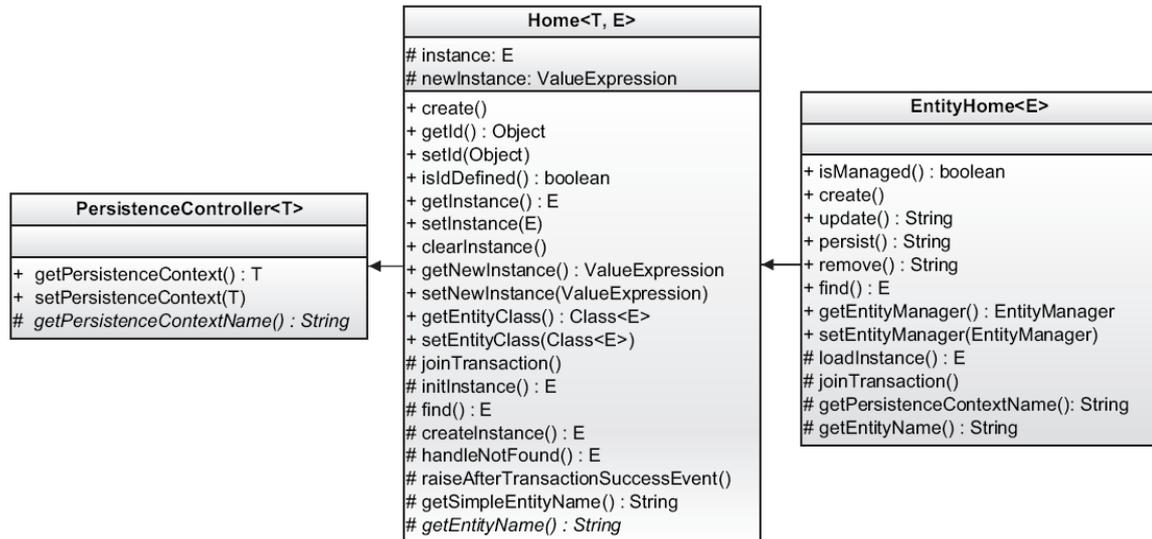


Figura 3.64: (Diagrama de clases de EntityHome)

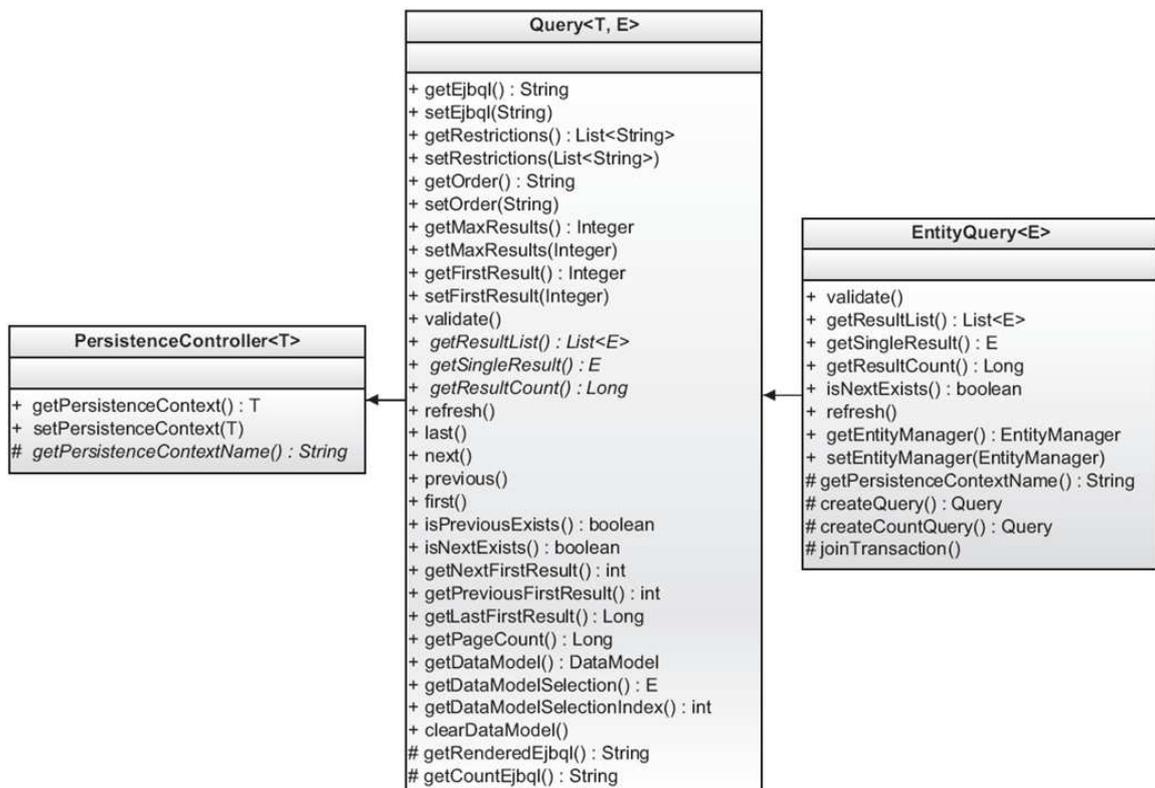


Figura 3.65: (Diagrama de clases de EntityQuery)

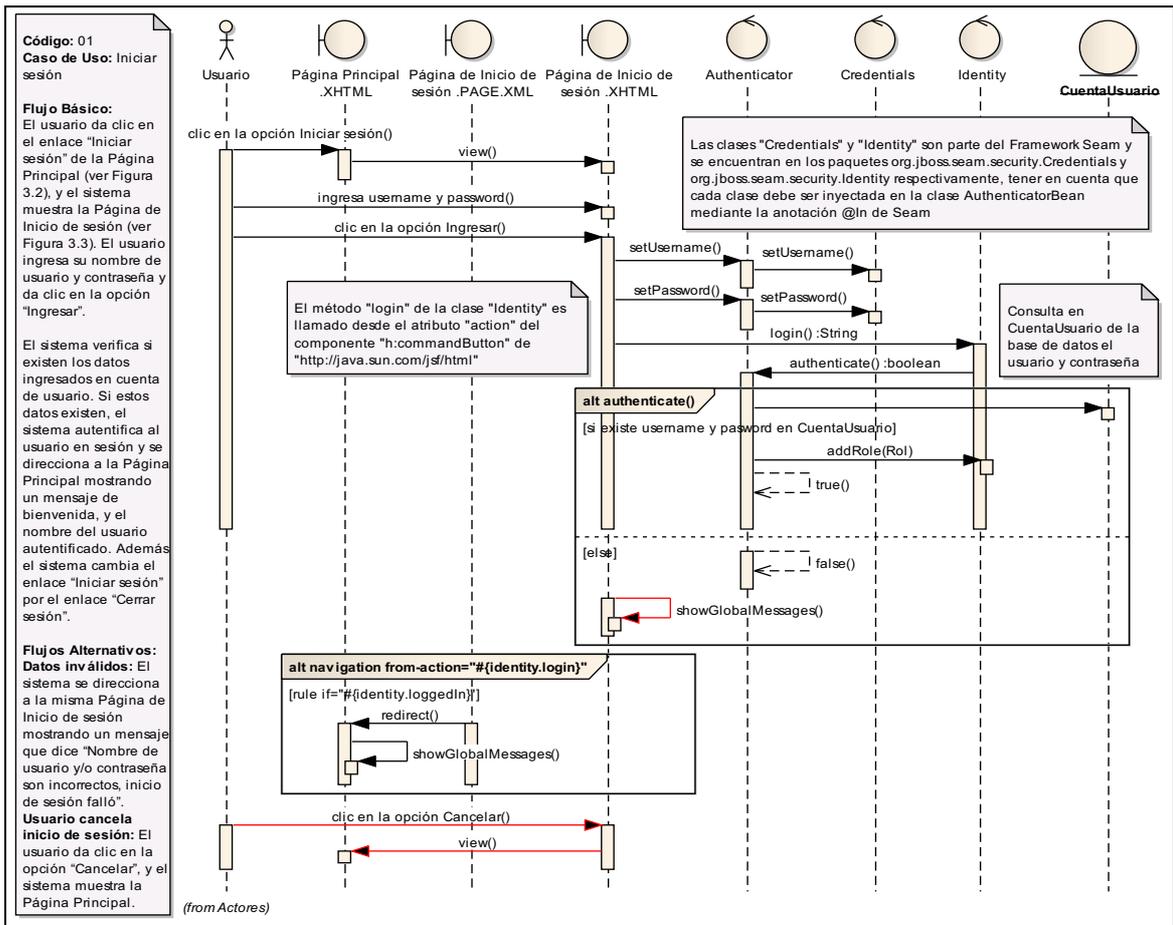


Figura 3.66: (Diagrama de Secuencia del caso de uso "Iniciar sesión")

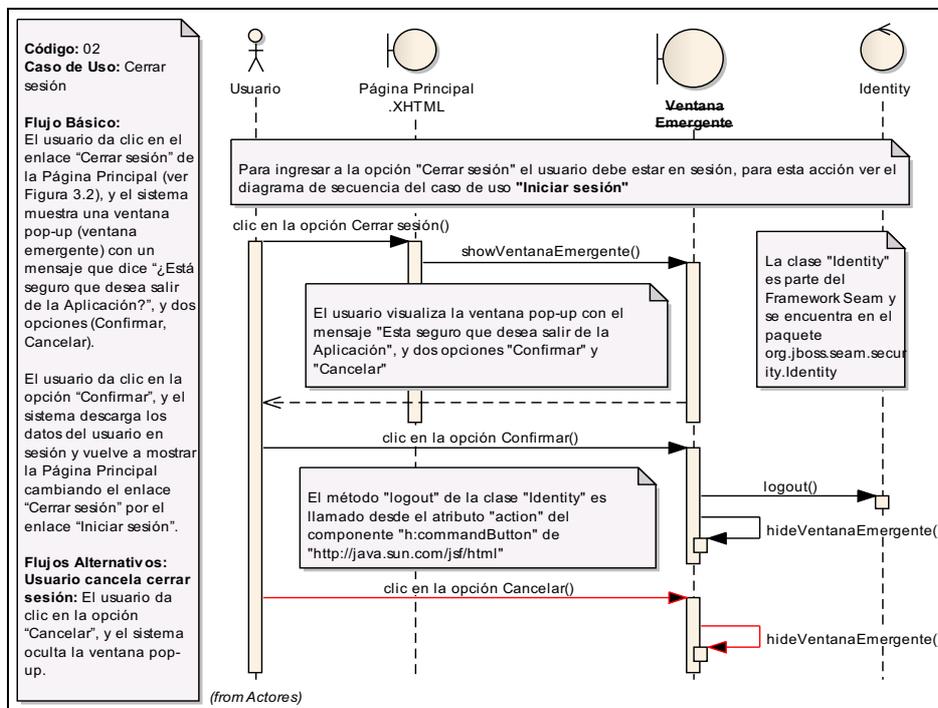


Figura 3.67: (Diagrama de Secuencia del caso de uso "Cerrar sesión")

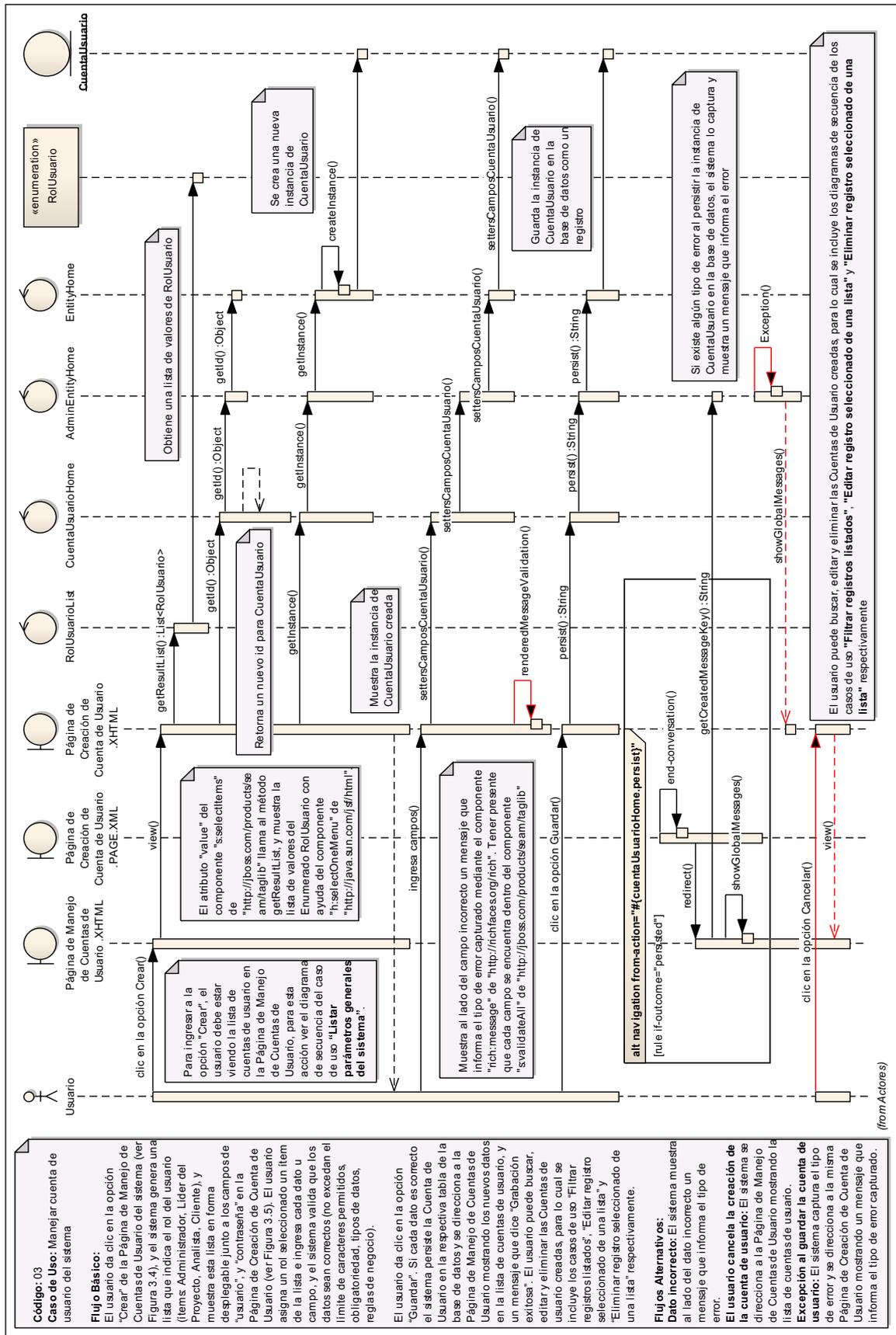


Figura 3.68: (Diagrama de Secuencia del caso de uso “Manejar cuenta de usuario del sistema”)

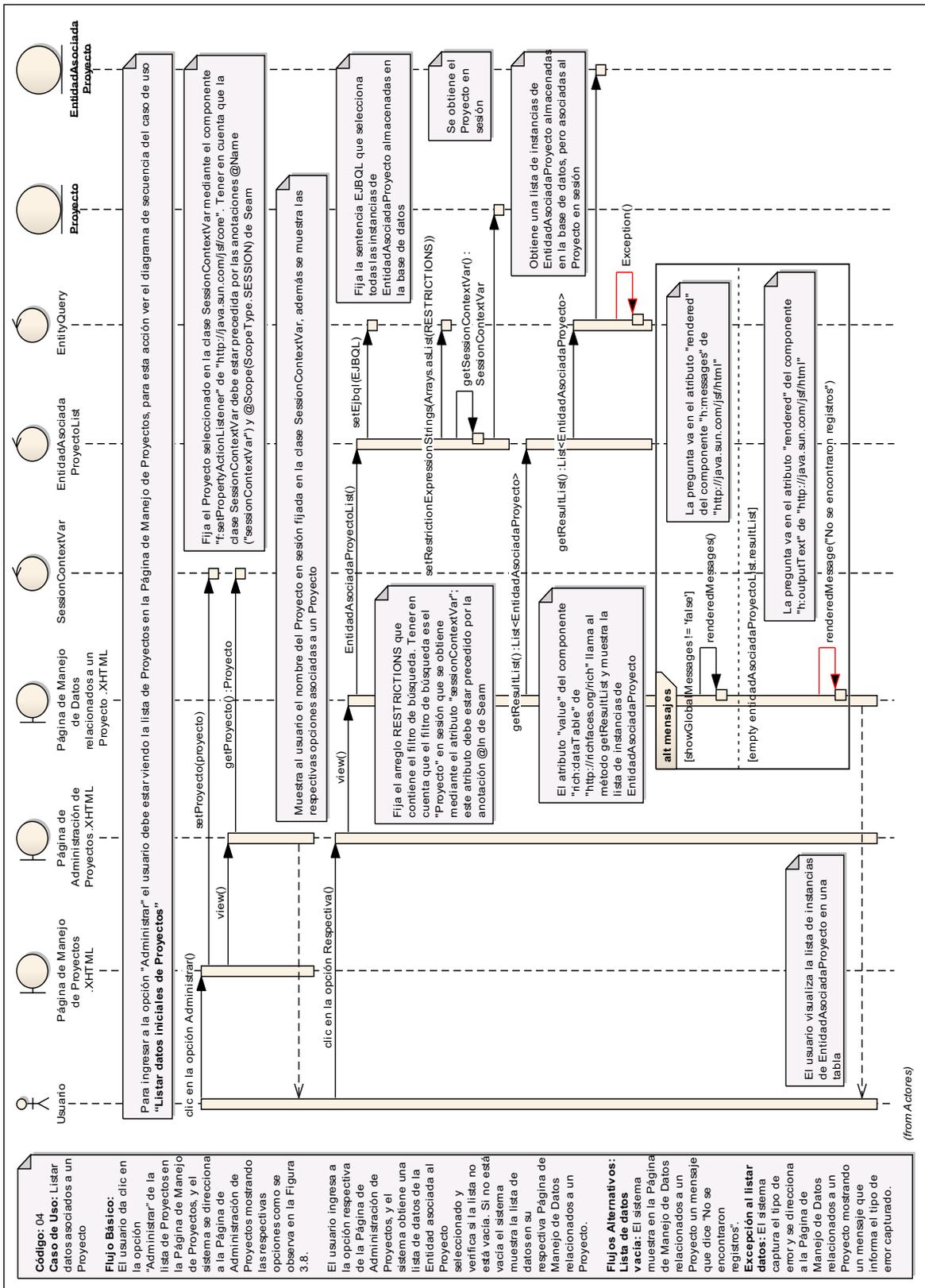


Figura 3.69: (Diagrama de Secuencia del caso de uso "Listar datos asociados a un Proyecto")

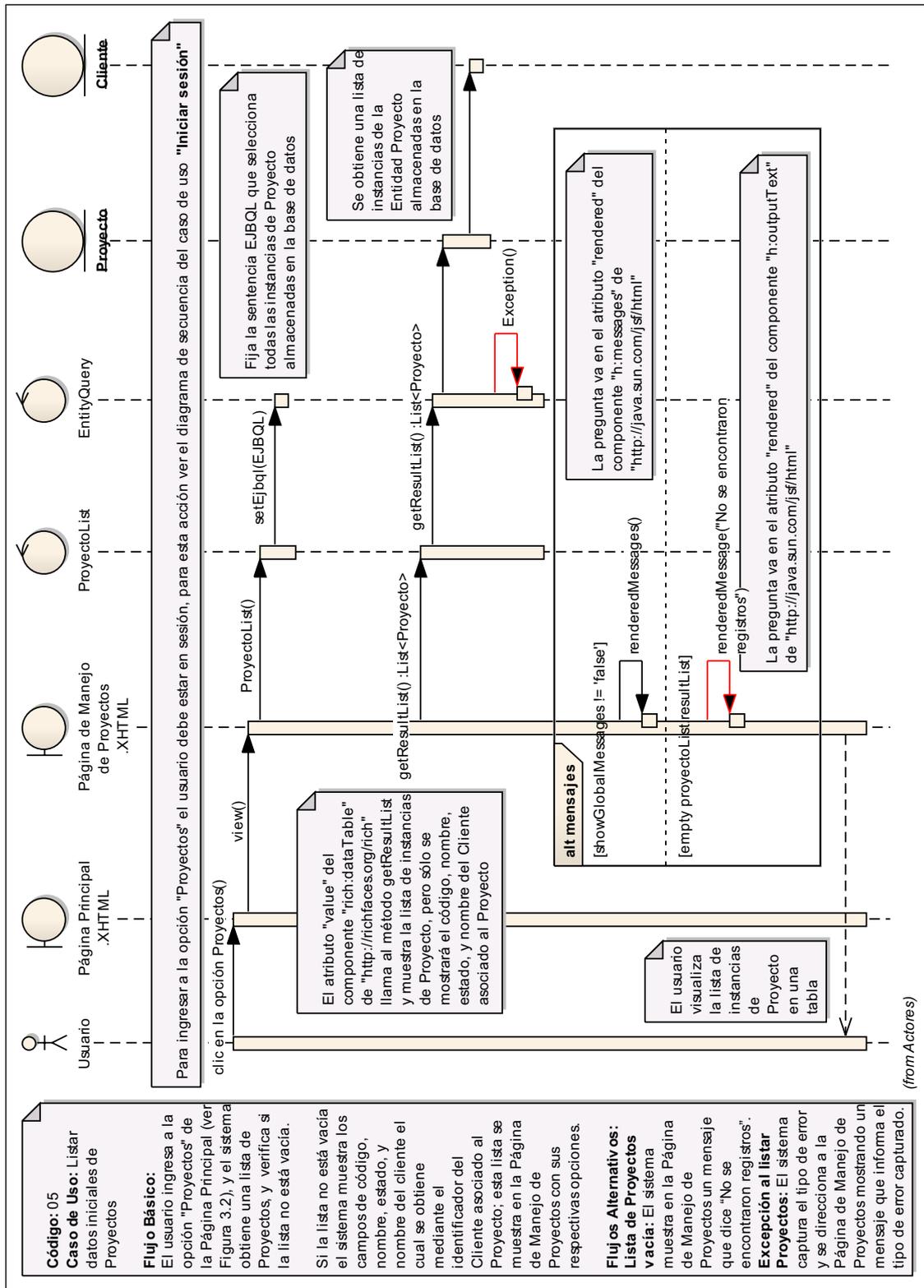


Figura 3.70: (Diagrama de Secuencia del caso de uso "Listar datos iniciales de Proyectos")

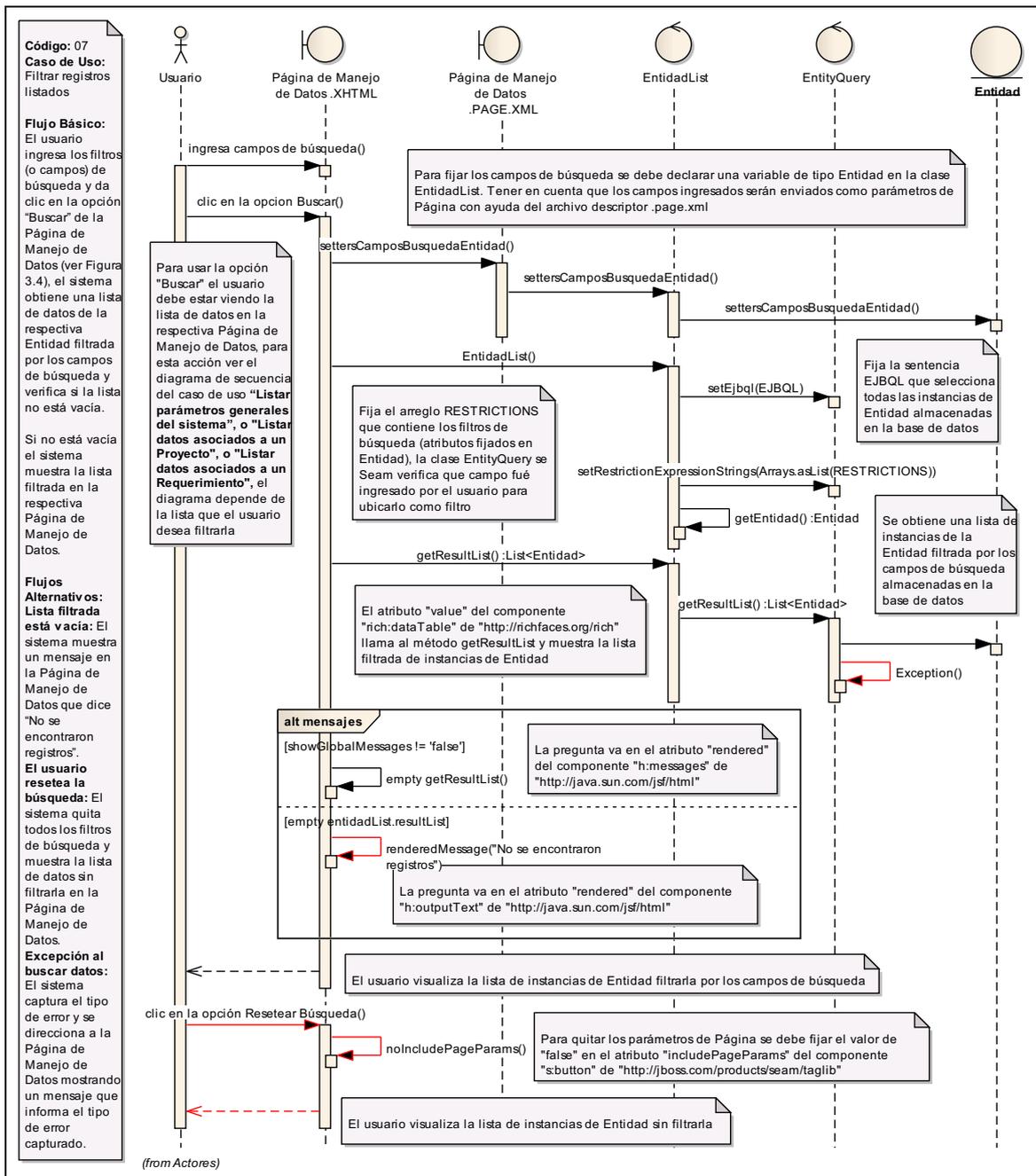


Figura 3.72: (Diagrama de Secuencia del caso de uso "Filtrar registros listados")

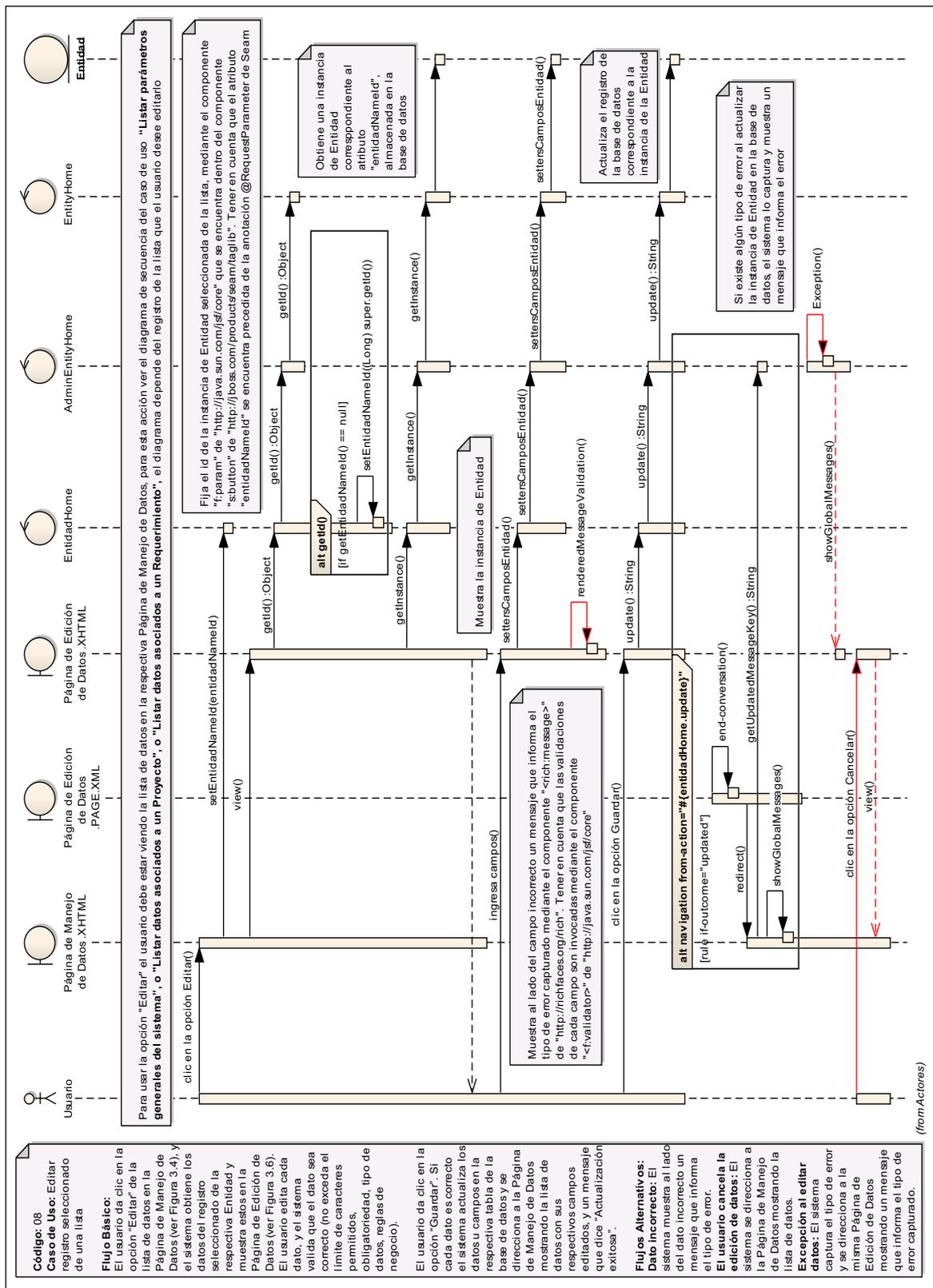


Figura 3.73: (Diagrama de Secuencia del caso de uso "Editar registro seleccionado de una lista")

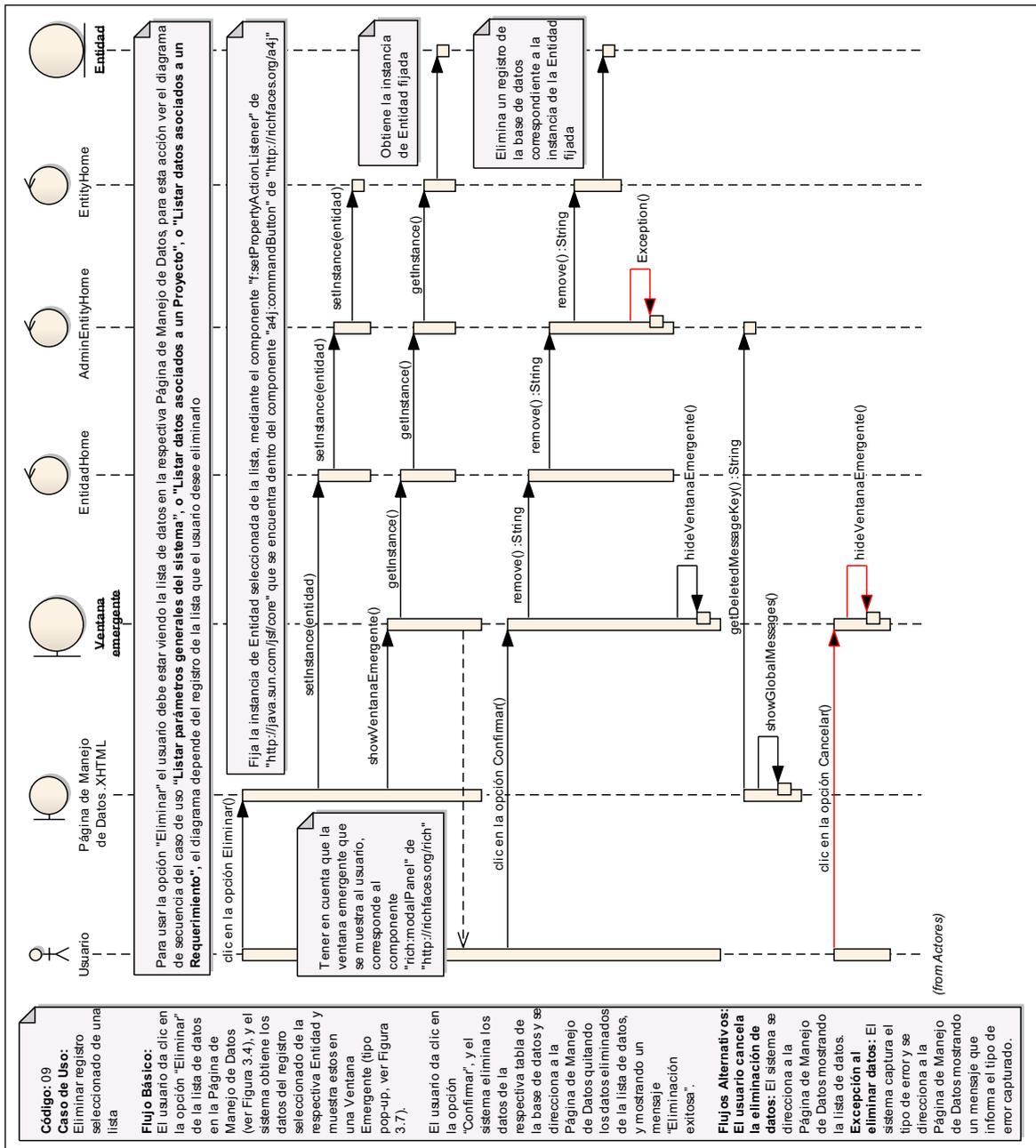


Figura 3.74: (Diagrama de Secuencia del caso de uso "Eliminar registro seleccionado de una lista")

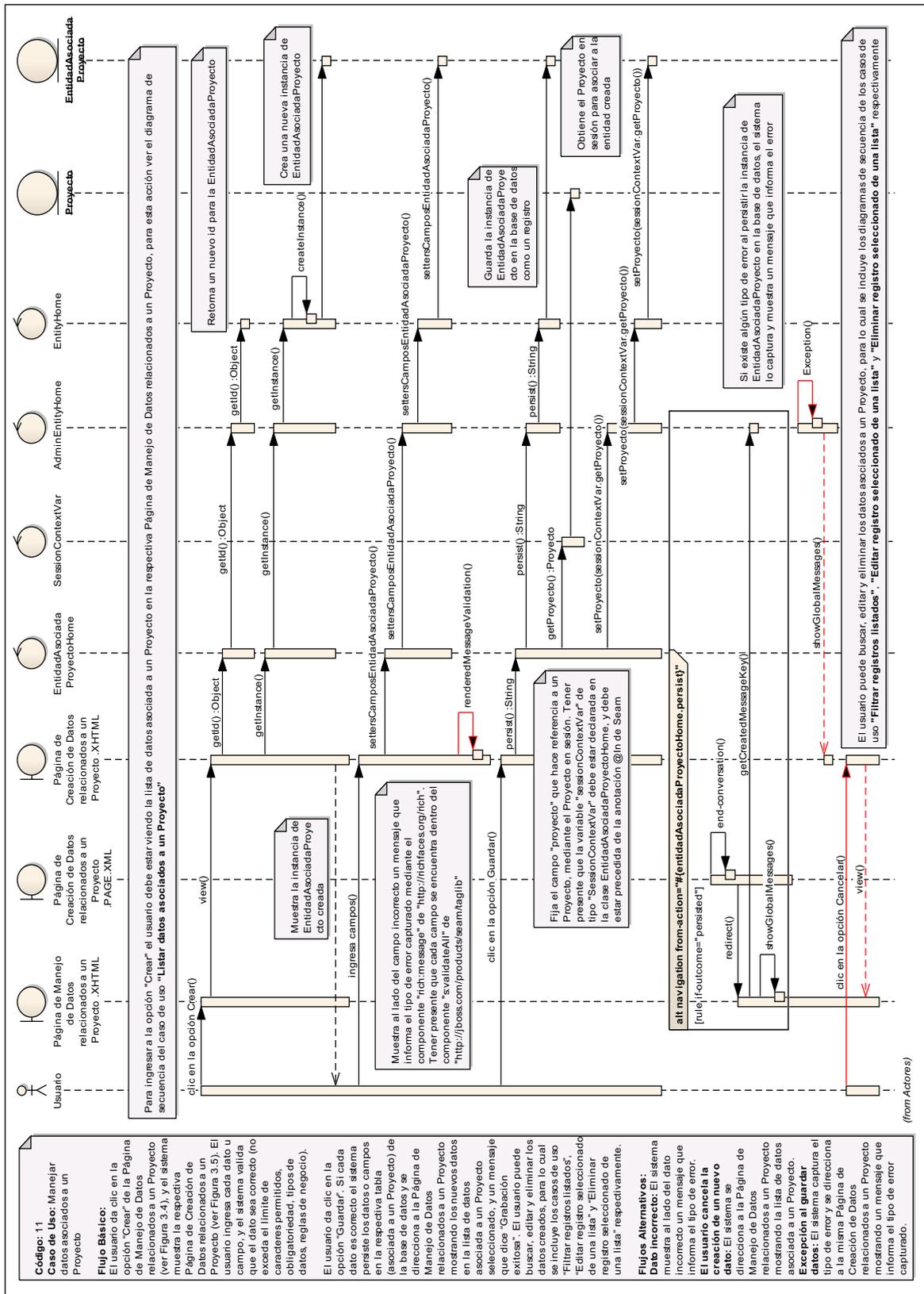


Figura 3.76: (Diagrama de Secuencia del caso de uso "Manejar datos asociados a un Proyecto")

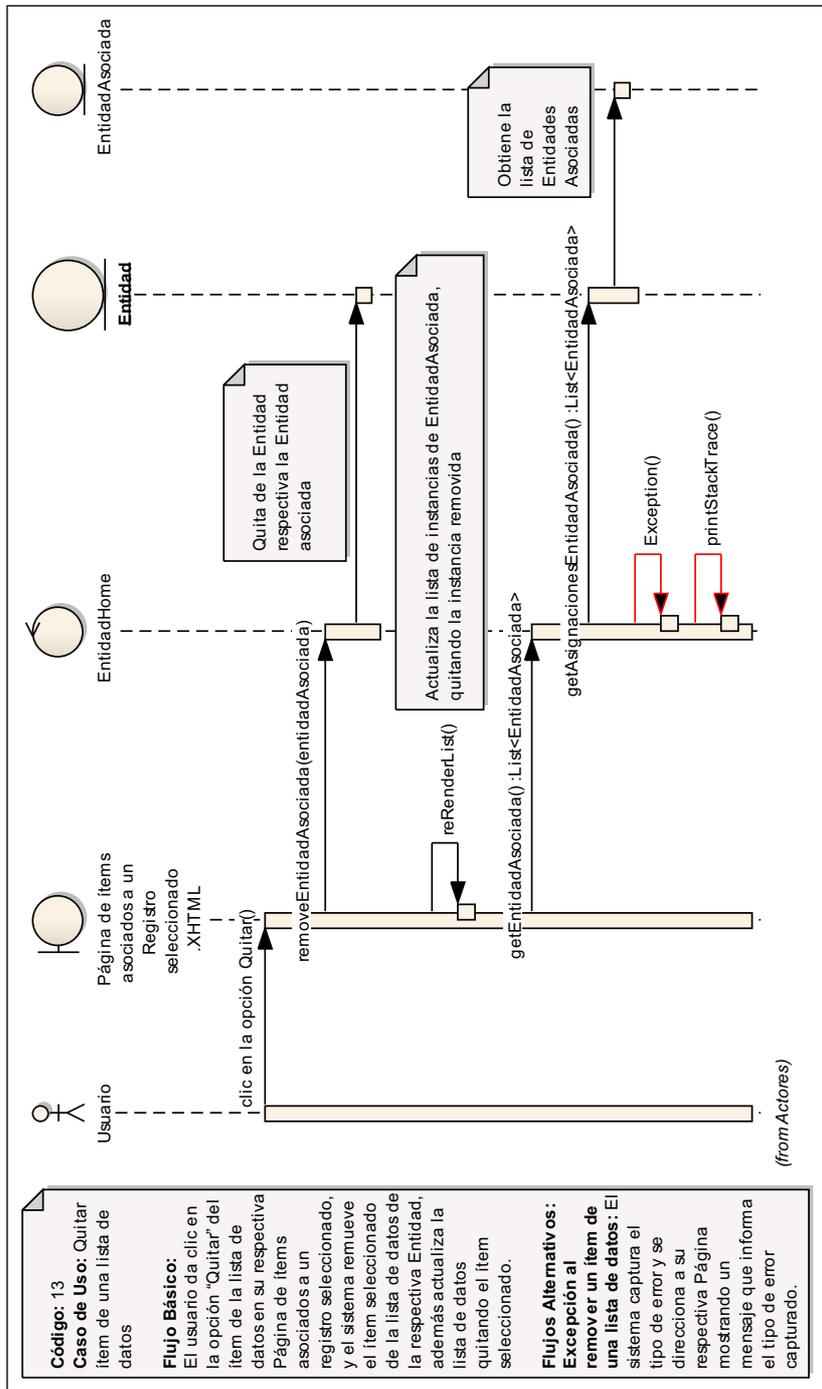


Figura 3.78: (Diagrama de Secuencia del caso de uso “Quitar ítem de una lista de datos”)

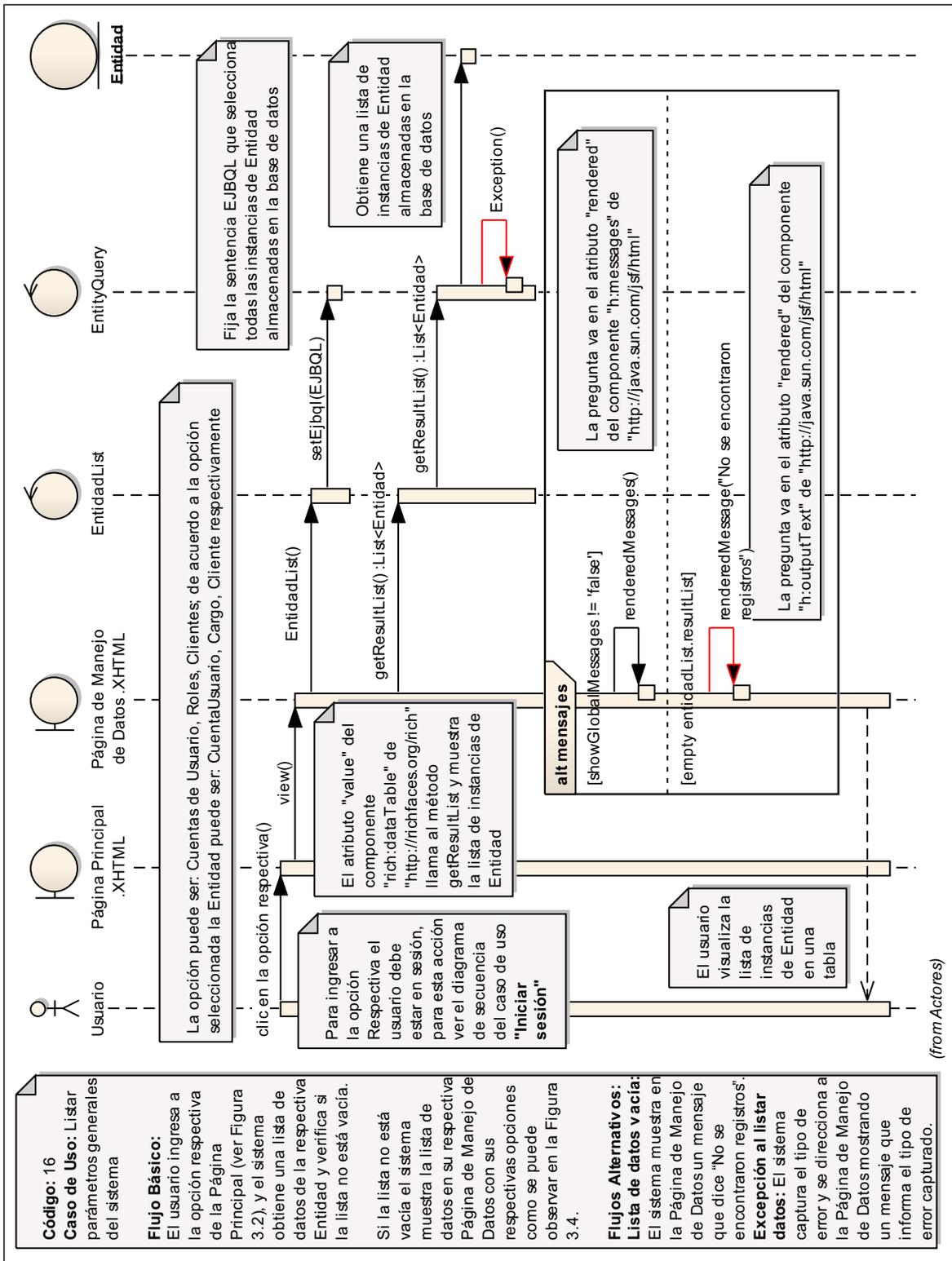


Figura 3.81: (Diagrama de Secuencia del caso de uso “Listar parámetros generales del sistema”)

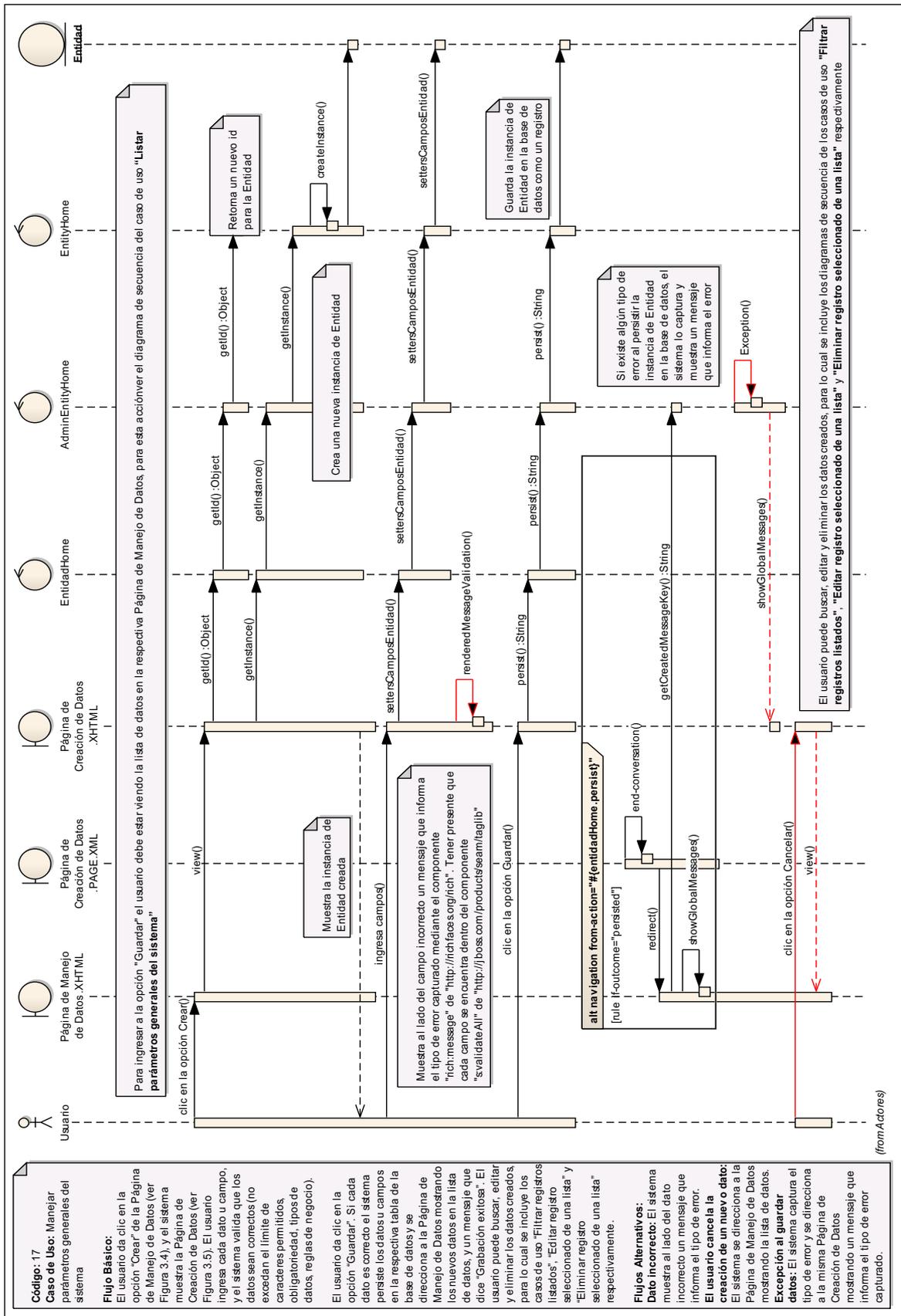


Figura 3.82: (Diagrama de Secuencia del caso de uso "Manejar parámetros generales del sistema")

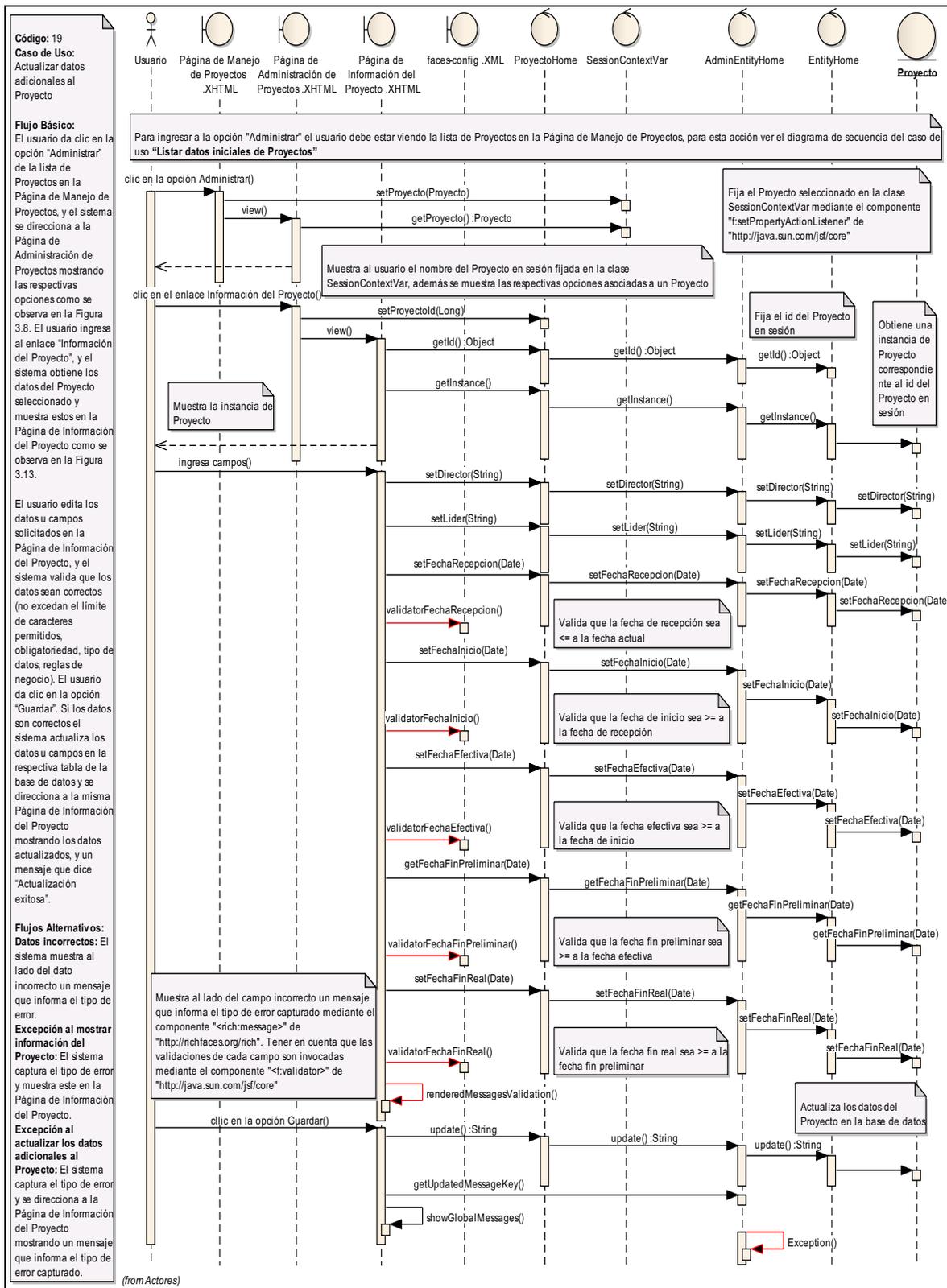


Figura 3.84: (Diagrama de Secuencia del caso de uso "Actualizar datos adicionales al Proyecto")

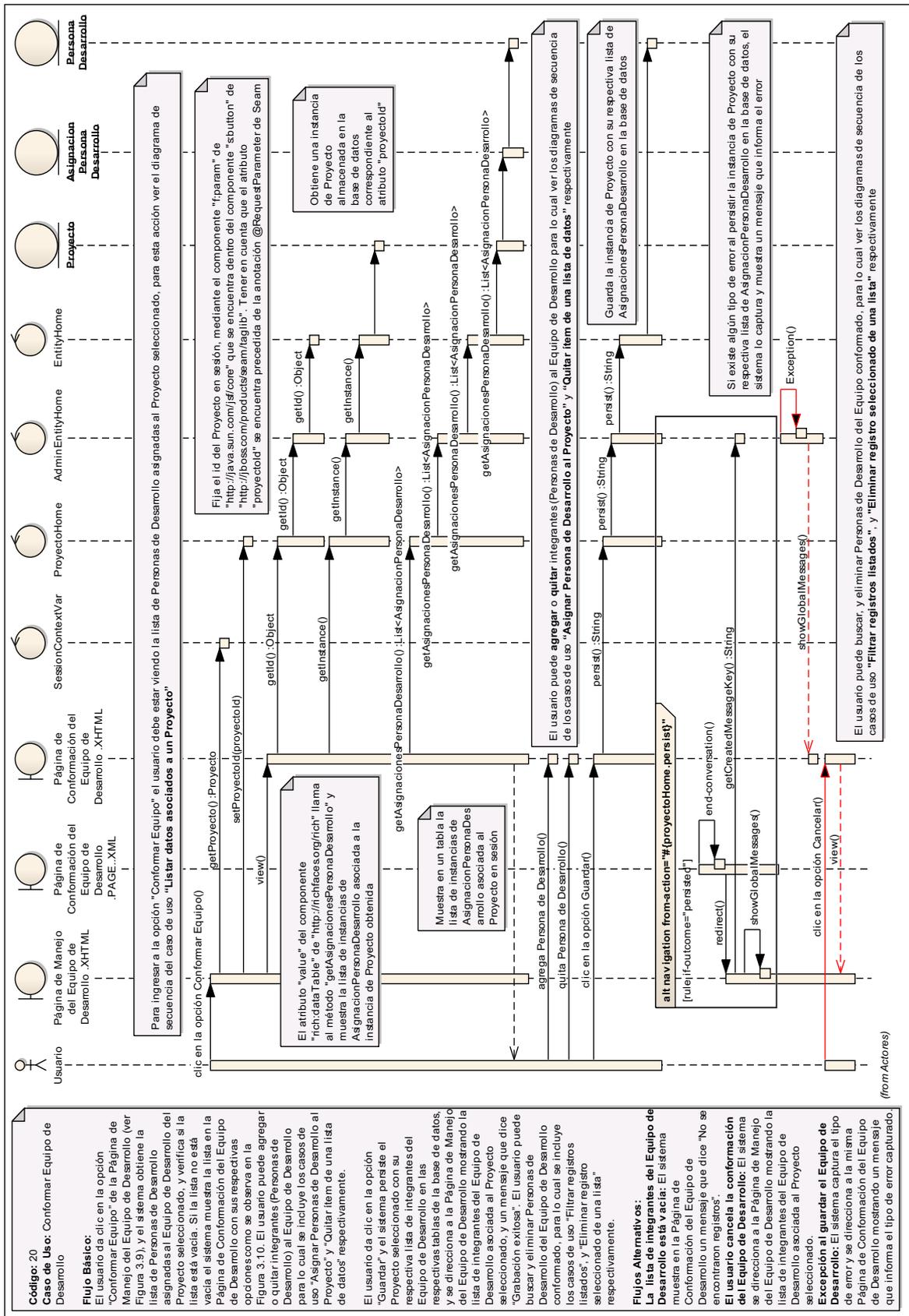


Figura 3.85: (Diagrama de Secuencia del caso de uso "Conformar Equipo de Desarrollo")

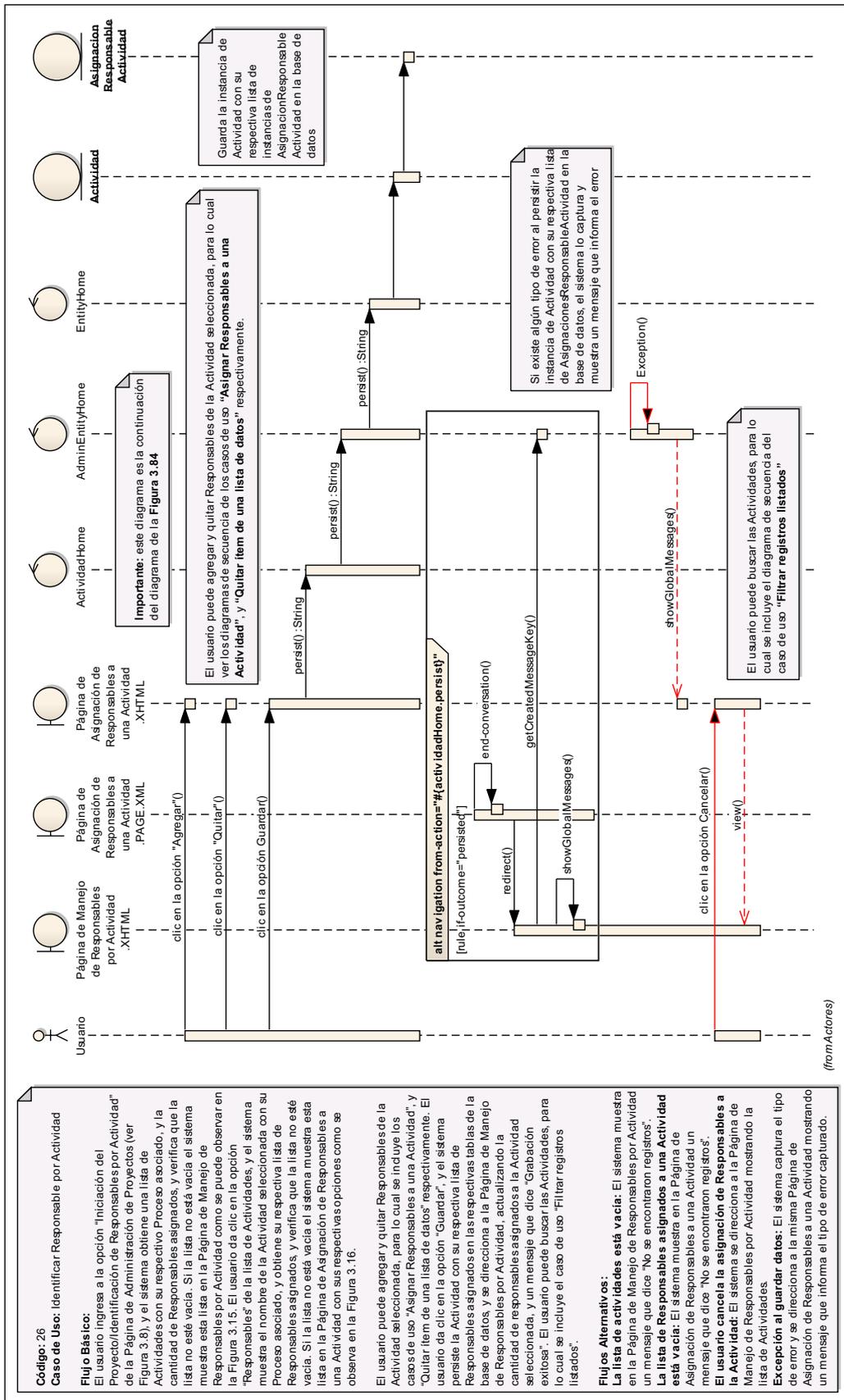


Figura 3.90: (Diagrama de Secuencia del Caso de Uso "Identificar Responsable por Actividad – Parte II")

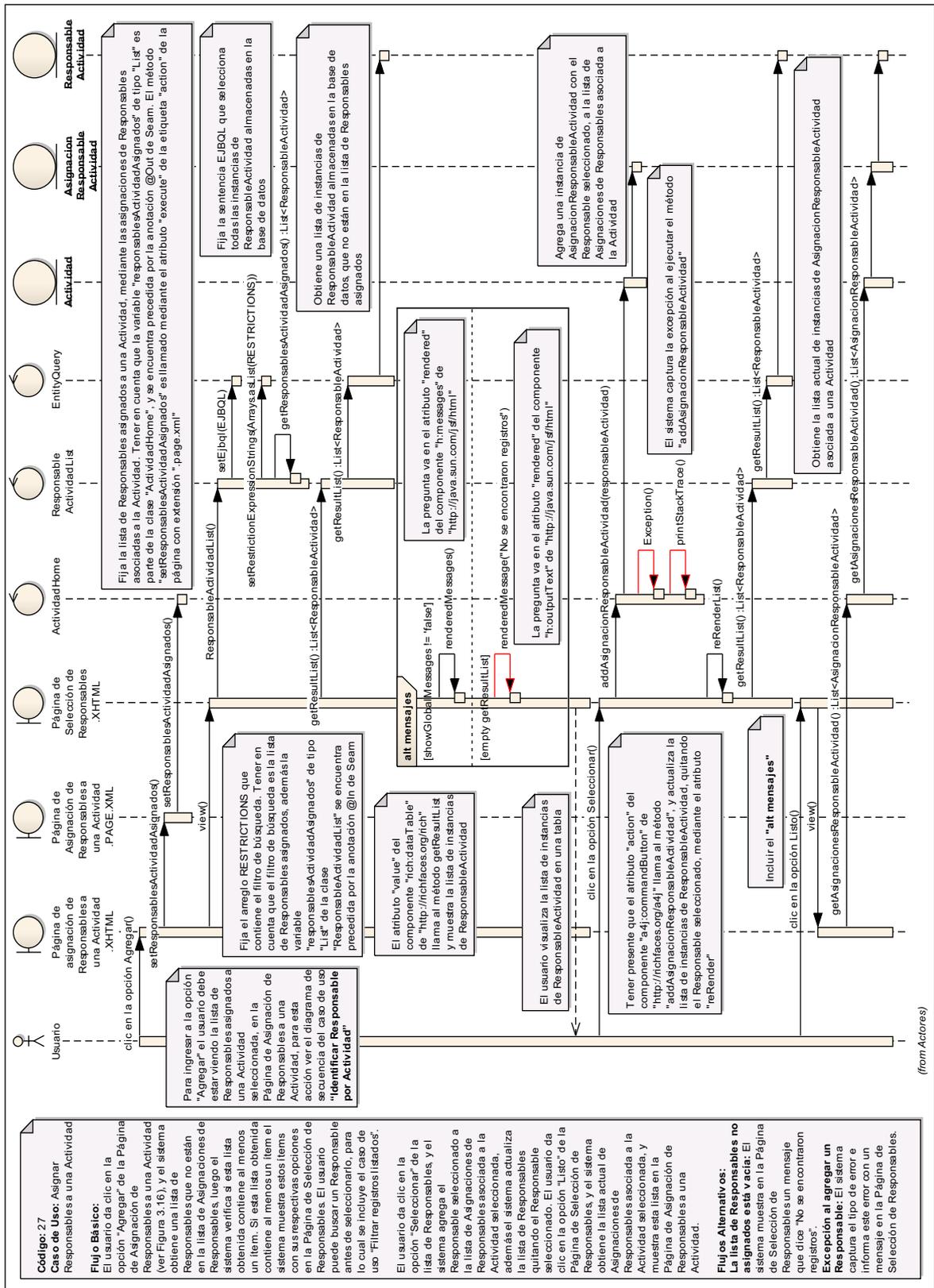


Figura 3.91: (Diagrama de Secuencia del caso de uso "Asignar Responsables a una Actividad")

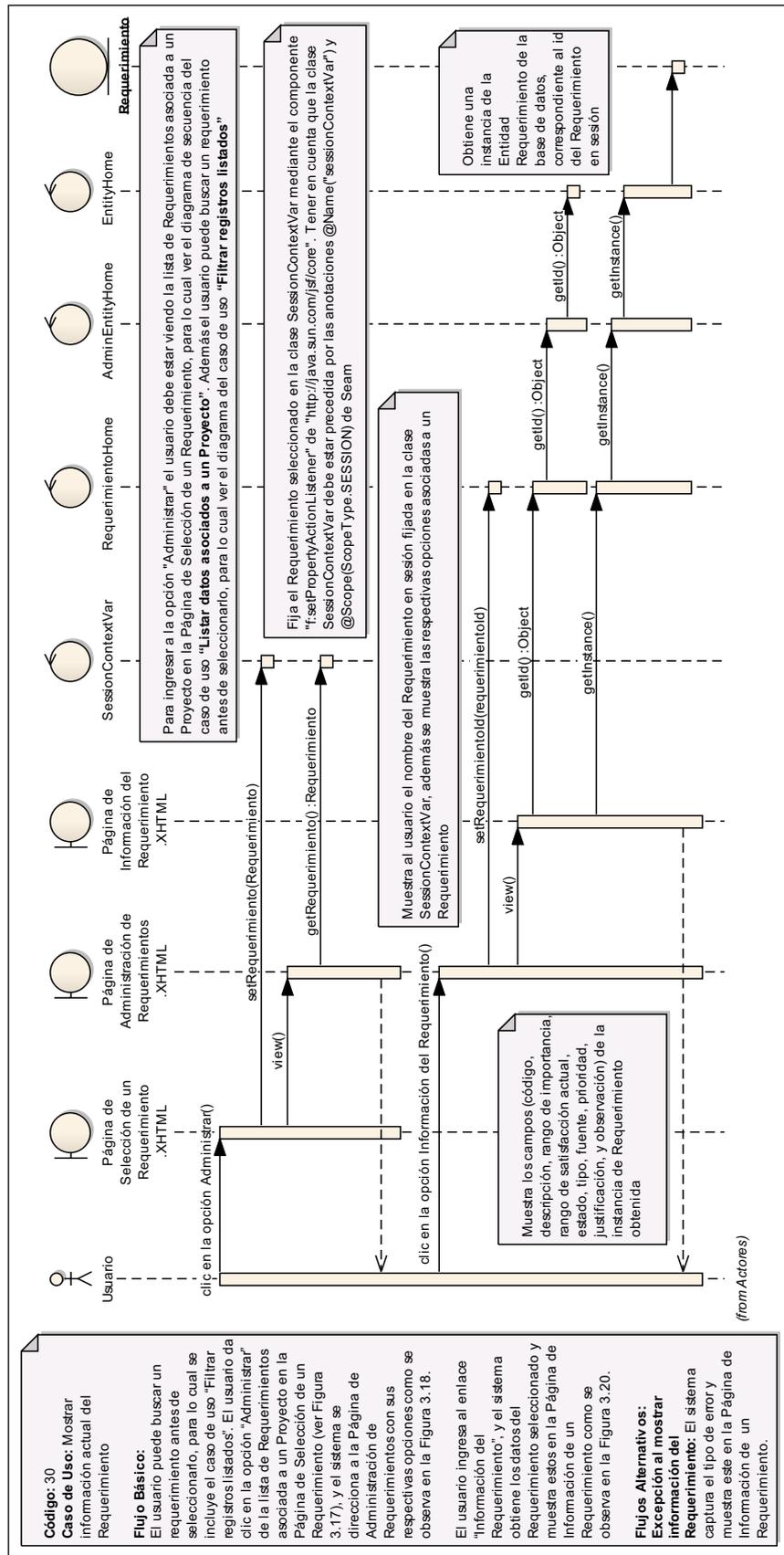


Figura 3.93: (Diagrama de Secuencia del caso de uso "Mostrar información actual del Requerimiento")

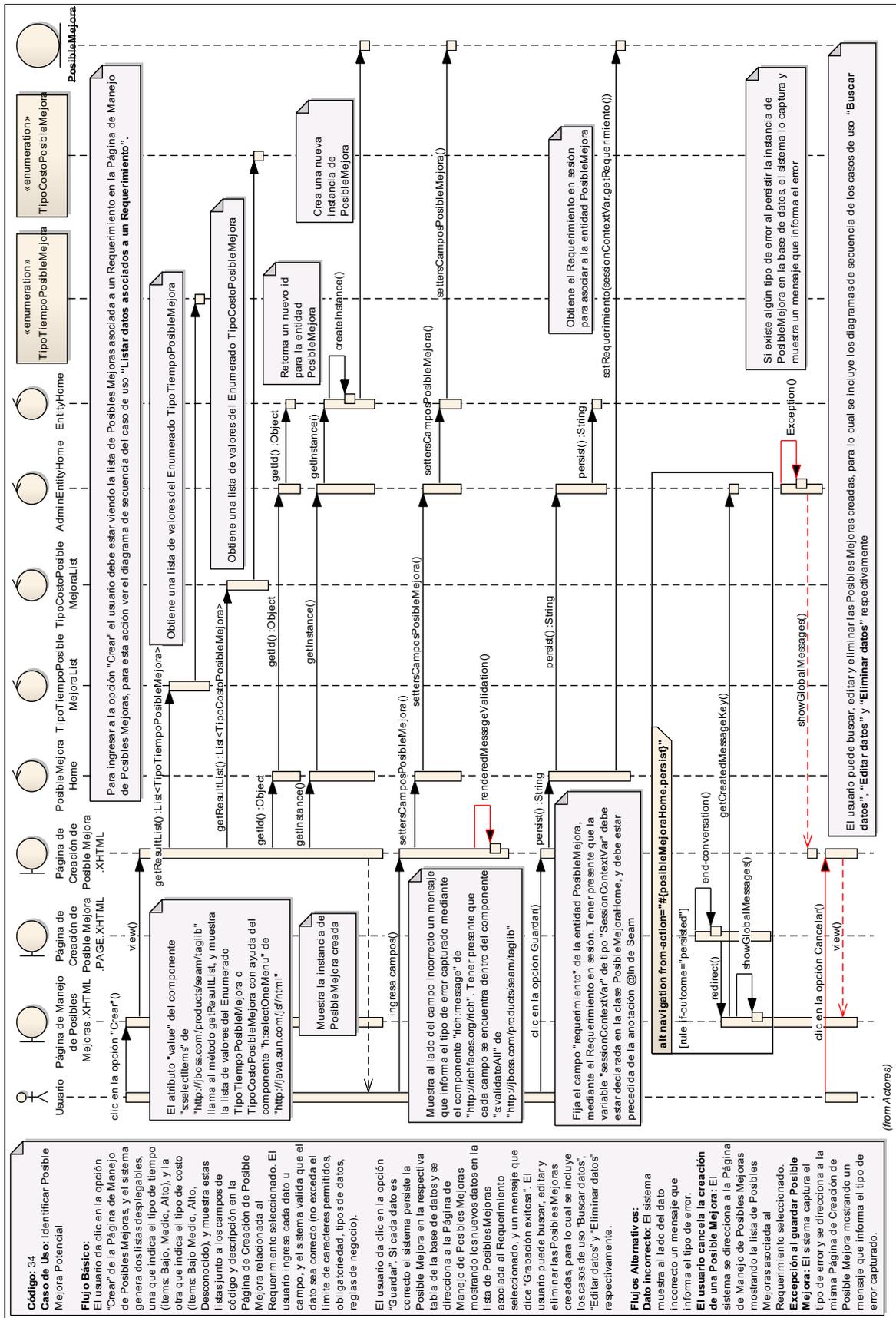


Figura 3.95: (Diagrama de Secuencia del Caso de Uso "Identificar Posible Mejora Potencial")
Página | 184

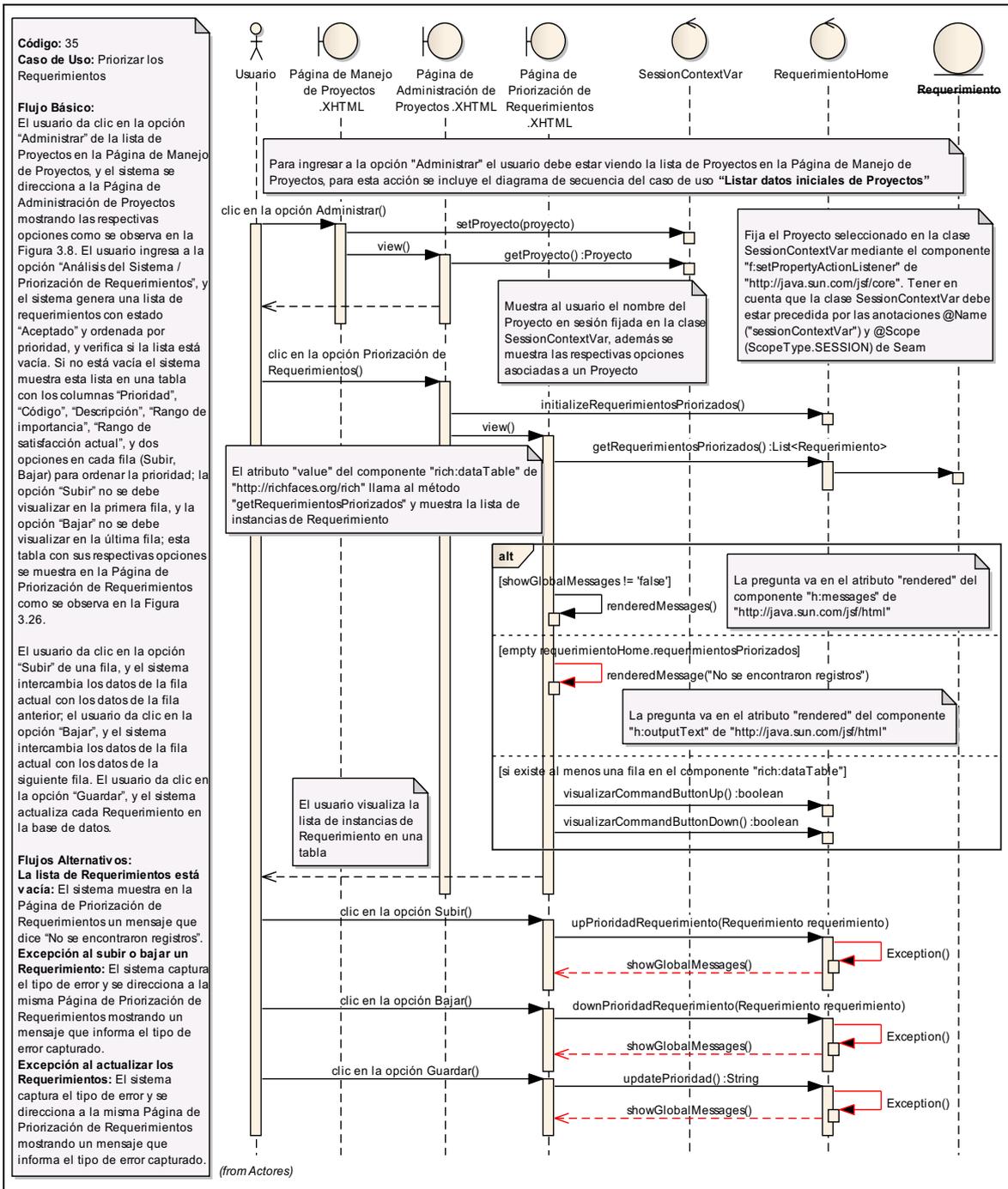


Figura 3.96: (Diagrama de Secuencia del caso de uso "Priorizar los Requerimientos")

3.3.2 Diagrama de clases

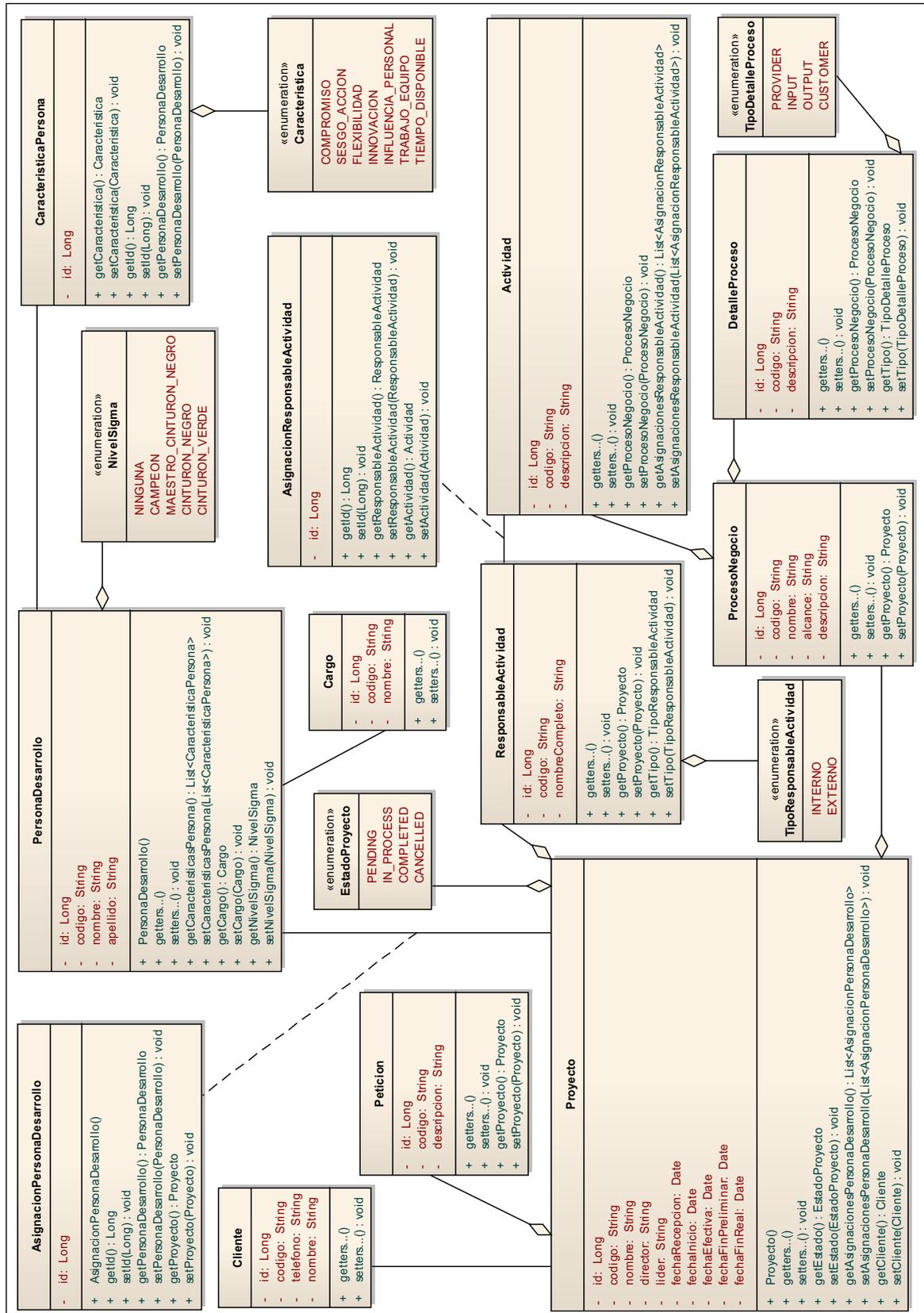


Figura 3.97: (Diagrama de Clases - Entidades - Parte I)

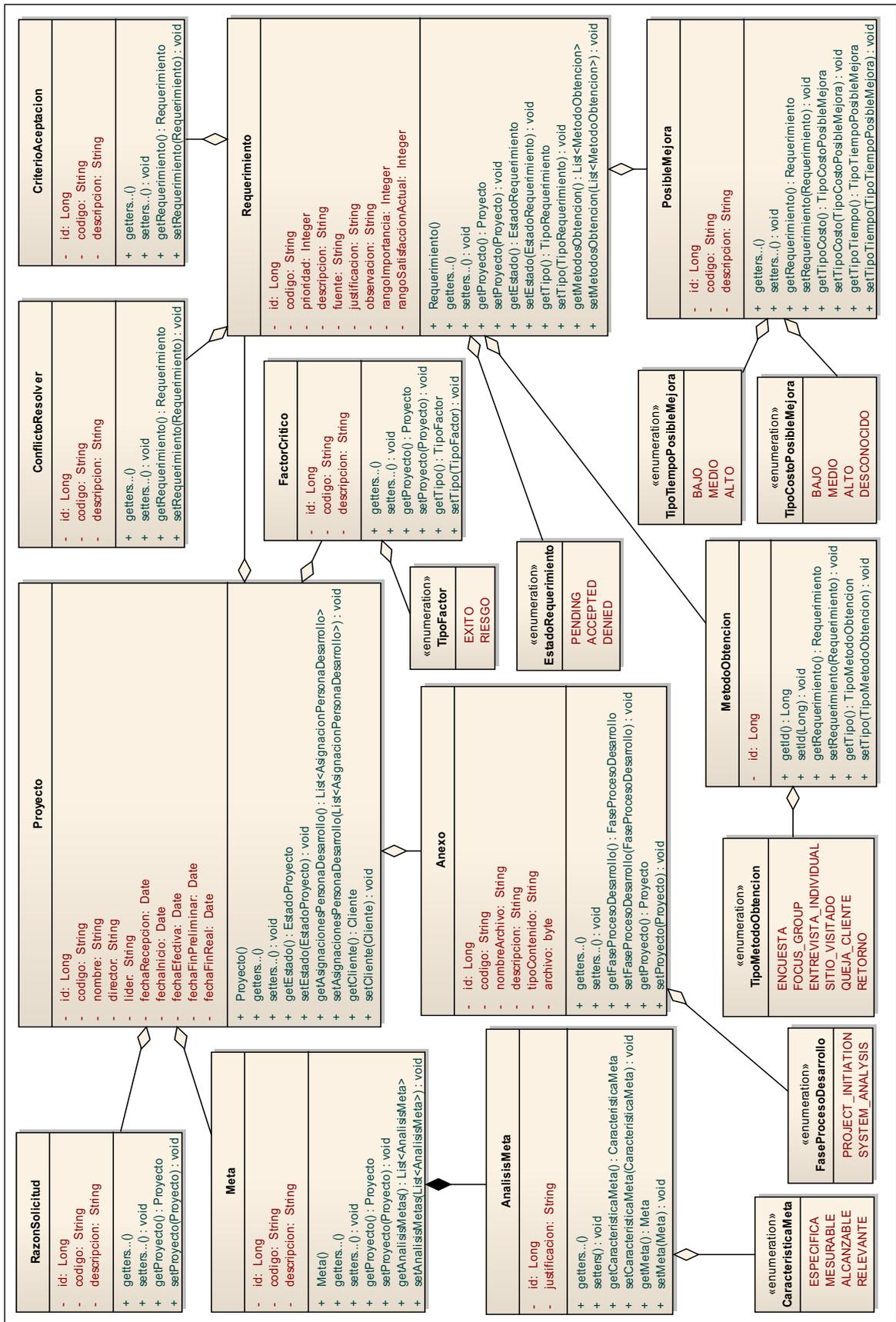


Figura 3.98: (Diagrama de Clases - Entidades - Parte II)

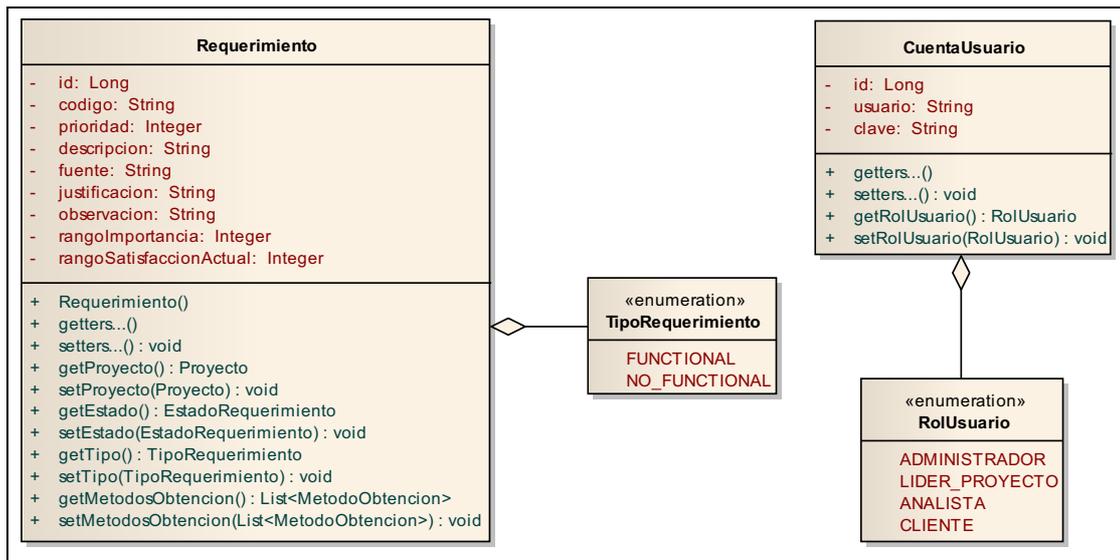


Figura 3.99: (Diagrama de Clases - Entidades - Parte III)

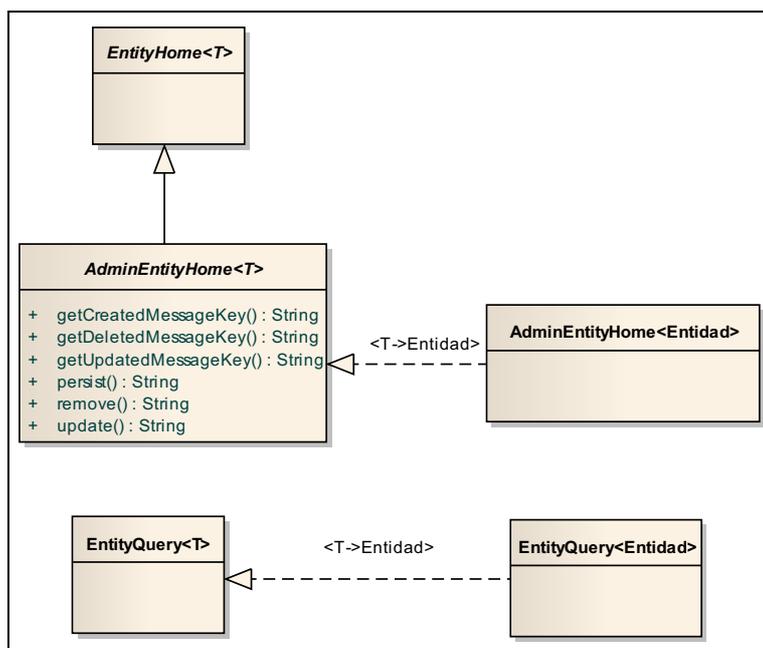


Figura 3.100: (Diagrama de Clases – Framework)

El diagrama de clases de EntityHome y EntityQuery (parte del framework de clases de JBoss Seam) se detalla en la sección 3.3 Diseño Detallado.

Los controladores Home (ver Figura 3.103) apoyan en las operaciones CRUD de su respectiva entidad, ya que se ha creado un controlador por Entidad. Tener

en cuenta que el nombre del controlador tiene la siguiente estructura “nombre_entidadHome”, es decir, si un controlador tiene el nombre “AnexoHome”, este apoyará las operaciones CRUD de la entidad “Anexo”. De igual forma, el nombre de los controladores List (ver Figura 3.104) tiene la misma estructura pero la palabra “Home” es reemplazada por “List”; estos controladores List administran los resultados de una sentencia JPQL aplicada a su respectiva entidad.

En la Figura 3.101 y 3.102 se muestra como definir un controlador Home y List respectivamente. Esta estructura fue utilizada por todos los controladores.

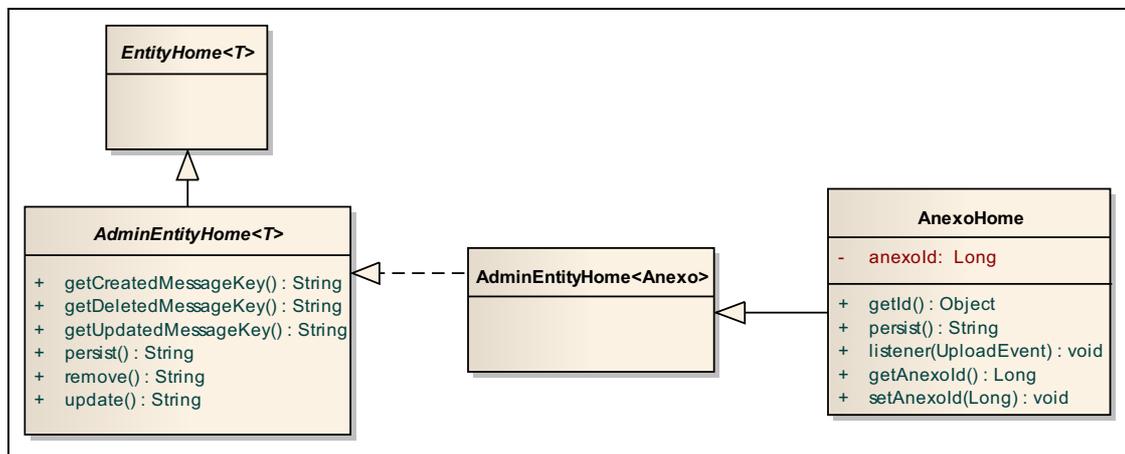


Figura 3.101: (Definición de un Controlador Home)

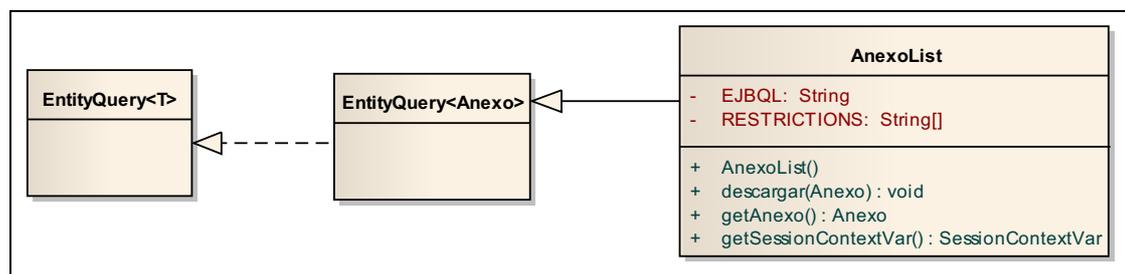


Figura 3.102: (Definición de un Controlador List)

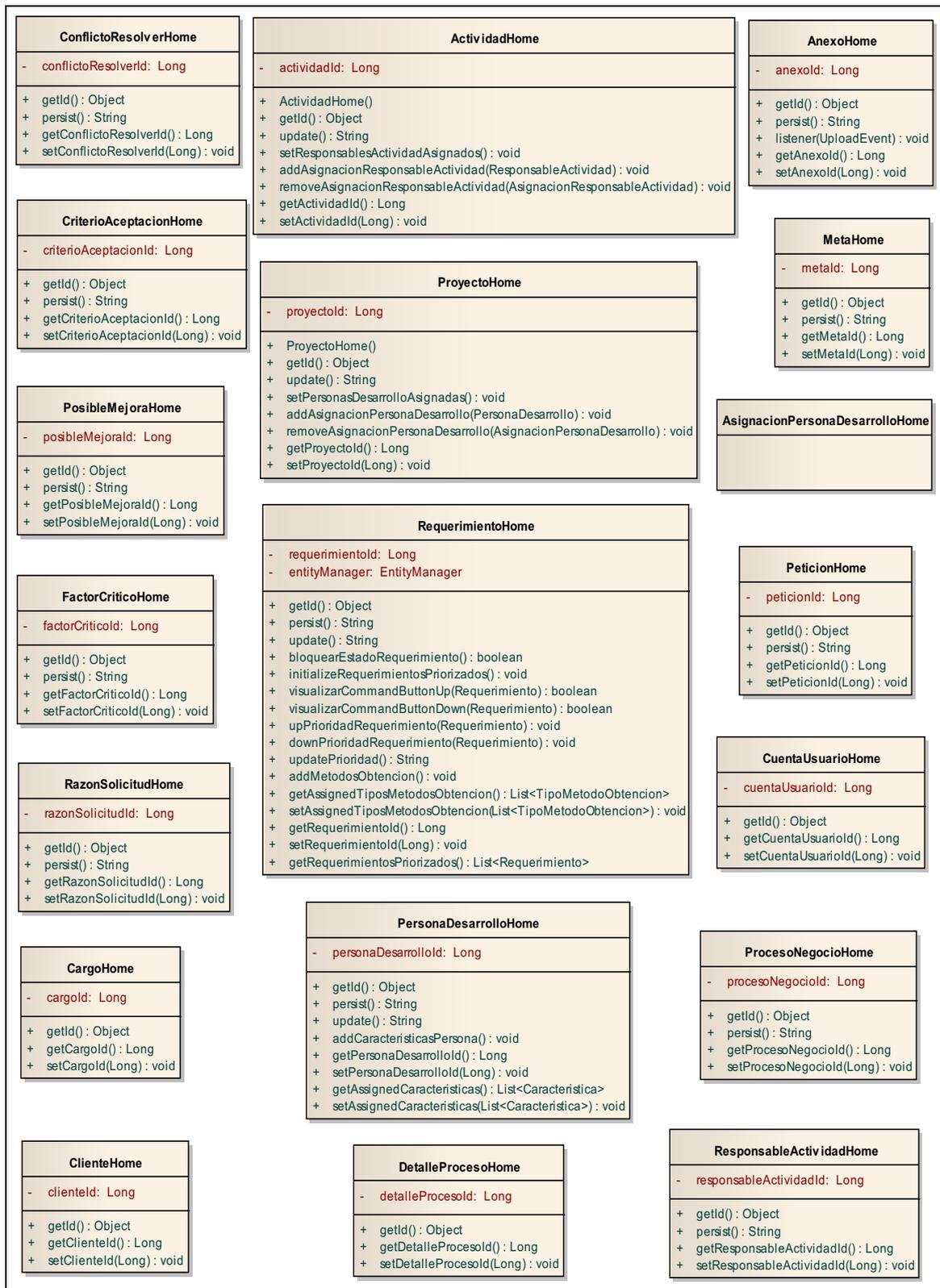


Figura 3.103: (Diagrama de Clases – Controladores – Home)



Figura 3.104: (Diagrama de Clases – Controladores – List)

Los controladores de la Figura 3.105 están asociados a un componente de interfaz de usuario, y muestran los valores de su respectiva clase de tipo Enumerado.

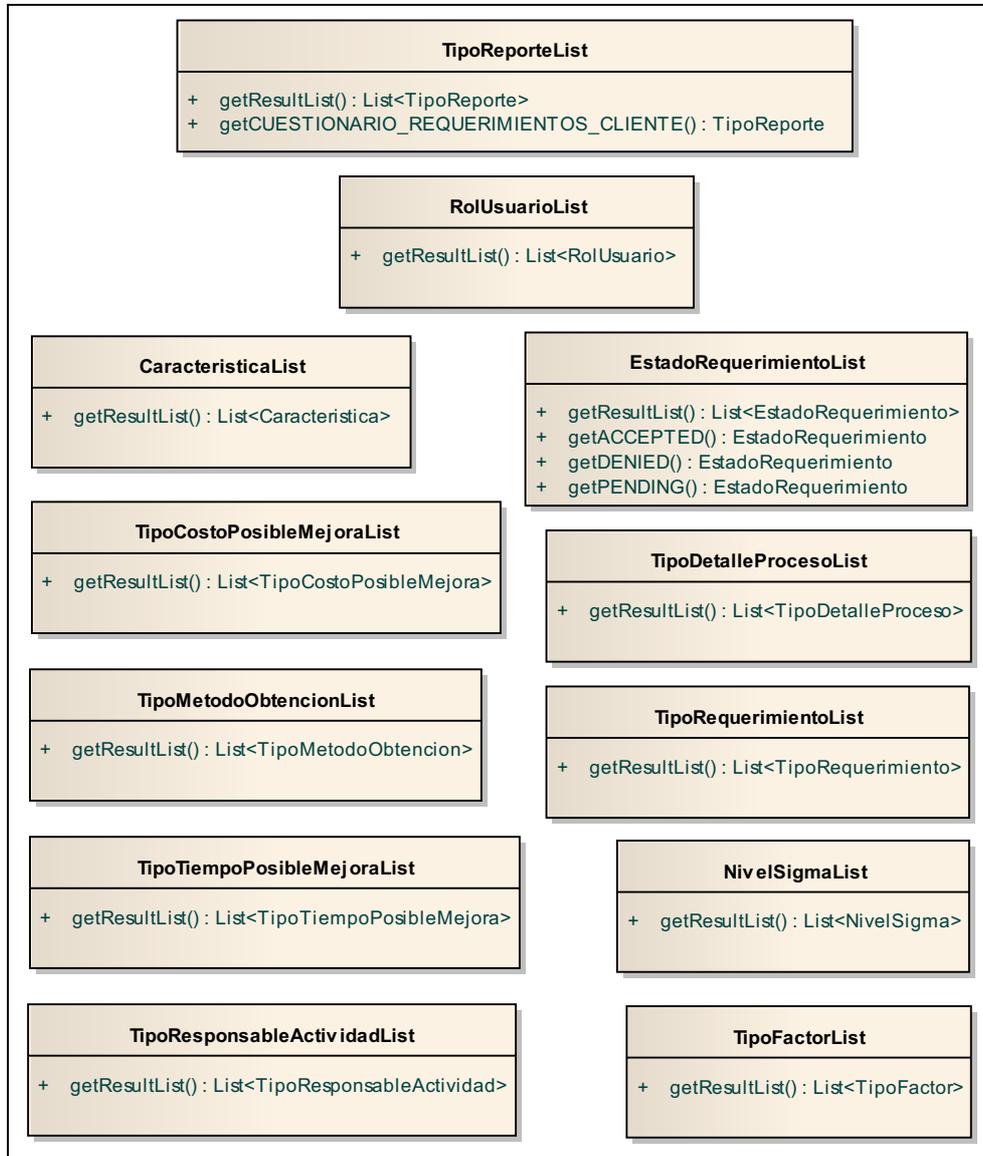


Figura 3.105: (Diagrama de Clases – Controladores – Enumerations List)

Los controladores de la Figura 3.106, fueron creados con fines de validar el ingreso de datos, almacenar variables de sesión, generar y mostrar reportes en formato PDF, y autenticar usuario del sistema en sesión.

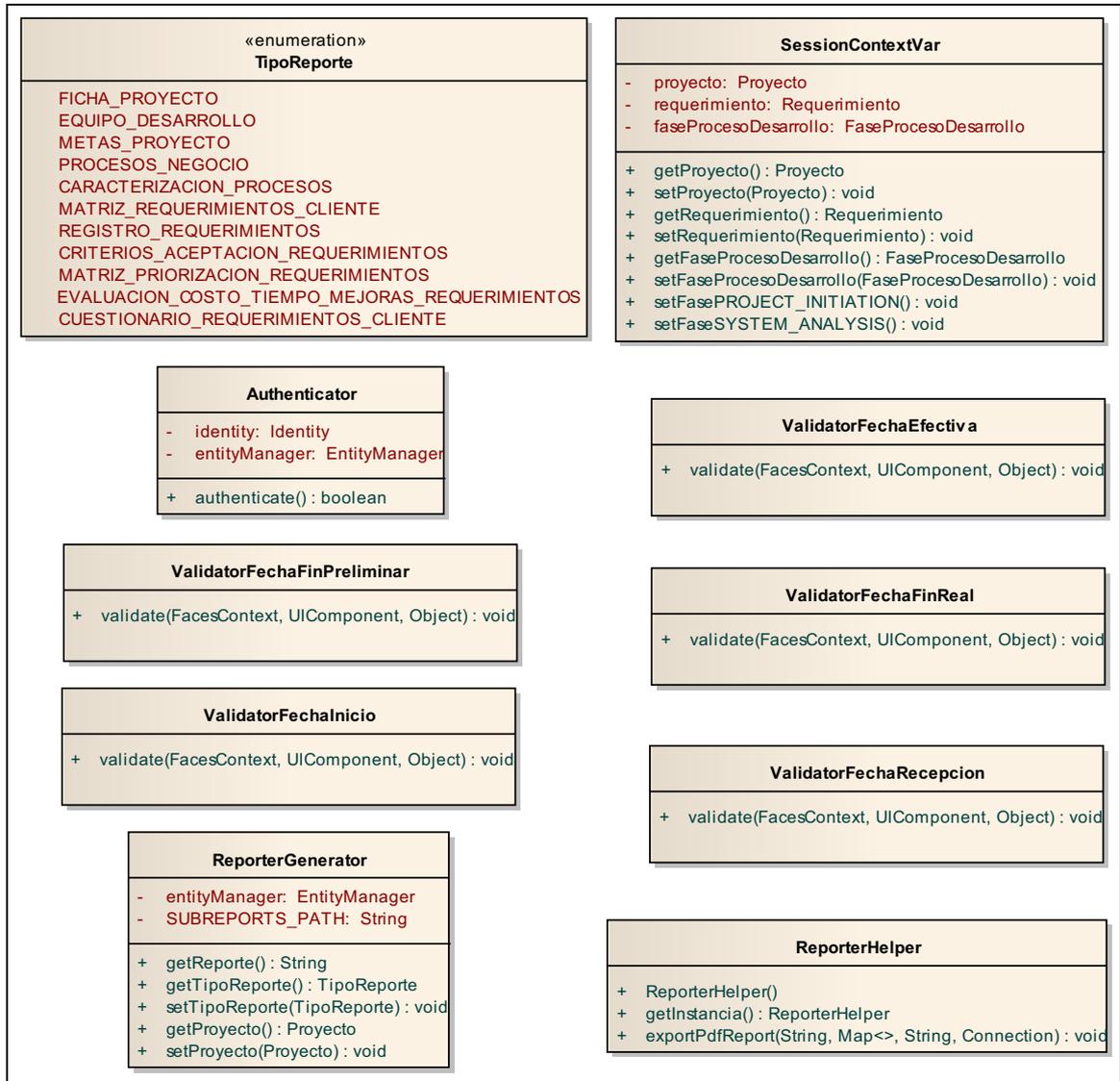


Figura 3.106: (Diagrama de Clases – Controladores – Otros)

3.3.3 Modelo de datos

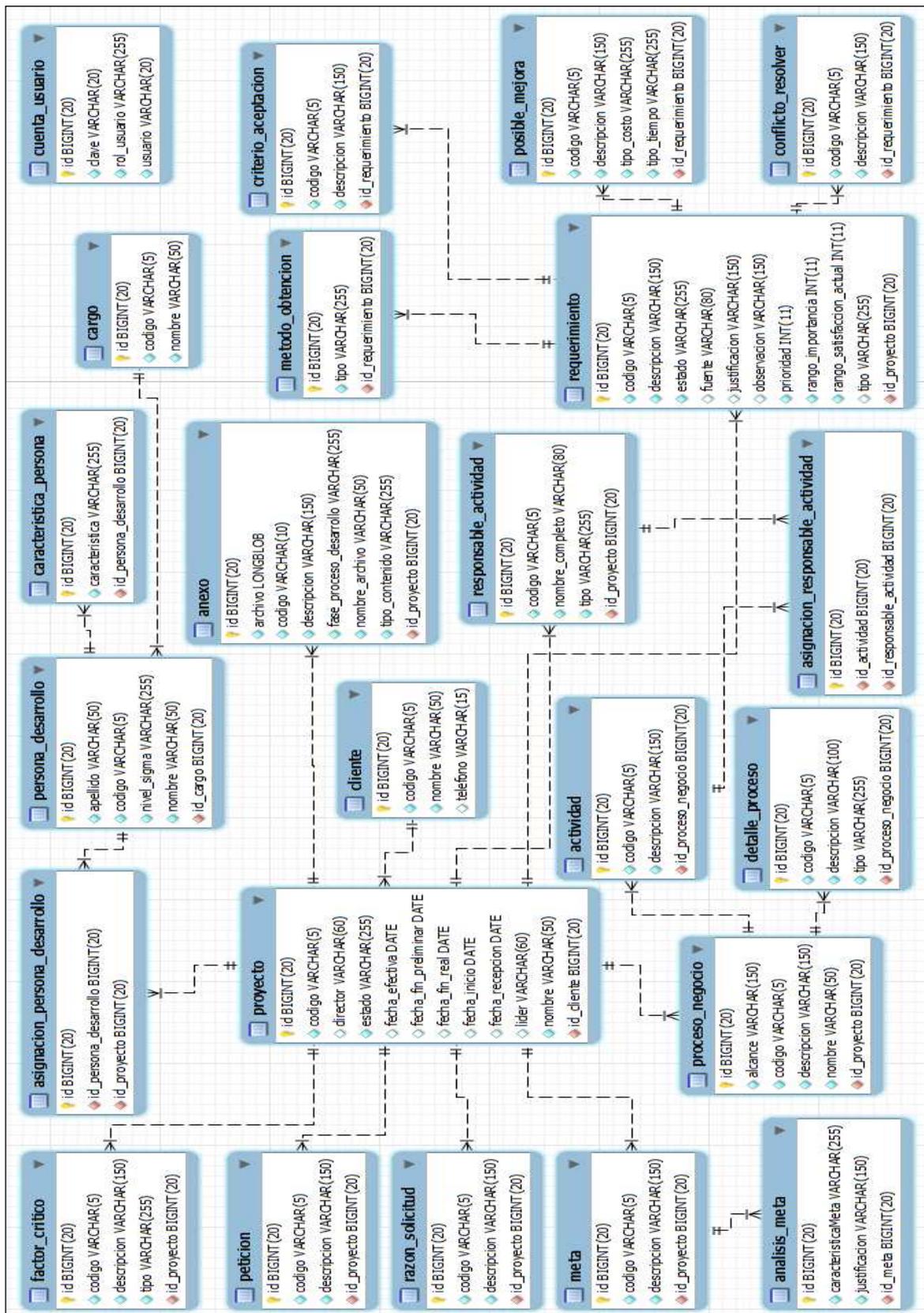


Figura 3.107: (Modelo de datos)

3.4 Implementación

En esta fase se realizó la codificación del software, en base al análisis y diseño realizado en las fases anteriores de la metodología ICONIX. Antes de empezar la codificación se instaló y configuró el ambiente de desarrollo, con las herramientas mencionadas en la sección “2.5 Herramientas de Desarrollo” del presente proyecto.

Luego se configuró y generó la estructura inicial del proyecto web con ayuda de la herramienta “Seam Generator” que forma parte del framework Seam. Posteriormente, la edición de este proyecto se realizó en Netbeans IDE.

Finalmente, para realizar las pruebas funcionales del software con el usuario, se desplegó este proyecto en el Servidor de Aplicaciones JBoss.

3.5 Evaluación del Software

En esta sección se expondrá los resultados de una encuesta, ejecutada con el fin de evaluar el software desarrollado en el presente proyecto.

Con esta finalidad se subió temporalmente al Internet el aplicativo y la encuesta. Luego se reunió a un grupo de personas para explicarles en su totalidad el funcionamiento del software, y concluir llenado la encuesta.

También se realizó la encuesta a otro grupo de personas de forma virtual (online), enviándoles por correo electrónico el link de acceso al software para que la utilicen, y de esta forma llenen la encuesta.

Al finalizar la evaluación, los resultados fueron procesados a través de una hoja de cálculo del programa Google Docs. Las preguntas con sus respectivos porcentajes obtenidos se muestran en la siguiente tabla.

Resultados de la evaluación del software				
Sección	Preguntas	Opciones	% obtenido	
Manejo de Parámetros Generales	En qué opción(s) tuvo mayor dificultad en el manejo de datos	Roles	17%	
		Clientes	0%	
		Personas de Desarrollo	8%	
		Ninguna	75%	
	Al buscar y listar datos	Todos los campos de búsqueda fueron los apropiados para filtrar la información, pero falta más opciones de búsqueda		8%
		Los campos y opciones de búsqueda fueron suficientes para filtrar la información requerida		75%
		Los campos seleccionados para mostrar en la lista de datos fue la apropiada		50%
		Debería mostrarse más campos en la lista de datos		0%
	En qué opción(s) se requirió el ingreso de algún dato importante que el sistema no disponía	Roles		8%
		Clientes		17%
Personas de Desarrollo			0%	
Ninguna			75%	
Manejo de Proyectos	Al crear o editar un proyecto	Se pudo ingresar todos los datos, sin perder la secuencia de ingreso	83%	
		La aplicación le guió en el ingreso de todos los datos	58%	
		En el ingreso de datos se detectó validaciones que la aplicación no controlaba	0%	
	Al buscar proyectos	Hubo algún campo que se requirió para buscar un proyecto que no estuvo presente en los filtros de búsqueda		0%
		Se necesitó otras opciones de búsqueda para encontrar un proyecto		17%
		No se detectó ningún contratiempo en la búsqueda de proyectos		83%
	Al listar Proyectos	Los campos mostrados en la lista de Proyectos son los adecuados		75%
		Debería mostrarse más campos en la lista de Proyectos		33%
		Se detectó problemas al listar los Proyectos		0%
Administración de las Fases “Iniciación del Proyecto” y “Análisis del Sistema” propuestas en la guía	Usted como Cliente desea realizar una notificación inicial sobre el desarrollo de un Proyecto de Software, para ello utiliza las opciones: “Proyectos”, “Peticiónes Generales del Cliente”, y “Razones de Solicitud del Proyecto”. Cuál fue su apreciación	Con los datos solicitados se podría dar inicio a la negociación del proyecto, pero se requiere más detalle	25%	
		Con los datos solicitados se podría dar inicio a la negociación del proyecto	75%	

		Se encontró la necesidad de ingresar otros datos que la aplicación no disponía	8%
Usted como Líder de un Proyecto de Desarrollo en que rango de porcentaje cree usted que los datos manejados en las opciones "Proyectos", "Peticiones Generales del Cliente", y "Razones de Solicitud del Proyecto", le ayudarían a comprender lo que Cliente está solicitando	Del 1% al 30%		0%
	Del 31% al 50%		8%
	Del 51% al 70%		33%
	Del 71% al 100%		58%
Está usted de acuerdo que los datos manejados en la opción "Información del Proyecto" son suficientes para tener una visión general de la identificación de un Proyecto de Desarrollo	Parcialmente de acuerdo		0%
	Medianamente de acuerdo		25%
	Completamente de acuerdo		75%
La conformación del equipo de desarrollo utilizando la opción "Equipo de Desarrollo" le resultó	Muy Fácil		42%
	Fácil		58%
	Complicado		0%
	Muy Complicado		0%
Los datos de las Personas de Desarrollo manejadas en la opción "Parámetros Generales/Personas de Desarrollo" le ayudaron con algún criterio para seleccionar a las Personas de deberían formar parte de un Equipo de Desarrollo. Seleccionar el rango de porcentaje de ayuda	Del 1% al 30%		8%
	Del 31% al 50%		8%
	Del 51% al 70%		33%
	Del 71% al 100%		50%
Teniendo en cuenta que los Factores Críticos de Éxito hacen único a un Proyecto porque le ayudan a identificar cómo va a competir en el mercado; y los Factores de Riesgo ayudan a identificar los posibles problemas que se pueden enfrentar en un Proyecto. En base a lo expuesto la aplicación permite el ingreso de estos factores, y el registro de metas aplicando el criterio SMART mediante las opciones "Factores Críticos" y "Análisis de Metas" respectivamente. Piensa usted que los datos manejados en estas opciones pueden influir de manera positiva en la toma de decisiones	Si		100%
	No		0%
A través de las opciones "Iniciación del Proyecto/Procesos del Negocio" y "Iniciación del Proyecto/Actividades", la aplicación permite manejar información de los Procesos que requiere un sistema con sus respectivas actividades. Tener en cuenta que si se dispone de información adicional para comprender mejor los Procesos que serán automatizados, esta podrán ser anexada al Proyecto mediante la opción "Iniciación del Proyecto/Anexos". Seleccionar el rango de importancia que cree usted que puede resultar en un Proyecto de Desarrollo la información manejada en estas opciones	1 = no muy importante		0%
	4 = moderadamente importante		8%
	7 = muy importante		75%
	10 = extremadamente importante		17%
En el rango del 1 al 5 siendo el 1 el más bajo. Cuanto cree usted que es necesario la identificación de las Personas encargadas de ejecutar las Actividades de los Procesos que requiere un sistema. Utilizar la opción "Iniciación del Proyecto/Responsables de	1		0%
	2		0%
	3		8%

	Actividades" para registrar estas Personas, y la opción "Iniciación del Proyecto/ Identificación de Responsables por Actividad" para identificar las Actividades que tendrán asignadas	4	42%
		5	50%
	La opción "Iniciación del Proyecto/Identificación Macro de Procesos" permite determinar las salidas claves que provean un nivel de comprensión macro sobre los Procesos que requiere un sistema. Qué tan útil puede resultar esta opción en la identificación de un Proceso	Poco útil	0%
		Moderadamente útil	42%
		Muy útil	58%
	La definición de requerimientos se realiza mediante la opción "Iniciación del Proyecto/Definición de Requerimientos". Tener presente que la aplicación permite descargar un cuestionario para orientarle en la obtención de requerimientos. Estos requerimientos pueden ser analizados uno por uno mediante la opción "Análisis del Sistema/Análisis de Requerimientos". Para utilizar todas las opciones de Análisis de Requerimientos, se recomienda iniciar sesión como usuario "Analista". Después de finalizar el análisis de requerimientos, en qué rango de porcentaje cree usted que la aplicación le ayudo a definir el alcance del Proyecto	1% al 10%	0%
		11% al 20%	0%
		21% al 30%	0%
		31% al 40%	0%
		41% al 50%	0%
		51% al 60%	0%
		61% al 70%	17%
		71% al 80%	8%
		81% al 90%	67%
	91% al 100%	8%	
	Los requerimientos aceptados para su desarrollo pueden ser priorizados a través de la opción "Análisis del Sistema/Priorización de Requerimientos". En el rango del 1 al 5 siendo el 1 el más bajo, cuanto cree usted que la aplicación le ayudó en priorizar los requerimientos del Cliente en orden de desarrollo	1	0%
		2	0%
		3	0%
		4	58%
5		42%	
En cada fase de desarrollo se ha incluido una opción denominada "Anexos" que permite registrar cualquier archivo que sea de importancia para el Proyecto. La aplicación le brindó las facilidades para subir y descargar todos sus documentos	Si, pero hubo contratiempos	0%	
	Si, totalmente	100%	
	No	0%	
Generación de Reportes	La información presentada en cada reporte tiene relación con los datos ingresados	Si	100%
		No	0%
	En qué formato(s) le gustaría obtener los reportes	PDF	83%
		Documento Word	58%
		XML	0%
		HTML	8%
		Documento Excel	58%
RTF	0%		
Apreciación general de la herramienta	Las ayudas proporcionadas por la herramienta facilitaron	Ingreso de datos	58%
		Manejo de todas las opciones	67%
		Uso de toda la aplicación	75%
		Manejo de algunas opciones	25%
	Qué pudo apreciar en los mensajes de validaciones de ingreso de datos	Son Confusos	0%
		Son Claros, pero falta detallarlos	33%
		Son aceptables como para no cometer errores	67%
En cuanto al uso de opciones de acuerdo al tipo de usuario en sesión (Líder del	Se encontró la forma de acceder a una opción no autorizada al usuario	0%	

	Proyecto, Analista, Cliente), cual fue el comportamiento de la herramienta	La aplicación garantizó solo el uso de opciones autorizadas al usuario	100%
	Encontró alguna opción que no funcionó o produjo cierta caída de la aplicación, seleccione el rango de cantidad de opciones defectuosas encontradas	Ninguna	75%
		De 1 a 3	25%
		De 4 a 6	0%
		De 7 a 9	0%
		De 10 a 15	0%
		De 16 en adelante	0%
	Cree usted que la aplicación le resultó intuitiva en el uso de sus diferentes opciones	Nada intuitiva	0%
		Poco intuitiva	8%
		Bastante intuitiva	83%
		Totalmente intuitiva	8%
	Teniendo en cuenta que la herramienta en su versión actual permite la administración de dos fases de desarrollo de las seis que propone el modelo de la Guía mencionada en la introducción de la presente encuesta. Estaría usted de acuerdo en probar esta herramienta aplicándola en sus proyectos de desarrollo, pero al finalizar la automatización de todas las fases de desarrollo. Seleccione el porcentaje de aceptación	Del 1% al 30%	0%
		Del 31% al 50%	0%
		Del 51% al 70%	8%
		Del 71% al 100%	92%
	En el rango del 1 al 5 siendo el 1 el más bajo. En qué rango recomendaría usted a otras personas el uso de esta herramienta, con el fin de evaluarla y probarla en sus proyectos de desarrollo	1	0%
		2	0%
		3	0%
		4	42%
		5	58%

Tabla 3.46: (Resultados de la evaluación del software)

CAPÍTULO IV

4.1 Conclusiones

- ❖ Al aplicar la metodología ICONIX en el desarrollo del software, se pudo apreciar que la parte más importante y sobre la cual se fundamenta el análisis de un sistema es la identificación y redacción de los casos de uso.
- ❖ Para eliminar las posibles ambigüedades en la obtención de los requerimientos funcionales del sistema, se aplicó algunas directrices propuestas por ICONIX que facilitaron la ejecución de esta tarea, y es así, que mediante la técnica de “Análisis de Robustez” se logró efectuar un análisis profundo de los casos de uso, y a la vez se facilitó la identificación de objetos como interfaces, entidades, y controles presentes en los requerimientos; esto fue de gran utilidad al momento de pasar del análisis al diseño del sistema.
- ❖ La herramienta case Enterprise Architect facilitó la aplicación de la metodología ICONIX en el desarrollo del software, cabe mencionar que esta herramienta dispone de un ejemplo que abarca todas las fases de ICONIX con su respectiva explicación, el cual puede ser utilizado como formato para iniciar un proyecto de desarrollo de software. Este ejemplo es una guía muy práctica para personas que están iniciando en esta metodología.
- ❖ Se implementó un software orientado a la web que gestione la aplicación de la técnica de calidad SEIS SIGMA al proceso de desarrollo de software, pero solo se automatizaron la dos primeras fases que son: “Iniciación del

Proyecto” y “Análisis del Sistema” de las seis que propone la guía mencionada en los antecedentes del presente proyecto, entonces queda abierta la posibilidad de finalizar la automatización de las fases restantes, y así obtener un sistema que controle adecuadamente todo el proceso de desarrollo de software.

- ❖ Mediante la especificación EJB 3.0 de la plataforma Java EE 5.0, se aceleró considerablemente la codificación de los componentes de la capa de negocio del sistema, ya que no hay que preocuparse por las transacciones, la persistencia, la concurrencia, u otros servicios de nivel de sistema, porque estos son responsabilidad del servidor de aplicaciones; esto permitió centrarse exclusivamente en la codificación de la lógica de negocio del sistema.
- ❖ La configuración y creación del proyecto web, se aceleró gracias a la herramienta “Seam Generator” que forma parte del framework Seam.
- ❖ El framework de clases (EntityHome y EntityQuery) que proporciona Seam, facilitó el desarrollo de controladores que apoyan la ejecución de operaciones CRUD y consultas JPQL sobre las entidades JPA.
- ❖ Los resultados de una encuesta reflejaron que el software desarrollado en el presente proyecto, es muy intuitivo para el usuario en un 83% gracias a las diferentes ayudas, mensajes, y control de errores que proporciona este software. Cabe mencionar que el 92% de los usuarios respondió que estaría dispuesto en utilizar esta herramienta, y además recomendó el uso de este software a otras personas, seleccionando un rango del 1 al 5

(siendo el 1 el más bajo) obteniéndose el siguiente resultado 4 – 42% y 5 – 58%.

4.2 Recomendaciones

- ❖ Utilizar herramientas que apoyen el control de Proyectos de Desarrollo de Software y permitan tomar decisiones de mejoramiento continuo, y por consiguiente obtener un Software de calidad.
- ❖ Emplear metodologías que fomenten la aplicación de buenas prácticas en el Proceso de Desarrollo de Software.
- ❖ Usar herramientas case que permitan la aplicación de metodologías en el análisis y diseño de Proyectos de Desarrollo de Software; esto facilitará el mantenimiento del mismo.
- ❖ Al desarrollar una aplicación web con Seam Framework, se recomienda recurrir a “Eclipse” como IDE de desarrollo, porque la mayoría de las herramientas que facilitan la creación y edición de este tipo de aplicaciones están hechas para este IDE.
- ❖ Para obtener una herramienta que permita el control de todo el proceso de desarrollo de software, se recomienda continuar con el desarrollo de las cuatro fases restantes que son: Diseño del Sistema, Construcción, Evaluación y Aseguramiento de Calidad, e Implementación, de las seis fases que propone el modelo de la Guía utilizada como base en el presente proyecto.

BIBLIOGRAFÍA

- ❖ Doug Rosenberg and Matt Stephens. (2007). **Use Case Driven Object Modeling with UML.** Berkeley: Apress.
- ❖ Kevin Mukhar & Chirs Zelenak, with James L. Weaver, Jim Crume, (2006). **Beginning Java EE 5.** United States of America: Apress.
- ❖ Mike Keith, Merrick Schincariol, (2006). **Pro EJB 3: Java Persistence API.** United States of America: Apress.
- ❖ Debu Panda, Reza Rahman, Derek Lane, (2007). **EJB 3 in Action.** United States of America: Manning Publications Co.
- ❖ Dan Allen, (2009). **Seam in Action.** United States of America: Manning Publications Co.
- ❖ Joseph Faisal Nusairat, (2007). **Beginning JBoss Seam.** United States of America: Apress.
- ❖ Jim Farley, (2007). **Practical JBoss Seam Projects.** United States of America: Apress.
- ❖ Jacobson I., Booch G., Rumbaugh J., (2000). **El proceso unificado de desarrollo de software.** Madrid: Pearson Educación, S.A.
- ❖ Cecilia Hinojosa, R., (2008). **Seis Sigma y el Proceso de Desarrollo de Software.** Obtenida el 20 de Noviembre del 2009, de http://www.espe.edu.ec/portal/files/sitiocongreso/congreso/c_computacion/SeisSigmaPDS.pdf.
- ❖ Pérsico, Carolina, Z., Daniel Oscar. (2009). **Modelo de Dominio de Larman-RUP vs Modelo de Dominio de ICONIX.** Obtenida el 5 de Marzo del 2010, de URL.
- ❖ Jorge Arias, (2004). **Aplicaciones Empresariales sobre Plataformas J2EE Open Source.** Obtenida el 20 de Noviembre del 2009, de <http://www.acis.org.co/memorias/SalonInformatica/XXIVSalonInformatica/C onferencia-J2EE.pdf>.
- ❖ Carla, S. **Metodología ICONIX.** Obtenida el 22 de Febrero del 2010, de <http://www.portalhuarpe.com.ar/Seminario09/archivos/MetodologiaICONIX.pdf>.
- ❖ Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, (2004). **Control de Versiones con Subversion.**

- ❖ Sun Microsystems, Inc. **The Java EE 5 Tutorial**. Santa Clara, U.S.A. Obtenida el 4 de Abril del 2010 del sitio oficial de Sun, <http://java.sun.com/javaee/5/docs/tutorial/doc>.

BIOGRAFÍA

LUIS MIGUEL YACCHIREMA ESPÍN

Nació en la ciudad de Quito el 9 de Agosto de 1984.

Inició su formación académica en la escuela Fiscal “Federico González Suárez” (1996) de Salcedo – Cotopaxi. Sus estudios secundarios los realizó en tres colegios, por motivos de la profesión militar de su padre, iniciando en el Colegio Nacional Técnico “JAMBELI” (1996 – 1998) de Santa Rosa – El Oro, continuando en el Colegio Fiscal Mixto “Dr. Jorge Icaza Coronel” (1998 – 2000) de Guayaquil – Guayas, finalizando la secundaria en el Colegio Nacional “Juan de Salidas” (2000 – 2002) de Sangolquí – Pichincha, y obtuvo el título de “Bachiller en ciencias especialidad Químico – Biológicas”.

Realizó sus estudios superiores en la Escuela Politécnica del Ejército (ESPE), egresando en la carrera de Ingeniería en Sistemas e Informática (2004 – 2009), y obtuvo el certificado de suficiencia en el idioma Inglés en la misma institución.

En el año 2006 colaboró como soporte técnico en el diseño de animaciones web para el Proyecto de Educación Virtual de la ESPE. Trabajó como Analista – Programador en el Sistema Administrativo Financiero de la empresa Termopichincha S.A., (2007 – 2008). Colaboró en la implantación del Sistema Administrativo Financiero en la empresa Termoesmeraldas S.A., realizando tareas de análisis, diseño, y capacitación de usuarios (2008). Desempeñó el cargo de “Líder de Desarrollo” en el Proyecto de Gestión de Recursos Humanos para la empresa Carrasco & Asociados Consultores (2008 – 2009).

Se graduó de Ingeniero en Sistemas e Informática, en la Escuela Politécnica del Ejército (2011) de Sangolquí – Pichincha.

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

Sr. Yacchirema Espín Luis Miguel

COORDINADOR DE LA CARRERA

Sr. Ing. Mauricio Campaña

Lugar y fecha: Sangolquí, Agosto del 2011