

**SISTEMA DE INFORMACIÓN
PARA LA AUTOMATIZACIÓN DEL MÉTODO DE
DESARROLLO DE LA ARQUITECTURA
EMPRESARIAL.
AUTOR:
JOSÉ OSUNA BARRIOS**



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

**“Sistema de Información para la automatización del
Método de Desarrollo de la Arquitectura Empresarial”**

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela,
por el bachiller:

José Osuna Barrios
C.I.: 16.022.535

Para optar por el Título de Licenciado en Computación

Tutor:

Dr. Pedro Bonillo

Junio, 2013

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Investigación en Sistemas de Información



ACTA DEL VEREDICTO

Quienes suscriben, Miembros del Jurado designados por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado, presentado por el Bachiller José Misael Osuna Barrios C.I. 16.022.535, con el título **“Sistema de Información para la automatización del Método de Desarrollo de la Arquitectura Empresarial”**, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de los siguiente:

Leído, como fue dicho trabajo, por cada uno de los Miembros del Jurado, se fijó el día 17 de junio de 2013, para que su autor lo defendiera en forma pública, lo cual éste realizó, mediante una exposición oral de su contenido, y luego respondió satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela.

Finalizada la defensa pública del Trabajo Especial de Grado, el Jurado decidió APROBARLO con 20 puntos.

Firmas del Tutor y Jurados Examinadores:

Dr. Pedro Bonillo
(Tutor)

Prof. Carlos Acosta
(Jurado Principal)

Prof. Wilfredo Rangel
(Jurado Principal)

DEDICATORIA

Por el amor a mi madre Rosalba Barrios de Osuna, quien con su esfuerzo, amor, paciencia y cariño hizo que este trabajo fuera más fácil de sobrellevar, también está dedicado a mi padre José Osuna Santiago, de quien aprendí que la responsabilidad es uno de los valores más importantes de la vida, a mi hermana, Gina Paola Osuna, quien ha llevado las riendas de la familia en situaciones importantes, permitiéndome crecer y madurar a mi ritmo, y a Ana Mireya Gómez, sin su dedicación y amor esto no hubiese sido posible.

En memoria de mi eterno y recordado tío Diomedes Osuna Santiago, “tío lo logré, gracias por haberme enseñado que existe algo más allá de lo que vivimos día a día, y para lo cual debemos prepararnos”.

“El talento sin probidad es un
azote”.

Simón Bolívar.

AGRADECIMIENTOS

Durante el camino que ha tocado recorrer para obtener el título de Licenciado en Computación, otorgado por la ilustre Universidad Central de Venezuela, muchas personas han contribuido directamente para que el requisito final que se materializa con este Trabajo Especial de Grado, se haga posible, entre dichas personas tengo el honor de mencionar a Franklin Laya, cuyo apoyo, guía y consejos me encaminaron a lograr mi objetivo, también tengo el agrado de mencionar muy especialmente a Joel Zerpa, cuya tutoría y consejos personales me ayudaron a entregar un trabajo bien conciso y de calidad profesional, a Daniel Cabrices, cuya oportunidad y apoyo me ayudaron a desarrollar la capacidad de mantenerme enfocado por un periodo de tiempo prolongado para cumplir un objetivo, y cuyo ojo agudo me condujo al mercado laboral en un momento crucial para mi vida, a Miguel Bautista, cuyo apoyo intelectual me ayudó a reencontrarme y enamorarme nuevamente de mi carrera, a mi tutor de tesis, Pedro Bonillo, quien me dio una gran oportunidad y me alentó a llevar el proyecto a su exitoso desenlace; muchas personas también han realizado aportes para lograr que este trabajo llegará a su culminación, entre ellas tenemos a Anggi Anzola, Enrique Buono, Ana Morales, Wilfredo Rangel, Ismael Tobeilla, Hernán Abreu, Yrcel Chaudary, Gabriela Gruber, Andrés Placeres, Deiberg López, Rafael Pieters, Ricardo Subero, Adriana Peralta, Anakarina Giménez, Carlos Molina. Igualmente quiero aprovechar la oportunidad para mencionar a otro grupo de personas cuyas visiones y decisiones me han cambiado la vida y han hecho su aporte para que hoy me faculte como Licenciado en Computación: Graciela, mi maestra de tareas dirigidas en primer grado, me abrió la imaginación y me hizo pensar, a tan corta edad, en que algún día podía obtener un grado universitario y que esto era algo importante, Adalberto Guerra, mi maestro de sexto grado, me enseñó la importancia de la excelencia estudiantil y me habló seriamente de la vida universitaria, Máyela Palacios, Coordinadora de control de estudios de mi liceo de educación básica, me dio la zonificación al Liceo Francisco Fajardo y esa decisión me cambió el destino, también a la profesora coordinadora del Samuel Robinson en el Liceo Francisco Fajardo cohorte 2001, su llamada me llevó directamente a la Universidad Central de Venezuela.

Muchas Gracias.

José Osuna Barrios

**UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN**

**Sistema de Información para la automatización del Método de
Desarrollo de la Arquitectura Empresarial**

Autor: Br. José Osuna Barrios

Tutor: Dr. Pedro Bonillo

RESUMEN

La finalidad de este Trabajo Especial de Grado es ofrecer un desarrollo de un Sistema de Información para la automatización del Método de Desarrollo de la Arquitectura Empresarial.

Para cumplir con el objetivo de este trabajo, se procederá a analizar las principales arquitecturas empresariales existentes en la actualidad, seleccionar una Arquitectura Empresarial para su análisis y presentar un desarrollo de un Sistema de Información para la automatización de dicha Arquitectura, donde: se define el alcance general del sistema a desarrollar, mediante Historias de Usuario, plasmándolo en un Plan de Entregas. Se desarrollan los Casos de Prueba del sistema dirigidos desde las Historias de Usuario. Se desarrolla el sistema basándose en un Plan de Iteraciones, dividido en Tareas, que sigue el cronograma establecido en el Plan de Entregas.

Por limitaciones de tiempo y complejidad, el Sistema de Información para la automatización de una Arquitectura Empresarial descrito en este Trabajo Especial de Grado, ha sido acotado para que sólo la Fase Preliminar sea diseñada y desarrollada con dos Casos de Pruebas de Aceptación por cada Historia de Usuario del sistema propuesta por el cliente.

Palabras claves: Arquitectura Empresarial.

ABSTRACT

The main goal of this work is to provide a development of an information system for automation of Development Method of Enterprise Architecture.

To meet this objective, will proceed to analyze the main enterprise architectures currently available, select an Enterprise Architecture for analysis and present a development of an information system for automation of the architecture, where: defines the general scope of the system to be developed, using User Stories, capturing it in the Release Plan. They develop system Test Cases designed from User Stories. System is developed based on an Iteration Plan, divided into Tasks, that follows the schedule established in the Release Plan.

Due to time constraints and complexity, Information System for automation of an Enterprise Architecture described in this work, has been limited to only the Preliminary Phase has been designed and developed with two Acceptance Test Cases for each User Stories proposed by client.

Keywords: Enterprise Architecture.

ÍNDICE

ACTA.....	i
DEDICATORIA.....	ii
AGRADECIMIENTOS	iii
RESUMEN	iv
ABSTRACT	v
ÍNDICE	vi
ÍNDICE DE FIGURAS	x
ÍNDICE DE TABLAS.....	xi
INTRODUCCIÓN	1
CAPÍTULO I: EL PROBLEMA.....	3
1.1 Contexto	3
1.2 Planteamiento del Problema	4
1.3 Manifestaciones y Evidencias del Problema.....	5
1.4 Objetivos.....	6
1.4.1 Objetivo General	6
1.4.2 Objetivos Específicos	6
1.5 Justificación	6
1.6 Alcance	8
CAPÍTULO II: MARCO TEÓRICO	9
2.1 Bases Teóricas	9
2.1.1 Marcos de Arquitectura Empresarial.....	9
2.1.1.1 El Marco de Arquitectura Empresarial de Zachman.....	9
2.1.1.1.1 Descripción Específica	9
2.1.1.1.2 Vistas o Filas	10
2.1.1.1.3 Modelos o Celdas.....	12
2.1.1.2 La Arquitectura Empresarial Federal (<i>FEA</i>)	12
2.1.1.2.1 La perspectiva de la vista de la Arquitectura Empresarial	13
2.1.1.2.2 Los Modelos de referencia	14
2.1.1.2.3 El Proceso de desarrollo de una Arquitectura Empresarial	16
2.1.1.2.4 La medición del éxito de la Arquitectura Empresarial Federal	17
2.1.1.3 El Marco de Arquitectura Empresarial de Oracle (<i>OEAF</i>)	18
2.1.1.3.1 Los Componentes	18
2.1.1.3.2 El Proceso de Desarrollo de la Arquitectura de Oracle.....	23
2.1.1.3.3 Valor y Beneficios	24
2.1.1.4 El Marco de Arquitectura de The Open Group (<i>TOGAF</i>).....	25
2.1.1.4.1 Los Dominios Arquitectónicos	26
2.1.1.4.2 La Clasificación de los Resultados	27
2.1.1.4.3 El Continuum Empresarial	27
2.1.1.4.4 El Repositorio de la Arquitectura	29
2.1.1.5 Comparativa entre Marcos de Arquitecturas Empresariales	31

2.1.1.5.1 ¿Por qué <i>TOGAF</i> ?	36
2.1.2 El Método de Desarrollo de la Arquitectura (ADM)	37
2.1.2.1 Las Fases del Método de Desarrollo de la Arquitectura	38
2.1.2.1.1 Fase Preliminar	39
2.1.2.1.1.1 Objetivos de la Fase Preliminar	40
2.1.2.1.1.2 Elementos Claves de la Fase Preliminar	41
2.1.2.1.1.2.1 La Empresa	41
2.1.2.1.1.2.2 El Contexto Organizacional	42
2.1.2.1.1.2.3 Los Requerimientos para desarrollar la Arquitectura	43
2.1.2.1.1.2.4 Los Principios de la Arquitectura	43
2.1.2.1.1.3 Pasos de la Fase Preliminar	44
2.1.2.1.1.3.1 Identificar el alcance de las Unidades de Negocio Impactadas	44
2.1.2.1.1.3.2 Verificar la Gobernabilidad y los Marcos de Trabajo Soportados	45
2.1.2.1.1.3.3 Definir el equipo que desarrollará la Arquitectura Empresarial	45
2.1.2.1.1.3.4 Identificar y establecer los Principios de la Arquitectura	45
2.1.2.1.1.3.5 Seleccionar y adaptar los Marcos de Trabajo Arquitectónicos	46
2.1.2.1.1.3.6 Implementar las Herramientas de la Arquitectura	46
2.1.2.1.1.4 Salidas de la Fase Preliminar	47
2.1.3 Los Principios de la Arquitectura	48
2.1.3.1 La plantilla de <i>TOGAF</i> para la definición de los Principios de la Arquitectura	49
2.1.3.2 Ejemplo de los Principios de la Arquitectura	50
2.1.3.2.1 Los Principios del Negocio	50
2.1.3.2.2 Los Principios de los Datos	51
2.1.3.2.3 Los Principios de las Aplicaciones	51
2.1.3.2.4 Los Principios Tecnológicos	51
2.1.4 Niveles de Madurez de Arquitectura Empresarial	52
2.1.4.1 Nivel 0: ninguno	52
2.1.4.2 Nivel 1: inicial	52
2.1.4.3 Nivel 2: en desarrollo	53
2.1.4.4 Nivel 3: definido	53
2.1.4.5 Nivel 4: gestionado	54
2.1.4.6 Nivel 5: en optimización	54
2.2 Metodología de Desarrollo de Software	55
2.2.1 Metodologías Estructuradas	56
2.2.1.1 Metodologías Orientadas a Procesos	56
2.2.1.2 Metodologías Orientadas a Datos	57
2.2.2 Metodologías No Estructuradas	57
2.2.2.1 Metodologías Orientadas a Objeto	57
2.2.2.2 Metodologías de Sistemas de Tiempo Real	57
2.2.3 Metodologías Tradicionales vs. Metodologías Ágiles	57
2.2.3.1 Metodologías Tradicionales	57

2.2.3.2 Metodologías Ágiles	58
2.2.3.2.1 UP Ágil (<i>Agile Unified Process, AUP</i>)	59
2.2.3.2.2 Programación Extrema (<i>XP</i>).....	61
2.3 Herramientas Tecnológicas.....	61
2.3.1 Navegador Web.....	61
2.3.2 Servidor Web	62
2.3.3 Plataforma Java Edición Empresarial	63
2.3.4 Tecnologías del lado del Cliente	64
2.3.4.1 HTML	64
2.3.4.2 HTML5	64
2.3.4.3 Hojas de Estilo en Cascada	65
2.3.4.4 JavaScript	65
2.3.4.5 AJAX	65
2.3.4.6 JQuery.....	66
2.3.5 Tecnologías del lado del Servidor	67
2.3.5.1 Contenedor Web	67
2.3.5.2 XML	67
2.3.5.3 XHTML	67
2.3.5.4 La Tecnología Java.....	67
2.3.5.5 Servlets.....	69
2.3.5.6 La Arquitectura MVC	71
2.3.5.7 JavaServer Faces.....	72
2.3.5.8 PrimeFaces	73
2.3.5.9 JavaBean	73
2.3.5.10 Enterprise JavaBeans	74
2.3.5.11 GlassFish.....	74
2.3.5.12 NetBeans	75
2.3.5.13 PostgreSQL	75
CAPÍTULO III: MARCO METODOLÓGICO.....	76
3.1 Nivel y tipo de Investigación	76
3.2 Diseño de la Investigación	77
3.3 Técnicas de Recolección de Datos	77
3.4 Procedimiento de Análisis de Datos.....	78
3.5 Metodología utilizada en función al proyecto.....	79
3.5.1 Fase de exploración	81
3.5.2 Fase de planificación	83
3.5.3 Fase de iteraciones.....	84
CAPÍTULO IV: MARCO APLICATIVO Y RESULTADOS	88
4.1 Caso de Estudio.....	88
4.1.1 Identificación de la Empresa	88
4.1.1.1 Capas360 C.A.....	88
4.1.1.2 Misión	88
4.1.1.3 Visión.....	88
4.1.1.4 Objetivos de la empresa	88
4.1.1.5 Organigrama de la Empresa	89
4.1.1.6 Descripción de la Estructura Organizacional de la Empresa.....	90

4.1.1.6.1 Direcciones	90
4.1.1.6.1.1 Dirección Ejecutiva	90
4.1.1.6.1.2 Subdirección	90
4.1.1.6.2 Departamentos y Unidades	90
4.1.1.6.2.1 Departamento de Procura	90
4.1.1.6.2.2 Departamento de Recursos Humanos.....	90
4.1.1.6.2.3 Departamento de Finanzas.....	91
4.1.1.6.2.4 Departamento de Mercadeo	92
4.1.1.6.2.5 Departamento de Sistemas de Información	92
4.1.1.6.2.6 Departamento de Consultoría Jurídica	93
4.1.1.6.2.7 Departamento de Desarrollo de Software.....	93
4.1.1.6.2.8 Departamento de Planificación	94
4.1.1.6.2.9 Departamento de Operaciones	94
4.2 Fase de exploración	95
4.2.1 Historias de Usuario	95
4.2.2 Plan de Estimación	96
4.3 Fase de planificación	97
4.3.1 Plan de Entregas	97
4.4 Fase de iteraciones	98
4.4.1 Plan de Iteraciones.....	98
4.4.2 Casos de Prueba de Aceptación.....	99
4.4.3 Tareas de Programación.....	100
4.4.4 Tarjetas C.R.C.....	100
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES.....	104
REFERENCIAS BIBLIOGRÁFICAS	107
GLOSARIO.....	110
ANEXOS	115

ÍNDICE DE FIGURAS

Figura 1: Mapa del Segmento del Gobierno Federal.....	15
Figura 2: Componentes del Marco de Arquitectura de Oracle	19
Figura 3: Proceso de Desarrollo de la Arquitectura de Oracle <i>OADP</i>	23
Figura 4: Categorías <i>TOGAF</i>	28
Figura 5: Continuum Empresarial	29
Figura 6: Repositorio de la Arquitectura	30
Figura 7: Método de Desarrollo de la Arquitectura.....	37
Figura 8: Pasos de la Fase Preliminar	40
Figura 9: Elementos Claves de la Fase Preliminar	42
Figura 10: Salidas de la Fase Preliminar.....	47
Figura 11: Ciclo de vida de UP Ágil	60
Figura 12: Plataforma Java EE 6.....	62
Figura 13: Ciclo de vida de un Servlet	70
Figura 14: Ciclo de vida de un proyecto en Programación Extrema.....	80
Figura 15: Organigrama de la Empresa Capas360 C.A.	89
Figura 16: Arquitectura del Sistema	101
Figura 17: Módulo de Seguridad y Módulo de Operaciones	102
Figura 18: Módulos de Analista, Especialista y Líder de Planificación.....	103

ÍNDICE DE TABLAS

Tabla 1: El Marco de Arquitectura Empresarial de Zachman	10
Tabla 2: Comparativa entre Marcos de Arquitecturas Empresariales	36
Tabla 3: La plantilla de TOGAF para la definición de los Principios de la Arquitectura .	49
Tabla 4: Ejemplo de un principio de arquitectura	50
Tabla 5: Plan de Estimación.....	81
Tabla 6: Historia de Usuario	82
Tabla 7: Plan de Entregas.....	83
Tabla 8: Plan de Iteraciones	84
Tabla 9: Descripción de Tarea.....	85
Tabla 10: Tarjeta CRC	86
Tabla 11: Caso de Prueba de Aceptación.....	87
Tabla 12: Historia de Usuario 1.....	95
Tabla 13: Historia de Usuario 2.....	115
Tabla 14: Historia de Usuario 3.....	115
Tabla 15: Historia de Usuario 4.....	116
Tabla 16: Historia de Usuario 5.....	116
Tabla 17: Historia de Usuario 6.....	117
Tabla 18: Historia de Usuario 7.....	117
Tabla 19: Historia de Usuario 8.....	118
Tabla 20: Historia de Usuario 9.....	118
Tabla 21: Historia de Usuario 10.....	119
Tabla 22: Historia de Usuario 11.....	119
Tabla 23: Historia de Usuario 12.....	120
Tabla 24: Historia de Usuario 13.....	120
Tabla 25: Historia de Usuario 14.....	121
Tabla 26: Historia de Usuario 15.....	121
Tabla 27: Historia de Usuario 16.....	122
Tabla 28: Historia de Usuario 17.....	122
Tabla 29: Historia de Usuario 18.....	123
Tabla 30: Historia de Usuario 19.....	123
Tabla 31: Historia de Usuario 20.....	124
Tabla 32: Historia de Usuario 21.....	124
Tabla 37: Caso de Prueba de Aceptación CA1-2	125
Tabla 38: Caso de Prueba de Aceptación CA2-1	125
Tabla 39: Caso de Prueba de Aceptación CA2-2	126
Tabla 40: Caso de Prueba de Aceptación CA3-1	126
Tabla 41: Caso de Prueba de Aceptación CA3-2	127
Tabla 42: Caso de Prueba de Aceptación CA4-1	127
Tabla 43: Caso de Prueba de Aceptación CA4-2	128
Tabla 44: Caso de Prueba de Aceptación CA5-1	128
Tabla 45: Caso de Prueba de Aceptación CA5-2	129

Tabla 46: Caso de Prueba de Aceptación CA6-1	129
Tabla 47: Caso de Prueba de Aceptación CA6-2	130
Tabla 48: Caso de Prueba de Aceptación CA7-1	130
Tabla 49: Caso de Prueba de Aceptación CA7-2	131
Tabla 50: Caso de Prueba de Aceptación CA8-1	131
Tabla 51: Caso de Prueba de Aceptación CA8-2	132
Tabla 52: Caso de Prueba de Aceptación CA9-1	132
Tabla 53: Caso de Prueba de Aceptación CA9-2	133
Tabla 54: Caso de Prueba de Aceptación CA10-1	133
Tabla 55: Caso de Prueba de Aceptación CA10-2	134
Tabla 56: Caso de Prueba de Aceptación CA11-1	134
Tabla 57: Caso de Prueba de Aceptación CA11-2	135
Tabla 58: Caso de Prueba de Aceptación CA12-1	135
Tabla 59: Caso de Prueba de Aceptación CA12-2	136
Tabla 60: Caso de Prueba de Aceptación CA13-1	136
Tabla 61: Caso de Prueba de Aceptación CA13-2	137
Tabla 62: Caso de Prueba de Aceptación CA14-1	137
Tabla 63: Caso de Prueba de Aceptación CA14-2	138
Tabla 64: Caso de Prueba de Aceptación CA15-1	138
Tabla 65: Caso de Prueba de Aceptación CA15-2	139
Tabla 66: Caso de Prueba de Aceptación CA16-1	139
Tabla 67: Caso de Prueba de Aceptación CA16-2	140
Tabla 68: Caso de Prueba de Aceptación CA17-1	140
Tabla 69: Caso de Prueba de Aceptación CA17-2	141
Tabla 70: Caso de Prueba de Aceptación CA18-1	141
Tabla 71: Caso de Prueba de Aceptación CA18-2	142
Tabla 72: Caso de Prueba de Aceptación CA19-1	142
Tabla 73: Caso de Prueba de Aceptación CA19-2	143
Tabla 74: Caso de Prueba de Aceptación CA20-1	143
Tabla 75: Caso de Prueba de Aceptación CA20-2	144
Tabla 76: Caso de Prueba de Aceptación CA21-1	144
Tabla 79: Tarea 2, Historia 1	145
Tabla 80: Tarea 3, Historia 1	145
Tabla 81: Tarea 1, Historia 2	146
Tabla 82: Tarea 2, Historia 2	146
Tabla 83: Tarea 3, Historia 2	147
Tabla 84: Tarea 4, Historia 2	147
Tabla 85: Tarea 1, Historia 3	148
Tabla 86: Tarea 2, Historia 3	148
Tabla 87: Tarea 3, Historia 3	149
Tabla 88: Tarea 1, Historia 4	149
Tabla 89: Tarea 2, Historia 4	150
Tabla 90: Tarea 3, Historia 4	150
Tabla 91: Tarea 1, Historia 5	151
Tabla 92: Tarea 2, Historia 5	151

Tabla 93: Tarea 3, Historia 5	152
Tabla 94: Tarea 4, Historia 5	152
Tabla 95: Tarea 1, Historia 6	153
Tabla 96: Tarea 2, Historia 6	153
Tabla 97: Tarea 3, Historia 6	154
Tabla 98: Tarea 1, Historia 7	154
Tabla 99: Tarea 2, Historia 7	155
Tabla 100: Tarea 3, Historia 7	155
Tabla 101: Tarea 4, Historia 7	156
Tabla 102: Tarea 1, Historia 8	156
Tabla 103: Tarea 2, Historia 8	157
Tabla 104: Tarea 3, Historia 8	157
Tabla 105: Tarea 1, Historia 9	158
Tabla 106: Tarea 2, Historia 9	158
Tabla 107: Tarea 3, Historia 9	159
Tabla 108: Tarea 1, Historia 10	159
Tabla 109: Tarea 2, Historia 10	160
Tabla 110: Tarea 3, Historia 10	160
Tabla 111: Tarea 4, Historia 10	161
Tabla 112: Tarea 5, Historia 10	161
Tabla 113: Tarea 1, Historia 11	162
Tabla 114: Tarea 2, Historia 11	162
Tabla 115: Tarea 3, Historia 11	163
Tabla 116: Tarea 4, Historia 11	163
Tabla 117: Tarea 1, Historia 12	164
Tabla 118: Tarea 2, Historia 12	164
Tabla 119: Tarea 3, Historia 12	165
Tabla 120: Tarea 1, Historia 13	165
Tabla 121: Tarea 2, Historia 13	166
Tabla 122: Tarea 3, Historia 13	166
Tabla 123: Tarea 1, Historia 14	167
Tabla 124: Tarea 2, Historia 14	167
Tabla 125: Tarea 3, Historia 14	168
Tabla 126: Tarea 4, Historia 14	168
Tabla 127: Tarea 1, Historia 15	169
Tabla 128: Tarea 2, Historia 15	169
Tabla 129: Tarea 3, Historia 15	170
Tabla 130: Tarea 1, Historia 16	170
Tabla 131: Tarea 2, Historia 16	171
Tabla 132: Tarea 3, Historia 16	171
Tabla 133: Tarea 4, Historia 16	172
Tabla 134: Tarea 1, Historia 17	172
Tabla 135: Tarea 2, Historia 17	173
Tabla 136: Tarea 3, Historia 17	173
Tabla 137: Tarea 1, Historia 18	174

Tabla 138: Tarea 2, Historia 18	174
Tabla 139: Tarea 3, Historia 18	175
Tabla 140: Tarea 1, Historia 19	175
Tabla 141: Tarea 2, Historia 19	176
Tabla 142: Tarea 3, Historia 19	176
Tabla 143: Tarea 4, Historia 19	177
Tabla 144: Tarea 1, Historia 20	177
Tabla 145: Tarea 2, Historia 20	178
Tabla 146: Tarea 3, Historia 20	178
Tabla 147: Tarea 4, Historia 20	179
Tabla 148: Tarea 1, Historia 21	179
Tabla 149: Tarea 2, Historia 21	180
Tabla 150: Tarea 3, Historia 21	180
Tabla 152: C.R.C. TipoUnidad.....	181
Tabla 153: C.R.C. Sistema	181
Tabla 154: C.R.C. Proyecto	181
Tabla 155: C.R.C. RequerimientoDeCambio	181
Tabla 156: C.R.C. Rol.....	182
Tabla 157: C.R.C. Persona	182
Tabla 158: C.R.C. MesaTrabajo	182
Tabla 159: C.R.C. Patrocinador.....	182
Tabla 160: C.R.C. Tarea	183
Tabla 161: C.R.C. Artefacto.....	183
Tabla 162: C.R.C. TipoArtefacto	183
Tabla 163: C.R.C. Comunidad	183
Tabla 164: C.R.C. Usuario	184
Tabla 165: C.R.C. RepositorioArquitectura.....	184
Tabla 166: C.R.C. NivelMadurez	184
Tabla 167: C.R.C. Mejora	184
Tabla 168: C.R.C. Alcance.....	185
Tabla 169: C.R.C. Terminologia	185
Tabla 170: C.R.C. Premisa.....	185
Tabla 171: C.R.C. Requerimiento	185
Tabla 172: C.R.C. Adaptacion	186
Tabla 173: C.R.C. Presupuesto.....	186
Tabla 174: C.R.C. Principio	186
Tabla 175: C.R.C. TipoPrincipio	186

INTRODUCCIÓN

En la actualidad, al gestionar sistemas empresariales de software de gran complejidad surge un caos de aplicaciones para los gerentes de Tecnologías de la Información, tales aplicaciones en su mayoría, ofrecen la misma funcionalidad replicándose en diferentes gerencias y subgerencias a lo largo de toda la organización; por otro lado, al intentar integrar todos estos sistemas con tecnologías diferentes, los gerentes se ven obligados a contratar especialistas de las distintas plataformas a integrar, estos especialistas, que resuelven la costosa integración, no documentan suficientemente las resoluciones de las mismas, haciendo dicha integración y mantenimiento dependientes de ellos. Durante este Trabajo Especial de Grado analizaremos una de las soluciones actuales a dicha problemática llamada “gestión de componentes del sistema”, en vez de la tradicional construcción de aplicaciones, esta gestión de componentes la llamaremos Arquitectura Empresarial.

Este Trabajo Especial de Grado se fundamenta en las siguientes líneas de investigación del área de Sistemas de Información: Desarrollo de Herramientas y Técnicas para el Análisis, Concepción y Diseño de Sistemas de Información, Análisis y Diseño de Sistemas de Información, Modelización de Sistemas de Información, Bases de Datos, Gerencia y Planificación de Sistemas de Información.

La finalidad de este Trabajo Especial de Grado es ofrecer un desarrollo de un Sistema de Información para la automatización del Método de Desarrollo de la Arquitectura Empresarial.

Para cumplir con el objetivo del presente Trabajo Especial de Grado, se procederá a analizar las principales arquitecturas empresariales existentes en la actualidad, seleccionar una Arquitectura Empresarial para su análisis y presentar un desarrollo de un Sistema de Información para la automatización de dicha Arquitectura.

A continuación se describe la estructura de este Trabajo Especial de Grado:

Capítulo I: El Problema. Se formula el problema del Trabajo Especial de Grado como resultado de la investigación realizada en el mismo. Se presenta todo lo relacionado al contexto, el problema, las manifestaciones y evidencias, seguido a esto, se plantean los objetivos, la justificación y Alcance de la investigación.

Capítulo II: Marco Teórico. En él se establecen las bases conceptuales que soportan la investigación. Se estudia detalladamente las principales Arquitecturas Empresariales y una comparativa entre las mismas, el Método de Desarrollo de la Arquitectura, la Metodología de Desarrollo de Software y las herramientas tecnológicas que dan soporte a la plataforma software de un Sistema de Información para la automatización del Método de Desarrollo de la Arquitectura Empresarial.

Capítulo III: Marco Metodológico. Se describe el tipo de investigación, su diseño, las técnicas de recolección de datos y los procedimientos de análisis de datos que se utilizaron durante la investigación.

Capítulo IV: Marco Aplicativo y Resultados. Se expone el desarrollo del sistema en función de la metodología propuesta.

Capítulo V: Conclusiones y Recomendaciones. Se presentan las conclusiones y recomendaciones de la investigación realizada en este Trabajo Especial de Grado.

Para finalizar, se exponen las referencias consultadas que dieron soporte a la investigación y un glosario de términos.

CAPÍTULO I: EL PROBLEMA

En este capítulo se describe ampliamente la situación objeto de estudio, ubicándola en un contexto para entender su origen y de esta manera proceder a plantear, justificar y delimitar el problema; también se definen los objetivos que se cubren en este Trabajo Especial de Grado.

1.1 Contexto

En la actualidad, al gestionar sistemas empresariales de software de gran complejidad surge un caos de aplicaciones para los gerentes de Tecnologías de la Información, dichas aplicaciones, en la mayoría de los casos, ofrecen la misma funcionalidad replicándose en diferentes gerencias y subgerencias a lo largo de toda la organización; por otro lado al querer integrar todos estos sistemas con tecnologías diferentes, los gerentes se ven obligados a contratar a especialistas de las distintas plataformas a integrar, los cuales, una vez resuelta la costosa integración, no documentan suficientemente las resoluciones de las mismas, haciendo dicha integración, su escalabilidad y mantenimiento dependientes de ellos.

Una Arquitectura Empresarial proporciona un marco en el que la planificación del desarrollo de aplicaciones pueda realizarse en el grupo y niveles del proyecto. Una arquitectura Empresarial puede guiar decisiones acerca de qué aplicaciones deberían ser construidas (Brancheau, 1996).

El propósito de una Arquitectura Empresarial es optimizar los procesos en toda la empresa (tanto manuales como automatizados) en un entorno integrado que responda a los cambios y sirva de apoyo a la estrategia empresarial.

En efecto, existen sistemas organizacionales en la Tecnología de la Información con el objetivo de asegurar que se implemente y controle la calidad de los procesos de la empresa de acuerdo con una Arquitectura Empresarial.

1.2 Planteamiento del Problema

En base a lo expuesto anteriormente, surgen las siguientes interrogantes:

- ¿Es posible determinar, bajo un marco referencial, los procesos de la organización y determinar qué funciones necesitan ser desarrolladas para ser un negocio competitivo?
- Si se consigue capturar y representar estos procesos, ¿Se podrían elaborar mapas de las estructuras de negocio en relación con las funciones del negocio para determinar qué gerentes son responsables de (o desempeñan un rol en) cada función del negocio?
- ¿Se podrían elaborar mapas de la información sobre aplicaciones existentes con respecto a las funciones de negocio para reunir información sobre las funciones proporcionadas por los sistemas existentes y su conveniencia para cubrir las necesidades de información de la organización?
- ¿Cómo verificar la calidad del programa de la Arquitectura Empresarial en las TI garantizando su eficacia en el dominio de implementación?

La búsqueda de respuestas a estas interrogantes permite plantear un problema de investigación basado en la hipótesis de que las organizaciones en general fallan en el programa de la Arquitectura Empresarial, principalmente por factores como:

- La relativa complejidad de la ejecución del programa de Arquitectura Empresarial depende de factores como el nivel de compromiso de la organización, la disponibilidad de recursos y el tamaño de orientación, la complejidad del modelo de negocio de la organización, y la agilidad de la organización. La realidad es que muchas organizaciones simplemente no son capaces de implementar y mantener un programa de Arquitectura Empresarial en un tiempo mínimo.

- Muchos marcos generales de aplicación de Arquitectura Empresarial están enfocados en los artefactos organizativos y procesos en el contexto de la meta-estructura, limitando la creatividad y añadiendo burocracia al proceso de aplicación de la Arquitectura Empresarial, en vez de basar la arquitectura de la empresa en procesos, cuyo enfoque se centra principalmente en las actividades en lugar de los artefactos, siendo este enfoque más fácil de entender y vincular con las metodologías existentes y las técnicas de solución de la empresa.

1.3 Manifestaciones y Evidencias del Problema

Al gestionar sistemas empresariales de software de gran complejidad surge un caos de aplicaciones para los gerentes de Tecnologías de la Información, dichas aplicaciones, en la mayoría de los casos, ofrecen la misma funcionalidad replicándose en diferentes gerencias y subgerencias a lo largo de toda la organización; por otro lado al querer integrar todos estos sistemas con tecnologías diferentes, los gerentes se ven obligados a contratar a especialistas de las distintas plataformas a integrar, los cuales, una vez resuelta la costosa integración, no documentan suficientemente las resoluciones de las mismas, haciendo dicha integración, su escalabilidad y mantenimiento dependientes de ellos.

Una Arquitectura Empresarial proporciona un marco en el que la planificación del desarrollo de aplicaciones pueda realizarse por niveles en el grupo del proyecto. Sin embargo, las Arquitecturas Empresariales en sí, su elección, implantación y mantenimiento se han convertido en un proyecto sumamente costoso en tiempo y dinero; y muchas organizaciones simplemente no son capaces de implementar y mantener un programa de Arquitectura Empresarial en un tiempo mínimo.

Entonces se evidencia la búsqueda de un Marco de Arquitectura Empresarial que se aproxime al desarrollo de arquitecturas de manera “Ágil” sin prescribir los modelos existentes en la organización y que pueda ser modificado por los usuarios para adaptarlos a las necesidades de arquitectura de su negocio.

1.4 Objetivos

Seguidamente se presenta el objetivo general y los objetivos específicos de esta investigación.

1.4.1 Objetivo General

Desarrollar un sistema de información para la automatización del Método de Desarrollo de la Arquitectura Empresarial.

1.4.2 Objetivos Específicos

- Analizar los cuatro Marcos de Arquitectura Empresarial más usados actualmente.
- Seleccionar un Marco de Arquitectura, de acuerdo al análisis realizado, para ser desarrollado.
- Definir el alcance general del sistema a desarrollar, mediante Historias de Usuario, plasmándolo en un Plan de Entregas.
- Desarrollar los Casos de Prueba del sistema dirigidos desde las Historias de Usuario.
- Desarrollar el sistema basándose en un Plan de Iteraciones, dividido en Tareas, que sigue el cronograma establecido en el Plan de Entregas.

1.5 Justificación

Desde el punto de vista institucional y social, las empresas toman decisiones referentes a la tecnología que utilizan pero no lo hacen con conciencia; utilizan enfoques que tienden a subestimar aspectos importantes de la selección y uso de la tecnología que compran y de los procesos que se administran. En el marco de esta afirmación, la justificación de esta investigación se refiere a los aportes y beneficios que tendrá la empresa en cuanto a un modelo formal a seguir para la automatización de la Arquitectura Empresarial en los procesos de TI sustentada en el marco de una estructura de apoyo a los procesos de negocio de la organización (Bonillo, 2012).

Los componentes son las funcionalidades que idealmente se necesita que la organización trabaje, esos componentes son como Legos que al hacerlos encajar generan patrones, y con un conjunto de patrones se puede trabajar más ágil con los procesos de la organización (Bonillo, 2012).

La solución teórica actual a los problemas de Arquitectura comprende: 1. Identificar los componentes funcionales. 2. Identificar los patrones se necesitan. 3. Seleccionar las aplicaciones que te permiten hacer lo que se quiere que haga el componente funcional. Los procesos se pueden representar a través de patrones, los patrones a través de componentes, las aplicaciones a través de los componentes, y todo esto se integra a través de una metodología (Bonillo, 2012).

La metodología tiene que ser un conjunto de métodos, herramientas y técnicas, sobre las cuales se haya reflexionado y esté probado que se puede repetir, con el objeto de medir la calidad y reutilizar de una manera más adecuada el software de la organización. Un marco metodológico para entender la organización de TI, básicamente la Gerencia de Sistemas, la Gerencia de Procesos, la Gerencia de Calidad del Proceso es lo que llamamos Arquitectura Empresarial (Bonillo, 2012).

El propósito de una Arquitectura Empresarial es optimizar los procesos en toda la empresa (tanto manuales como automatizados) en un entorno integrado que responda a los cambios y sirva de apoyo a la estrategia empresarial.

La gestión eficaz y la explotación de la información a través de tecnología de la información son un factor clave para el éxito del negocio, y un medio indispensable para lograr una ventaja competitiva. Una arquitectura empresarial responde a esta necesidad, proporcionando un marco estratégico para la evolución del sistema de tecnología de la información en respuesta a las cambiantes necesidades del entorno empresarial.

1.6 Alcance

Este estudio está circunscrito al área de gerencia de procesos de negocio y los sistemas de información, dentro del dominio tecnológico existente en las ciencias computacionales para el año 2013: Software Libre, Ambientes de Soporte de Decisiones y Modelización de Sistemas de Información.

De los procesos de Desarrollo de la Arquitectura existentes, se seleccionó como objeto de estudio el Método de Desarrollo de la Arquitectura (*ADM*) de *TOGAF*, debido a que hay numerosos trabajos sobre Marcos para Arquitecturas Empresariales, pero pocas toman en cuenta el problema de la una Arquitectura Ágil.

Por limitaciones de tiempo y complejidad, el Sistema de Información para la automatización de una Arquitectura Empresarial descrito en este Trabajo Especial de Grado, ha sido acotado para que sólo la Fase Preliminar sea diseñada y desarrollada con dos Casos de Pruebas de Aceptación por cada Historia de Usuario del sistema propuesta por el cliente, el caso de estudio es la empresa Capas360 C.A. la cual será descrita en la sección 4.1 del capítulo 4.

CAPÍTULO II: MARCO TEÓRICO

En éste capítulo se presentan los antecedentes y las bases teóricas que sustentan la investigación.

2.1 Bases Teóricas

A continuación se presentan los fundamentos para la investigación, donde tendrá relevancia el Método de Desarrollo de la Arquitectura basado en las mejores prácticas de *TOGAF*.

2.1.1 Marcos de Arquitectura Empresarial

2.1.1.1 El Marco de Arquitectura Empresarial de Zachman

El Marco de Arquitectura Empresarial de Zachman provee una manera formal y sumamente estructurada de ver y definir la estructura de una empresa. Fue creado por John Zachman en los 1980's, quien se encontraba trabajando en *IBM* en el Sistema de Planeación de Negocios (*BSP*)

2.1.1.1.1 Descripción Específica

El Marco de Arquitectura Empresarial de Zachman provee 36 categorías necesarias para describir de manera completa cualquier cosa, especialmente, cosas complejas como bienes manufacturados (por ejemplo, componentes electrónicos), estructuras construidas (Edificios) y empresas (la organización y todos sus objetivos, personas y tecnologías). También provee de 6 vistas detalladas o niveles de abstracción desde 6 perspectivas diferentes. De esta manera, diferentes personas pueden mirar el mismo ítem de manera diferente, esto crea una vista holística del entorno, una capacidad sumamente importante en un ambiente multidisciplinario como el que se encuentra hoy en día en las empresas, en la Tabla 1 encontramos la vista actual del Marco de Arquitectura Empresarial de Zachman.

Tabla 1: El Marco de Arquitectura Empresarial de Zachman
Fuente: Zachman (2012)

	DATOS <i>Qué</i>	FUNCIÓN <i>Cómo</i>	REDES <i>Dónde</i>	PERSONAS <i>Quién</i>	TIEMPO <i>Cuándo</i>	MOTIVO <i>Por qué</i>
Planeación/Alcance (contextual) Rol: Planificador	Lista de cosas importantes para el negocio	Lista de Procesos del Negocio	Lista de Lugares del Negocio	Lista de organizaciones importantes	Lista de Eventos	Lista de Metas y Estrategias del Negocio
Modelo Empresarial (conceptual) Rol: Propietario	Modelo Objeto de Datos Conceptuales	Modelo de Negocios	Sistema Logístico del Negocio	Modelo del Flujo de Trabajo	Agenda Principal	Plan del negocio
Modelo del Sistema (lógico) Rol: Diseñador	Modelo de Lógico de los datos	Modelo de la Arquitectura del Sistema	Arquitectura de los Sistemas Distribuidos	Arquitectura de la Interface Humana	Estructura de Procesamiento	Modelo de las Reglas del Negocio
Modelo Tecnológico (físico) Rol: Constructor	Modelo de Clases/Datos Físicos	Modelo del Diseño de la Tecnología	Arquitectura Tecnológica	Arquitectura de Presentación	Estructura de Control	Reglas de Diseño
Especificación Detallada (fuera de contexto) Rol: Programador	Definición de los datos	Programa	Arquitectura de Red	Arquitectura de Seguridad	Definición de tiempos	Reglas de Tráfico

2.1.1.1.2 Vistas o Filas

Cada fila representa una vista total de la solución desde una vista particular. Una fila superior o perspectiva no tiene necesariamente un entendimiento de toda la perspectiva inferior. Cada fila representa una perspectiva única, sin embargo, los contenidos de cada perspectiva deben proveer suficiente detalle para definir la solución al nivel de la perspectiva y estos se deben transferir a la próxima fila inferior. Cada perspectiva debe tener en cuenta los requerimientos de las otras perspectivas y las limitaciones que éstas imponen. Las limitaciones de cada perspectiva se suman. Por ejemplo, las limitaciones de las filas superiores afectan a las inferiores. Las limitaciones de las filas inferiores pueden, pero no necesariamente afectan las filas superiores. Entender los requerimientos y limitaciones implica conocimiento y entendimiento de perspectiva a perspectiva.

Esta versión simplificada nos servirá para explicar el funcionamiento del marco:

- Fila 1 – Vista de Planeación / Alcance: El primer borrador de arquitectura es un diagrama de Venn el cual muestra en términos de tamaño, forma y relaciones parciales el propósito final de la estructura. Corresponde a un sumario ejecutivo para un planeador o inversionista que requiere una

perspectiva general del sistema, cuánto costaría y como se relacionaría con el sistema general donde este operará.

- Fila 2- Vista del Propietario / Modelo Empresarial: Lo siguiente son los planos del arquitecto que muestran cómo la construcción final será desde la perspectiva del usuario, el cual tendrá que interactuar con esta. Corresponden a los modelos de la empresa/negocio, los cuales constituyen los diseños del negocio y muestran las entidades del negocio y como se relacionan los procesos.
- Fila 3 – Vista del Diseñador / Modelo del Sistema: Los planos del arquitecto corresponden con la traducción de los dibujos a representaciones detalladas de los requerimientos desde una perspectiva de un diseñador. Ellos corresponden al modelo del sistema diseñado por un Analista el cual debe determinar los elementos de datos, el flujo de la lógica de los procesos y las funciones que representan entidades o procesos de negocios.
- Fila 4 – Vista del Constructor / Modelo Tecnológico: El contratista debe redibujar los planos del arquitecto para representar la perspectiva del constructor con suficiente detalle para entender las limitaciones de las herramientas, tecnologías y materiales. Los planos corresponden a los modelos tecnológicos, los cuales se deben adaptar al modelo de sistemas de información, estos tienen en cuenta los lenguajes de programación, los dispositivos de E/S y otras tecnologías de soporte.
- Fila 5 – Vista del Programador / Especificación Detallada: Los subcontratistas trabajan desde las plantas, en las cuales se especifican los detalles en partes o subsecciones. Estas corresponden a las especificaciones detalladas que se les dan a los programadores que desarrollan modelos específicos sin tener en cuenta el contexto general.
- Fila 6 – Vista Actual del Sistema / Columnas de Enfoque de la Empresa: Cada perspectiva le da un enfoque a una pregunta fundamental, donde estas preguntas se resuelven desde esa perspectiva, creando diferentes representaciones (modelos), las cuales se interpretan desde perspectivas de

alto a bajo nivel. Contamos con seis categorías con sus respectivas interrogativas:

- 1) Descripción de datos – Qué
- 2) Descripción de función – Cómo
- 3) Descripción de Redes – Dónde
- 4) Descripción del personal – Quién
- 5) Descripción del tiempo – Cuándo
- 6) Descripción de la motivación – Por qué

2.1.1.1.3 Modelos o Celdas

Los modelos se hacen explícitos en las intersecciones entre filas y columnas, a estas se les conoce como celdas, las cuales son únicas, su contenido es normalizado según el enfoque de la perspectiva. Las descripciones de estas utilizan un lenguaje general enfocado a un set específico de objetivos.

2.1.1.2 Arquitectura Empresarial Federal (FEA)

La Arquitectura Empresarial Federal (*FEA*) es el último intento del gobierno de los Estados Unidos para unir sus múltiples agencias y sus funciones bajo una Arquitectura Empresarial única que sea común y ubicua. *FEA* se encuentra todavía en desarrollo, ya que la mayoría de sus componentes más importantes han estado disponibles desde el 2006.

FEA es una metodología muy completa. Tiene una taxonomía y un proceso arquitectónico y consta de cinco modelos de referencia básicos a saber que son:

1. El Modelo de Referencia del Negocio (*BRM*)
2. El Modelo de Referencia de los componentes (*CRM*)
3. El Modelo de Referencia Técnica (*TRM*)
4. El modelo de referencia de datos (*DRM*)
5. El modelo de referencia de rendimiento (*PRM*)

Sin embargo, un tratamiento más completo de las necesidades de *FEA* puede incluir las siguientes características:

- Una perspectiva de la vista de la Arquitectura Empresarial (llamada el modelo de *segmento*)
- Un conjunto de modelos de referencia para la descripción de las diferentes perspectivas de la arquitectura de la empresa (son los cinco modelos mencionado anteriormente)
- Un proceso para la creación de una arquitectura empresarial
- Un proceso de transición para la migración de una pre-Arquitectura Empresarial a un paradigma de post-Arquitectura Empresarial
- Una taxonomía de los activos de catalogación que se encuentran dentro del ámbito de la arquitectura de la empresa
- Una aproximación a la medición del éxito de la utilización de la arquitectura empresarial para impulsar el valor agregado en la empresa.

FEA Incluye todo lo necesario para construir una arquitectura empresarial para la organización que probablemente sea la más compleja en la tierra: el gobierno de los Estados Unidos. La Oficina de Gestión del Programa *FEA* (*FEAPMO*) establece lo siguiente: “*FEA* es un lenguaje común y un marco para describir y analizar las inversiones en TI, mejorar la colaboración y en última instancia transformar el gobierno en una organización centrada en el ciudadano, orientado a resultados, y basado en el mercado como se establece en la Agenda Ejecutiva del Presidente”. (*FEA*, 2006). A continuación se presenta una descripción de sus componentes:

2.1.1.2.1 La perspectiva de la vista de la Arquitectura Empresarial

La perspectiva de la vista de la Arquitectura Empresarial de *FEA* indica que una empresa está constituida en segmentos (*FEA*, 2006).

Un segmento es una de las principales líneas de funcionalidad del negocio, tales como los recursos humanos. Hay dos tipos de segmentos: el segmento central y los segmentos de servicios de negocios.

Un segmento central es aquel que es fundamental para la misión o propósito de una línea política particular dentro de la empresa. Por ejemplo, en la Agencia de Salud y

Servicios Humanos del gobierno de los Estados Unidos (*HHS*), la salud es un segmento central.

Un segmento de servicios de negocios es aquel que es fundamental para la mayoría de las organizaciones políticas. Por ejemplo, la gestión de recursos humanos es un segmento de servicios de negocios que se requiere a todas las agencias federales.

Dentro de los activos empresariales, la arquitectura es un servicio de la empresa.

Un servicio de la empresa es una función bien definida que se extiende por las líneas políticas. Un ejemplo de un servicio de la empresa es la gestión de la seguridad. La gestión de la seguridad es un servicio que funciona de una manera unificada a través de toda la línea de la empresa.

El hecho de que los segmentos se definen globalmente facilita su reutilización a través de las líneas políticas. Uno puede planear el uso de segmentos a través de las líneas políticas de la empresa, luego, utilizar ese mapa para buscar oportunidades de reutilización arquitectónica. En la figura 1, por ejemplo, se muestra un mapa de segmentos del gobierno de los Estados Unidos (*FEA*, 2006). Como se puede observar, hay muchos segmentos (las columnas verticales) que se utilizan en múltiples organismos, y cualquiera o todos estos son buenos candidatos para reutilizar.

2.1.1.2.2 Los Modelos de referencia

Los cinco modelos de referencia *FEA* tratan de establecer lenguajes comunes. El objetivo es facilitar la comunicación, la cooperación y la colaboración a través de líneas políticas. De acuerdo con la *FEAPMO*:

FEA está compuesta por un conjunto de modelos de referencia interrelacionados diseñados para facilitar a las agencias el análisis y la identificación de inversiones duplicadas o agotadas, y las oportunidades para la colaboración dentro y entre agencias. Colectivamente, los modelos de referencia componen un marco para describir elementos importantes de *FEA* de una manera común y consistente. (*FEA*, 2006)

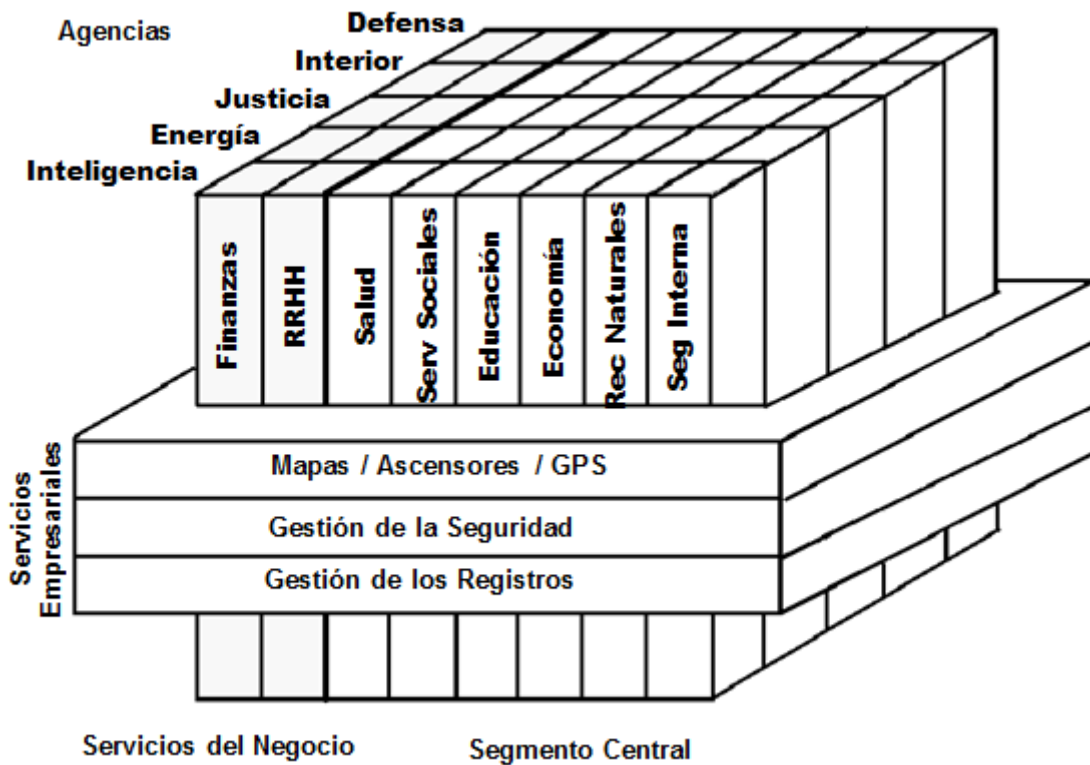


Figura 1: Mapa del Segmento del Gobierno Federal
Fuente: FEA (2012)

El problema que los modelos de referencia *FEA* están tratando de resolver son aquellos relativos a la comunicación y ambigüedad de términos entre las agencias así como de los componentes software de todas tus arquitecturas. Supongamos que el Servicio de Rentas Internas (*IRS*) decide que necesita un sistema de seguimiento de los datos demográficos de los contribuyentes. Solicitan a otras agencias si alguien tiene uno que puedan modificar para sus propósitos y ninguna agencia responde al llamado. Lo que no saben es que, justo en la puerta del lado, la Oficina de Imprenta del Gobierno (*GPO*) tiene un sistema de demografía que es exactamente lo que el *IRS* necesita. Por lo tanto, el *IRS* va y construye su sistema desde cero, en vez de modificar el ya construido (y pagado) por el *GPO*. Y, al hacerlo, el *IRS* pierde dinero y tiempo innecesario.

La meta de los cinco modelos de referencia de *FEA* es: dar términos y definiciones para los dominios de la arquitectura de la empresa y, con ello, facilitar la colaboración

y el intercambio a través del gobierno de los Estados Unidos. Los cinco modelos de referencia son los siguientes:

1. El Modelo de Referencia del Negocio (BRM) da una visión de negocio de las distintas funciones del gobierno. Por ejemplo, el *BRM* define una capacidad de negocio estándar llamada gestión de los recursos hídricos que es una subfunción de los recursos naturales que se considera un segmento central para el área de negocios ciudadanos. (FEA, 2006)
2. El Modelo de Referencia de los componentes (CRM) da una vista de los sistemas de TI que pueden soportar la funcionalidad del negocio. Por ejemplo, en el CRM se define un sistema como el que se ha descrito anteriormente en el hipotético intercambio entre el *IRS* y la *GPO*. (FEA, 2006)
3. El Modelo de Referencia Técnica (TRM) define las diversas tecnologías y estándares que se pueden utilizar en la construcción de los sistemas de TI. Por ejemplo, la *TRM* define *HTTP* como un protocolo de acceso de servicio y entrega de datos Internet. (FEA, 2006)
4. El modelo de referencia de datos (DRM) define los procedimientos estándares para describir los datos. Por ejemplo, la *DRM* define una entidad como algo que contiene atributos y participa en las relaciones. (FEA, 2006)
5. El modelo de referencia de rendimiento (PRM) define los procedimientos estándar para describir el valor entregado por las arquitecturas empresariales a la empresa. Por ejemplo, el *PRM* describe la *calidad* como un área de tecnología de medición que se define como "el grado en que la tecnología cumple con los requisitos de funcionalidad o capacidad." (FEA, 2006)

2.1.1.2.3 El Proceso de desarrollo de una Arquitectura Empresarial

El Proceso de desarrollo de una Arquitectura Empresarial en *FEA* se centra principalmente en la creación de una arquitectura de segmento para un subconjunto de toda la empresa (en caso de *FEA*, la empresa es el gobierno de los Estados Unidos y el subconjunto es una agencia gubernamental) (FEA, 2006).

A un nivel muy general el proceso de desarrollo de una Arquitectura Empresarial se describe como sigue:

- Paso 1: Análisis arquitectónico para definir una visión simple y concisa para el segmento.
- Paso 2: Definición de arquitectura: para definir el estado arquitectónico deseado del segmento, documentar los objetivos de rendimiento, considerar las alternativas de diseño y desarrollar una arquitectura empresarial para el segmento, incluyendo los negocios, datos, servicios y arquitecturas tecnológicas.
- Paso 3: Considerar la Inversión y el Financiamiento: cómo será financiado el proyecto.
- Paso 4: Crear un plan para la gestión y ejecución del proyecto, incluyendo los hitos y las medidas de desempeño que evaluarán el éxito del proyecto.

2.1.1.2.4 La medición del éxito de la Arquitectura Empresarial Federal

El Marco de la Arquitectura Empresarial Federal para medir el éxito de la organización en el uso de la Arquitectura Empresarial se define en el Programa de Evaluación del Marco de la Arquitectura Empresarial Federal en su versión 2.1 (FEA, 2006). Las agencias federales están valoradas en sus niveles de madurez general en tres categorías principales:

1. Madurez Arquitectónica, nivel de la propia arquitectura
2. Uso Arquitectónico, la eficacia con que la agencia utiliza su arquitectura para impulsar la toma de decisiones
3. Los Resultados Arquitectónicos, los beneficios conseguidos por el uso de la arquitectura

La Oficina de Administración y Presupuesto de Estados Unidos (*OMB*) asigna a cada agencia una calificación de éxito, en función de sus puntuaciones en cada categoría y un puntaje acumulado de la siguiente manera:

- Verde - La agencia muy bien consolidada (tiene una arquitectura empresarial muy madura). También un buen uso (se utiliza eficazmente la arquitectura empresarial para impulsar la estrategia en curso) y la zona de resultados (el uso de esa arquitectura está impulsando el valor agregado del negocio).

- Amarillo - La agencia con arquitectura aceptable. También cuyo uso y resultados son aceptables.
- Rojo - La agencia que no tiene una arquitectura completa y / o no usa eficazmente la arquitectura.

2.1.1.3 El Marco de Arquitectura Empresarial de Oracle (OEAF)

Es un esfuerzo de Oracle por proporcionar un marco eficaz de dirección del negocio para ayudar a sus clientes a alinear sus estrategias de negocios con TI. El tema central del Marco de Arquitectura Empresarial de Oracle es ofrecer la suficiente estructura, la cual se crea bajo una estrategia justo a tiempo para satisfacer las necesidades de la organización. Además, el *OEAF* ofrece una estructura arquitectónica muy conocida para compartir el gran capital intelectual de Oracle en torno a las soluciones empresariales de TI con sus clientes y socios, por lo tanto, mejorando aún más la propuesta estratégica de Oracle para agregar valor al negocio, en la Figura 2 se describen los componentes que forman parte del *OEAF*.

2.1.1.3.1 Los Componentes

Arquitectura de Negocios

Cualquier intento arquitectónico debe comenzar con la arquitectura de negocios. La arquitectura de negocios realiza la alineación del modelo del negocio, sus estrategias y objetivos con TI. También crea un caso de negocio para las transformaciones de TI y proporciona una visión de negocio centrada en la empresa desde una perspectiva funcional. Este componente del marco proporciona un seguimiento a las siguientes tres áreas claves del negocio:

- Estrategia del negocio: los requisitos claves de negocio, objetivos, estrategias, indicadores claves de rendimiento, los riesgos del negocio y el modelo operativo de negocio.
- Función del negocio: Los servicios clave del negocio, procesos y capacidades que se verán afectados por el esfuerzo de arquitectura empresarial.

- Organización del negocio: La estructura de la organización, y los roles (Público interno, clientes y socios externos), la toma de decisiones, y la información sobre el presupuesto de la organización.

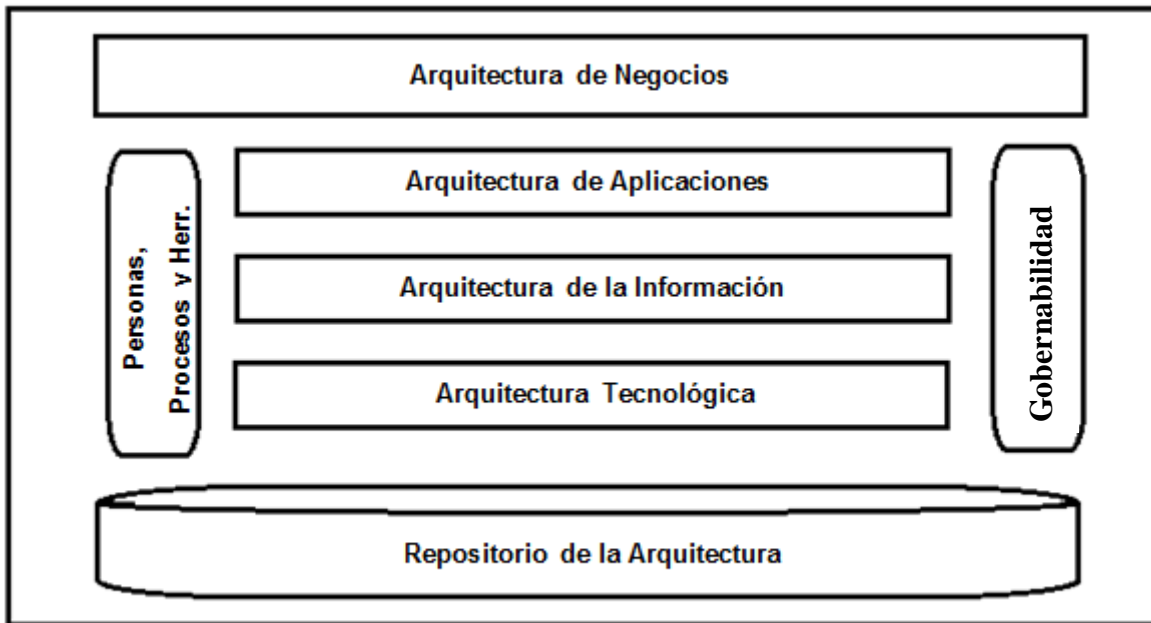


Figura 2: Componentes del Marco de Arquitectura de Oracle
Fuente: Oracle (2012)

Arquitectura de Aplicaciones

La arquitectura de aplicaciones proporciona una vista de la aplicación y los servicios centralizados de una organización, une a las funciones de negocios, a los servicios de los procesos que cubre la aplicación y los servicios de los componentes de las aplicaciones, en búsqueda de la alineación de las aplicaciones con la estrategia. El ámbito de la Arquitectura de la aplicación, la estrategia, y los estándares son una consecuencia de la arquitectura de negocios. La arquitectura de la aplicación se compone de las siguientes categorías:

- Estrategia de aplicaciones: Los principios clave de la arquitectura de aplicaciones (construir versus comprar, hosteado versus en sitio, los estándares abiertos versus propietario, etc), la gobernanza y la gestión del

portafolio de aplicaciones, y un conjunto de arquitecturas de aplicaciones relevantes para el cliente.

- Servicios de las aplicaciones: un inventario de los servicios expuestos e internos de las aplicaciones clave y audiencias externas que apoyan a los servicios de negocio.
- Procesos de la Aplicación: Una serie de procesos específicos de la aplicación que soportan los procesos de negocio en la arquitectura del negocio
- Componentes lógicos: un inventario de productos independientes del sistema que sean relevantes para los objetivos de negocio establecidos.
- Componentes físicos: Los productos reales que soportan los componentes lógicos de la aplicación y su relación con los componentes y servicios relevantes en la Arquitectura de la tecnología y la información.

Arquitectura de la Información

La Arquitectura de la Información describe todas las piezas en movimiento, las partes para la gestión de información de la empresa, y el intercambio de esa información con las personas adecuadas en el momento adecuado para alcanzar los objetivos de negocio establecidos en la arquitectura de negocio. Los componentes clave para describir la arquitectura de la información son:

- Estrategia de información: Los principios de la arquitectura de información, gobernanza de la información y los requisitos de cumplimiento, obligatorios en los modelos de datos.
- Activos de Información: Un catálogo de los datos críticos del negocio y sus modelos (por ejemplo, los clientes el perfil, la orden de compra, los datos del producto, la cadena de suministro, etc) además de las relaciones entre los tipos de datos, los servicios y procesos que interactúan con estos datos.

La Arquitectura de la Información ofrece una visión de la información, centrada en los datos de una organización, y en los activos de información clave que se utilizan para apoyar las funciones críticas de negocio.

Arquitectura Tecnológica

La arquitectura de la tecnología describe como se organizan la infraestructura subyacente del negocio, las arquitecturas de aplicaciones y la información. Los componentes básicos son:

- **Estrategia de la Tecnología:** Los principios de la arquitectura de tecnología, gestión de activos de tecnología, las normas de tecnología, patrones y referencia de arquitecturas utilizadas para el desarrollo de soluciones tecnológicas específicas.
- **Servicios de Tecnología:** un inventario de los servicios de tecnología específicos y sus relaciones con los servicios de negocio, los servicios de aplicaciones, los activos de información y los componentes lógicos o físicos que realizan esos servicios.
- **Componentes lógicos:** Los componentes tecnológicos a nivel de infraestructura para apoyar a cada servicio de la tecnología.
- **Componentes físicos:** El conjunto de productos de tecnología que existen detrás de cada uno de los componentes lógicos de tecnología para implementar el servicio de la tecnología.

La Arquitectura de Tecnología ofrece un modelo de referencia técnica, como se documenta en Marco de Software Empresarial de Oracle (*ESF*), que se utiliza para alinear la adquisición de tecnología, la infraestructura, y la implementación de soluciones de la empresa con las estrategias de TI, la arquitectura de principios, las normas, las arquitecturas de referencia y el modelo de gobernanza.

Personas, Procesos y Herramientas

Esta área del marco identifica a las personas, procesos y herramientas que se utilizan para definir la arquitectura del negocio y las soluciones de arquitectura.

- **Personas:** se refiere a los equipos y a las personas que se encuentren con responsabilidades en el desarrollo de la arquitectura empresarial ya sea en varias perspectivas de arquitectura, mantenimiento, aplicación, y/o en la gobernanza.

- **Procesos:** se refiere a un conjunto de procesos arquitectónicos que se adaptan al acuerdo de servicios de la arquitectura a través de un camino que maximice la probabilidad de éxito de las aplicaciones y reduzca los gastos de recursos.
- **Herramientas:** se refiere a conjunto de herramientas y tecnologías que aceleran el proceso de desarrollo y gestión de la arquitectura empresarial. La mayoría de estos recursos se clasifican en la categoría de modelos (por ejemplo, *ARIS IT Architect de Oracle BPA Suite*), la gestión del portafolio (por ejemplo, *Oracle Primavera*), y los repositorios de los activos de la arquitectura (por ejemplo, *Oracle Enterprise Repository*).

Gobernabilidad de la Arquitectura Empresarial

La Gobernabilidad de la Arquitectura Empresarial proporciona la estructura y los procesos para la implementación de los objetivos y de la estrategia de negocios de la organización a través de una Arquitectura Empresarial. Un ente para la gobernabilidad de la Arquitectura Empresarial se utiliza para guiar a cada proyecto y asegurar la alineación de dicha arquitectura con las transformaciones de TI y la implementación de soluciones. El éxito de la gobernabilidad de la Arquitectura Empresarial incluye:

- **Personas:** Los equipos, los individuos, las funciones y responsabilidades de la junta de gobierno
- **Procesos y Políticas:** la gestión del ciclo de vida de la Arquitectura, la gestión del cambio, los ciclos de revisión, etc
- **Tecnología:** las infraestructura para la aplicación de los procesos y políticas de la gobernanza de la Arquitectura Empresarial
- **Finanzas:** Asignación de costos de TI, modelos de financiación de proyectos, herramientas de negocios para supervisar de forma continua la rentabilidad de la inversión, etc.

Repositorio de la Arquitectura Empresarial

Es un repositorio interno de Oracle que contiene todos los artefactos de la arquitectura y los entregables que se desarrollan a lo largo del ciclo de vida de la Arquitectura Empresarial. El propósito de este repositorio es proporcionar información que describa al estado actual de la arquitectura y una biblioteca de arquitecturas de referencia, los modelos y los principios que describen el objetivo deseado de la arquitectura, teniendo en cuenta los objetivos del negocio.

2.1.1.3.2 El Proceso de Desarrollo de la Arquitectura de Oracle

Para crear los componentes de la arquitectura empresarial, Oracle ha creado un proceso para facilitar su desarrollo. El Proceso de Desarrollo de la Arquitectura de Oracle (*OADP*) que define un enfoque práctico para trabajar en colaboración con los clientes y alinear sus arquitecturas de empresa, y soluciones para sus estrategias de negocio y objetivos como se muestra en la Figura 3.

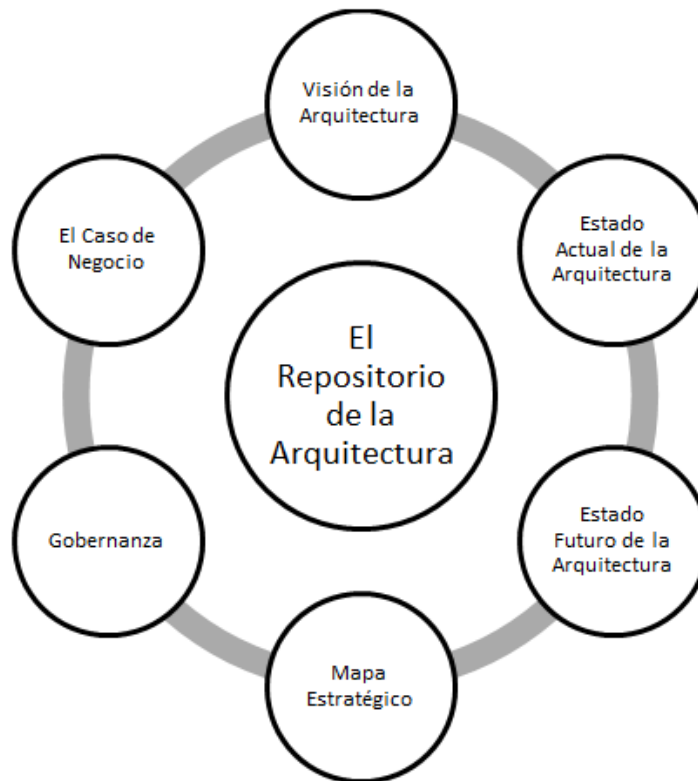


Figura 3: Proceso de Desarrollo de la Arquitectura de Oracle OADP
Fuente: Oracle (2012)

El *OADP* establece un proceso genérico, el a su vez es la base para el desarrollo de la arquitectura como parte de *OEAF* y contiene los siguientes componentes:

- Seis fases de alto nivel:
 - La Visión de la Arquitectura
 - El Estado Actual de la Arquitectura
 - El Estado Futuro de la arquitectura
 - El Mapa Estratégico
 - La gobernanza de Arquitectura Empresarial
 - El Caso de Negocio

El enfoque de Oracle permite que muchas de estas fases se ejecuten al mismo tiempo para reducir el tiempo asociado con la creación de arquitecturas de diversos ámbitos. Además, *OADP* es un proceso muy iterativo, ya que las arquitecturas son desarrolladas y refinadas.

- Las tareas: Son realizadas en cada fase y para la realización de las tareas de una manera práctica y eficiente Oracle aprovecha su repositorio (de artefactos reutilizables de la arquitectura).
- Los entregables de cada fase: Por lo general un documento de PowerPoint que resume los resultados de cada tarea y las referencias de los artefactos producidos en cada fase.
- Los artefactos creados en cada fase: los modelos y diagramas, un enfoque de simplificación en la documentación, que proporciona suficiente detalle sin necesidad de gastos excesivos asociadas con la documentación.

2.1.1.3.3 Valor y Beneficios

El Marco de Aplicación Empresarial de Oracle proporciona un valor significativo a la organización, incluyendo:

- La alineación continua del negocio y de TI
- El aumento del retorno de la inversión (*ROI*) a través de una mejor ejecución de la estrategia empresarial, con eficiente rehusó de los recursos de TI
- El uso de la tecnología para crear nuevas estrategias de negocios

- El uso de un marco único para combinar los productos, servicios y arquitecturas
- Un enfoque práctico que permite a los arquitectos centrarse en la arquitectura y no estancarse en los procesos y los artefactos, o la creación de sus propios procesos.
- La naturaleza ágil del *OEAF* permite la mejora continua para adaptarse a los cambios de las condiciones de negocio y nuevas tecnologías.
- El *OEAF* utiliza conceptos de la industria de la Arquitectura Empresarial, y su terminología y aprovecha lo mejor de otros marcos.
- Un acceso a un conjunto de mejores prácticas, juegos de herramientas, arquitecturas de referencia, y la arquitectura a la medida de los procesos en torno a problemas específicos (racionalización de las aplicaciones, optimización de TI, etc.), reduciendo significativamente el tiempo necesario para desarrollar arquitecturas de empresa.

2.1.1.4 El Marco de Arquitectura de The Open Group (*TOGAF*)

El Marco de Arquitectura de The Open Group, traducido del inglés *The Open Group Architecture Framework (TOGAF)*, es un marco de trabajo para el desarrollo de una arquitectura empresarial.

TOGAF proporciona los métodos y herramientas de apoyo para la aceptación, uso y actualización de una arquitectura empresarial. Está basado en un modelo de proceso iterativo de las mejores prácticas y un conjunto de componentes reutilizables de la arquitectura existente (The Open Group, 2009).

Un esquema o marco de trabajo es un método detallado y un conjunto de herramientas de apoyo.

El término empresa en el contexto de la arquitectura empresarial puede ser usado para referirse tanto a toda la empresa - que abarca la totalidad de sus servicios de información y la tecnología, los procesos y la infraestructura - y de un dominio específico dentro de la empresa (The Open Group, 2009).

En *TOGAF*, la arquitectura tiene dos significados dependiendo del contexto:

1. Una descripción formal de un sistema o un plan detallado del sistema a nivel de componentes para guiar su implementación.
2. La estructura de los componentes y las relaciones entre dichos componentes.

El Marco de Arquitectura es una estructura fundamental o un conjunto de estructuras, que pueden ser utilizadas para el desarrollo de una amplia gama de arquitecturas diferentes. Se debe describir un método para el diseño de un estado final de la empresa en términos de un conjunto de bloques de construcción, y para mostrar cómo los bloques de construcción encajan entre sí. Debe contener un conjunto de herramientas y proporcionar un vocabulario común. También debe incluir una lista de normas recomendadas y los productos compatibles que se pueden utilizar para implementar los bloques de construcción (The Open Group, 2009).

2.1.1.4.1 Los dominios arquitectónicos de *TOGAF*

Hay cuatro niveles arquitectónicos que son comúnmente aceptados como partes de una arquitectura empresarial en general, *TOGAF* está diseñado para soportar los cuatro dominios que se mencionan a continuación:

1. La arquitectura de negocios, que define la estrategia, gestión, organización y procesos claves del negocio.
2. La arquitectura de datos, que describe la estructura de los datos físicos y lógicos de una organización y los recursos de gestión de los mismos.
3. La arquitectura de aplicación, que proporciona un modelo para cada sistema de aplicación a ser desplegado, así como sus interacciones y sus relaciones con los procesos de negocio claves de la organización.
4. La arquitectura tecnológica, que describe las capacidades lógicas de software y hardware que se requieren para apoyar el despliegue de negocio, datos y servicios de aplicación (The Open Group, 2009).

2.1.1.4.2 La clasificación de los resultados

TOGAF utiliza tres categorías para describir el tipo de resultado producido según sea el contexto:

1. Un entregable, es un tipo de resultado que se especifica a través de una cláusula y a su vez es revisado formalmente, aprobado, y firmado por las partes involucradas.
2. Un artefacto, es un tipo de resultado más detallado que describe una arquitectura desde un punto de vista específico.
3. Un bloque de construcción, representa un componente del negocio, tecnologías de la información o la capacidad de la arquitectura para combinarse con otros bloques de construcción. Los bloques de construcción se pueden relacionar a las “arquitecturas” o a las “soluciones”: Los bloques de construcción de la arquitectura (*ABB*) suelen describir una capacidad requerida y dar forma a la especificación de los bloques de construcción de la solución (*SBB*). Los bloques de construcción de la solución (*SBB*) representan los componentes que se utilizarán para implementar una capacidad requerida.

La relación entre entregables, artefactos y bloques de construcción puede ser observada en la Figura 4.

2.1.1.4.3 El Continuum Empresarial

TOGAF incluye el concepto de Continuum Empresarial, que establece el contexto más amplio para un arquitecto y explica cómo se pueden aprovechar y especializar las soluciones genéricas con el fin de atender las necesidades individuales de una organización (The Open Group, 2009). El Continuum Empresarial puede ser interpretado como un repositorio virtual de todos los artefactos arquitectónicos disponibles en una organización. Incluye modelos arquitectónicos, patrones de arquitectura, descripciones arquitectónica, entre otros. Estos artefactos pueden existir específicamente al interior de la empresa, o en general en la industria de Tecnologías de Información. El Continuum Empresarial consiste tanto del Continuum Arquitectónico como del Continuum de Soluciones.

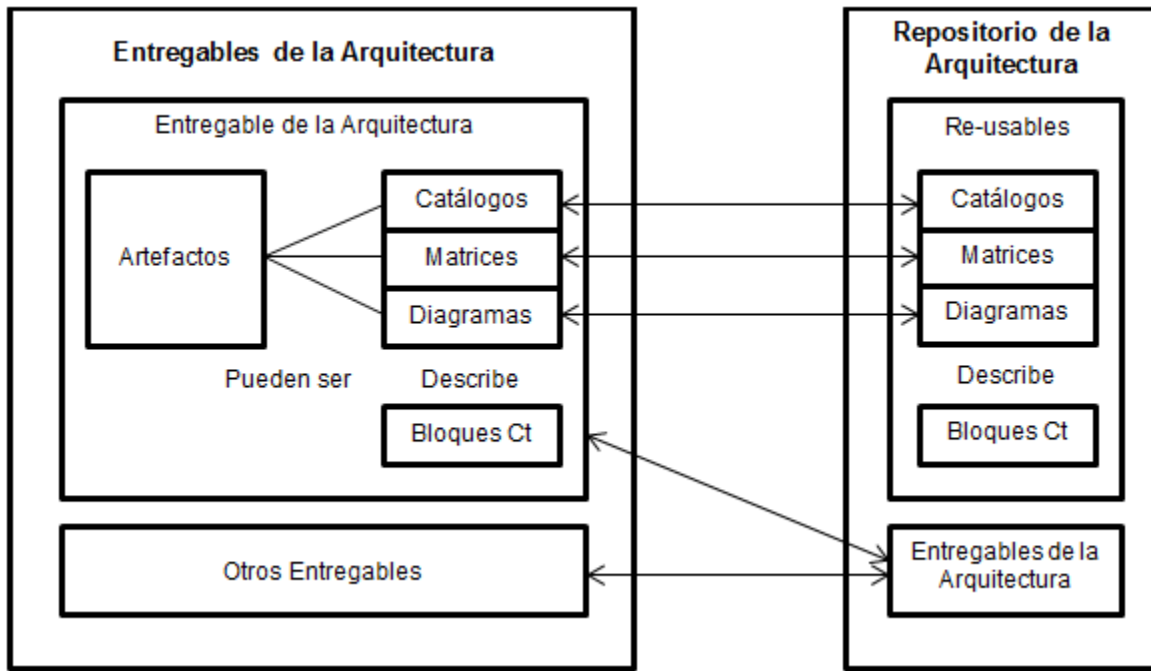


Figura 4: Categorías TOGAF.
Fuente: The Open Group (2009).

El Continuum Arquitectónico especifica la estructura de los artefactos arquitectónicos reutilizables, incluyendo reglas, representaciones y relaciones de los sistemas de información disponibles en la organización.

El Continuum de Soluciones describe la implementación del Continuum Arquitectónico mediante la definición de bloques de construcción de una solución. Una vista de la estructura y el contexto del Continuum Empresarial se muestran en la figura 5.

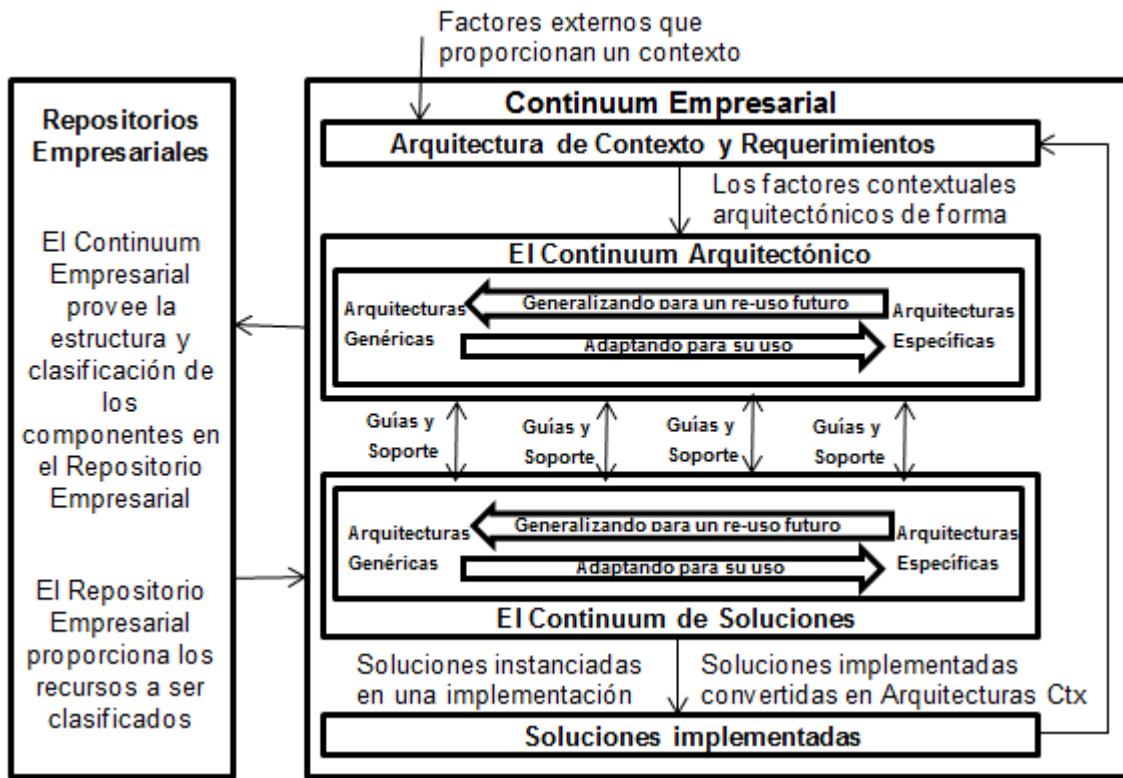


Figura 5: Continuum Empresarial.

Fuente: The Open Group (2009).

2.1.1.4.4 El Repositorio de la Arquitectura

Sustentando al Continuum Empresarial se encuentra el concepto del Repositorio de la Arquitectura, que se puede utilizar para almacenar las diferentes clases de producción arquitectónica en diferentes niveles de abstracción creados por el ADM (The Open Group, 2009).

La estructura del repositorio de la arquitectura de TOGAF se muestra en la figura 6.

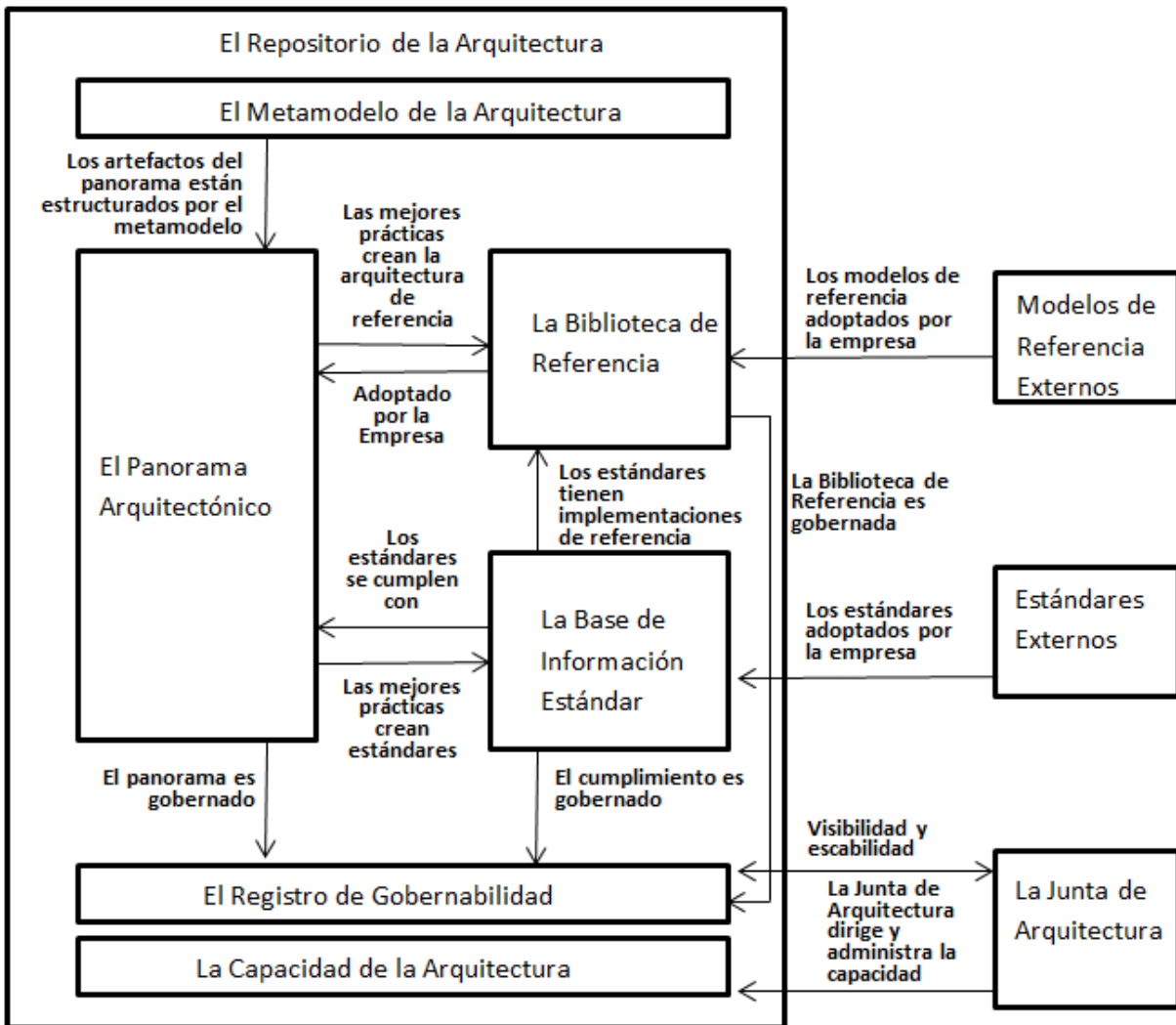


Figura 6: Repositorio de la Arquitectura.
Fuente: The Open Group (2009).

Los componentes principales dentro del Repositorio de la Arquitectura son los siguientes:

- El Metamodelo de la Arquitectura, que describe la aplicación adaptada a un esquema de arquitectura, incluyendo un metamodelo para el contenido de la arquitectura.
- La Capacidad de la Arquitectura, que define los parámetros, estructuras y procesos que apoyan la gestión del Repositorio de la Arquitectura.

- El panorama arquitectónico, que muestra una vista arquitectónica de los bloques de construcción que están en uso dentro de la organización hoy en día.
- La Base de Información Estándar (SIB), que captura las normas que deben cumplir las nuevas arquitecturas, que pueden incluir normas de la industria, productos y servicios seleccionados de proveedores, o servicios compartidos ya implementados dentro de la organización.
- La Biblioteca de Referencia, que ofrece directrices, plantillas, patrones, y otras formas de material de referencia que puede ser utilizado con el fin de acelerar la creación de nuevas arquitecturas para la empresa.
- El Registro de Gobernabilidad, que proporciona un registro de la actividad de la gobernabilidad en toda la empresa (The Open Group, 2009).

2.1.1.5 Comparativa entre Marcos de Arquitecturas Empresariales

Hace veinte años nació el campo de la Arquitectura empresarial, desarrollado inicialmente para hacer frente a dos problemas:

- Los sistema de gran complejidad: las empresas gastaban cada vez más dinero en la construcción de sistemas de TI
- La deficiente alineación entre el negocio y las TIs: a las empresas les resultaba cada vez más difícil mantener los cada vez más costosos sistemas de TI alineados con las necesidades del negocio.

El resultado final: más gasto en TI, menos valor agregado para el negocio. Estos problemas, que ya se conocían desde hace 20 años hoy han llegado a un punto de crisis. El coste y la complejidad de los sistemas informáticos han aumentado de manera exponencial, mientras que las posibilidades de obtener valor agregado de dichos sistemas han disminuido dramáticamente.

Las grandes empresas ya no pueden permitirse el lujo de hacer caso omiso de estos problemas. Se ha construido muchos marcos de Arquitectura Empresarial en los últimos 20 años. En este punto, más del 90 por ciento del campo utiliza una de estos cuatro marcos de Arquitectura Empresarial (MSDN, 2012):

- El Marco de Arquitectura Empresarial de Zachman.
- La Arquitectura Empresarial Federal (*FEA*).
- El Marco de Arquitectura Empresarial de Oracle (*OEAF*).
- El Marco de Arquitectura Empresarial de The Open Group (*TOGAF*).

Para establecer una comparativa entre los marcos de Arquitectura Empresarial anteriormente nombrados lo enmarcaremos en el contexto de una empresa ficticia que se enfrenta a algunos problemas operativos (MSDN, 2012). Estos problemas incluyen:

- Sistemas informáticos que se han convertido en complejos y cada vez más costosos de mantener.
- Los sistemas de TI que están obstaculizando la capacidad de la organización para responder a las condiciones actuales y futuras de mercado.
- Una cultura de la desconfianza entre las partes empresariales y tecnológicas de la organización (MSDN, 2012).

Primero hay que definir los criterios de evaluación, los 12 criterios que con mayor frecuencia se utilizan para comparar y evaluar un marco de Arquitectura Empresarial son:

1. Alcance de la Taxonomía
2. Exhaustividad
3. Orientación por Modelos de Referencia
4. Orientación Práctica
5. Modelo de Madurez
6. Área de Negocios
7. Gobernabilidad
8. Particionamiento
9. Catálogo Prescriptivo
10. Neutralidad de Proveedores
11. Disponibilidad de la Información
12. Curva de valor (MSDN, 2012).

La escala de clasificación de cada marco en cada criterio se asignará de la siguiente manera:

- 1: Tiene un trabajo muy deficiente en esta área
- 2: Tiene un mal trabajo en esta área
- 3: Tiene un trabajo aceptable en esta área
- 4: Tiene un buen trabajo en esta área (MSDN, 2012).

Alcance de la Taxonomía: se refiere a qué tan bien se puede utilizar la metodología para clasificar los distintos artefactos arquitectónicos. Valoraciones:

- Zachman: 4
- TOGAF: 2
- FEA: 2
- OEAF: 1

Exhaustividad se refiere a la guía paso a paso a través del proceso de creación de una Arquitectura Empresarial. Valoraciones:

- Zachman: 1
- TOGAF: 4
- FEA: 2
- OEAF: 3

Orientación por Modelos de Referencia: se refiere a la construcción de un conjunto relevante de los modelos de referencia. Valoraciones:

- Zachman: 1
- TOGAF: 3
- FEA: 4
- OEAF: 1

Orientación Práctica: se refiere a la cantidad de ayuda para que la organización asimile la arquitectura empresarial. Valoraciones:

- Zachman: 1
- TOGAF: 2
- FEA: 2
- OEAF: 4

Modelo de Madurez: se refiere a la cantidad de la orientación de madurez organizacional que se provee a la empresa. Valoraciones:

- Zachman: 1
- TOGAF: 1
- FEA: 3
- OEAF: 2

Área de Negocios: se refiere al uso de la tecnología para impulsar el valor del negocio. Valoraciones:

- Zachman: 1
- TOGAF: 2
- FEA: 1
- OEAF: 4

Gobernabilidad: se refiere a la cantidad de ayuda que provee la metodología para la creación de un modelo de gobernabilidad eficaz para la arquitectura empresarial.

Valoraciones:

- Zachman: 1
- TOGAF: 2
- FEA: 3
- OEAF: 3

Particionamiento: se refiere a la capacidad de la metodología para lidiar con la complejidad a través de particiones autónomas de la empresa. Valoraciones:

- Zachman: 1
- TOGAF: 2
- FEA: 4
- OEAF: 3

Catálogo Prescriptivo: se refiere a qué tan bien la metodología guía en la creación de un catálogo de bienes arquitectónicos que se puedan reutilizar en futuras actividades. Valoraciones

- Zachman: 1
- TOGAF: 2
- FEA: 4

- OEAF: 2

Neutralidad de Proveedores: se refiere a la probabilidad de independencia de productos tecnológicos mediante la adopción de esta metodología. Valoraciones:

- Zachman: 2
- TOGAF: 4
- FEA: 3
- OEAF: 1

Disponibilidad de la Información: se refiere a la cantidad y calidad de información gratuita o de bajo costo disponible de esta metodología. Valoraciones:

- Zachman: 2
- TOGAF: 4
- FEA: 2
- OEAF: 1

Curva de valor: se refiere a la cantidad de tiempo que se va a utilizar esta metodología antes de empezar a crear soluciones que proporcionen un alto valor agregado para la empresa, un valor alto implica un tiempo bajo. Valoraciones:

- Zachman: 1
- TOGAF: 3
- FEA: 1
- OEAF: 4

Los criterios y las puntuaciones se resumen en la tabla 2 (MSDN, 2012).

Tabla 2: Comparativa entre Marcos Arquitecturas Empresariales

Fuente: MSDN (2012)

Crterios	Zachman	TOGAF	FEA	OEAF
Alcance de la Taxonomía	4	2	2	1
Exhaustividad	1	4	2	3
Orientación por Modelos de Referencia	1	3	4	1
Orientación Práctica	1	2	2	4
Modelo de Madurez	1	1	3	2
Área de Negocios	1	2	1	4
Gobernabilidad	1	2	3	3
Particionamiento	1	2	4	3
Catálogo Prescriptivo	1	2	4	2
Neutralidad de Proveedores	2	4	3	1
Disponibilidad de la Información	2	4	2	1
Curva de valor	1	3	1	4

Escala de Valoración:

1. Tiene un trabajo muy deficiente en esta área
2. Tiene un mal trabajo en esta área
3. Tiene un trabajo aceptable en esta área
4. Tiene un buen trabajo en esta área

2.1.1.5.1 ¿Por qué *TOGAF*?

TOGAF busca ser una aproximación al desarrollo y gobierno de Arquitecturas Empresariales de manera “Ágil”. No prescribe los modelos que deberían ser usados para representar la arquitectura de la empresa y guía el proceso de creación de la Arquitectura. Debido a su escalabilidad, puede ser usado por organizaciones de gobierno, empresas pequeñas, medianas o grandes. Es muy importante destacar que el marco puede ser modificado por todos sus usuarios, como pasa con un producto Open Source, sin olvidar nunca la retroalimentación y la información obtenida en procesos de la vida real debido a que se ha desarrollado gracias a la colaboración de más de 300 empresas las cuales son miembros del Foro de Arquitectura, algunos de ellos son los principales clientes de tecnología en el mundo de la información y los vendedores y representa las mejores prácticas en el desarrollo de la arquitectura. Es de suma importancia recalcar que el objetivo principal de este Trabajo Especial de Grado es Desarrollar un Sistema de Información para la automatización del Método de Desarrollo de la Arquitectura Empresarial, entonces *TOGAF* provee un método Open Source, que puede ser adaptado, Ágil, escalable y con mucha documentación disponible a través de la red,

además de poseer el proceso de creación de una Arquitectura Empresarial más completo hasta el momento de presentación de este trabajo, el *ADM*.

2.1.2 El Método de Desarrollo de la Arquitectura (*ADM*)

El Método de Desarrollo de la Arquitectura de *TOGAF* traducido del inglés *The TOGAF Architecture Development Method (ADM)* es un método para el desarrollo de arquitecturas que incluye el establecimiento de un marco de trabajo, el desarrollo de contenidos, la transición y la gestión de la implementación de la arquitectura. Todas estas actividades se llevan a cabo dentro de un ciclo iterativo de definición continua de la arquitectura y su implementación, que permite a las organizaciones transformar sus empresas de forma controlada, en respuesta a los objetivos del negocio y las oportunidades (The Open Group, 2009). El flujo del *ADM* puede ser visto en la Figura 7.

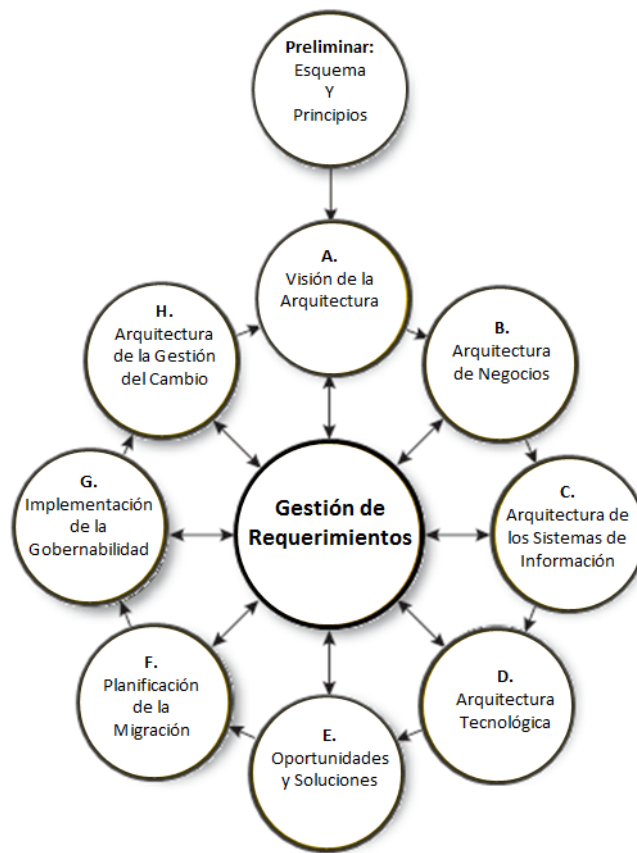


Figura 7: Método de Desarrollo de la Arquitectura.

Fuente: The Open Group (2009).

Por cada iteración del Método de Desarrollo de la Arquitectura se verifica lo siguiente:

- El alcance de la arquitectura a ser definida.
- El nivel de detalle a ser definido.
- La extensión del período de tiempo a ser destinado.

2.1.2.1 Las Fases del Método de Desarrollo de la Arquitectura:

Las Fases del Método de Desarrollo de la Arquitectura son:

La Fase Preliminar Describe las actividades de preparación e iniciación requeridas para el cumplimiento de la directiva de negocio de una arquitectura empresarial nueva, incluyendo la definición de un esquema de arquitectura para una organización específica y la definición de los principios.

La Fase A: La Visión de la Arquitectura Describe la fase inicial del ciclo de desarrollo de la arquitectura incluyendo información sobre la definición del alcance, la identificación de los involucrados, la creación de la visión de la arquitectura, y la obtención de las aprobaciones.

La Fase B: La Arquitectura de Negocios Describe el desarrollo de una arquitectura de negocios para apoyar la visión de la arquitectura previamente acordada.

La Fase C: La Arquitectura de los Sistemas de Información Describe el desarrollo de las arquitecturas de los sistemas de información para un proyecto de arquitectura, incluyendo el desarrollo de los datos y las arquitecturas de aplicaciones.

La Fase D: La Arquitectura Tecnológica Describe el desarrollo de la arquitectura tecnológica para un proyecto de arquitectura.

La Fase E: Oportunidades y Soluciones Lleva a cabo la planificación de la implementación inicial y la identificación de vías de publicación para la arquitectura definida en las fases anteriores.

La Fase F: Planificación de la Migración Se refiere a la formulación de un conjunto de secuencias detalladas de Arquitecturas de transición que apoyan al Plan de Migración.

La Fase G: Implementación de la Gobernabilidad Proporciona una supervisión arquitectónica de la aplicación.

La Fase G: Arquitectura de la Gestión del Cambio establece los procedimientos para la gestión del cambio a la nueva arquitectura.

La Gestión de Requerimientos examina el proceso de gestión de requerimientos de arquitectura a lo largo del Método de Desarrollo de la Arquitectura.

Las fases del ciclo de Método de Desarrollo de la Arquitectura se dividen en pasos, como puede ser visto en la Figura 8, por ejemplo, los pasos dentro de la Fase Preliminar son los siguientes:

- Identificar el alcance de las Unidades de Negocio impactadas
- Verificar la Gobernabilidad y los Marcos de Trabajo soportados
- Definir el equipo que desarrollará la Arquitectura Empresarial
- Identificar y establecer los Principios de la Arquitectura
- Seleccionar y adaptar los Marcos de Trabajo Arquitectónicos
- Implementar las Herramientas de la Arquitectura

2.1.2.1.1 Fase Preliminar

Describe las actividades de preparación e iniciación requeridas para el cumplimiento de la directiva de negocio de una arquitectura empresarial nueva, incluyendo la definición de un esquema de arquitectura para una organización específica y la definición de los principios. Las actividades claves de esta fase son las siguientes:

- Comprender el entorno empresarial.
- Asegurar un compromiso de gestión de alto nivel.
- Obtener un acuerdo sobre el alcance.
- Establecer los Principios de la Arquitectura.
- Establecer la estructura de la gobernabilidad.

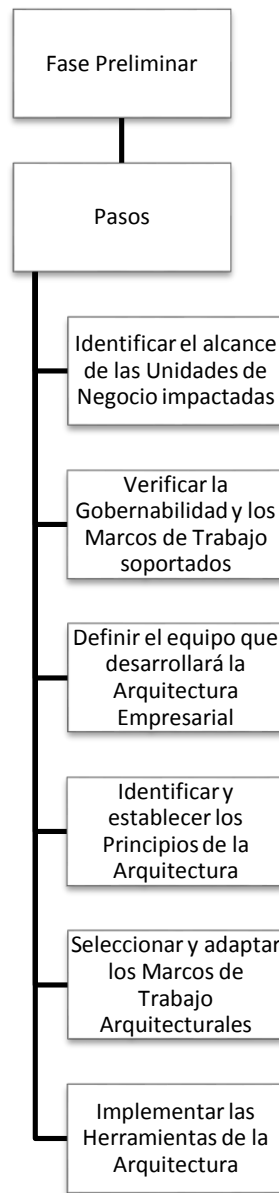


Figura 8: Pasos de la Fase Preliminar.

Fuente: Elaboración propia.

2.1.2.1.1.1 Objetivos de la Fase Preliminar

- Revisar el contexto organizacional a fin de asociarlo al ejercicio de desarrollo de la Arquitectura Empresarial.
- Identificar los usuarios involucrados y a otros usuarios impactados por la decisión de desarrollar la Arquitectura Empresarial, a fin de determinar sus requerimientos y prioridades según los objetivos del Negocio

- Asegurar que todos los usuarios que deben involucrarse en el desarrollo de la Arquitectura Empresarial sean incluidos efectivamente.
- Habilitar a los usuarios patrocinadores del desarrollo de la Arquitectura Empresarial y a las tareas que estos deben promover en las áreas de Negocio afectadas.
- Identificar el alcance de los elementos organizacionales afectados por las directivas del Negocio y definir las premisas y requerimientos.
- Seleccionar e implementar herramientas e infraestructuras tecnológicas que permitan soportar las actividades de desarrollo de la Arquitectura Empresarial.
- Definir los Principios de la Arquitectura que podrían formar parte de los requerimientos del desarrollo de la Arquitectura Empresarial.

2.1.2.1.1.2 Elementos Claves de la Fase Preliminar

Como podemos observar en la Figura 9 entre los elementos Claves de la Fase Preliminar tenemos:

- La Empresa.
- El Contexto Organizacional.
- Los Requerimientos para desarrollar la Arquitectura.
- Los Principios de la Arquitectura.

2.1.2.1.1.2.1 La Empresa

- El principal desafío en el desarrollo de la Arquitectura Empresarial es la definición del alcance de la Arquitectura.
- La Arquitectura empresarial puede considerarse como un elemento del plan estratégico, que comienza a incrementar una parte integral de la gerencia del negocio.
- El alcance determina a los usuarios involucrados, quienes podrían obtener mayores beneficios de una nueva o mejorada Arquitectura Empresarial.

- Es importante asegurar que los patrocinadores, puedan gerenciar e impulsar las actividades, y disponer de los recursos necesarios, para desarrollar efectivamente la Arquitectura Empresarial.



Figura 9: Elementos Claves de la Fase Preliminar.
Fuente: Elaboración propia.

2.1.2.1.1.2.2 El Contexto Organizacional

- Para tomar decisiones eficaces sobre el marco de trabajo utilizado para desarrollar la Arquitectura que se utilizará dentro de la organización, es necesario entender el contexto:
 - Los Modelos comerciales para la Arquitectura Empresarial.
 - Los Planes presupuestarios para la Arquitectura Empresarial.
 - Los Requerimientos y necesidades de los usuarios involucrados.
 - Las Prioridades del Negocio, estrategias, principios y metas.
 - Los Procesos que soportan la ejecución de cambios:

- La Gestión de Proyectos y Gestión del Portafolio.
 - La Gestión de Sistemas.
 - El Análisis y Diseño del Negocio.
 - La Gestión del Portafolio de Información, aplicaciones y tecnología.
- El nivel de formalidad y rigor a ser aplicado
 - Los puntos de interacción con otras organizaciones, procesos, roles y responsabilidades.

2.1.2.1.1.2.3 Los Requerimientos para desarrollar la Arquitectura

- Las necesidades del Negocio, se traducen en requerimientos de la Arquitectura Empresarial a desarrollar, y de las métricas de rendimiento, sobre el desempeño de esta.
- Las necesidades del Negocio deben ser definidas de forma suficientemente clara en la Fase Preliminar, de tal forma de poder alcanzar las metas del Negocio, cumplir con los requerimientos de recursos y de información y asociarse a la estrategia de desarrollo de la Arquitectura Empresarial.

2.1.2.1.1.2.4 Los Principios de la Arquitectura

- La definición de los Principios de la Arquitectura es clave para el desarrollo de una Arquitectura Empresarial.
- El desarrollo de la Arquitectura Empresarial es influenciado tanto por los Principios del Negocio, como por los Principios de la Arquitectura.
- Los Principios de la Arquitectura normalmente se basan en los principios del Negocio.
- La definición de los Principios del Negocio usualmente está fuera del alcance de las funciones de la arquitectura.
- El conjunto de Principios de la Arquitectura puede referirse a los principios del Negocio, los objetivos y las estrategias del Negocio se definen en otra parte dentro de la organización.

- La aplicación de la gobernabilidad de la Arquitectura está relacionada con los Principios de la Arquitectura.
- Los responsables de la gobernabilidad usualmente podrían ser responsables de aprobar los Principios de la Arquitectura y las mejoras de la Arquitectura Empresarial.

2.1.2.1.1.3 Pasos de la Fase Preliminar

Como puede ser visto en la Figura 8, los pasos dentro de la Fase Preliminar son los siguientes:

- Identificar el alcance de las Unidades de Negocio impactadas
- Verificar la Gobernabilidad y los Marcos de Trabajo soportados
- Definir el equipo que desarrollará la Arquitectura Empresarial
- Identificar y establecer los Principios de la Arquitectura
- Seleccionar y adaptar los Marcos de Trabajo Arquitectónicos
- Implementar las Herramientas de la Arquitectura

2.1.2.1.1.3.1 Identificar el alcance de las Unidades de Negocio impactadas

- Identificar las principales Unidades de Negocio, aquellas que son más afectadas y tienen mayor valor en el desarrollo de la Arquitectura Empresarial.
- Identificar las Unidades de Negocio secundarias, aquellas que percibirán el cambio en sus capacidades por que trabajan con las Unidades de Negocio principales, pero no serán afectadas directamente por el desarrollo de la Arquitectura Empresarial.
- Identificar las Unidades de Negocio extendidas, aquellas Unidades de Negocio que están fuera del alcance de la empresa, y que pueden ser afectadas en su propia Arquitectura Empresarial.
- Identificar las Comunidades involucradas, aquellos usuarios involucrados quienes pueden ser afectados y que se estructuran en grupos de comunidades.

- Identificar la Gobernabilidad requerida, incluyendo Marcos de Trabajo regulatorios, legales y geográficos.

2.1.2.1.1.3.2 Verificar la Gobernabilidad y los Marcos de Trabajo soportados

- Entender como los artefactos arquitectónicos pueden ser gobernados y administrados.
- Revisar la gobernabilidad existente, los modelos que soportan la organización y como estos podrían requerir de cambios en base a la adopción de un nuevo Marco de Trabajo arquitectónico.
- Determinar, entender y discutir los puntos de mejora de la Arquitectura y los impactos probables.

2.1.2.1.1.3.3 Definir el equipo que desarrollará la Arquitectura Empresarial

- Determinar las capacidades Arquitectónicas de la Empresa y el Negocio.
- Realizar de ser necesario, estudio del Nivel de Madurez Arquitectónica del Negocio.
- Identificar brechas entre las áreas de trabajo existentes.
- Asignar los roles y responsabilidades principales de la Gestión y Gobierno de la Arquitectura Empresarial.
- Definir los requerimientos para los cambios en los proyectos y programas de negocio existentes.
- Definir el alcance del desarrollo de la nueva Arquitectura Empresarial.
- Definir las premisas y requerimientos sobre el desarrollo de la nueva Arquitectura Empresarial.
- Definir y comprometer a los patrocinadores y las mesas de trabajo para el desarrollo de la nueva Arquitectura Empresarial.
- Determinar los requerimientos de presupuesto.

2.1.2.1.1.3.4 Identificar y establecer los Principios de la Arquitectura

- Definir un conjunto de principios arquitectónicos apropiados para la organización:

- Principios de Negocio
- Principios sobre los Datos
- Principios sobre las Aplicaciones
- Principios sobre la Tecnología

2.1.2.1.1.3.5 Seleccionar y adaptar los Marcos de Trabajo Arquitectónicos

- Determinar lo que la adaptación necesita.
- La adaptación debe producir un conjunto de terminologías acordadas para la descripción de contenido arquitectónico.
- Adaptar los procesos:
 - Retirar las tareas que ya están realizadas en otras partes de la organización.
 - Añadir a la organización tareas específicas, tales como puntos de control específicos.
 - Alinear los procesos para entornos de procesos externos y puntos de contacto
- Adaptar la estructura del contenido y la clasificación para permitir la adopción del contenido de marcos de terceros y permitir la personalización del marco para brindar soporte a los requerimientos específicos de la organización.

2.1.2.1.1.3.6 Implementar las Herramientas de la Arquitectura

- Las herramientas de la arquitectura pueden basarse en estándares de las aplicaciones de productividad de oficina, o pueden estar basadas en una implementación personalizada de herramientas especializadas de arquitectura.
- Dependiendo del nivel de sofisticación, la implementación de herramientas puede variar de una tarea trivial a una actividad de implementación de soluciones más complejas.

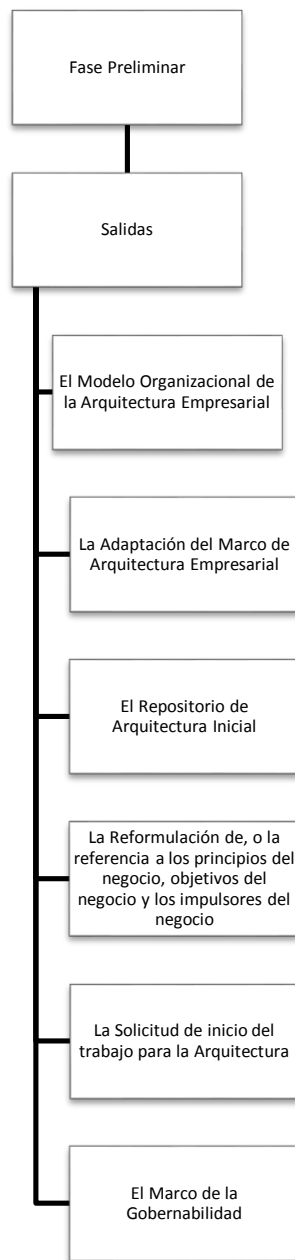


Figura 10: Salidas de la Fase Preliminar.

Fuente: Elaboración propia.

2.1.2.1.1.4 Salidas de la Fase Preliminar

En la Figura 10 se encuentran descritas las salidas de la Fase Preliminar entre las cuales se encuentran las siguientes:

- El Modelo Organizacional de la Arquitectura Empresarial.

- La Adaptación del Marco de Arquitectura Empresarial.
- El Repositorio de Arquitectura Inicial.
- La Reformulación de, o la referencia a los principios del negocio, objetivos del negocio y los impulsores del negocio.
- La Solicitud de inicio del trabajo para la Arquitectura.
- El Marco de la Gobernabilidad.

2.1.3 Los Principios de la Arquitectura

Los Principios de la Arquitectura son un conjunto de normas y directrices generales para la arquitectura se será desarrollada.

Se pretende que sea duradera y que sea modificada pocas veces, muestra y da soporte a la forma en que una organización cumple su misión. A menudo son un elemento que forma parte de un conjunto estructurado de ideas que unidas definen y guían a la organización, desde los valores hasta las acciones y resultados.

Los principios forman parte de la salida de la Fase Preliminar y se utilizan en todo el Método de Desarrollo de la Arquitectura para proporcionar un marco de referencia orientado a la toma de decisiones dentro de la empresa.

Dependiendo de la organización, los principios pueden establecerse en cualquiera de los siguientes tres niveles:

- Los Principios Empresariales proporcionan una base para la toma de decisiones y dictan cómo cumple la organización con su misión. Tales principios se encuentran comúnmente en las organizaciones gubernamentales y sin fines de lucro, pero también se encuentran en las organizaciones comerciales, como medio de armonizar la toma de decisiones. Son elementos clave en una estrategia de Gobernabilidad de la Arquitectura exitosa.
- Los Principios de las Tecnologías de la Información proporcionan orientación sobre el uso y despliegue de todos los recursos de TI y de los activos de la empresa. Están desarrollados para hacer que el entorno de la información sea lo más productivo y rentable posible.

- Los Principios de la Arquitectura son un subconjunto de principios de TI que se relacionan con la arquitectura. Son el reflejo de un consenso en toda la empresa, y encarnan el espíritu de la arquitectura empresarial. Los Principios de la Arquitectura pueden ser divididos en:
 - Los principios que gobiernan el proceso de la arquitectura, los que afectan el desarrollo, mantenimiento y uso de la arquitectura empresarial.
 - Los principios que rigen la implementación de la arquitectura.

2.1.3.1 La plantilla de *TOGAF* para la definición de los Principios de la Arquitectura

TOGAF define una manera estándar de describir los principios, como se muestra en la Tabla 3. Además de una definición, cada principio debe tener asociado fundamentos e implicaciones, tanto para promover la comprensión y la aceptación de los propios principios, como para apoyar el uso de los mismos y justificar el por qué se toman algunas decisiones específicas.

Tabla 3: La plantilla de *TOGAF* para la definición de los Principios de la Arquitectura

Fuente: The Open Group (2009).

Sección	Descripción
Nombre	Debe representar la esencia de la regla y ser fácil de recordar. Las plataformas tecnológicas específicas no deben ser mencionadas en el nombre o en la definición de un principio. Evite las palabras ambiguas en el nombre y en la definición, no busque adjetivos y adverbios innecesarios.
Definición	Debe ser clara al comunicar la regla fundamental. En su mayor parte, las definiciones de los principios para gestionar la información son similares entre las organizaciones. Es de vital importancia que la definición no sea ambigua.
Fundamento	Debe destacar los beneficios empresariales de la adhesión al principio, utilizando la terminología de negocios. Señala la similitud entre los principios de las tecnologías de la información y los principios que gobiernan las operaciones del negocio. También describe la relación con otros principios, y las intenciones en relación con una interpretación equilibrada. Describe las situaciones en las que un principio dado tiene prioridad sobre otro para tomar una decisión.
Implicaciones	Deben destacar las necesidades, tanto para el negocio como para las tecnologías de la Información, para la realización del principio - en términos de recursos, costos y actividades / tareas. A menudo será evidente que los actuales sistemas, normas o prácticas sean incongruentes con el principio que se está adoptando. El impacto en el negocio y las consecuencias de la adopción de un principio debe quedar claramente establecidas. Se debe discernir fácilmente la respuesta a: "¿Cómo me afecta esto?"

2.1.3.2 Ejemplo de los Principios de la Arquitectura

En la Tabla 4 se muestra un ejemplo de un principio de una arquitectura.

Tabla 4: Ejemplo de un principio de arquitectura

Fuente: The Open Group (2009).

Auto-servicio	
Definición	Los clientes deben ser capaces de servirse a sí mismos.
Fundamento	La aplicación de este principio va a mejorar la satisfacción del cliente, reducir los gastos generales de administración y, mejorar potencialmente los ingresos.
Implicaciones	Hay una implicación para mejorar la facilidad de uso y reducir al mínimo las necesidades de capacitación, por ejemplo, los miembros deben ser capaces de actualizar sus datos de contacto, etc y deben ser capaces de comprar productos adicionales de membresía en línea.

2.1.3.2.1 Los Principios del Negocio

- Los principios de gestión de la información, son compartidos por todas las unidades de negocio.
- Las decisiones de gestión de la información se hacen para proporcionar el máximo beneficio a la organización en su conjunto.
- Todas las unidades de negocio de la organización participan en las decisiones de gestión de la información necesaria para lograr los objetivos de negocio.
- Las operaciones de empresa se mantienen a pesar de las interrupciones del sistema.
- El desarrollo de aplicaciones que se utilizan en toda la organización se prefiere sobre el desarrollo de aplicaciones similares o duplicadas que solamente se faciliten a una unidad de negocio.
- La arquitectura se basa en un diseño de servicios que reflejan las actividades de negocio del mundo real que comprenden la organización de los procesos de negocio.
- Los procesos de gestión de la información empresarial cumplen con todas las leyes, políticas y reglamentos.
- La función de las tecnologías de la información es responsable de la propiedad y la aplicación de los procesos de TI y la infraestructura que

permiten soluciones para cumplir con los requisitos definidos por el usuario para la funcionalidad, niveles de servicio, costo y tiempos de entrega.

- La Propiedad Intelectual de la organización (IP) debe ser protegida y esta protección debe reflejarse en los procesos de arquitectura de TI, su implementación y los procesos de gobernabilidad.

2.1.3.2.2 Los Principios de los Datos

- Los datos son un activo que tiene valor para la organización y son administrados en consecuencia.
- Los usuarios tienen acceso a los datos necesarios para llevar a cabo sus funciones y por lo tanto, los datos se comparten a través de las funciones de la organización y de las unidades de negocio.
- Los datos son accesibles para los usuarios al realizar sus funciones.
- Los datos tienen un administrador responsable de la calidad de los mismos.
- Los datos se define coherente en toda la organización y las definiciones son comprensibles y están disponibles para todos los usuarios.
- Los datos están protegidos contra uso y la divulgación no autorizada.

2.1.3.2.3 Los Principios de las Aplicaciones

- Las aplicaciones son independientes de las opciones tecnológicas específicas y por lo tanto puede operar en una gran variedad de plataformas tecnológicas.
- Las aplicaciones son fáciles de usar y la tecnología subyacente es transparente para los usuarios, para que puedan concentrarse en sus tareas.

2.1.3.2.4 Los Principios Tecnológicos

- Los cambios en las aplicaciones y en la tecnología se hacen sólo en respuesta a las necesidades de la empresa.
- Los Cambios en el entorno de la información de la empresa se llevan a cabo de manera oportuna.

- La diversidad tecnológica es controlada para minimizar el costo de mantenimiento y experiencia en la conectividad entre múltiples entornos de procesamiento.
- El software y hardware deben ajustarse a normas definidas que promuevan la interoperabilidad de los datos, las aplicaciones y la tecnología.

2.1.4 Niveles de Madurez de Arquitectura Empresarial

A continuación se muestran las características detalladas de los Niveles de Madurez de Arquitectura Empresarial que se aplican en los procesos de Arquitectura Empresarial:

2.1.4.1 Nivel 0: ninguno

No hay ningún programa de Arquitectura Empresarial.

2.1.4.2 Nivel 1: inicial

Hay un proceso de Arquitectura Empresarial en marcha:

1. Los procesos son ad hoc y localizados. Se definen algunos procesos de Arquitectura Empresarial. No existe un proceso arquitectura unificada a través de tecnologías o procesos de negocio. El éxito depende de los esfuerzos individuales.
2. Los procesos de la Arquitectura Empresarial, documentación y normas son establecidos por una variedad de medios y son informales.
3. Hay un conocimiento limitado del equipo de gestión en el proceso de la Arquitectura.
4. Hay poca aceptación del funcionamiento del proceso de la Arquitectura Empresarial.
5. La última versión de la documentación de la Arquitectura Empresarial está en la web, pero existe poca comunicación sobre el proceso de arquitectura y las posibles mejoras en los procesos.
6. No hay Gobernabilidad explícita de las normas arquitectónicas.
7. Hay poca participación de la planificación estratégica en el proceso de desarrollo de la arquitectura empresarial.

2.1.4.3 Nivel 2: en desarrollo

El proceso de la Arquitectura Empresarial se encuentra en desarrollo.

1. El proceso básico de la arquitectura empresarial está documentado y se han desarrollado funciones y responsabilidades claras.
2. La visión de las tecnologías de la Información, los principios, los vínculos comerciales, la línea de base y el destino de la arquitectura están identificados. Existen estándares de arquitectura, pero no están necesariamente vinculados a la arquitectura objetivo. Hay un Modelo de Referencia Técnica (TRM) y un marco perfil de las normas establecidas.
3. Hay vinculación explícita con las estrategias de negocio.
4. Hay conciencia de la gestión del esfuerzo de la arquitectura.
5. Las responsabilidades se asignan y se está trabajando.
6. Las páginas web donde está contenida la información de la Arquitectura Empresarial se actualizan periódicamente y se utilizan para documentar los entregables de la arquitectura.
7. La Arquitectura de Seguridad ha definido los roles y las responsabilidades de manera clara.
8. Existe la Gobernabilidad con estándares arquitectónicos y adherencia al perfil de las normas existentes.

2.1.4.4 Nivel 3: definido

La Arquitectura Empresarial está definida, incluyendo los procedimientos escritos detallados y el Modelo de Referencia Técnica (TRM).

1. La arquitectura está bien definida y comunicada al personal de Tecnologías de la Información. El proceso es seguido en gran medida.
2. El análisis de las deficiencias y el plan de migración se han completado. El Modelo de Referencia Técnica y el perfil de las normas están plenamente desarrollados.
3. La Arquitectura Empresarial está integrada con la planificación del capital y el control de las inversiones.
4. El equipo directivo conoce y apoya el proceso de arquitectura de toda la empresa. La gestión apoya activamente las normas arquitectónicas.
5. La mayoría de los elementos de funcionamiento de la unidad muestran aceptación o están participando activamente en el proceso de arquitectura empresarial.
6. Los documentos de arquitectura se actualizan periódicamente en la página web de la Arquitectura Empresarial.

7. La Arquitectura de Seguridad está totalmente desarrollado y se integra con la Arquitectura Empresarial.
8. Existe Gobernabilidad explícita documentada para la mayoría de las inversiones en Tecnologías de la Información.
9. Existe una estrategia de adquisición e incluye medidas de cumplimiento para la Arquitectura Empresarial. Los costos-beneficio son considerados en los proyectos.

2.1.4.5 Nivel 4: gestionado

Se está administrando y midiendo un proceso de Arquitectura Empresarial.

1. El proceso de Arquitectura Empresarial forma parte de la cultura organizacional. Las métricas de calidad asociados con el proceso de la arquitectura son capturadas.
2. La documentación de la Arquitectura Empresarial se actualiza en un ciclo regular para reflejar la arquitectura de la empresa actualizada.
3. La planificación de capital y control de las inversiones se ajustan en base a los comentarios recibidos y las lecciones aprendidas de la Arquitectura Empresarial actualizada. Hay una periódica re-examinación de los objetivos del negocio.
4. El equipo directivo participa directamente en el proceso de revisión de la arquitectura.
5. Todo el equipo de operación acepta y participa activamente en el proceso de la Arquitectura Empresarial.
6. Los documentos de arquitectura se actualizan regularmente.
7. Los indicadores de desempeño asociados a la Arquitectura de Seguridad son capturados.
8. Hay Gobernabilidad explícita de todas las inversiones en Tecnologías de la Información.
9. Todas las adquisiciones y compras planificadas son guiadas y regidas por la Arquitectura Empresarial.

2.1.4.6 Nivel 5: en optimización

Existe una mejora continua de los procesos de la Arquitectura Empresarial.

1. Los esfuerzos están concertados para optimizar y mejorar continuamente el proceso de arquitectura.
2. Las exenciones a las normas y procedimientos se utilizan para mejorar el proceso de desarrollo de la arquitectura.

3. Las métricas de procesos de la Arquitectura se utilizan para optimizar e impulsar los vínculos comerciales. El negocio está involucrado en las mejoras continuas de los procesos de arquitectura empresarial.
4. Hay participación alta de la dirección en la optimización de mejoras de los procesos de desarrollo de la arquitectura y la gobernabilidad.
5. Los comentarios sobre el proceso de la arquitectura de todos los elementos de la unidad de operación se utilizan para impulsar mejoras en los procesos de arquitectura.
6. Los documentos de la Arquitectura son utilizados por cada tomador de decisiones en la organización para cada decisión de negocios relacionados con Tecnologías de la Información.
7. La retroalimentación de la misma métrica arquitectura de seguridad se utiliza para impulsar mejoras en los procesos de arquitectura.
8. Hay gobernabilidad explícita de todas las inversiones en Tecnologías de la Información. Las exenciones a las normas y procedimientos se utilizan para hacer las mejoras de gobierno en el proceso.
9. No hay ninguna inversión o adquisición que no se haya planificado.

2.2 Metodología de Desarrollo de Software

Una metodología de desarrollo de software se define como un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a producir nuevo software. (Piattini, 2003).

Una metodología debe especificar los siguientes aspectos:

- División del proyecto en etapas.
- Tareas a ser realizadas en cada etapa.
- Restricciones a ser aplicadas.
- Técnicas y herramientas a utilizar.
- Control y gestión del proyecto.

Con el paso del tiempo, las metodologías de desarrollo de software han ido evolucionando de la siguiente manera:

- Desarrollo convencional: se utilizó alrededor de los años 50; en ella no existían metodologías ni se realizaba análisis alguno, sólo se limitaban a programar.

Inconvenientes del desarrollo convencional:

- El resultado final era impredecible.
 - No hay forma de control sobre el proyecto.
 - Ausencia de información actualizada.
 - Insatisfacción del usuario.
- Desarrollo estructurado: fue muy utilizado en el entorno académico alrededor de los años 60 y a mediados de los 70 ingresó al entorno industrial, en ella se establecen métodos de ingeniería y de programación estructurada. Surge el diseño y el análisis estructurado, se introduce el concepto de módulos, estructura, relaciones y calidad; lo cual facilita la comprensión de programas y provee mayor nivel de abstracción.
 - Desarrollo Orientado a Objetos: utilizado en los años 80, trata los procesos y los datos en forma conjunta, se habla de principios de abstracción, modularidad y lenguajes orientados a objetos.

2.2.1 Metodologías Estructuradas

Proponen la creación de modelos del sistema que representan: procesos, flujos y la estructura de datos. Emplean el enfoque *Top-Down*, que consiste en formular el sistema sin especificar los detalles.

2.2.1.1 Metodologías Orientadas a Procesos

Se apoyan en técnicas gráficas para obtener una especificación estructurada; emplea el modelo básico de entrada/proceso/salida de un sistema.

Está compuesto por:

- Diagrama de flujo de datos (DFD).
- Diccionario de datos.
- Especificaciones del proceso

2.2.1.2 Metodologías Orientadas a Datos

Son metodologías basadas en la información; primero se definen las estructuras de datos (entradas y salidas) y de ellas se derivan los componentes procedimentales. Ejemplos: Metodologías de Jackson, Warnier.

2.2.2 Metodologías no Estructuradas

2.2.2.1 Metodologías Orientadas a Objeto

El modelado del sistema se realiza examinando el dominio del problema como un conjunto de objetos que interactúan entre sí. Mediante las metodologías OO se logra unificar los procesos, ya que los objetos encapsulan funciones y datos. (García y Bravo, 2008).

Enfoques:

- Revolucionario, puro u ortodoxo: rompen con lo establecido en las metodologías tradicionales. Ejemplos: Metodología ODD de Booch, CRC/Rdd de Wirfs-Brock.
- Sintetista o evolutivo: el análisis y diseño estructurado son considerados como la base para el desarrollo OO. Ejemplos: Metodología OMT de Rumbaugh, Proceso Unificado.

2.2.2.2 Metodologías de Sistemas de Tiempo Real

Están orientadas al control de un ambiente caracterizado por concurrencia, priorización de procesos, comunicación y sincronización entre tareas, acceso simultáneo a datos y respuesta ante algún evento externo.

2.2.3 Metodologías Tradicionales vs. Metodologías Ágiles

2.2.3.1 Metodologías Tradicionales

Al inicio el desarrollo de software era artesanal en su totalidad, la fuerte necesidad de mejorar el proceso y llevar los proyectos a la meta deseada, tuvieron que importarse la concepción y fundamentos de metodologías existentes en otras áreas y adaptarlas

al desarrollo de software. Esta nueva etapa de adaptación contenía el desarrollo dividido en etapas de manera secuencial que de algo mejoraba la necesidad latente en el campo del software. (Figuroa, 2002).

Las metodologías tradicionales se enfocaban principalmente en la documentación, planificación y procesos, definen un proceso secuencial en el que cada proceso se alimenta del anterior y el software está disponible únicamente al final del proceso. Este enfoque no está preparado para el cambio, es por ello que obliga al cliente a tomar todas las decisiones al principio para definir el alcance del proyecto.

Las metodologías tradicionales definen un proceso secuencial en el que se realiza el desarrollo de software. Como cada proceso se alimenta del anterior, el coste del cambio será mayor cuanto más tarde se produzca, ya que es necesario volver a repetir las tareas realizadas en los procesos previos.

2.2.3.2 Metodologías Ágiles

Las metodologías ágiles surgen como respuesta a los problemas de las metodologías tradicionales, basándose en retrasar las decisiones y en realizar planificaciones adaptativas.

El hecho de retrasar las decisiones es ventajoso tanto para el cliente como para la empresa ya que de esta manera se reduce:

- El número de decisiones de alta inversión.
- El número de cambios necesarios en el proyecto.
- El coste del cambio.

La planificación adaptativa permite tomar decisiones a lo largo del proyecto, con lo cual se irá trabajando el proyecto como un conjunto de proyectos pequeños. Esta planificación a corto plazo nos permitirá tener software disponible para el cliente y podemos ir aprendiendo del *feedback* para hacer una planificación más sensible

dado el caso de que se presenten situaciones que aceleren o retrasen el producto. (Pérez, 2005).

En el año 2001 los principales promotores de esta teoría se reunieron para crear el Manifiesto Ágil, cuyas principales ideas son:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto de software que funcione que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos.
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

2.2.3.2.1 UP Ágil (*Agile Unified Process, AUP*)

UP Ágil es una versión simplificada de la *Rational Unified Process* (RUP), basado en disciplinas y entregables incrementales con el tiempo. El ciclo de vida en proyectos grandes es serial mientras que en los pequeños es iterativo. (Figuroa, 2002).

Esta versión ligera fue desarrollada por Scott Ambler en el año 2005, se caracteriza por aplicar técnicas ágiles tales como: el Desarrollo Dirigido por Pruebas (TDD), Modelado Ágil, Gestión de Cambios Ágil y Refactorización de Base de Datos con la finalidad de mejorar la productividad.

En la Figura 11 se puede apreciar el ciclo de vida de UP Ágil.

El proceso unificado consta de cuatro fases:

- Fase 1 Inicio: tiene como objetivo determinar si vale la pena desarrollar el sistema en estudio, identificar el alcance inicial del proyecto, la arquitectura del sistema y obtener la financiación inicial del proyecto y la aceptación por parte de los interesados.

- **Fase 2 Elaboración:** su objetivo es probar la arquitectura del sistema, refinar las prioridades y desarrollar un plan de trabajo para el proyecto.
- **Fase 3 Construcción:** el objetivo es construir el software de manera iterativa y gradual, reuniendo las prioridades más altas del proyecto.
- **Fase 4 Transición:** tiene como objetivo validar e implementar el sistema en el entorno de producción para que en definitiva esté disponible para los usuarios finales. Esta fase viene dada por el *feedback* de los usuarios y permite la corrección de fallas, ajustar elementos, actualizar la documentación y descubrir riesgos no detectados anteriormente.

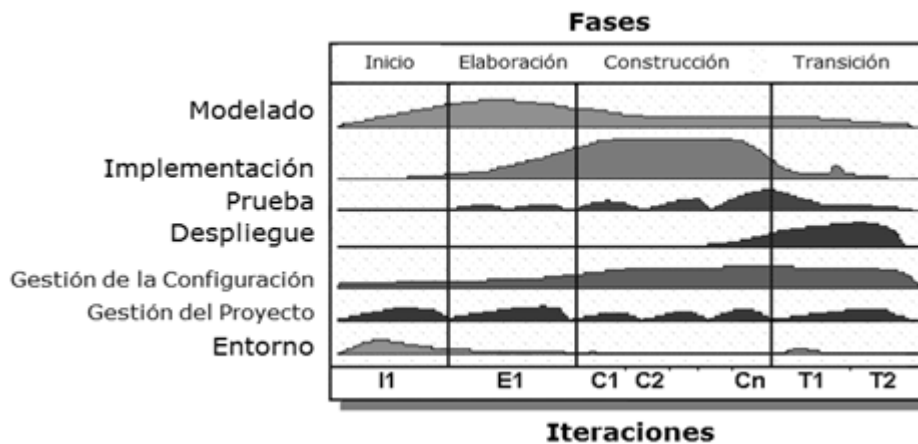


Figura 11: Ciclo de vida de UP Ágil

Fuente: ambyssoft (2009)

Las disciplinas de UP Ágil son:

- **Modelado:** su objetivo es entender el negocio de la organización, el dominio del problema del proyecto y determinar una solución viable para resolver el problema.
- **Implementación:** transforma el modelo en código ejecutable y realiza pruebas básicas.
- **Prueba:** realiza una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funcione como se estima y verificar el cumplimiento de requisitos.

- **Despliegue:** se encarga de planificar la forma en que el sistema estará a la disposición de los usuarios finales.
- **Gestión de la configuración:** maneja el acceso a los artefactos del proyecto. Esto no sólo implica un seguimiento a las versiones del artefacto, sino también el control y la gestión de cambio de los mismos.
- **Gestión del proyecto:** dirige las actividades que se llevan a cabo en el proyecto. Incluye la gestión de riesgos, coordinación con el personal y sistemas que estén fuera del alcance del proyecto para garantizar que el proyecto se entregue a tiempo y dentro del presupuesto establecido.
- **Entorno:** garantiza que el proceso adecuado, la orientación y las herramientas necesarias estén disponibles para el equipo cuando lo requieran.

2.2.3.2 Programación Extrema (XP)

Será presentada en el capítulo 3: Marco Metodológico.

2.3 Herramientas Tecnológicas

En el desarrollo de aplicaciones Web se maneja el concepto cliente – servidor multicapas, por lo cual es necesario mencionar tanto la tecnología del lado del cliente como del lado del servidor, se ha tomado como referencia a la Plataforma Empresarial Java EE 6 cuya especificación permite utilizar arquitecturas de N capas distribuidas como se muestra en la Figura 12. En esta sección se hablará de conceptos tales como Java EE, HTML5, Navegador Web, Servidor Web, JavaServer Faces, JPA, EJB y Base de Datos entre otros.

2.3.1 Navegador Web

Un navegador es una aplicación que ofrece una interfaz gráfica para ver, leer e incluso escuchar toda la información disponible en formato de hipertexto, que se encuentra disponible en servidores Web de todo el mundo y que se acceden a través de Internet, usando el protocolo de transferencia de hipertexto (Hypertext Transfer Protocol, http) para hacer solicitudes a servidores en Internet. Ejemplos de

navegadores web lo son: Mozilla Firefox, Internet Explorer, Opera, Chrome, entre otros.

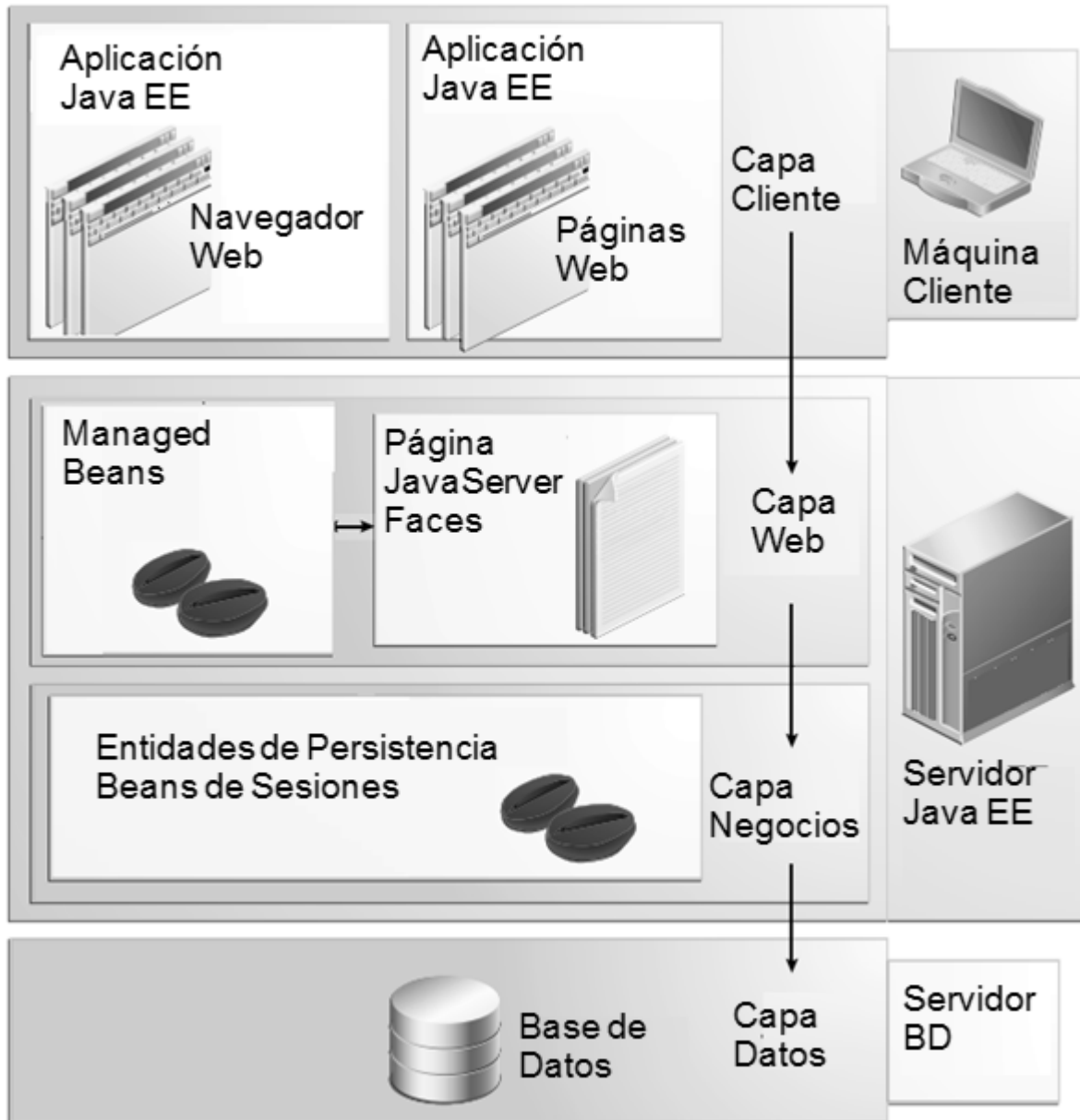


Figura 12: Plataforma Java EE 6

Fuente: Oracle (2012)

2.3.2 Servidor Web

Un servidor web, es un programa que implementa el protocolo http (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que se denomina

hipertextos, páginas web escritas generalmente en lenguaje HTML (hypertext markup language), las cuales contienen en la mayoría de los casos textos complejos con enlaces, figuras, formularios, botones y objetos incrustados de tipo multimedia como animaciones o reproductores de música.

Su trabajo consiste en mantenerse a la espera de peticiones http llevadas a cabo por un cliente http conocido como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

2.3.3 Plataforma Java Edición Empresarial

La Plataforma Java Edición Empresarial o Java, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del *Java Community Process*, Java EE es también considerada informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE; estandarizado por *The Java Community Process / JCP*.

Java EE tiene varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans, servlets, portlets (siguiendo la especificación de Portlets Java), JavaServer Pages y varias tecnologías de servicios web. Ello permite al desarrollador crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.

2.3.4 Tecnologías del lado del Cliente

Cuando se habla de las tecnologías del lado del cliente, se refiere a aquellas que son manejadas por el navegador web. Entre estas se encuentran:

2.3.4.1 HTML

HTML (HyperText Markup Languaje) es un lenguaje de marcado usado para publicar información en Internet. Se define mediante una serie de etiquetas que crean el formato con el que se visualizará un documento en el navegador. HTML tiene una limitante desventajosa que reside en su carácter estático, ya que no permite crear sitios interactivos. Cuando se escribe una página en lenguaje HTML y se coloca en un servidor web los usuarios podrán descargarla y visualizar su contenido, no obstante la única posibilidad de obtener nuevos contenidos mediante HTML es al solicitar una nueva página, por medio de un enlace o un formulario.

2.3.4.2 HTML5

HTML5 (*HyperText Markup Language*, versión 5) es la quinta revisión importante del lenguaje básico de la *World Wide Web*, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml).

Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas.

Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se le recomienda al usuario común actualizar a la versión más nueva, para poder disfrutar de todo el potencial que provee HTML5.

El desarrollo de este lenguaje de marcado es regulado por el Consorcio W3C.

2.3.4.3 Hojas de Estilo en Cascada

Las Hojas de Estilo en Cascada (Cascading Style Sheets) ó CSS, es el lenguaje usado para describir la presentación de documentos HTML o XML (CSS, 2012), Es una herramienta que proporciona al diseñador de documentos HTML un control total sobre la forma en que estos se visualizan o imprimen.

Las hojas de estilo controlan todos los elementos de la presentación de un documento HTML: márgenes de página, espaciado entre párrafos, tamaños y tipos de letra, color y fondos, entre otros.

Las hojas de estilos permiten agrupar varios formatos en nuevas clases de texto, para evitar tener que repetir las mismas etiquetas HTML en diferentes partes del documento. La presentación final de una página HTML se puede alterar, alterando su hoja de estilos, que es común a todos los documentos.

2.3.4.4 JavaScript

JavaScript es un lenguaje interpretado orientado a objetos, que permite crear interacciones entre los componentes de una página Web, el estado del navegador y las operaciones realizadas por los usuarios. Funciona del lado del cliente y el mismo puede ser usado para la interacción con cualquiera de los componentes de las páginas web, tales como asegurarnos de que el usuario introduce datos en el formato correcto, para crear efectos sobre las páginas, crear ventanas que muestren algún determinado mensaje al usuario, entre otras cosas.

2.3.4.5 AJAX

AJAX, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las

páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y *Document Object Model (DOM)*.

2.3.4.6 JQuery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

2.3.5 Tecnologías del lado del Servidor

2.3.5.1 Contenedor Web

En la plataforma Java EE, un contenedor web es la implementación que hace cumplimiento del contrato de componentes web de la arquitectura JEE. Este contrato especifica un entorno de ejecución para componentes web que incluye seguridad, concurrencia, gestión del ciclo de vida, procesamiento de transacciones, despliegue y otros servicios. Un contenedor web se suministra incluido en un servidor web JEE.

2.3.5.2 XML

Es un lenguaje usado para estructurar información en un documento o en general en cualquier archivo o fichero que contenga texto, como por ejemplo archivos de configuración de un programa. Su función principal es la de describir datos y no mostrarlos como es el caso de HTML (XML, 2012).

Este es un formato que permite la lectura de datos a través de diferentes aplicaciones y sirve para estructurar, almacenar e intercambiar información. Cuando se habla de información estructurada quiere decir la información se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces da como resultado que la información se encuentre estructurada en forma de árbol.

2.3.5.3 XHTML

XHTML, es el lenguaje de marcado que fue pensado para sustituir a HTML como estándar para las páginas web. Esta es la versión XML de HTML. Por lo cual tiene las mismas funcionalidades, pero cumpliendo con las estrictas especificaciones de XML

2.3.5.4 La Tecnología Java

La tecnología Java es una plataforma informática presentada por Sun Microsystems en 1995. Su denominación original fue OAK, y más adelante fue rebautizado como Java en 1995. Esta tecnología ofrece una gran variedad de posibilidades para los

usuarios, pues permite que prácticamente cualquier aplicación (ya sean juegos, herramientas o programas de información y servicios) se ejecute en casi cualquier equipo o dispositivo. Refiriéndonos desde computadoras de escritorio hasta dispositivos de mano. La tecnología java se encuentra compuesta de dos partes:

1. El lenguaje de Programación Java:

Java es un lenguaje de programación orientado a objetos, es decir, que define los programas en términos de clases de objetos, objetos que son entidades los cuales combinan datos, métodos o funciones y una identidad, propiedad la cual hace que un objeto se diferencie del resto. Y estos objetos colaboran entre sí para realizar determinadas tareas.

Las aplicaciones java son compiladas en un bytecode⁷ para luego ser ejecutadas por una máquina virtual que traduce el byte code a código nativo para su ejecución. En este sentido cualquier aplicación java puede ejecutarse independientemente de la plataforma donde se encuentre.

Otra característica bien importante del lenguaje de programación es que permite la ejecución multihilo de las aplicaciones, es decir que se crea la ilusión que dos o más tareas se pueden ejecutar simultáneamente dentro de un mismo programa en plataformas con un solo CPU⁸.

2. La plataforma Java:

Cuando se habla de la plataforma se refiere al ambiente de hardware y software en donde los programas escritos en lenguaje java se ejecutan, por ejemplo, plataformas como Linux, Solaris, Windows y MacOS.

En casi todos los casos las plataformas son descritas como la combinación del sistema operativo y el hardware. La plataforma Java se diferencia de estas plataformas, ya que es una plataforma sólo de software y se ejecuta sobre las otras plataformas de hardware. La plataforma Java a su vez tiene dos componentes:

- La máquina virtual de Java (JVM): es un programa ejecutable en una plataforma específica, capaz de interpretar y ejecutar las instrucciones expresadas en un código binario especial, el Java bytecode, el cual es generado por el compilador del lenguaje de programación Java.
- El Java API (Application Programming Interface): es una colección de componentes de software que proporcionan utilidades para el programador, como por ejemplo, los API's para las interfaces gráficas. Los API's de Java

2.3.5.5 Servlets

Un servlet es un programa escrito en java el cual corre del lado del servidor, con lo cual se es capaz de dotar de dinamismo a todas las páginas contenidas dentro del web server, todos los servlets se ejecutan en un contenedor que es un componente más de la arquitectura del web Server y se denomina Contenedor web.

Cualquier computadora ejecutando Servlets debe poseer un Contenedor Web. El cual es la pieza de software responsable de cargar, ejecutar y descargar los Servlets, representa el proceso único que crea, maneja y destruye los hilos de ejecución (threads) que ejecutan Servlets.

Un contenedor web provee un robusto API de desarrollo para Servlets con todos los beneficios de la plataforma Java, esto incluye la gestión del ciclo de vida de los servlets, seguridad, concurrencia y procesamientos de transacciones, entre otros.

Los servlets son clases escritas en el lenguaje de programación Java, estas se encuentran alojadas y funcionan del lado del servidor web, allí procesan peticiones (requerimientos del cliente) y de manera dinámica construyen respuestas http, comúnmente en formato HTML.

Durante su vida los servlets pasan por tres estados: inicialización, servicio y destrucción como se puede observar en la Figura 13:

1. Inicialización: en esta etapa el servidor carga e inicializa el servlet asignado los recursos necesarios que este requiera para que así posteriormente pueda atender

peticiones en la fase de servicio. En esta estado se ejecuta el método `init()` el cual es llamado una sola vez y es el responsable de llevar a tal efecto el proceso de instanciación del mismo.

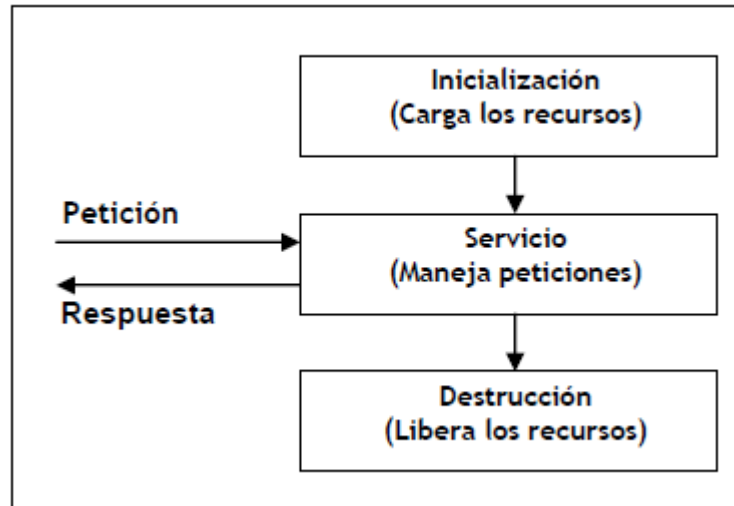


Figura 13: Ciclo de Vida de los Servlets
Fuente: UPM (2012)

2. Servicio: Una vez inicializado el servlet, este es capaz de manejar las peticiones de los clientes, en este caso la interfaz para que dicho servicio se haga efectivo, se realiza a través del método `service()`; el mismo usa dos parámetros: un objeto “`javax.servlet.ServletRequest`” y el objeto “`javax.servlet.ServletResponse`”. Estos dos objetos respectivamente representan la petición del cliente para la solicitud del recurso y la respuesta del Servlet al cliente. Cabe destacar que el método `service()` es invocado una vez por cada petición generada por el cliente. Además, por defecto los servlets son multihilo, lo que quiere decir que sólo una instancia del mismo es cargada por el contenedor web en un tiempo dado.

3. Destrucción: en esta fase el servlet libera los recursos que se le fueron otorgados durante el proceso de inicialización, para posteriormente ser removido del contenedor web, este proceso se hace efectivo haciendo la llamada del método `destroy()`.

2.3.5.6 La Arquitectura MVC

La Arquitectura MVC es un patrón de desarrollo que organiza una aplicación en tres subsistemas separados: el Modelo, la Vista y el Controlador.

El patrón MVC se centra en la separación de los datos de la aplicación o modelo, y del conjunto de pantallas que representan la interfaz de usuario, las cuales conforma a la Vista, mientras que el Controlador es el encargado relacionar a estos dos. La principal característica del patrón MVC es la separación a la vista del modelo.

Esta es una arquitectura muy apropiada para las aplicaciones web y especialmente a las tecnologías proporcionadas por la plataforma Java, de forma que:

- El modelo es la representación específica de la información con la cual opera el sistema. Son los componentes encargados del acceso a datos, estos contienen la lógica de negocio y son independientes de cualquier representación de salida. En el caso de la tecnología java sería modelado por un conjunto de clases, existiendo dos alternativas de implementación, utilizando objetos java o EJB (Enterprise JavaBeans).
- La vista presenta el modelo en un formato adecuado para interactuar, proporciona una serie de páginas web dinámicamente al cliente, que son para él simples páginas HTML. Existen múltiples frameworks que generan estas páginas web a partir de distintos formatos, siendo el más extendido el de páginas JSP (JavaServerPages).
- El controlador en la plataforma J2EE se desarrolla mediante servlets, que hacen de intermediarios entre la vista y el modelo. Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Para facilitar la implementación de estos servlets existen una serie de frameworks de desarrollo que proporcionan soporte a los desarrolladores de aplicaciones.

Una vez claros los conceptos de la arquitectura, funcionamiento de una aplicación web J2EE que utilice MVC (Letelier, 2012):

1. El usuario realiza una acción en su navegador, que llega al servidor mediante una petición HTTP y es recibida por un servlet (controlador). Esa petición es interpretada y se transforma en la ejecución de código java que delegará al modelo la ejecución de una acción de éste.

2. El modelo recibe las peticiones del controlador, a través de un interfaz o fachada que encapsulará y ocultará la complejidad del modelo al controlador. El resultado de esa petición será devuelto al controlador.
3. El controlador recibe del modelo el resultado, y en función de éste, selecciona la vista que será mostrada al usuario, y le proporcionará los datos recibidos del modelo y otros datos necesarios para su transformación a HTML.
4. En la vista se realiza la transformación tras recibir los datos del controlador, elaborando la respuesta HTML adecuada para que el usuario la visualice.

2.3.5.7 JavaServer Faces

JavaServer Faces (JSF) es una tecnología para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JavaServer Faces usa XHTML como la tecnología que permite hacer el despliegue de las páginas. JavaServer Faces incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Un conjunto por defecto de componentes para la interfaz de usuario.
- Bibliotecas de etiquetas personalizadas para XHTML que permiten expresar una interfaz JavaServer Faces dentro de una página XHTML.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.

2.3.5.8 PrimeFaces

PrimeFaces es un componente para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes ricos que facilitan la creación de las aplicaciones web. Primefaces está bajo la licencia de Apache License V2. Una de las ventajas de utilizar Primefaces, es que permite la integración con otros componentes como por ejemplo RichFaces y entre sus propiedades se encuentran:

- Un conjunto de componentes ricos (Editor de HTML, autocompletar, cartas, gráficas o paneles, entre otros)
- Soporte de AJAX con despliegue parcial, lo que permite controlar cuáles componentes de la página actual se actualizarán y cuáles no
- 25 temas prediseñados
- Un componente para desarrollar aplicaciones web para móviles-celulares, especiales para Iphones, Palm, Android y teléfonos móviles Nokia.

2.3.5.9 JavaBean

Los JavaBeans son un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java.

La especificación de JavaBeans de Sun Microsystems los define como "componentes de software reutilizables que se puedan manipular visualmente en una herramienta de construcción".

Para funcionar como una clase JavaBean, una clase debe obedecer ciertas convenciones sobre nomenclatura de métodos, construcción, y comportamiento.

Estas convenciones permiten tener herramientas que puedan utilizar, reutilizar, sustituir, y conectar JavaBeans.

Las convenciones requeridas son:

- Debe tener un constructor sin argumentos.

- Sus propiedades deben ser accesibles mediante métodos get y set que siguen una convención de nomenclatura estándar.
- Debe ser Serializable

Dentro de un JavaBean podemos distinguir tres partes:

- Propiedades: Los atributos que contiene.
- Métodos: Se establecen los métodos get y set para acceder y modificar los atributos.
- Eventos: Permiten comunicarnos con otros JavaBeans.

2.3.5.10 Enterprise JavaBeans

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

- Comunicación remota utilizando CORBA.
- Transacciones.
- Control de la concurrencia.
- Eventos utilizando JMS (Java messaging service).
- Servicios de nombres y de directorio.
- Seguridad.
- Ubicación de componentes en un servidor de aplicaciones.

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

2.3.5.11 GlassFish

GlassFish es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que

siguen esta especificación. Es gratuito, de código libre y se distribuye bajo un licenciamiento dual a través de la licencia CDDL y la GNU GPL. La versión comercial es denominada Oracle GlassFish Enterprise Server (antes Sun GlassFish Enterprise Server).

GlassFish está basado en el código fuente donado por Sun y Oracle Corporation; este último proporcionó el módulo de persistencia TopLink. GlassFish tiene como base al servidor Sun Java System Application Server de Oracle Corporation, un derivado de Apache Tomcat, y que usa un componente adicional llamado Grizzly que usa Java NIO para escalabilidad y velocidad.

2.3.5.12 NetBeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

2.3.5.13 PostgreSQL

PostgreSQL es un Sistema de Gestión de Base de Datos relacional orientado a objetos y libre, publicado bajo la licencia BSD. El desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

CAPÍTULO III: MARCO METODOLÓGICO

3.1 Nivel y tipo de Investigación

Para precisar que es la investigación se han escogido las siguientes definiciones:

"Genéricamente, la investigación es una actividad del hombre, orientada a descubrir algo desconocido." (Sierra Bravo, 1991).

"Una investigación puede definirse como un esfuerzo que se emprende para resolver un problema, claro está, un problema de conocimiento." (Sabino, 1992).

"Se define la investigación como una actividad encaminada a la solución de problemas. Su objetivo consiste en hallar respuesta a preguntas mediante el empleo de procesos científicos. (Cervo y Bervian, 1989).

La investigación debe estar presente cuando se tiene un objetivo de probar la efectividad un proyecto, en síntesis:

"La investigación científica es un proceso metódico y sistemático dirigido a la solución de problemas o preguntas científicas, mediante la producción de nuevos conocimientos, los cuales constituyen la solución o respuesta a tales interrogantes." (Fidias, 2006).

Para ubicar el presente Trabajo Especial de Grado dentro del contexto de clases de la investigación se detalla que: "El nivel de investigación se refiere al grado de profundidad con se aborda un fenómeno u objeto de estudio" (Fidias, 2006). Partiendo de dicha definición se señala que la investigación contenida en este trabajo es de Nivel Descriptiva.

"La investigación descriptiva consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento. Los resultados de este tipo de investigación se ubican en un nivel intermedio en cuanto a la profundidad de los conocimientos se refiere." (Fidias, 2006).

Según el propósito de la investigación se puntualiza que el tipo de investigación de este trabajo es Aplicada, encaminada a la solución de problemas prácticos, dirigida al desarrollo de un producto.

“La investigación aplicada se refiere a la producción de nuevo conocimiento el cual pretende tener una aplicación inmediata en la solución de problemas prácticos.”(Fidias, 2006).

3.2 Diseño de Investigación

“El diseño de la investigación es la estrategia general que adopta el investigador para responder al problema planteado.”(Fidias, 2006).

Esta investigación está diseñada bajo el esquema de una Investigación documental monográfica, de nivel descriptivo y de propósito aplicativo.

“La investigación documental es un proceso basado en la búsqueda, recuperación, análisis, crítica e interpretación de datos secundarios...”(Fidias, 2006).

“El tipo de investigación documental monográfica consiste en el desarrollo amplio y profundo de un tema específico.” (Fidias, 2006).

3.3 Técnicas de Recolección de Datos

“Se entenderá por técnica, el procedimiento o forma particular de obtener datos e información.” (Fidias, 2006).

La aplicación de una técnica conduce a la obtención de información, la cual deber ser almacenada en un medio físico, para que luego puedan ser recuperados, procesados, analizados e interpretados posteriormente. A este soporte de le conoce como instrumento.

“Un instrumento de recolección de datos es cualquier recurso, dispositivo o formato (en papel o digital), que se utiliza para obtener, registrar o almacenar información.” (Fidias, 2006).

Como el diseño de la investigación de este Trabajo Especial de Grado es documental se procede a elegir como técnicas de recolección de datos e instrumentos a la revisión bibliográfica y a la entrevista, las cuales se describen a continuación:

- La Revisión Bibliográfica: “Se debe recurrir a la técnica de revisión bibliográfica; tanto de libros, folletos, documentos, revistas, artículos y seminarios, los cuales brindan todo el soporte del marco teórico.”(Bonillo, 2008). La Revisión Bibliográfica, se utiliza como base complementaria a la investigación central, con el fin de recopilar y revisar todos aquellos documentos que permitan confrontar el aspecto teórico con la situación real o práctica dentro del modelo de evaluación de las capacidades tecnológicas necesarias en el mercado de la gerencia de tecnologías de la información a través del uso de arquitecturas empresariales.
- La Entrevista: “la entrevista, más que un simple interrogatorio, es un técnica basada en un diálogo o conversación ‘cara a cara’, entre el entrevistador y el entrevistado acerca de un tema previamente determinado, de tal manera que el entrevistador pueda obtener la información requerida.” (Fidias, 2006). La entrevista se usa en el desarrollo del sistema de información propuesto por este trabajo y su nombre particular es Historias de Usuario.

3.4 Procedimiento de Análisis de Datos

Al investigar se busca y se selecciona la bibliografía para su análisis crítico y descripción, precisando toda la información posible sobre el problema que se está investigando, con un doble objetivo:

- Conocer los antecedentes de ese tema de investigación y el estado actual del conocimiento: los hallazgos de investigaciones anteriores o los resultados obtenidos en estudios previos, informa de lo que se sabe en la actualidad sobre un problema de interés. Se trata de que el investigador se pregunte:

¿Qué han estudiado otros sobre el tema que vamos a investigar y qué resultados han obtenido?

- Desarrollar el contexto conceptual en el cual se inserta el problema de investigación (ubicar la investigación en un contexto conceptual): conocer las descripciones teóricas de investigaciones previas, nos proporciona el contexto conceptual para un problema de investigación. En la revisión bibliográfica, igualmente se pregunta: ¿Qué teorías marco teórico, marcos modelos esquemas conceptuales usan las investigaciones similares o de un mismo campo?

Para el presente Trabajo Especial de Grado se usó un análisis de datos cualitativo cuyo proceso se describe de la forma siguiente (Álvarez-Gayou, 2005):

1. Obtener la información: a través de la revisión bibliográfica y la entrevista.
2. Capturar, transcribir y ordenar la información: la captura de la información se hace a través de diversos medios. Específicamente, en el caso de entrevistas (Historias de Usuario), a través de un registro electrónico (en formato digital). En el caso de documentos, a través de la recolección de material original en físico y digital, o de la realización de fotocopias o el escaneo de esos originales.
3. Seleccionar la información: es el proceso mediante el cual se agrupa la información obtenida en categorías que concentran las ideas, conceptos o temas similares descubiertos por el investigador, y se descarta el resto de la información que no tiene que nada que ver con el tema de investigación.
4. Integrar la información: se toma la información seleccionada y se ordena para presentar un marco teórico de forma lógica.

3.5 Metodología utilizada en función al proyecto

El procedimiento a seguir para lograr los objetivos específicos de esta investigación se constituirá en la aplicación de una instanciación del proceso de desarrollo de software Programación Extrema (XP) propuesto por Kent Beck en 1998 a través del proyecto de desarrollo *Chrysler Comprehensive Compensation*. “XP es principalmente usado para proyectos a corto plazo, basado en un conjunto de valores

y de prácticas que persigue el objetivo de aumentar la productividad a la hora del desarrollo de software.” (Beck, 2000). En la Figura 14 se puede observar el ciclo de vida de un proyecto en *XP*.

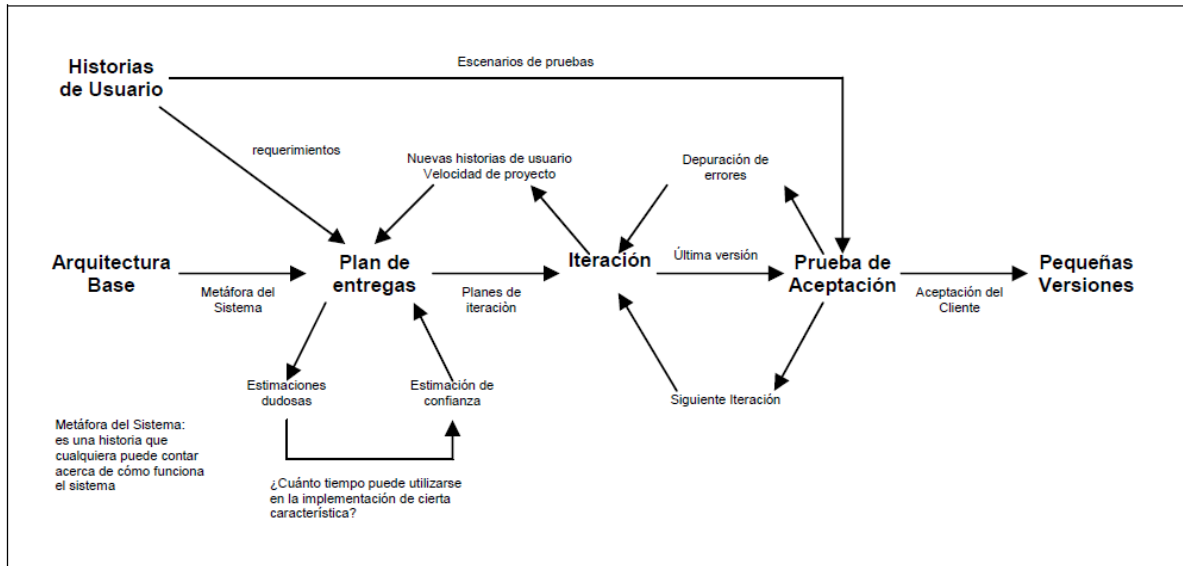


Figura 14: Ciclo de vida de un proyecto en Programación Extrema
Fuente: UPM (2012).

Debido a las limitaciones de tiempo que presenta el desarrollo del prototipo del sistema de información propuesto para este Trabajo Especial de Grado, la búsqueda de una metodología Ágil permite plasmar un prototipo en un tiempo razonable y adecuado para un trabajo que se desarrolla en un lapso estipulado de 6 meses académicos, que incluyen: el desarrollo de la investigación, el desarrollo del prototipo del sistema, la elaboración y revisión del informe y su posterior publicación y defensa. Dentro de las metodologías ágiles la Programación Extrema fue seleccionada debido a la extensa documentación y acceso a la misma que se puede encontrar tanto física, a través de libros y otras publicaciones, como a través de formato digital en Internet, y las otras bondades del desarrollo ágil expuestas anteriormente.

Las fases del ciclo de vida de la Programación Extrema son (Wake, 2002):

1. Fase de exploración.
2. Fase de planificación.
3. Fase de iteraciones.

3.5.1 Fase de exploración

Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuarios”. Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración. Esta fase dura típicamente un par de semanas, y el resultado es una visión general del sistema, y un Plan de Estimación como se observa en la tabla 5 (Wake, 2002).

Tabla 5: Plan de Estimación
Fuente: Kent Beck (2000)

Plan de Estimación	
Historia	Estimación

El proyecto comienza recopilando “Historias de usuarios”, las que sustituyen a los tradicionales “casos de uso”. Una vez obtenidas las “historias de usuarios”, los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Si alguna de ellas tiene “riesgos” que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba (“spikes”), para reducir estos riesgos. El formato de historias de usuario que se presenta en este trabajo se muestra en la tabla 6.

Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios. Las historias de usuarios deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia. (Letelier, 2012).

Tabla 6: Historia de Usuario
Fuente: DSIIC (2003)

Historia de Usuario	
Número:	Usuario:
Nombre historia:	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados:	Iteración asignada:
Programador responsable:	
Descripción:	
Observaciones:	

3.5.2 Fase de planificación

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas como se muestra en la tabla 7 (Escribano, 2002).

Tabla 7: Plan de Entregas
Fuente: Kent Beck (2000)

Plan de Entregas			
Historia	Tiempo Estimado	Nº Iteración Asignada	Nº Entrega Asignada

Una vez realizadas las estimaciones de riesgos de la fase de exploración, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un Plan de Entregas en los que todos estén de acuerdo.

El Plan de Entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes, etc.). XP denomina a esta reunión “Juego de planificación”. Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. El plan de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores. Luego de algunas iteraciones es recomendable realizar

nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario.

3.5.3 Fase de iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance. El plan de Iteraciones se muestra en la tabla 8.

Tabla 8: Plan de Iteraciones
Fuente: Kent Beck (2000)

Plan de Iteraciones	
Evento	Fecha

Una vez acordado el Plan de Entregas de la fase de planificación, comienza una fase de iteraciones, en dónde en cada una de estas iteraciones se desarrolla, prueba e instalan unas pocas “historias de usuarios”. El formato de descripción de una tarea de la fase de iteración se muestra en la tabla 9.

Se busca estructurar el código mediante el paradigma Orientado a Objetos usando tarjetas C.R.C (Clases, Responsabilidades y Colaboraciones). Las Tarjetas C.R.C tendrán en la parte superior la Clase a la que pertenece el Objeto, en una columna a la izquierda se especificaran los objetivos y/o responsabilidades del Objeto y en una columna a la derecha las Clases que intervienen en cada objetivo como se muestra en la tabla 10 (Escribano, 2002).

Tabla 9: Descripción de Tarea
Fuente: DSIIC (2003)

Tarea	
Número tarea:	Número historia:
Nombre tarea:	
Tipo de tarea : Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos estimados:
Fecha inicio:	Fecha fin:
Programador responsable:	
Descripción:	

En el Plan de Iteraciones las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación. Asimismo, para cada historia de usuario se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero

también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores. Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para prever que no vuelvan a ocurrir.

Tabla 10: Tarjeta CRC
Fuente: Elaboración propia

Nombre de la Clase	
Responsabilidades	Colaboradores

En las metodologías tradicionales, la fase de pruebas, incluyendo la definición de las pruebas, es usualmente realizada sobre el final del proyecto, o sobre el final del desarrollo de cada módulo. La metodología XP propone un modelo inverso, en el que, lo primero que se escribe son las pruebas que el sistema debe pasar. Luego, el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas. Las pruebas a las que se refiere esta práctica, son las pruebas unitarias, realizados por los desarrolladores. La definición de estas pruebas al comienzo, condiciona o “dirige” el desarrollo.

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (“Programación dirigida por pruebas, TDD”). Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo.

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información. El formato de Caso de Prueba de Aceptación se muestra en la tabla 11.

Tabla 11: Caso de Prueba de Aceptación
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código:	Historia de Usuario (Nº y Nombre):
Nombre:	
Descripción:	
Condiciones de Ejecución:	
Entrada / Paso de Ejecución:	
Resultado Esperado:	
Evaluación de la Prueba:	

CAPÍTULO IV: MARCO APLICATIVO Y RESULTADOS

4.1 Caso de Estudio

En esta sección se presentará la empresa del caso de estudio: Capas360 C.A.

4.1.1 Identificación de la Empresa

4.1.1.1 Capas360 C.A.

Capas360 C.A. es la unión de un grupo de emprendedores profesionales con mucha experiencia en el diseño de plataformas de servicio y con experiencia en el desarrollo de soluciones para el sector salud (Capas360, 2013).

4.1.1.2 Misión

Ser la comunidad de servicios de salud y bienestar más importante de Latinoamérica destacándose por ofrecer una excelente calidad de servicio y una constante innovación tecnológica de la plataforma (Capas360, 2013).

4.1.1.3 Visión

Ser la plataforma de servicios de salud en línea líder en Latino América con la mayor comunidad de profesionales, instituciones, empresas y personas interactuando de forma segura y con un alto nivel de satisfacción producto de nuestro servicio y nuestra innovación tecnológica (Capas360, 2013).

4.1.1.4 Objetivos de la empresa

- Brindar servicios, integración y colaboración entre los diferentes actores de la comunidad Médicos, Ciudadanos y Prestadores de Servicios.
- Mantener información actualizada en nuestra comunidad.
- Ser una plataforma de integración de los servicios de Salud.
- Prestar servicios a las comunidades (Capas360, 2013).

4.1.1.5 Organigrama de la Empresa

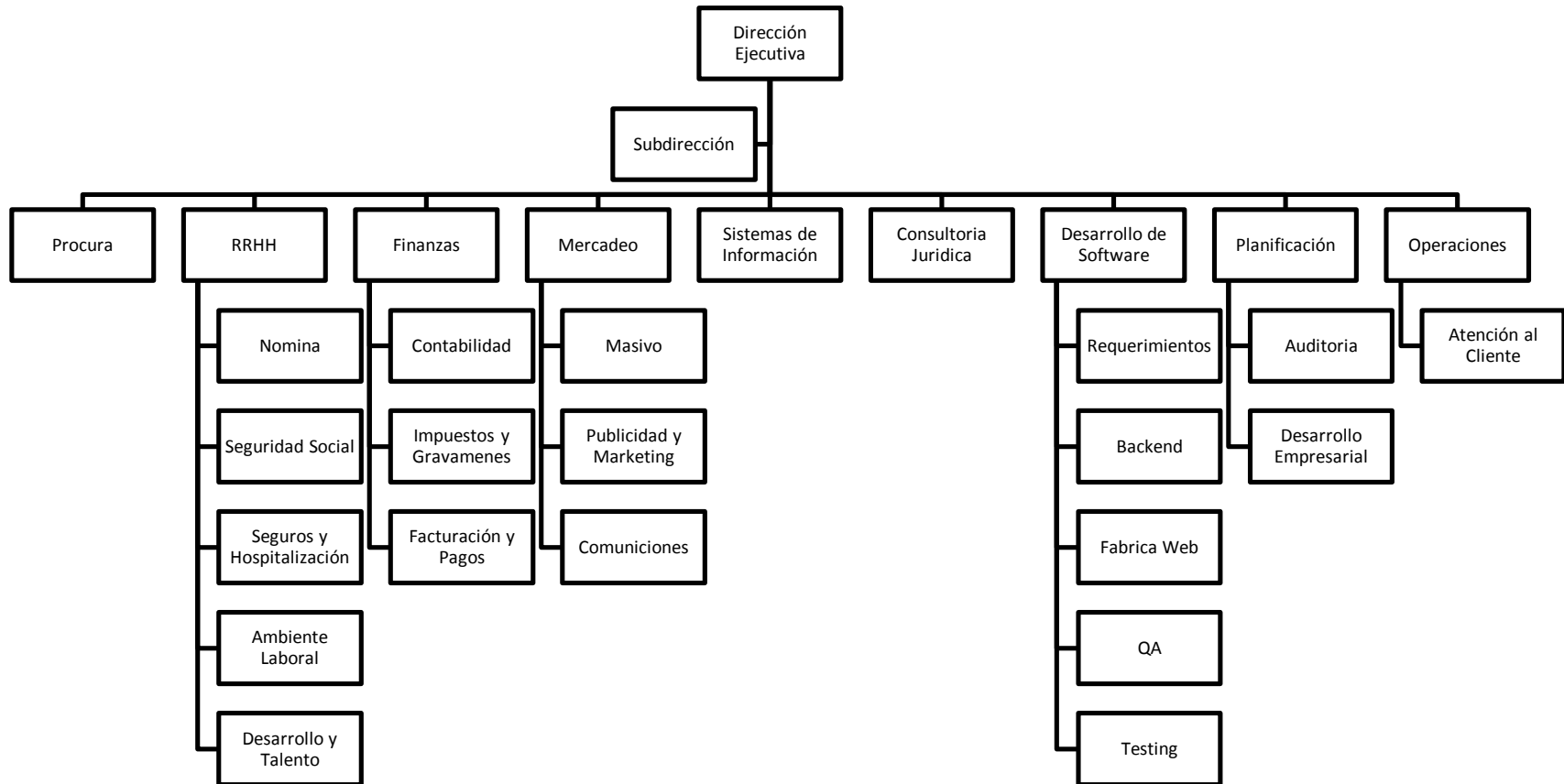


Figura 15: Organigrama de la Empresa Capas360 C.A.

Fuente: Capas360 (2013).

4.1.1.6 Descripción de la Estructura Organizacional de la Empresa

4.1.1.6.1 Direcciones

4.1.1.6.1.1 Dirección Ejecutiva

(Director Ejecutivo, CEO: Leonardo Capas): es el encargado de máxima autoridad de la gestión y dirección administrativa de la empresa (Capas360, 2013).

4.1.1.6.1.2 Subdirección

(Subdirector Ejecutivo: Wilmer Vidal): es el asistente del Director Ejecutivo, y al ausentarse el mismo, este le reportará todos los acontecimientos del día, toma decisiones, tiene las responsabilidad de las actividades llevadas a cabo, planifica, organiza, dirige y controla actividades de su cargo, dirige personal, resuelve conflictos y cualquier actividad delegada por parte del Director Ejecutivo (Capas360, 2013).

4.1.1.6.2 Departamentos y Unidades

4.1.1.6.2.1 Departamento de Procura

(Gerente General de Procura y Abastecimiento: Alexander Ibarrén): es el encargado de cubrir las necesidades de consumo de la empresa en tiempo, forma y calidad (Capas360, 2013).

4.1.1.6.2.2 Departamento de Recursos Humano

(Gerente General de Recursos Humanos: Adriana Peralta): es la encargada de alinear el área o profesionales de Recursos Humanos con la estrategia de la organización. El Departamento de Recursos Humanos consta de las siguientes unidades:

- Unidad de Nomina (Jefe de la Unidad de Nomina: Mary Díaz): es la encargada de elaborar y pagar la nómina del personal de conformidad con el presupuesto aprobado y contratos establecidos.
- Unidad de Seguridad Social (Jefe de la Unidad de Seguridad Social: Susana Teról): es la encargada de ejercer el control y dar cumplimiento a las

prestaciones y haberes que por Ley le corresponde al personal con apego a la normativa vigente.

- Unidad de Seguros y Hospitalización (Jefe de la Unidad de Seguros y Hospitalización: David García): es el encargado de optimizar el capital que anualmente se destina a seguros generales (auto, hogar, vida, accidentes, salud, ahorro), a seguros industriales/empresariales (responsabilidades, construcción, transporte, comercio, oficinas) y a hospitalización y medicina ambulatoria del personal.
- Unidad de Ambiente Laboral (Jefe de la Unidad de Ambiente Laboral: Anita Cáceres): es la encargada de desarrollar y fomentar un Servicio de Seguridad y Salud en el Trabajo, además de diseñar, planificar, organizar y ejecutar un programa de educación e información preventiva, en materia de seguridad y salud en el trabajo.
- Unidad de Desarrollo y Talento (Jefe de la Unidad de Desarrollo y Talento: Luisa Cabello): es la encargada de desarrollar modelos de competencias, mantenimiento de perfiles con competencias actualizados y vigentes, procurar del mejoramiento del clima y la cultura organizacional (Capas360, 2013).

4.1.1.6.2.3 Departamento de Finanzas

(Gerente General de Finanzas, CFO: Gabriel Villanueva): es el ejecutivo a cargo de la gestión financiera de la organización. Es responsable de la planificación, ejecución e información financieras. El Departamento de Finanzas consta de las siguientes unidades:

- Unidad de Contabilidad (Jefe de la Unidad de Contabilidad: Wilson Icaza): es el encargado de instrumentar y operar las políticas, normas, sistemas y procedimientos necesarios para garantizar la exactitud y seguridad en la captación y registro de las operaciones financieras.
- Unidad de Impuestos y Gravámenes (Jefe de la Unidad de Impuestos y Gravámenes: Javier González): es el responsable de velar por los aspectos

tributarios de la organización conforme a la ley, incluyendo impuestos a la propiedad, indirectos, de renta, municipales y nacionales.

- **Unidad de Facturación y Pagos** (Jefe de la Unidad de Facturación y Pagos: Felipe Pinto): es el encargado de garantizar la facturación de todos los servicios realizados por la compañía y los pagos a proveedores, por medio de un minucioso seguimiento en cada uno de los procesos de la organización (Capas360, 2013).

4.1.1.6.2.4 Departamento de Mercadeo

(Gerente General de Mercadeo y Publicidad: Mónica Ribas): es la encargada de dirigir la ejecución de los procesos de mercadeo y comunicación de la empresa. El Departamento de Mercadeo consta de las siguientes unidades:

- Unidad de Mercadeo Masivo (Jefe de la Unidad de Mercadeo Masivo: Eusebio Aguilera): es el encargado de identificar los tipos de mercado, sus características, el comportamiento humano, así como lograr diseñar e implementar estrategias de consumo masivo con las cuales puedan posicionarse con éxito los productos, servicios o ideas en los mercados nacionales e internacionales.
- Unidad de Publicidad y Marketing (Jefe de la Unidad de Publicidad y Marketing: Rosa Gómez): es la responsable de identificar las necesidades y deseos del mercado objetivo y adaptarse para ofrecer las satisfacciones deseadas por el mercado de forma más eficiente que la competencia.
- Unidad de Comunicaciones (Jefe de la Unidad de Comunicaciones: Daniela Martínez): es la encargada de diseñar comunicaciones que difundan un mensaje acerca de los productos y que éste tenga una respuesta del público objetivo al que va destinado (Capas360, 2013).

4.1.1.6.2.5 Departamento de Sistemas de Información

(Gerente General de Sistemas de Información: Deiberg González): es el encargado de gestionar eficiente y eficazmente los recursos, la infraestructura y servicios

tecnológicos de la empresa, mediante la administración, mantención y desarrollo de sistemas de información y servicios informáticos que apoyen los procesos realizados por usuarios internos y la realización de trámites y obtención de servicios por parte de usuarios externos (Capas360, 2013).

4.1.1.6.2.6 Departamento de Consultoría Jurídica

(Gerente General de Consultoría Jurídica: Yolanda Angulo): es la encargada de velar por el cumplimiento de todas las leyes, normas o reglamentos que regulan las operaciones y/o actividades de la empresa (Capas360, 2013).

4.1.1.6.2.7 Departamento de Desarrollo de Software

(Gerente General de Tecnología, CTO: Rafael Pieters): es el ejecutivo encargado de desarrollar los productos y servicios que requieren los clientes de la empresa, el desarrollo de software es el corazón y negocio de la compañía. El Departamento de Desarrollo de Software consta de las siguientes unidades:

- Unidad de Requerimientos y Análisis (Jefe de la Unidad de Requerimientos y Análisis: Alex Mazziotta): es el responsable de analizar las necesidades de los clientes y usuarios finales del software para determinar qué objetivos debe cubrir.
- Backend (Jefe de Backend: Miguel Bautista): es el responsable de descomponer y organizar el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo y definir los algoritmos empleados y la organización del código para comenzar la implementación de la capa de back-end.
- Factoría Web (Jefe de la Factoría Web: Ricardo Subero): es el responsable de descomponer y organizar el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo y definir los algoritmos empleados y la organización del código para comenzar la implementación de la capa de front-end.

- QA (Jefe de QA: Antonio Rodríguez): es el encargado de asegurar la garantía de calidad de software.
- Testing (Jefe de Testing: Ernesto Pérez): es el responsable de presentar conceptos, métodos, técnicas y procesos relativos a las múltiples actividades de las pruebas dentro del proceso de desarrollo, además de emitir la certificación de pruebas del sistema en desarrollo (Capas360, 2013).

4.1.1.6.2.8 Departamento de Planificación

(Gerente General de Estrategia y Planificación Empresarial: Ramón Suárez): es el encargado de desarrollar e implementar planes para alcanzar los propósitos y objetivos de la empresa. El Departamento de Planificación consta de las siguientes unidades:

- Unidad de Auditoría Interna (Jefe de la Unidad de Auditoría Interna: Natalia González): es la responsable de implementar las políticas dictadas por la Junta Administrativa en materia de auditoría, monitorear la correcta secuencia de los ciclos de transacciones, los sistemas de operación aplicables, revisar y aprobar el programa anual de trabajo de auditoría interna, recibir y comentar los informes de auditorías interna.
- Unidad de Desarrollo Empresarial (Jefe de la Unidad de Desarrollo Empresarial: José Luis Monzón): es el encargado de desarrollar la visión futura de la empresa, reflejada en la orientación, participación y promoción para la formulación, seguimiento, control y evaluación de la ejecución de las políticas, planes, programas, estudios y proyectos de inversión, de todos los departamentos de la empresa.

4.1.1.6.2.9 Departamento de Operaciones

(Gerente General de Operaciones, COO: Juan Carlos Gallego): es el ejecutivo responsable del control de las actividades diarias de la empresa y del manejo de las operaciones. El Departamento de Operaciones consta de las siguientes unidades:

- Unidad de Atención al Cliente (Jefe de la Unidad de Atención al Cliente: Jairo García): es el responsable de proporcionar total asistencia, así como un servicio de información exhaustivo, y garantizar que todas las preguntas o consultas relativas de los clientes a todos los productos, servicios o pedidos hechos a la empresa sean atendidas.

4.2 Fase de exploración

Esta es la planificación de historias que se realizó al inicio del proyecto, tras estudiar el proyecto y mantener conversaciones con el cliente. De esta redacción inicial de historias de usuario se realizó una planificación inicial y posteriormente fue cambiada a lo largo del proyecto. En la tabla 12 se describe la Historia de Usuario 1, en la sección de anexos se presenta el resto. En la tabla 33 se presenta el Plan de Estimación.

Tabla 12: Historia de Usuario 1
Fuente: Elaboración propia

Historia de Usuario	
Número: 1	Usuario: Analista de Planificación
Nombre historia: Identificación de las Unidades de Negocio principales.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: José Osuna/Miguel Bautista	
Descripción: Se crea cada unidad de negocio principal a través de un formulario ingresando nombre de la unidad de negocio, lista de sistemas y proyectos de dicha unidad, nombre del gerente encargado y valor de dicha unidad para la el desarrollo de la arquitectura.	
Observaciones:	

Tabla 33: Plan de Estimación
Fuente: Elaboración propia

Plan de Estimación	
Historia	Estimación
1 Identificación de las Unidades de Negocio principales	2 horas
2 Identificación de las Unidades de Negocio secundarias	3 horas
3 Identificación de las Unidades de Negocio extendidas	2 horas
4 Identificación de las Comunidades involucradas	4 horas
5 Identificación de la Gobernabilidad requerida	8 horas
6 Administración y Gobernabilidad de los artefactos	3 horas
7 Revisión de la Gobernabilidad existente	8 horas
8 Introducción de los puntos de mejora de la Arquitectura	1 hora
9 Generación de un estudio del Nivel de Madurez	4 horas
10 Asignación de roles y responsabilidades	4 horas
11 Introducción de los requerimientos para los cambios	4 horas
12 Introducción del alcance del desarrollo	1 hora
13 Introducción de las premisas y requerimientos	1 hora
14 Definición de la relación entre los patrocinadores	4 horas
15 Introducción de los requerimientos de presupuesto	3 horas
16 Introducción de los Principios de la Arquitectura	3 horas
17 Introducción de lo que la adaptación necesita	1 hora
18 Producción de un conjunto de terminologías	2 horas
19 Retiro de las tareas que ya están realizadas	3 horas
20 Introducción de las tareas específicas	3 horas
21 Alineación de los procesos	4 horas

4.2 Fase de planificación

Tras la fase de exploración el cliente, los gerentes y el grupo de desarrolladores acordaron el orden en que se implementarían las historias de usuario, y, asociadas a éstas, las entregas, plasmadas en Plan de Entregas descrito en la tabla 34.

Tabla 34: Plan de Entregas
Fuente: Elaboración propia

Plan de Entregas			
Historia	Tiempo Estimado	Nº Iteración Asignada	Nº Entrega Asignada
1	2 horas	1	1
2	3 horas	1	1
3	2 horas	1	1
4	4 horas	1	1
5	8 horas	1	1
6	3 horas	1	1
7	8 horas	1	1
8	1 hora	2	1
9	4 horas	2	1
10	4 horas	2	1
11	4 horas	2	1
12	1 hora	2	1
13	1 hora	2	1
14	4 horas	2	1
15	3 horas	3	1
16	3 horas	3	1
17	1 hora	3	1
18	2 horas	3	1
19	3 horas	3	1
20	3 horas	3	1
21	4 horas	3	1

4.3 Fase de iteraciones

Una vez realizada la fase de planificación se procede a diseñar un Plan de Iteraciones, tal cual como muestra la tabla 35, las historias de usuarios son desarrolladas y probadas en un ciclo de iteración.

Tabla 35: Plan de Iteraciones
Fuente: Elaboración propia

Plan de Iteraciones	
Evento	Fecha
Establecer las Tareas de Programación para la Iteración 1	4 oct 2012
Diseñar las Pruebas de Aceptación para la Iteración 1	8 oct 2012
Desarrollar las Tareas de Programación de la Iteración 1	13 oct 2012
Realizar las Pruebas de Aceptación de la Iteración 1	5 nov 2012
Analizar las Pruebas de Aceptación que hallan fallado	5 nov 2012
Corregir las Incidencias de las Pruebas de Aceptación	24 nov 2012
Establecer las Tareas de Programación para la Iteración 2	21 dic 2012
Diseñar las Pruebas de Aceptación para la Iteración 2	26 dic 2012
Desarrollar las Tareas de Programación de la Iteración 2	29 dic 2012
Realizar las Pruebas de Aceptación de la Iteración 2	26 ene 2013
Realizar las Pruebas de Aceptación de la Iteración 1	26 ene 2013
Analizar las Pruebas de Aceptación que hallan fallado	26 ene 2013
Corregir las Incidencias de las Pruebas de Aceptación	27 ene 2013
Establecer las Tareas de Programación para la Iteración 3	17 feb 2013
Diseñar las Pruebas de Aceptación para la Iteración 3	23 feb 2013
Desarrollar las Tareas de Programación de la Iteración 3	2 mar 2013
Realizar las Pruebas de Aceptación de la Iteración 3	6 abr 2013
Realizar las Pruebas de Aceptación de la Iteración 1	6 abr 2013
Realizar las Pruebas de Aceptación de la Iteración 2	6 abr 2013
Analizar las Pruebas de Aceptación que hallan fallado	7 abr 2013
Corregir las Incidencias de las Pruebas de Aceptación	17 abr 2013

La Programación Extrema XP propone escribir las pruebas de aceptación que el sistema debe pasar para luego comenzar a codificar, a continuación se presentan las pruebas de aceptación con todos los casos de pruebas propuestos por el cliente. La tabla 36 describe el Caso de Prueba de Aceptación CA1-1, en la sección de anexos se presenta el resto.

Tabla 36: Caso de Prueba de Aceptación CA1-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA1-1	Historia de Usuario (Nº y Nombre): 1
Nombre: Introducción de unidad de negocio principal con errores	
Descripción: El analista una vez autenticado en el sistema seleccionará la opción del panel “Unidades Principales” y hará clic en el botón “Nueva Unidad de Negocios”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El analista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El analista introducirá su identificador y contraseña para autenticarse. - Del panel seleccionará “Unidades Principales” y hará clic en el botón “Nueva Unidad de Negocios” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará un mensaje diciendo “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: Las unidades de negocio introducidas de forma incorrecta no son almacenadas en la base de datos mientras que las correctas si son almacenadas.	
Evaluación de la Prueba: Prueba satisfactoria.	

Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación. La tabla 78 presenta la Tarea 1 para la Historia de Usuario 1, en la sección de anexos se presenta el resto.

Tabla 78: Tarea 1, Historia 1
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 1
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 13/10/2012	Fecha fin: 14/10/2012
Programador responsable: Equipo XP	
Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.	

Para estructurar el código mediante el paradigma Orientado a Objetos se usan tarjetas C.R.C (Clases, Responsabilidades y Colaboraciones). En la Tabla 151 se muestra la tarjeta C.R.C. Unidad, en la sección de anexos se presenta el resto.

Tabla 151: C.R.C. Unidad
Fuente: Elaboración propia

Unidad	
Responsabilidades	Colaboradores
Gestionar las Unidades de Negocio de la Arquitectura Empresarial.	TipoUnidad, Sistema, Proyecto, Persona.

La Arquitectura base del Sistema está enmarcada dentro del patrón MVC tipo 1, donde cada vista está relacionada con uno o más controladores, los cuales se apoyan en servicios remotos que suelen implementar CRUD por cada modelo del sistema. La Arquitectura se describe en la figura 16, las clases que ayudan a modelar el módulo de seguridad y operaciones del sistema se describen en la figura 17, mientras que las clases para los módulos de análisis, especialista y líder están descritos en la figura 18.

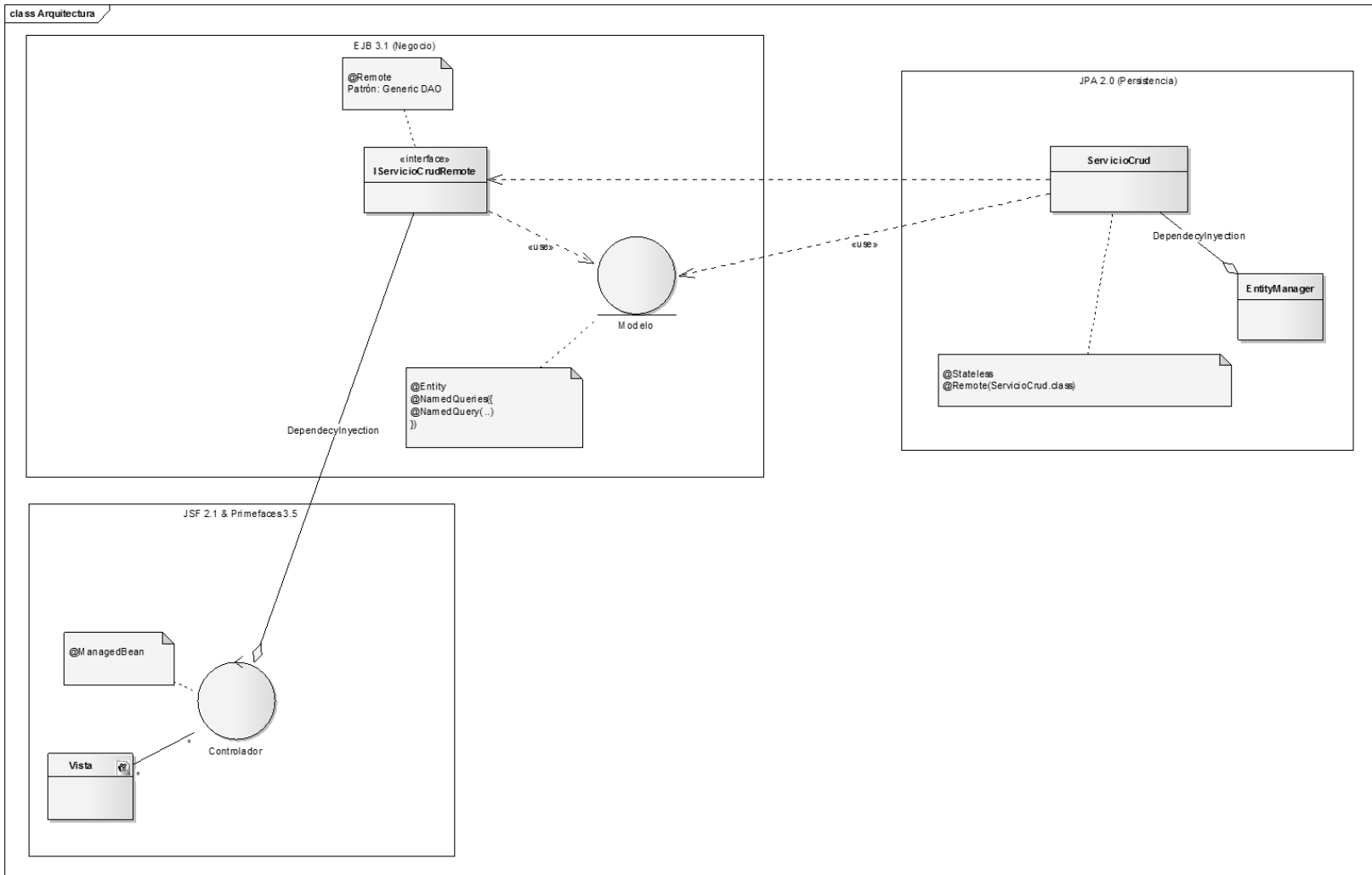


Figura 16: Arquitectura del Sistema

Fuente: Elaboración Propia

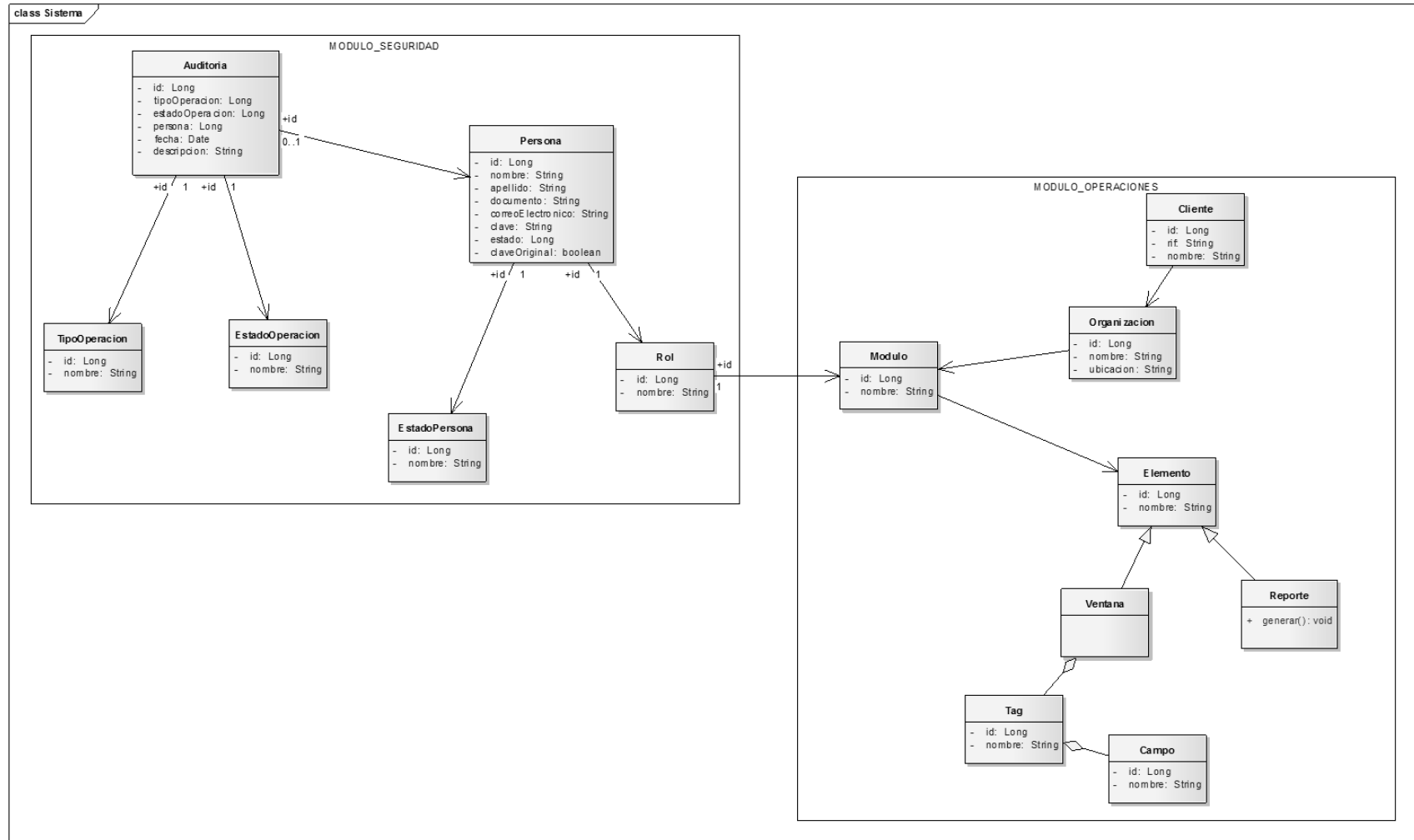


Figura 17: Módulo de Seguridad y Módulo de Operaciones

Fuente: Elaboración Propia

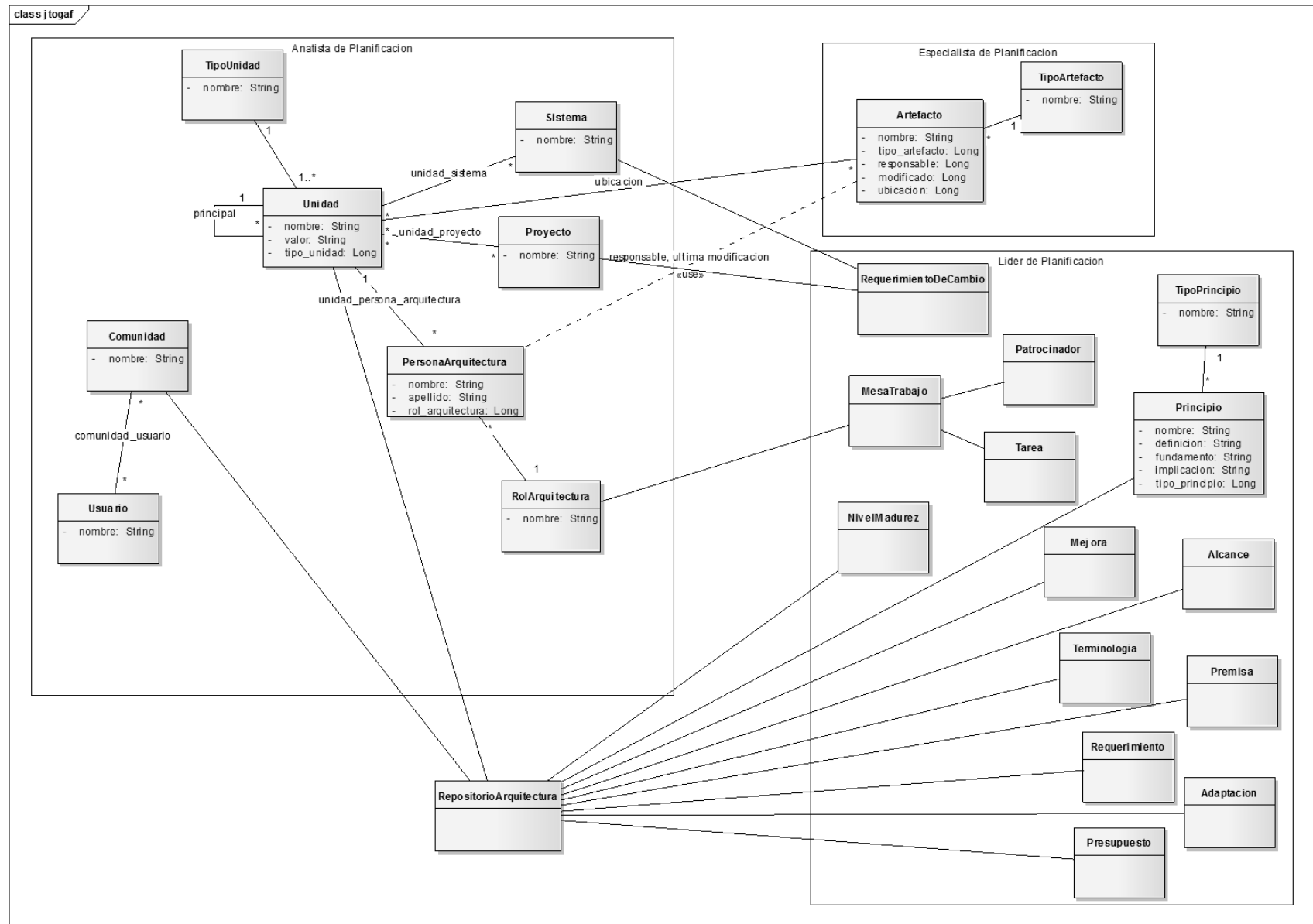


Figura 18: Módulos de Analista, Especialista y Líder de Planificación

Fuente: Elaboración Propia

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

Tomando como referencia los objetivos específicos de este trabajo: (1) “*Analizar los cuatro Marcos de Arquitectura Empresarial más usados actualmente.*”: se ha cubierto una amplia introducción al campo de la Arquitectura Empresarial. Dos de los cuatro marcos principales (*OEAF* y *FEA*) han experimentado grandes cambios en los últimos dos años. Estas metodologías son bastante diferentes unas de otras, tanto en los objetivos como en el enfoque.

Uno de los objetivos más importantes de cualquier Arquitectura Empresarial es unir el modelo de negocios y la tecnología, por lo que ambas están trabajando con eficacia hacia los mismos objetivos. Las metodologías no resuelven los problemas de la *gente*, sólo proporcionan un marco en el que esos problemas pueden ser resueltos.

(2) “*Seleccionar un Marco de Arquitectura, de acuerdo al análisis realizado, para ser desarrollado.*”: luego de haber estudiado y analizado los cuatro Marcos de Arquitectura Empresarial más usados a la presentación de este Trabajo Especial de Grado, surge la necesidad de construir una herramienta para estandarizar el proceso de gestión de la Arquitectura y lograr una mejor administración de los entornos de Tecnologías de la Información.

De acuerdo al estudio y la comparación de las Arquitecturas (*OEAF*, *FEA*, *TOGAF*, Zachman) realizada en este trabajo, se concluye que la más adecuada es *TOGAF*, ya que busca ser una aproximación al desarrollo de arquitecturas y al gobierno de manera “Ágil”. No prescribe los modelos que deberían ser usados para representar la arquitectura, guía el proceso cuando esta se crea. Debido a su escalabilidad, puede ser usado por organizaciones de gobierno, empresas pequeñas, medianas o grandes.

(3) *“Definir el alcance general del sistema a desarrollar, mediante Historias de Usuario, plasmándolo en un Plan de Entregas.”*: para llevar a cabo el desarrollo del “Sistema de Información para la Automatización de la Arquitectura” se eligió la metodología de desarrollo ágil XP. A través de sus etapas se estableció un alcance general del sistema a desarrollar, mediante las Historias de Usuario que proporcionó el cliente caso de uso CAPAS360 C.A., dicho alcance quedó plasmado en un Plan de Entregas. CAPAS360 C.A.

(4) *“Desarrollar los Casos de Prueba del sistema dirigidos desde las Historias de Usuario.”*: el cliente también suministró un conjunto de Casos de Prueba de Integración para cada Historia de Usuario.

(5) *“Desarrollar el sistema basándose en un Plan de Iteraciones, dividido en Tareas, que sigue el cronograma establecido en el Plan de Entregas.”*: finalmente el sistema se implementó siguiendo un Plan de Iteraciones, el esfuerzo de programación fue dividido en Tareas.

La aplicación provee un sistema de automatización de gran importancia para CAPAS360 C.A. y la comunidad de software libre ya que a través de la misma podrán gestionar el desarrollo inicial de una Arquitectura Empresarial tal como describe el alcance de este Trabajo Especial de Grado.

Las siguientes son una serie de recomendaciones que bien podrían ser tomadas en cuenta con el fin de optimizar y agregar nuevas funcionalidades al Sistema de Información para la Automatización de una Arquitectura Empresarial descrito en el presente trabajo:

- Desacoplar los módulos a través de una programación por componentes distribuidos usando servicios REST protegidos con OAUTH2 para hacer el sistema escalable, distribuido y menos dependiente de una tecnología.

- Usar CriteriaQuery en vez de JPQL para un uso total del paradigma orientado a objetos en la persistencia de datos a fin de hacer posible el cambio o manejo de distintas bases de datos sin problemas de portabilidad.
- Usar tablas de históricos que reflejen el cambio de estado en los objetos.
- Cambiar los servicios de persistencia síncronos a asíncronos mediante colas de persistencias que avisen el resultado de las operaciones, para liberar al usuario de la espera hasta el fin de la operación.
- Crear un modelo Lenguaje a fin de manejar los mensajes IGU desde la base de datos y así evitar la pérdida de archivos de propiedades y evitar tener que reiniciar el servidor cada vez que se actualizan los mismos.

Como trabajo futuro se recomienda la implementación e integración del resto de las Fases del Método de Desarrollo de la Arquitectura de TOGAF.

REFERENCIAS BIBLIOGRÁFICAS

- Agile Spain/ Germinus. (2005). Metodologías Ágiles: la ventaja competitiva de estar preparado para tomar decisiones lo más tarde posible y cambiarlas en cualquier momento. Pérez, J.
- Ajax un nuevo acercamiento a aplicaciones web. Disponible en (Fecha: Agosto 2012) <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>.
- AJAX. Disponible en (Fecha: Agosto 2012) <http://developer.mozilla.org/es/docs/AJAX>
- Arias, F. (2006). El Proyecto de Investigación: Introducción a la metodología científica. (Quita edición). Caracas: Episteme.
- Beck, K. (2000). Extreme Programming Explained. Embrace Change. Addison Wesley.
- Bertalanffy, L. (2006). Teoría general de los sistemas: fundamentos, desarrollo, aplicaciones. México: Fondo de Cultura Económica.
- Booch G., Rumbaugh J. y Jacobson I. (1999). El Lenguaje Unificado de Modelado. Madrid: Addison Wesley.
- CSS. Disponible en (Fecha: Agosto 2012) <http://developer.mozilla.org/es/docs/CSS>

- Desarrollo de aplicaciones web con JSP y XML. Disponible en (Fecha: Agosto 2012) <http://programacion.com/java/tutorial/jspxml>.
- FEA Consolidated Reference Model Document Version 2.1, Diciembre de 2006, published by the Federal Enterprise Architecture Program Management Office, Office of Management of Budget.
- García, F. y Bravo, C. (2008). Metodologías de Desarrollo de Software.
- Gutiérrez, G. (1984). Metodología de las Ciencias Sociales I. Universidad Nacional Autónoma de México.
- Introducción a extreme programming. Disponible en (Fecha: Agosto 2012) <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>.
- Introducción a XML. Disponible en (Fecha: Agosto 2012) <http://quark.fe.up.pt/cursoxml/curso.pdf>.
- Jacobson, I., Booch, G. y Rumbaugh, J. (2000). El Proceso Unificado de Desarrollo de Software. Madrid: Addison Wesley.
- Johansen, O. (2004). Introducción a la teoría general de sistemas. Ciudad de México: Limusa.
- MDSN, A Comparison of the Top Four Enterprise-Architecture Methodologies. Disponible en (Fecha: Marzo 2012) <http://msdn.microsoft.com/en-us/library/bb466232.aspx>

- MySQL 5.1 Reference Manual Disponible en (Fecha: Agosto 2012) www.MySQL.com
- Que es Ajax. Disponible en (Fecha: Agosto 2012) <http://es.wikipedia.org/wiki/AJAX>.
- The Agile UnifiedProcess. Disponible en (Fecha: Agosto 2012) <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.
- The Open Group Architecture Framework (TOGAF) Manual. Disponible en (Fecha: Agosto 2012) <http://www.thelonius.com.au/TOGAF-Downloads/index.htm>.
- The Zachman Framework Evolution. Disponible en (Fecha: Marzo 2012) <http://www.zachmaninternational.com/index.php/eaarticles/100-the-zachman-framework-evolution>
- Universidad Politécnica de Valencia. (2003) Metodologías Ágiles en el Desarrollo de Software. Valencia: Canós, J., Letelier, P. y Penadés, M.
- Universidad Técnica Particular de Loja. (2002). Metodologías Tradicionales vs. Metodologías Ágiles. Figueroa, R., Solis, C. y Cabrera, A.
- UPM, Ingeniería SW para Sistemas Disponible en: (Fecha: Agosto 2012) http://www-lsi.die.upm.es/~carreras/ISSE/programacion_extrema_2.x2.pdf.
- Wake, W.C. (2002). Extreme Programming Explored. Addison-Wesley.
- XHTML. Disponible en (Fecha: Agosto 2012) <http://es.wikipedia.org/wiki/XHTML>.

GLOSARIO

1.1 **Actividad**

Una tarea o conjunto de tareas que soportan las funciones de una organización.

1.2 **Actor**

Una persona, organización o sistema que tiene un rol que inicia o interactúa con las actividades.

1.3 **Ambiente**

Se refiere al área de sucesos y condiciones que influyen sobre el comportamiento de un sistema.

1.4 **Atributo**

Se entiende por atributo, las características y propiedades estructurales o funcionales que identifican las partes o componentes de un sistema.

1.5 **Arquitectura empresarial**

Es la estrategia de negocios, gestión, organización y la información clave de los procesos de negocio, así como la interacción entre estos conceptos.

1.6 **Brecha**

Una expresión de la diferencia entre dos estados.

1.7 **Caja negra**

La caja negra se utiliza para representar a los sistemas cuando no sabemos qué elementos o cosas componen al sistema o proceso, pero sabemos que determinadas entradas les corresponden determinadas salidas y con ello poder inducir, presumiendo que a determinados estímulos, las variables funcionarían en cierto sentido.

1.8 **Capacidad**

Una habilidad que una organización, persona o sistema posee.

1.9 **Ciencia**

Conjunto de conocimientos verificables, sistemáticamente organizados y metodológicamente obtenidos, relativos a un determinado objeto de estudio o rama del saber.

1.10 **Complejidad**

Por un lado, indica la cantidad de elementos de un sistema (complejidad cuantitativa) y, por el otro, sus potenciales interacciones (conectividad) y el número de estados posibles que se producen a través de éstos (variedad, variabilidad).

1.11 **Componente**

Los componentes de un sistema son simplemente las partes identificables del mismo.

1.12 **Conglomerado**

Cuando la suma de las partes, componentes y atributos en un conjunto es igual al todo, estamos en presencia de una totalidad desprovista de sinergia, es decir, de un conglomerado.

1.13 **Conocimiento**

Proceso en el cual se relacionan el sujeto que conoce, que percibe mediante sus sentidos, y el objeto conocido o percibido.

1.14 **Dato**

Es la unidad de información que se obtiene durante la ejecución de una investigación.

1.15 **Documento**

Es el soporte material o *formato digital* en el que se registra y conserva una información.

1.16 **Elemento**

Se entiende por elemento de un sistema, las partes o componentes que lo constituyen.

1.17 **Entrada**

Se denomina entrada a la importación de los recursos (energía, materia, información) que se requieren para dar inicio al ciclo de actividades del sistema.

1.18 **Entropía**

La entropía es el movimiento de un sistema hacia un desgaste, desorden o discrepancia totales.

1.19 **Equifinalidad**

Se refiere al hecho que un sistema vivo a partir de distintas condiciones iniciales y por distintos caminos llega a un mismo estado final.

1.20 **Estructura**

La estructura de un sistema es el conjunto de relaciones entre los objetos y atributos de los objetos de un sistema.

1.21 **Fuente**

Es todo lo que suministra datos o información.

1.22 **Función**

Se denomina función la salida de un sistema que está dirigido a la mantención del sistema mayor en el que se encuentra inscrito.

1.23 **Interfaz**

La interfaz es una conexión entre dos sistemas, la región de contacto.

1.24 **Investigación científica**

Proceso metódico y sistemático dirigido a la solución de problemas o preguntas científicas, mediante la producción de nuevos conocimientos, los cuales constituyen la solución o respuesta a tales interrogantes.

1.25 **Metadato**

Datos sobre los datos, de cualquier tipo en cualquier medio, que describe las características de una entidad.

1.26 **Método**

Vía o camino que se utiliza para llegar a un fin o para lograr un objetivo.

1.27 **Método científico**

Es una abstracción de las actividades que los investigadores realizan, concentrando la atención exclusivamente en los procesos de adquisición del conocimiento, desentendiéndose del contenido particular de los resultados obtenidos, salvo por el hecho de que sean válidos. En cierto modo, el método es un camino que se va haciendo, o completando al menos, cuando se recorre específicamente al realizar cada investigación.

1.28 **Metodología**

Se transforma en una disciplina que estudia, analiza, promueve y depura el método, mismo que se va multiplicando y particularizando de conformidad con las ramas de las disciplinas científicas existentes. Cuando un método demuestra ser eficaz, o requiere que se le hagan algunas consideraciones especiales para apoyar el desarrollo de una disciplina en particular, se gesta una discusión sobre él; a estas consideraciones, a este tratamiento sobre el método suele llamársele discusión metodológica. En el dominio de la metodología, la actividad filosófica consiste en elaborar las teorías de los diversos métodos empleados en la investigación científica, comprendiendo su fundamentación, sus leyes generales, sus teoremas específicos, sus operaciones y sus reglas de cálculo.

1.29 **Modelo**

Una representación de un tema de interés. Un modelo proporciona una escala más pequeña, simplificada, y/o representación abstracta de la realidad.

1.30 **Negentropía**

Se refiere a la energía que el sistema importa del ambiente para mantener su organización y sobrevivir.

1.31 **Plataforma**

Una combinación de productos de infraestructura de tecnología y componentes que establece los requisitos para alojar el software de una aplicación.

1.32 **Plataforma de Aplicaciones**

La colección de componentes de tecnología de hardware y software que proporcionan los servicios utilizados para apoyar a las aplicaciones.

1.33 **Plataforma de servicios**

Capacidad técnica necesaria para proporcionar la infraestructura que permite la ejecución de aplicaciones.

1.34 **Restricción**

Un factor externo que impide que una organización alcance determinados enfoques para lograr sus objetivos.

1.35 **Retroalimentación**

Son los procesos mediante los cuales un sistema abierto recoge información sobre los efectos de sus decisiones internas en el medio, información que actúa sobre las decisiones (acciones) sucesivas.

1.36 **Retroalimentación negativa**

Este concepto está asociado a los procesos homeostáticos o de autorregulación. Los sistemas con retroalimentación negativa se caracterizan por la mantención de determinados objetivos.

1.37 **Retroalimentación positiva**

Indica una cadena cerrada de relaciones causales en donde la variación de uno de sus componentes se propaga en otros componentes del sistema, reforzando la variación inicial y propiciando un comportamiento sistémico caracterizado por un autorreforzamiento de las variaciones.

1.38 **Sinergia**

Todo sistema es sinérgico en tanto el examen de sus partes en forma aislada no puede explicar o predecir su comportamiento.

1.39 **Sistema**

Es un conjunto de entidades caracterizadas por ciertos atributos, que tienen relaciones entre sí y están localizadas en un cierto ambiente, de acuerdo con un cierto objetivo.

ANEXOS

Tabla 13: Historia de Usuario 2
Fuente: Elaboración propia

Historia de Usuario	
Número: 2	Usuario: Analista de Planificación
Nombre historia: Identificación de las Unidades de Negocio secundarias.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: José Osuna/Miguel Bautista	
Descripción: Se crea cada unidad de negocio secundaria a través de un formulario, seleccionando la unidad principal a la cual pertenece, ingresando nombre de la unidad de negocio, lista de sistemas y proyectos de dicha unidad y nombre del gerente o jefe encargado.	
Observaciones:	

Tabla 14: Historia de Usuario 3
Fuente: Elaboración propia

Historia de Usuario	
Número: 3	Usuario: Analista de Planificación
Nombre historia: Identificación de las Unidades de Negocio extendidas.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: José Osuna/Miguel Bautista	
Descripción: Se crea cada unidad de negocio extendida a través de un formulario, ingresando nombre de la unidad de negocio.	
Observaciones:	

Tabla 15: Historia de Usuario 4
Fuente: Elaboración propia

Historia de Usuario	
Número: 4	Usuario: Analista de Planificación
Nombre historia: Identificación de las Comunidades involucradas.	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Osuna/Miguel Bautista	
Descripción: Se crea cada comunidad involucrada a través de un formulario que permita ingresar el nombre de la comunidad y una lista de usuario involucrados en dicha comunidad.	
Observaciones:	

Tabla 16: Historia de Usuario 5
Fuente: Elaboración propia

Historia de Usuario	
Número: 5	Usuario: Especialista de Planificación
Nombre historia: Identificación de la Gobernabilidad requerida.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: José Osuna/Miguel Bautista	
Descripción: a través de un formulario asignarle a cada unidad de negocio: sus roles, autoridad de la unidad, competencias de sus líderes, responsabilidad de la unidad, responsabilidades de los jefes, que se requiere para el desarrollo de la arquitectura.	
Observaciones:	

Tabla 17: Historia de Usuario 6
Fuente: Elaboración propia

Historia de Usuario	
Número: 6	Usuario: Especialista de Planificación
Nombre historia: Administración y Gobernabilidad de los artefactos arquitectónicos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: José Osuna/Miguel Bautista	
Descripción: Crear los distintos artefactos arquitectónicos de la organización a través de un formulario que permita ingresar el nombre del artefacto, su tipo y sus características de gobierno tales como: una descripción acerca de su responsable, quien lo modificó, donde está ubicado.	
Observaciones:	

Tabla 18: Historia de Usuario 7
Fuente: Elaboración propia

Historia de Usuario	
Número: 7	Usuario: Especialista de Planificación
Nombre historia: Revisión de la Gobernabilidad existente.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: José Osuna/Miguel Bautista	
Descripción: a través de un formulario asignarle a cada unidad de negocio: sus roles, autoridad de la unidad, competencias de sus líderes, responsabilidad de la unidad, responsabilidades de los jefes, existentes en la organización.	
Observaciones:	

Tabla 19: Historia de Usuario 8
Fuente: Elaboración propia

Historia de Usuario	
Número: 8	Usuario: Líder de Planificación
Nombre historia: Introducción de los puntos de mejora de la Arquitectura.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: José Osuna/Miguel Bautista	
Descripción: En un campo de texto introducir cuál será un punto de mejora de la arquitectura e introducir en un campo de área los impactos probables de cada punto de mejora, seguir hasta que los puntos a mejorar hayan sido cubiertos.	
Observaciones:	

Tabla 20: Historia de Usuario 9
Fuente: Elaboración propia

Historia de Usuario	
Número: 9	Usuario: Líder de Planificación
Nombre historia: Generación de un estudio del Nivel de Madurez Arquitectónica del Negocio.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: José Osuna/Miguel Bautista	
Descripción: A través de un formulario generar una serie de preguntas que ayuden a establecer el nivel de madurez arquitectónico del negocio e indicar el mismo.	
Observaciones:	

Tabla 21: Historia de Usuario 10
Fuente: Elaboración propia

Historia de Usuario	
Número: 10	Usuario: Especialista de Planificación
Nombre historia: Asignación de roles y responsabilidades principales de la Gestión y Gobierno de la Arquitectura Empresarial.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: José Osuna/Miguel Bautista	
Descripción: crear una lista de roles a través de un formulario que permita ingresar el nombre del rol, las responsabilidades, las unidades a donde pertenece dicho rol y las relaciones con los distintos artefactos arquitectónicos de la organización, además de los usuarios de dicho.	
Observaciones:	

Tabla 22: Historia de Usuario 11
Fuente: Elaboración propia

Historia de Usuario	
Número: 11	Usuario: Líder de Planificación
Nombre historia: Introducción de los requerimientos para los cambios en los proyectos y programas de negocio existentes.	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: José Osuna/Miguel Bautista	
Descripción: a través de un formulario, listar los proyectos y programas de negocios e ingresar los requerimientos para los cambios del mismo.	
Observaciones:	

Tabla 23: Historia de Usuario 12
Fuente: Elaboración propia

Historia de Usuario	
Número: 12	Usuario: Líder de Planificación
Nombre historia: Introducción del alcance del desarrollo de la nueva Arquitectura Empresarial.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: José Osuna/Miguel Bautista	
Descripción: A través de un campo de área introducir el texto del alcance de la nueva arquitectura empresarial.	
Observaciones:	

Tabla 24: Historia de Usuario 13
Fuente: Elaboración propia

Historia de Usuario	
Número: 13	Usuario: Líder de Planificación
Nombre historia: Introducción de las premisas y requerimientos sobre el desarrollo de la nueva Arquitectura Empresarial.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: José Osuna/Miguel Bautista	
Descripción: A través de un campo de área introducir el texto de las premisas y requerimientos sobre el desarrollo de la nueva Arquitectura Empresarial.	
Observaciones:	

Tabla 25: Historia de Usuario 14
Fuente: Elaboración propia

Historia de Usuario	
Número: 14	Usuario: Líder de Planificación
Nombre historia: Definición de la relación entre los patrocinadores y las mesas de trabajo para el desarrollo de la nueva Arquitectura Empresarial.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: José Osuna/Miguel Bautista	
Descripción: A través de un formulario crear a cada patrocinador de la nueva arquitectura empresarial. Crear las mesas de trabajo donde se relacionan los patrocinadores con los roles. Marcar la relación entre el patrocinador y los roles en una mesa de trabajo.	
Observaciones:	

Tabla 26: Historia de Usuario 15
Fuente: Elaboración propia

Historia de Usuario	
Número: 15	Usuario: Líder de Planificación
Nombre historia: Introducción de los requerimientos de presupuesto.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: José Osuna/Miguel Bautista	
Descripción: Ingresar a través de un formulario un presupuesto que permita determinar, tiempo, dinero y recursos humanos para emprender el desarrollo de la arquitectura.	
Observaciones:	

Tabla 27: Historia de Usuario 16
Fuente: Elaboración propia

Historia de Usuario	
Número: 16	Usuario: Líder de Planificación
Nombre historia: Introducción de los Principios de la Arquitectura.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: José Osuna/Miguel Bautista	
Descripción: Crear a través de un formulario cada principio ingresando Nombre, definición, fundamento e implicaciones y seleccionando el tipo de principio (Negocio, Datos, Aplicaciones y Tecnología).	
Observaciones:	

Tabla 28: Historia de Usuario 17
Fuente: Elaboración propia

Historia de Usuario	
Número: 17	Usuario: Líder de Planificación
Nombre historia: Introducción de lo que la adaptación necesita.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: José Osuna/Miguel Bautista	
Descripción: A través de un campo de área introducir lo que la adaptación necesita.	
Observaciones:	

Tabla 29: Historia de Usuario 18
Fuente: Elaboración propia

Historia de Usuario	
Número: 18	Usuario: Líder de Planificación
Nombre historia: Producción de un conjunto de terminologías acordadas para la descripción de contenido arquitectónico.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: José Osuna/Miguel Bautista	
Descripción: Crear cada terminología a través de un formulario que te permita ingresar el nombre y la definición de la terminología.	
Observaciones:	

Tabla 30: Historia de Usuario 19
Fuente: Elaboración propia

Historia de Usuario	
Número: 19	Usuario: Líder de Planificación
Nombre historia: Retiro de las tareas que ya están realizadas en otras partes de la organización.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: José Osuna/Miguel Bautista	
Descripción: A través de un formulario agregar un conjunto de tareas para cada mesa de trabajo y marcar aquellas que han sido realizadas.	
Observaciones:	

Tabla 31: Historia de Usuario 20
Fuente: Elaboración propia

Historia de Usuario	
Número: 20	Usuario: Líder de Planificación
Nombre historia: Introducción de las tareas específicas, tales como puntos de control específicos.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: José Osuna/Miguel Bautista	
Descripción: A través de un formulario agregar un conjunto de tareas de puntos de control, tales como: avance lógico, inversión y gasto, RRHH, tiempo, para cada mesa de trabajo.	
Observaciones:	

Tabla 32: Historia de Usuario 21
Fuente: Elaboración propia

Historia de Usuario	
Número: 21	Usuario: Líder de Planificación
Nombre historia: Alineación de los procesos para entornos de procesos externos y puntos de contacto	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: José Osuna/Miguel Bautista	
Descripción: Introducir, a través de un formulario, los procesos internos de la organización y seleccionar o describir la relación con procesos externos y puntos de contacto.	
Observaciones:	

Tabla 37: Caso de Prueba de Aceptación CA1-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA1-2	Historia de Usuario (Nº y Nombre): 1
Nombre: Introducción correcta de unidad de negocio principal	
Descripción: El analista una vez autenticado en el sistema seleccionará la opción del panel “Unidades Principales” y hará clic en el botón “Nueva Unidad de Negocios”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El analista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El analista introducirá su identificador y contraseña para autenticarse. - Del panel “Unidades Principales” y hará clic en el botón “Nueva Unidad de Negocios” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que la unidad ha sido creada correctamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerá la unidad de negocios en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 38: Caso de Prueba de Aceptación CA2-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA2-1	Historia de Usuario (Nº y Nombre): 2
Nombre: Introducción de unidad de negocio secundaria con errores	
Descripción: El analista una vez autenticado en el sistema seleccionará la opción del menú “Unidades de Negocio” – “Secundaria”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El analista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El analista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Unidades de Negocio” – “Secundaria” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si ocurre algún error se mostrará un mensaje “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: Las unidades de negocio introducidas de forma incorrecta no son almacenadas en la base de datos mientras que las correctas si son almacenadas.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 39: Caso de Prueba de Aceptación CA2-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA2-2	Historia de Usuario (Nº y Nombre): 2
Nombre: Introducción correcta de unidad de negocio secundaria	
Descripción: El analista una vez autenticado en el sistema seleccionará la opción del menú “Unidades de Negocio” – “Secundaria”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El analista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El analista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Unidades de Negocio” – “Secundaria” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que la unidad ha sido creada correctamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerá la unidad de negocios en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 40: Caso de Prueba de Aceptación CA3-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA3-1	Historia de Usuario (Nº y Nombre): 3
Nombre: Introducción de unidad de negocio extendida con errores	
Descripción: El analista una vez autenticado en el sistema seleccionará la opción del menú “Unidades de Negocio” – “Extendida”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El analista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El analista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Unidades de Negocio” – “Extendida” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará un mensaje diciendo “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: Las unidades de negocio introducidas de forma incorrecta no son almacenadas en la base de datos mientras que las correctas si son almacenadas.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 41: Caso de Prueba de Aceptación CA3-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA3-2	Historia de Usuario (Nº y Nombre): 3
Nombre: Introducción correcta de unidad de negocio extendida	
Descripción: El analista una vez autenticado en el sistema seleccionará la opción del menú “Unidades de Negocio” – “Extendida”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El analista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El analista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Unidades de Negocio” – “Extendida” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que la unidad ha sido creada correctamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerá la unidad de negocios en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 42: Caso de Prueba de Aceptación CA4-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA4-1	Historia de Usuario (Nº y Nombre): 4
Nombre: Introducción de las comunidades involucradas con errores	
Descripción: El analista una vez autenticado en el sistema seleccionará la opción del menú “Comunidades Involucradas”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingreso de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El analista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El analista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Comunidades Involucradas” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará un mensaje diciendo “Ingreso de forma correcta los datos del campo XXX” 	
Resultado Esperado: Las comunidades involucradas introducidas de forma incorrecta no son almacenadas en la base de datos mientras que las correctas si son almacenadas.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 43: Caso de Prueba de Aceptación CA4-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA4-2	Historia de Usuario (Nº y Nombre): 4
Nombre: Introducción correcta de las comunidades involucradas	
Descripción: El analista una vez autenticado en el sistema seleccionará la opción del menú “Comunidades Involucradas”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El analista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El analista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Comunidades Involucradas” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que la comunidad ha sido creada correctamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerá la comunidad en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 44: Caso de Prueba de Aceptación CA5-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA5-1	Historia de Usuario (Nº y Nombre): 5
Nombre: Introducción de la Gobernabilidad requerida con errores	
Descripción: El especialista una vez autenticado en el sistema seleccionará la opción del menú “Gobernabilidad” – “Gobernabilidad Requerida”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El especialista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El especialista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Gobernabilidad” – “Gobernabilidad Requerida” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: La Gobernabilidad Requerida introducida de forma incorrecta no es almacenada en la base de datos mientras que la correcta si es almacenada.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 45: Caso de Prueba de Aceptación CA5-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA5-2	Historia de Usuario (Nº y Nombre): 5
Nombre: Introducción correcta la Gobernabilidad requerida	
Descripción: El especialista una vez autenticado en el sistema seleccionará la opción del menú “Gobernabilidad” – “Gobernabilidad Requerida”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El especialista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El especialista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Gobernabilidad” – “Gobernabilidad Requerida” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que la Gobernabilidad ha sido actualizada correctamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerá la Gobernabilidad en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 46: Caso de Prueba de Aceptación CA6-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA6-1	Historia de Usuario (Nº y Nombre): 6
Nombre: Introducción de los artefactos arquitectónicos con errores	
Descripción: El especialista una vez autenticado en el sistema seleccionará la opción del panel “Artefactos” y hará clic en el botón “Nuevo Artefacto”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingreso de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El especialista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El especialista introducirá su identificador y contraseña para autenticarse. - Del panel seleccionará “Artefactos” y hará clic en el botón “Nuevo Artefacto” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingreso de forma correcta los datos del campo XXX” 	
Resultado Esperado: Los Artefactos introducidos de forma incorrecta no son almacenados en la base de datos mientras que los correctos si son almacenados.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 47: Caso de Prueba de Aceptación CA6-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA6-2	Historia de Usuario (Nº y Nombre): 6
Nombre: Introducción correcta de los artefactos arquitectónicos	
Descripción: El especialista una vez autenticado en el sistema seleccionará la opción del panel “Artefactos” y hará clic en el botón “Nuevo Artefacto”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El especialista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El especialista introducirá su identificador y contraseña para autenticarse. - Del panel seleccionará “Artefactos” y hará clic en el botón “Nuevo Artefacto” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que el artefacto ha sido creado satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerá el artefacto en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 48: Caso de Prueba de Aceptación CA7-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA7-1	Historia de Usuario (Nº y Nombre): 7
Nombre: Introducción de la Gobernabilidad existente con errores	
Descripción: El especialista una vez autenticado en el sistema seleccionará la opción del menú “Gobernabilidad” – “Gobernabilidad Existente”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El Líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El especialista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Gobernabilidad” – “Gobernabilidad Existente” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: La Gobernabilidad introducida de forma incorrecta no es almacenada en la base de datos mientras que la correcta si es almacenada.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 49: Caso de Prueba de Aceptación CA7-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA7-2	Historia de Usuario (Nº y Nombre): 7
Nombre: Introducción correcta de la Gobernabilidad existente	
Descripción: El especialista una vez autenticado en el sistema seleccionará la opción del menú “Gobernabilidad” – “Gobernabilidad Existente”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El Líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El especialista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Gobernabilidad” – “Gobernabilidad Existente” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que la Gobernabilidad ha sido creada satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerá la Gobernabilidad en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 50: Caso de Prueba de Aceptación CA8-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA8-1	Historia de Usuario (Nº y Nombre): 8
Nombre: Introducción de los puntos de mejora de la arquitectura con errores	
Descripción: El Líder de Planificación una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Puntos de Mejora”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El Líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El Líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Puntos de Mejora” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: Los puntos de mejora introducidos de forma incorrecta no son almacenados en la base de datos mientras que los correctos si son almacenados.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 51: Caso de Prueba de Aceptación CA8-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA8-2	Historia de Usuario (Nº y Nombre): 8
Nombre: Introducción correcta de los puntos de mejora de la arquitectura	
Descripción: El Líder de Planificación una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Puntos de Mejora”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El Líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El Líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Puntos de Mejora” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que los puntos de mejora han sido creados satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerán los puntos de mejora en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 52: Caso de Prueba de Aceptación CA9-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA9-1	Historia de Usuario (Nº y Nombre): 9
Nombre: Introducción del nivel de madures arquitectónica del negocio con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “CMMI”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingreso de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El especialista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “CMMI” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingreso de forma correcta los datos del campo XXX” 	
Resultado Esperado: El estudio del nivel de madurez de la arquitectura introducido de forma incorrecta no es almacenado en la base de datos mientras que el correcto sí.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 53: Caso de Prueba de Aceptación CA9-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA9-2	Historia de Usuario (Nº y Nombre): 9
Nombre: Introducción correcta del nivel de madures arquitectónica del negocio	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “CMMI”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El especialista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “CMMI” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que el nivel de madurez han sido creado satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerá el estudio en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 54: Caso de Prueba de Aceptación CA10-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA10-1	Historia de Usuario (Nº y Nombre): 10
Nombre: Introducción de los roles y responsabilidades de gestión y gobierno con errores	
Descripción: El especialista una vez autenticado en el sistema seleccionará la opción del menú “Gobernabilidad” – “Roles”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El especialista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El especialista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Gobernabilidad” – “Roles” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: El Rol introducido de forma incorrecta no es almacenado en la base de datos mientras que el correcto si es almacenado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 55: Caso de Prueba de Aceptación CA10-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA10-2	Historia de Usuario (Nº y Nombre): 10
Nombre: Introducción correcta de los roles y responsabilidades de gestión y gobierno	
Descripción: El especialista una vez autenticado en el sistema seleccionará la opción del menú “Gobernabilidad” – “Roles”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El especialista debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El especialista introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Gobernabilidad” – “Roles” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que el rol ha sido creado satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerá el rol en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 56: Caso de Prueba de Aceptación CA11-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA11-1	Historia de Usuario (Nº y Nombre): 11
Nombre: Introducción de los requerimientos para el cambio con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Requerimientos de Cambio”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Requerimientos de Cambio” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará un mensaje diciendo “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: Los requerimientos de cambio introducidos de forma incorrecta no son almacenados en la base de datos mientras que los correctos si son almacenados.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 57: Caso de Prueba de Aceptación CA11-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA11-2	Historia de Usuario (Nº y Nombre): 11
Nombre: Introducción correcta de los requerimientos para el cambio	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Requerimientos de Cambio”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Requerimientos de Cambio” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que los requerimientos de cambio han sido creados satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerán los requerimientos en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 58: Caso de Prueba de Aceptación CA12-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA12-1	Historia de Usuario (Nº y Nombre): 12
Nombre: Introducción del alcance de la arquitectura con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Alcance”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Alcance” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: El alcance introducido de forma incorrecta no es almacenado en la base de datos mientras que el correcto si es almacenado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 59: Caso de Prueba de Aceptación CA12-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA12-2	Historia de Usuario (Nº y Nombre): 12
Nombre: Introducción correcta del alcance de la arquitectura	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Alcance”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Alcance” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que el alcance ha sido creado satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerá el alcance en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 60: Caso de Prueba de Aceptación CA13-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA13-1	Historia de Usuario (Nº y Nombre): 13
Nombre: Introducción de las premisas y requerimientos con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Premisas y Requerimientos”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Premisas y Requerimientos” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: Las premisas y requerimientos introducidos de forma incorrecta no son almacenados en la base de datos mientras que los correctos si son almacenados.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 61: Caso de Prueba de Aceptación CA13-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA13-2	Historia de Usuario (Nº y Nombre): 13
Nombre: Introducción correcta de las premisas y requerimientos	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Premisas y Requerimientos”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Premisas y Requerimientos” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que el alcance ha sido creado satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerán las premisas y requerimientos en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 62: Caso de Prueba de Aceptación CA14-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA14-1	Historia de Usuario (Nº y Nombre): 14
Nombre: Introducción de la relación patrocinadores-mesas de trabajo con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Patrocinadores y Mesas de Trabajo”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Patrocinadores y Mesas de Trabajo” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: Los patrocinadores y mesas de trabajo introducidos de forma incorrecta no son almacenados en la base de datos mientras que los correctos si son almacenados.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 63: Caso de Prueba de Aceptación CA14-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA14-2	Historia de Usuario (Nº y Nombre): 14
Nombre: Introducción correcta de la relación patrocinadores-mesas de trabajo	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Patrocinadores y Mesas de Trabajo”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Patrocinadores y Mesas de Trabajo” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que el alcance ha sido creado satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerán los patrocinadores y mesas de trabajo en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 64: Caso de Prueba de Aceptación CA15-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA15-1	Historia de Usuario (Nº y Nombre): 15
Nombre: Introducción de los requerimientos de presupuesto con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Presupuesto”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Presupuesto” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará un mensaje diciendo “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: El presupuesto introducido de forma incorrecta no es almacenado en la base de datos mientras que el correcto si es almacenado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 65: Caso de Prueba de Aceptación CA15-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA15-2	Historia de Usuario (Nº y Nombre): 15
Nombre: Introducción correcta de los requerimientos de presupuesto	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Presupuesto”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Presupuesto” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que el presupuesto ha sido creado satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto presupuesto en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 66: Caso de Prueba de Aceptación CA16-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA16-1	Historia de Usuario (Nº y Nombre): 16
Nombre: Introducción de los principios de la arquitectura con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del panel “Principios” y hará clic en el botón “Nuevo Principio”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del panel seleccionará “Principios” y hará clic en el botón “Nuevo Principio” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará un mensaje diciendo “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: El principio introducido de forma incorrecta no es almacenado en la base de datos mientras que el correcto si es almacenado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 67: Caso de Prueba de Aceptación CA16-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA16-2	Historia de Usuario (Nº y Nombre): 16
Nombre: Introducción correcta de los principios de la arquitectura	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del panel “Principios” y hará clic en el botón “Nuevo Principio”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del panel seleccionará “Principios” y hará clic en el botón “Nuevo Principio” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que el principio ha sido creado satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerá el principio en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 68: Caso de Prueba de Aceptación CA17-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA17-1	Historia de Usuario (Nº y Nombre): 17
Nombre: Introducción de lo que la adaptación necesita con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Adaptación y Necesidades”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Adaptación y Necesidades” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará un mensaje diciendo “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: Las necesidades introducidas de forma incorrecta no son almacenadas en la base de datos mientras que las correctas si son almacenadas.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 69: Caso de Prueba de Aceptación CA17-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA17-2	Historia de Usuario (Nº y Nombre): 17
Nombre: Introducción correcta de lo que la adaptación necesita	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Adaptación y Necesidades”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Adaptación y Necesidades” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que las necesidades han sido creadas satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerán las necesidades en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 70: Caso de Prueba de Aceptación CA18-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA18-1	Historia de Usuario (Nº y Nombre): 18
Nombre: Introducción de las terminologías acordadas con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Terminología”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Terminología” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: Las terminologías introducidas de forma incorrecta no son almacenadas en la base de datos mientras que las correctas si son almacenadas.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 71: Caso de Prueba de Aceptación CA18-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA18-2	Historia de Usuario (Nº y Nombre): 18
Nombre: Introducción correcta de las terminologías acordadas	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Terminología”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Terminología” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que las terminologías han sido creadas satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerán las terminologías en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 72: Caso de Prueba de Aceptación CA19-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA19-1	Historia de Usuario (Nº y Nombre): 19
Nombre: Introducción de las tareas realizadas con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Tareas Realizadas”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Tareas Realizadas” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: Las tareas introducidas de forma incorrecta no son almacenadas en la base de datos mientras que las correctas si son almacenadas.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 73: Caso de Prueba de Aceptación CA19-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA19-2	Historia de Usuario (Nº y Nombre): 19
Nombre: Introducción correcta de las tareas realizadas	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Tareas Realizadas”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Tareas Realizadas” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que las tareas han sido creadas satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerán las tareas en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 74: Caso de Prueba de Aceptación CA20-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA20-1	Historia de Usuario (Nº y Nombre): 20
Nombre: Introducción de las tareas específicas con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Tareas Específicas”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingresa de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Tareas Específicas” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará “Ingresa de forma correcta los datos del campo XXX” 	
Resultado Esperado: Las tareas introducidas de forma incorrecta no son almacenadas en la base de datos mientras que las correctas si son almacenadas.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 75: Caso de Prueba de Aceptación CA20-2
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA20-2	Historia de Usuario (Nº y Nombre): 20
Nombre: Introducción correcta de las tareas específicas	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Tareas Específicas”, se mostrará un formulario y una vez pulsado el botón “Guardar” y si no hay ningún error se avisará con un mensaje de introducción satisfactoria y se almacenará en la base de datos.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Tareas Específicas” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y si no hay errores se mostrará un mensaje indicando que las tareas han sido creadas satisfactoriamente. 	
Resultado Esperado: Si el procesado ha sido correcto aparecerán las tareas en la base de datos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 76: Caso de Prueba de Aceptación CA21-1
Fuente: Elaboración propia

Caso de Prueba de Aceptación	
Código: CA21-1	Historia de Usuario (Nº y Nombre): 21
Nombre: Introducción de la alineación de los procesos con errores	
Descripción: El líder una vez autenticado en el sistema seleccionará la opción del menú “Arquitectura” – “Alineación de Procesos”, se mostrará un formulario y una vez pulsado el botón “Guardar” se procederá a detectar errores (formato, campos vacíos) se avisará con un mensaje “Ingrese de forma correcta los datos del campo XXX”.	
Condiciones de Ejecución: El líder debe estar autenticado en el sistema.	
Entrada / Paso de Ejecución: <ul style="list-style-type: none"> - El líder introducirá su identificador y contraseña para autenticarse. - Del menú seleccionará “Arquitectura” – “Alineación de Procesos” - Procederá a llenar el formulario - Tras pulsar el botón de “Guardar” se procederá a validar los campos y en caso de que ocurra algún error se mostrará un mensaje diciendo “Ingrese de forma correcta los datos del campo XXX” 	
Resultado Esperado: Las alineaciones introducidas de forma incorrecta no son almacenadas en la base de datos mientras que las correctas si son almacenadas.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 79: Tarea 2, Historia 1
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 1
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 15/10/2012	Fecha fin: 16/10/2012
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 80: Tarea 3, Historia 1
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 1
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 17/10/2012	Fecha fin: 18/10/2012
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 81: Tarea 1, Historia 2
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 2
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 19/10/2012	Fecha fin: 20/10/2012
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 82: Tarea 2, Historia 2
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 2
Nombre tarea: Listado de las Unidades Principales desde la Base de Datos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 21/10/2012	Fecha fin: 21/10/2012
Programador responsable: Equipo XP	
<p>Descripción: Se muestra en la vista la lista de las Unidades Principales que actualmente se encuentra en la base de datos</p>	

Tabla 83: Tarea 3, Historia 2
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 2
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 22/10/2012	Fecha fin: 23/10/2012
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 84: Tarea 4, Historia 2
Fuente: Elaboración propia

Tarea	
Número tarea: 4	Número historia: 2
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 24/10/2012	Fecha fin: 26/10/2012
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 85: Tarea 1, Historia 3
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 3
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 27/10/2012	Fecha fin: 28/10/2012
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 86: Tarea 2, Historia 3
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 3
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 29/10/2012	Fecha fin: 31/10/2012
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 87: Tarea 3, Historia 3
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 3
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 01/11/2012	Fecha fin: 02/11/2012
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 88: Tarea 1, Historia 4
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 4
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 03/11/2012	Fecha fin: 05/11/2012
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 89: Tarea 2, Historia 4
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 4
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 06/11/2012	Fecha fin: 08/11/2012
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 90: Tarea 3, Historia 4
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 4
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 09/11/2012	Fecha fin: 11/11/2012
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 91: Tarea 1, Historia 5
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 5
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 12/11/2012	Fecha fin: 13/11/2012
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 92: Tarea 2, Historia 5
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 5
Nombre tarea: Listado de las Unidades de Negocio desde la Base de Datos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 14/11/2012	Fecha fin: 17/11/2012
Programador responsable: Equipo XP	
<p>Descripción: Se muestra en la vista la lista de las Unidades de Negocio que actualmente se encuentra en la base de datos</p>	

Tabla 93: Tarea 3, Historia 5
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 5
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 18/11/2012	Fecha fin: 24/11/2012
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 94: Tarea 4, Historia 5
Fuente: Elaboración propia

Tarea	
Número tarea: 4	Número historia: 5
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 25/11/2012	Fecha fin: 27/11/2012
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 95: Tarea 1, Historia 6
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 6
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 28/11/2012	Fecha fin: 29/11/2012
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 96: Tarea 2, Historia 6
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 6
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 30/11/2012	Fecha fin: 03/12/2012
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 97: Tarea 3, Historia 6
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 6
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 04/12/2012	Fecha fin: 14/12/2012
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 98: Tarea 1, Historia 7
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 7
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 15/12/2012	Fecha fin: 16/12/2012
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 99: Tarea 2, Historia 7
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 7
Nombre tarea: Listado de las Unidades de Negocio desde la Base de Datos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 17/12/2012	Fecha fin: 17/12/2012
Programador responsable: Equipo XP	
<p>Descripción: Se muestra en la vista la lista de las Unidades de Negocio que actualmente se encuentra en la base de datos</p>	

Tabla 100: Tarea 3, Historia 7
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 7
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 18/12/2012	Fecha fin: 18/12/2012
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 101: Tarea 4, Historia 7
Fuente: Elaboración propia

Tarea	
Número tarea: 4	Número historia: 7
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 19/12/2012	Fecha fin: 20/12/2012
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 102: Tarea 1, Historia 8
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 8
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 29/12/2012	Fecha fin: 29/12/2012
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 103: Tarea 2, Historia 8
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 8
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 30/12/2012	Fecha fin: 31/12/2012
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 104: Tarea 3, Historia 8
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 8
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 01/01/2013	Fecha fin: 02/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 105: Tarea 1, Historia 9
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 9
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 03/01/2013	Fecha fin: 04/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 106: Tarea 2, Historia 9
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 9
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 05/01/2013	Fecha fin: 06/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 107: Tarea 3, Historia 9
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 9
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 07/01/2013	Fecha fin: 08/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 108: Tarea 1, Historia 10
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 10
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 09/01/2013	Fecha fin: 10/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 109: Tarea 2, Historia 10
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 10
Nombre tarea: Listado de las Unidades de Negocio desde la Base de Datos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 11/01/2013	Fecha fin: 11/01/2013
Programador responsable: Equipo XP	
Descripción: Se muestra en la vista la lista de las Unidades de Negocio que actualmente se encuentra en la base de datos	

Tabla 110: Tarea 3, Historia 10
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 10
Nombre tarea: Listado de Roles desde la Base de Datos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 12/01/2013	Fecha fin: 13/01/2013
Programador responsable: Equipo XP	
Descripción: Se muestra en la vista la lista de Roles que actualmente se encuentra en la base de datos	

Tabla 111: Tarea 4, Historia 10
Fuente: Elaboración propia

Tarea	
Número tarea: 4	Número historia: 10
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 14/01/2013	Fecha fin: 15/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 112: Tarea 5, Historia 10
Fuente: Elaboración propia

Tarea	
Número tarea: 5	Número historia: 10
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 16/01/2013	Fecha fin: 17/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 113: Tarea 1, Historia 11
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 11
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 18/01/2013	Fecha fin: 21/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 114: Tarea 2, Historia 11
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 11
Nombre tarea: Listado de Programas y Proyectos desde la Base de Datos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 22/01/2013	Fecha fin: 22/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Se muestra en la vista la lista de que actualmente se encuentra en la base de datos</p>	

Tabla 115: Tarea 3, Historia 11
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 11
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 23/01/2013	Fecha fin: 25/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 116: Tarea 4, Historia 11
Fuente: Elaboración propia

Tarea	
Número tarea: 4	Número historia: 11
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 26/01/2013	Fecha fin: 30/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 117: Tarea 1, Historia 12
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 12
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 31/01/2013	Fecha fin: 01/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 118: Tarea 2, Historia 12
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 12
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 02/01/2013	Fecha fin: 02/01/2013
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 119: Tarea 3, Historia 12
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 12
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 03/01/2013	Fecha fin: 04/02/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 120: Tarea 1, Historia 13
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 13
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 05/02/2013	Fecha fin: 06/02/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 121: Tarea 2, Historia 13
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 13
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 07/02/2013	Fecha fin: 07/02/2013
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 122: Tarea 3, Historia 13
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 13
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 08/02/2013	Fecha fin: 10/02/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 123: Tarea 1, Historia 14
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 14
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 11/02/2013	Fecha fin: 11/02/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 124: Tarea 2, Historia 14
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 14
Nombre tarea: Listado de Roles desde la Base de Datos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 12/02/2013	Fecha fin: 13/02/2013
Programador responsable: Equipo XP	
<p>Descripción: Se muestra en la vista la lista Roles de que actualmente se encuentra en la base de datos</p>	

Tabla 125: Tarea 3, Historia 14
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 14
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 14/02/2013	Fecha fin: 14/02/2013
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 126: Tarea 4, Historia 14
Fuente: Elaboración propia

Tarea	
Número tarea: 4	Número historia: 14
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 15/02/2013	Fecha fin: 16/02/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 127: Tarea 1, Historia 15
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 15
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 02/03/2013	Fecha fin: 04/03/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 128: Tarea 2, Historia 15
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 15
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 05/03/2013	Fecha fin: 07/03/2013
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 129: Tarea 3, Historia 15
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 15
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 08/03/2013	Fecha fin: 12/03/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 130: Tarea 1, Historia 16
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 16
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 13/03/2013	Fecha fin: 13/03/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 131: Tarea 2, Historia 16
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 16
Nombre tarea: Listado de Tipo de Principio desde la Base de Datos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 14/03/2013	Fecha fin: 15/03/2013
Programador responsable: Equipo XP	
Descripción: Se muestra en la vista la lista de que actualmente se encuentra en la base de datos	

Tabla 132: Tarea 3, Historia 16
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 16
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 16/03/2013	Fecha fin: 18/03/2013
Programador responsable: Equipo XP	
Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.	

Tabla 133: Tarea 4, Historia 16
Fuente: Elaboración propia

Tarea	
Número tarea: 4	Número historia: 16
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 19/03/2013	Fecha fin: 20/03/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 134: Tarea 1, Historia 17
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 17
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 21/03/2013	Fecha fin: 22/03/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 135: Tarea 2, Historia 17
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 17
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 23/03/2013	Fecha fin: 23/03/2013
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 136: Tarea 3, Historia 17
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 17
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 24/03/2013	Fecha fin: 26/03/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 137: Tarea 1, Historia 18
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 18
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 27/03/2013	Fecha fin: 28/03/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 138: Tarea 2, Historia 18
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 18
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 29/03/2013	Fecha fin: 31/03/2013
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 139: Tarea 3, Historia 18
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 18
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 01/04/2013	Fecha fin: 04/04/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 140: Tarea 1, Historia 19
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 19
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 05/04/2013	Fecha fin: 09/04/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 141: Tarea 2, Historia 19
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 19
Nombre tarea: Listado de las Mesas de Trabajo desde la Base de Datos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 10/04/2013	Fecha fin: 12/04/2013
Programador responsable: Equipo XP	
Descripción: Se muestra en la vista la lista de que actualmente se encuentra en la base de datos	

Tabla 142: Tarea 3, Historia 19
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 19
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 13/04/2013	Fecha fin: 15/04/2013
Programador responsable: Equipo XP	
Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.	

Tabla 143: Tarea 4, Historia 19
Fuente: Elaboración propia

Tarea	
Número tarea: 4	Número historia: 19
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 16/04/2013	Fecha fin: 18/04/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 144: Tarea 1, Historia 20
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 20
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 19/04/2013	Fecha fin: 19/04/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 145: Tarea 2, Historia 20
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 20
Nombre tarea: Listado de las Mesas de Trabajo desde la Base de Datos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 20/04/2013	Fecha fin: 22/04/2013
Programador responsable: Equipo XP	
<p>Descripción: Se muestra en la vista la lista de las Mesas de Trabajo que actualmente se encuentra en la base de datos</p>	

Tabla 146: Tarea 3, Historia 20
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 20
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 23/04/2013	Fecha fin: 27/04/2013
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 147: Tarea 4, Historia 20
Fuente: Elaboración propia

Tarea	
Número tarea 4	Número historia: 20
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 28/04/2013	Fecha fin: 30/04/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 148: Tarea 1, Historia 21
Fuente: Elaboración propia

Tarea	
Número tarea: 1	Número historia: 21
Nombre tarea: Generar formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 01/05/2013	Fecha fin: 01/05/2013
Programador responsable: Equipo XP	
<p>Descripción: Se genera una ventana que muestre un formulario con los campos de datos requeridos para implementar la historia de usuario. Se habilitará un botón para procesar el registro de los datos suministrados en la base de datos.</p>	

Tabla 149: Tarea 2, Historia 21
Fuente: Elaboración propia

Tarea	
Número tarea: 2	Número historia: 21
Nombre tarea: Lectura y procesado del formulario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: 02/05/2013	Fecha fin: 02/05/2013
Programador responsable: Equipo XP	
<p>Descripción: Se leen los datos desde el formulario, se comprueba que los datos requeridos como obligatorios estén presentes y que el formato sea el correcto y en función de esto se insertan los datos en las tablas correspondientes en la base de datos.</p>	

Tabla 150: Tarea 3, Historia 21
Fuente: Elaboración propia

Tarea	
Número tarea: 3	Número historia: 21
Nombre tarea: Comprobación de resultados	
Tipo de tarea : Verificación	Puntos estimados:
Fecha inicio: 03/05/2013	Fecha fin: 05/05/2013
Programador responsable: Equipo XP	
<p>Descripción: Una vez leídos los datos suministrados a través del formulario se comprueba que se hace de forma correcta y que el proceso no falla debido a alguna restricción no satisfecha. Si el proceso falla en algún dato del formulario, ningún dato se inserta en la base de datos y se notifica al usuario.</p>	

Tabla 152: C.R.C. TipoUnidad
Fuente: Elaboración propia

TipoUnidad	
Responsabilidades	Colaboradores
Almacenar la clasificación de los Tipos de Unidades de Negocio de la Arquitectura Empresarial.	

Tabla 153: C.R.C. Sistema
Fuente: Elaboración propia

Sistema	
Responsabilidades	Colaboradores
Gestionar los Sistemas que posee cada Unidad de Negocios.	RequerimientoDeCambio

Tabla 154: C.R.C. Proyecto
Fuente: Elaboración propia

Proyecto	
Responsabilidades	Colaboradores
Gestionar los Proyectos que posee cada Unidad de Negocios.	RequerimientoDeCambio

Tabla 155: C.R.C. RequerimientoDeCambio
Fuente: Elaboración propia

RequerimientoDeCambio	
Responsabilidades	Colaboradores
Gestionar los requerimientos para el cambio de los Proyectos y/o Sistemas de las Unidades de Negocio.	

Tabla 156: C.R.C. Rol
Fuente: Elaboración propia

Rol	
Responsabilidades	Colaboradores
Gestionar los Roles de cada Persona perteneciente a cada Unidad de Negocio.	

Tabla 157: C.R.C. Persona
Fuente: Elaboración propia

Persona	
Responsabilidades	Colaboradores
Gestionar cada Persona perteneciente a cada Unidad de Negocio.	Rol

Tabla 158: C.R.C. MesaTrabajo
Fuente: Elaboración propia

MesaTrabajo	
Responsabilidades	Colaboradores
Gestionar las relaciones entre los Patrocinadores, Roles y sus Tareas para el desarrollo de la Arquitectura	Rol, Patrocinador, Tarea

Tabla 159: C.R.C. Patrocinador
Fuente: Elaboración propia

Patrocinador	
Responsabilidades	Colaboradores
Gestionar los Patrocinadores comprometidos con el desarrollo de la nueva Arquitectura Empresarial.	

Tabla 160: C.R.C. Tarea
Fuente: Elaboración propia

Tarea	
Responsabilidades	Colaboradores
Gestionar las tareas de los procesos de la organización que deben realizarse para el nuevo desarrollo	

Tabla 161: C.R.C. Artefacto
Fuente: Elaboración propia

Artefacto	
Responsabilidades	Colaboradores
Gestionar los Artefactos Arquitectónicos de la organización y su gobierno.	TipoArtefacto, Persona, Unidad.

Tabla 162: C.R.C. TipoArtefacto
Fuente: Elaboración propia

TipoArtefacto	
Responsabilidades	Colaboradores
Almacenar la clasificación de los distintos tipos de artefactos arquitectónicos de la organización.	

Tabla 163: C.R.C. Comunidad
Fuente: Elaboración propia

Comunidad	
Responsabilidades	Colaboradores
Gestionar a las Comunidades Involucradas con el desarrollo de la nueva Arquitectura.	Usuario.

Tabla 164: C.R.C. Usuario
Fuente: Elaboración propia

Usuario	
Responsabilidades	Colaboradores
Gestionar los Usuarios parte de las Comunidades involucradas, quienes son afectados por el desarrollo.	

Tabla 165: C.R.C. RepositorioArquitectura
Fuente: Elaboración propia

RepositorioArquitectura	
Responsabilidades	Colaboradores
Gestionar la relación entre los distintos elementos de la Arquitectura Empresarial a desarrollar.	Comunidad, Unidad, NivelMadurez, Principio, Mejora, Alcance, Terminología, Premisa, Requerimiento, Adaptacion, Presupuesto.

Tabla 166: C.R.C. NivelMadurez
Fuente: Elaboración propia

NivelMadurez	
Responsabilidades	Colaboradores
Gestionar el estudio del Nivel de Madurez Arquitectónica del Negocio.	

Tabla 167: C.R.C. Mejora
Fuente: Elaboración propia

Mejora	
Responsabilidades	Colaboradores
Gestionar los puntos de mejora de la Arquitectura.	

Tabla 168: C.R.C. Alcance
Fuente: Elaboración propia

Alcance	
Responsabilidades	Colaboradores
Gestionar el alcance del desarrollo de la nueva Arquitectura Empresarial.	

Tabla 169: C.R.C. Terminología
Fuente: Elaboración propia

Terminología	
Responsabilidades	Colaboradores
Gestionar el conjunto de terminologías acordadas para la descripción del contenido arquitectónico.	

Tabla 170: C.R.C. Premisa
Fuente: Elaboración propia

Premisa	
Responsabilidades	Colaboradores
Gestionar las premisas sobre el desarrollo de la nueva Arquitectura Empresarial.	

Tabla 171: C.R.C. Requerimiento
Fuente: Elaboración propia

Requerimiento	
Responsabilidades	Colaboradores
Gestionar los requerimientos sobre el desarrollo de la nueva Arquitectura Empresarial.	

Tabla 172: C.R.C. Adaptacion
Fuente: Elaboración propia

Adaptacion	
Responsabilidades	Colaboradores
Gestionar los que la adaptación de la nueva Arquitectura necesita.	

Tabla 173: C.R.C. Presupuesto
Fuente: Elaboración propia

Presupuesto	
Responsabilidades	Colaboradores
Gestionar los requerimientos de presupuesto del desarrollo de la nueva arquitectura.	

Tabla 174: C.R.C. Principio
Fuente: Elaboración propia

Principio	
Responsabilidades	Colaboradores
Gestionar los principios arquitectónicos apropiados para la organización.	TipoPrincipio.

Tabla 175: C.R.C. TipoPrincipio
Fuente: Elaboración propia

TipoPrincipio	
Responsabilidades	Colaboradores
Almacenar la clasificación de los tipos de principios presentes en la organización.	