

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**DESARROLLO DE UN SISTEMA DE DIFUSIÓN DE GESTIÓN DEL  
CONOCIMIENTO DE LA ESPE, APLICANDO LA METODOLOGÍA  
OPENUP Y EL FRAMEWORK RUBY ON RAILS**

**Previo a la obtención del Título de:**

**INGENIERO EN SISTEMAS E INFORMÁTICA**

**POR:**

**SANTIAGO RAMIRO RÍOS SALGADO**

**SANGOLQUÍ, MARZO DE 2013**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. SANTIAGO RAMIRO RÍOS SALGADO como requerimiento parcial para la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

---

Fecha

---

ING. CECILIA HINOJOSA

# **ESCUELA POLITÉCNICA DEL EJÉRCITO**

## **DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

### **CARRERA: INGENERÍA EN SISTEMAS E INFORMÁTICA**

#### **AUTORIZACIÓN**

Yo, **Santiago Ramiro Ríos Salgado**, autorizo a la Escuela Politécnica del Ejército, la publicación en la biblioteca virtual de la Institución del trabajo titulado **“Desarrollo de un Sistema de Difusión de Gestión del Conocimiento de la ESPE, Aplicando la Metodología OpenUP y el Framework Ruby On Rails”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Quito, Marzo del 2013.

---

Santiago Ramiro Ríos Salgado

## **DEDICATORIA**

A mi familia que con su ejemplo e incondicional apoyo a través del tiempo, se han convertido en el pilar fundamental en mi vida.

Todo este trabajo ha sido posible gracias a ellos.

**Santiago Ramiro Ríos Salgado**

## **AGRADECIMIENTOS**

### **A Dios.**

Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

### **A mis padres y mi hermano**

Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

### **A la Escuela Politécnica del Ejército**

A las autoridades, docentes y personal administrativo que forman parte de la Escuela Politécnica del Ejército, y en especial del Departamento de Ciencias de la Computación, quienes con su dedicación y experiencia me han permitido obtener una educación integral y ética.

**Santiago Ramiro Ríos Salgado**

## ÍNDICE DE CONTENIDOS

CERTIFICACIÓN .....	II
DEDICATORIA .....	IV
AGRADECIMIENTOS .....	V
ÍNDICE DE CONTENIDOS .....	VI
ÍNDICE DE FIGURAS .....	XXI
ÍNDICE DE TABLAS .....	XXV
RESUMEN .....	1
<b>CAPÍTULO 1: INTRODUCCIÓN .....</b>	<b>3</b>
1.1. TEMA .....	3
1.2. ANTECEDENTES .....	3
1.3. SITUACIÓN ACTUAL .....	3
1.4. FORMULACIÓN DEL PROBLEMA .....	4
1.5. CAUSAS QUE MOTIVAN EL DESARROLLO DEL SISTEMA .....	4
1.6. OBJETIVOS .....	5
1.6.1. GENERAL .....	5
1.6.2. ESPECÍFICOS .....	5
1.7. JUSTIFICACIÓN .....	6
1.8. ALCANCE .....	6

<b>CAPÍTULO 2: MARCO TEÓRICO DE REFERENCIA .....</b>	<b>8</b>
2.1. DATO, INFORMACIÓN Y CONOCIMIENTO .....	8
2.1.1.DATO .....	8
2.1.2.INFORMACIÓN .....	9
2.1.3.ARQUITECTURA DE LA INFORMACIÓN .....	11
2.1.3.1. ARQUITECTURA DE LA INFORMACIÓN COMO PROCESO.....	12
2.1.4.CONOCIMIENTO.....	13
2.1.5.GESTIÓN DE CONOCIMIENTO.....	15
2.1.5.1. DIFUSIÓN DE LA GESTIÓN DE CONOCIMIENTO .....	16
2.2. METODOLOGÍA OPENUP .....	18
2.2.1.PRINCIPIOS DE OPENUP .....	20
2.2.2.CICLO DE VIDA DE OPENUP .....	23
2.2.3.PRÁCTICAS DE OPENUP .....	26
2.2.4.ROLES DE OPENUP .....	27
2.3. DESARROLLO GUIADO POR COMPORTAMIENTO .....	28
2.3.1.HISTORIA.....	30
2.3.2.PRINCIPIOS DE BDD .....	31
2.3.2.1. ESPECIFICACIONES DE COMPORTAMIENTO.....	32
2.3.2.2. ESPECIFICACIÓN COMO UN LENGUAJE UBICUO .....	34
2.3.3.PRÁCTICAS RECOMENDADAS POR BDD .....	35
2.3.4.CICLO DE DESARROLLO BDD .....	35
2.4. ENTREGABLES DEL PROYECTO.....	36
2.5. RUBY ON RAILS.....	36
2.5.1.FILOSOFÍA DE RUBY ON RAILS .....	37
2.5.2.LA ARQUITECTURA MVC.....	38

2.5.2.1.	MODELO.....	39
2.5.2.2.	VISTA.....	39
2.5.2.3.	CONTROLADOR .....	39
2.5.3.	COMPONENTES DE RAILS .....	40
2.5.3.1.	ACTION PACK.....	40
2.5.3.1.1.	ACTION CONTROLLER .....	40
2.5.3.1.2.	ACTION VIEW.....	41
2.5.3.1.3.	ACTION DISPATCH.....	41
2.5.3.2.	ACTION MAILER .....	41
2.5.3.3.	ACTIVE MODEL.....	41
2.5.3.4.	ACTIVE RECORD.....	42
2.5.3.5.	ACTIVE RESOURCE .....	42
2.5.3.6.	ACTIVE SUPPORT .....	42
2.5.3.7.	RAILTIES .....	42
2.5.4.	REPRESENTATIONAL STATE TRANSFER (REST) .....	42
2.5.4.1.	PRINCIPIOS DE REST .....	43
2.5.4.1.1.	REST UTILIZA LOS MÉTODOS HTTP DE MANERA EXPLÍCITA... 43	
2.5.4.1.2.	REST NO MANTIENE ESTADO .....	44
2.5.4.1.2.1.	SERVICIOS CON ESTADO VS. SIN ESTADO.....	45
2.5.4.1.3.	REST EXPONE URIS CON FORMA DE DIRECTORIOS .....	47
2.5.4.1.4.	REST TRANSFIERE XML, JSON, O AMBOS.....	48
2.6.	HERRAMIENTAS PARA EL DESARROLLO DEL PROYECTO .....	50
2.6.1.	SISTEMA DE CONTROL DE VERSIONES DISTRIBUIDO .....	50
2.6.1.1.	DEFINICIÓN .....	51
2.6.1.2.	IMPORTANCIA DEL CONTROL DE VERSIONES .....	51

2.6.1.3.	TERMINOLOGÍA BÁSICA.....	52
2.6.1.4.	GIT .....	53
2.6.1.4.1.	OBJETOS DE GIT .....	54
2.6.1.5.	GITHUB.....	55
2.6.2.	COFFEESCRIPT .....	55
2.6.3.	TWITTER BOOTSTRAP .....	56
2.6.3.1.	CARACTERÍSTICAS PRINCIPALES.....	56
2.6.3.2.	COMPONENTES .....	57
2.6.4.	RESPONSIVE DESIGN .....	57
2.6.4.1.	ORIGEN .....	58
2.6.4.2.	VENTAJAS.....	58
2.6.5.	DISQUS .....	59
2.6.6.	TEAMWORKPM.NET .....	60
2.6.7.	OMNIGRAFFLE .....	60
2.6.8.	OMNIPLAN .....	61
2.6.9.	SUBLIME TEXT 2 .....	61
2.6.10.	RUBYMINE.....	62
2.7.	HERRAMIENTAS DE ANÁLISIS.....	62
2.7.1.	GOOGLE ANALYTICS.....	62
2.7.2.	MIXPANEL .....	63
2.7.3.	NEW RELIC .....	64
2.7.3.1.	CAPACIDADES TÉCNICAS .....	64
<b>CAPÍTULO 3:</b>	<b>INICIO .....</b>	<b>66</b>
3.1.	PLAN DE DESARROLLO DE SOFTWARE .....	66
3.1.1.	INTRODUCCIÓN .....	66

3.1.1.1.	PROPÓSITO .....	66
3.1.1.2.	ALCANCE .....	66
3.1.1.3.	DEFINICIONES, SIGLAS Y ABREVIACIONES .....	66
3.1.1.4.	REFERENCIAS.....	67
3.1.1.5.	DESCRIPCIÓN .....	67
3.1.2.	PLAN DE PROYECTO .....	68
3.1.2.1.	PROPÓSITO, ALCANCE Y OBJETIVOS DEL PROYECTO .....	68
3.1.2.2.	SUPOSICIONES Y RESTRICCIONES .....	68
3.1.2.3.	PRODUCTOS DEL PROYECTO .....	69
3.1.3.	ORGANIZACIÓN DEL PROYECTO .....	69
3.1.3.1.	ESTRUCTURA ORGANIZACIONAL.....	69
3.1.4.	GERENCIA DE PROYECTOS .....	70
3.1.4.1.	ESTIMACIÓN DEL PROYECTO .....	70
3.1.4.2.	PLAN DE PROYECTO .....	71
3.1.4.2.1.	PLAN DE FASES .....	71
3.1.4.2.2.	OBJETIVOS DE LAS ITERACIONES .....	71
3.1.4.2.3.	LIBERACIÓN DE PROTOTIPOS .....	72
3.2.	ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE .....	72
3.2.1.	INTRODUCCIÓN .....	72
3.2.1.1.	PROPÓSITO .....	72
3.2.1.2.	ALCANCE .....	73
3.2.1.3.	PERSONAL INVOLUCRADO .....	73
3.2.1.4.	DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS.....	73
3.2.1.5.	REFERENCIAS.....	76
3.2.1.6.	RESUMEN .....	77

3.2.2.DESCRIPCIÓN GENERAL .....	77
3.2.2.1. PERSPECTIVA DEL PRODUCTO .....	77
3.2.2.2. FUNCIONALIDAD DEL PRODUCTO.....	77
3.2.2.3. CARACTERÍSTICAS DE LOS USUARIOS .....	78
3.2.2.4. RESTRICCIONES.....	79
3.2.2.4.1. POLÍTICAS REGULATORIAS .....	79
3.2.2.4.2. LIMITACIONES DE HARDWARE .....	79
3.2.2.4.3. INTERFACES CON OTRAS APLICACIONES.....	80
3.2.2.4.4. CONSIDERACIONES DE SEGURIDAD .....	80
3.2.2.5. SUPOSICIONES Y DEPENDENCIAS .....	80
3.2.2.6. EVOLUCIÓN PREVISIBLE DEL SISTEMA.....	81
3.2.3.REQUISITOS ESPECÍFICOS.....	81
3.2.3.1. REQUISITOS COMUNES DE LAS INTERFACES.....	81
3.2.3.1.1. INTERFACES DE USUARIO .....	81
3.2.3.1.2. INTERFACES DE HARDWARE .....	81
3.2.3.2. REQUISITOS FUNCIONALES.....	82
3.2.3.2.1. MÓDULO: USUARIOS.....	82
3.2.3.2.1.1. AUTENTICACIÓN EN EL SISTEMA.....	82
3.2.3.2.1.2. CREACIÓN DE CUENTAS EN EL SISTEMA .....	82
3.2.3.2.1.3. CAMBIO DE CONTRASEÑA DE UN USUARIO .....	82
3.2.3.2.1.4. CAMBIO DE CONTRASEÑA A OTROS USUARIOS .....	83
3.2.3.2.1.5. RESTITUCIÓN DE CONTRASEÑAS .....	83
3.2.3.2.1.6. PERFIL DE UN USUARIO .....	83
3.2.3.2.2. MÓDULO: PROGRAMAS DE INVESTIGACIÓN .....	84
3.2.3.2.2.1. CREACIÓN DE PROGRAMAS DE INVESTIGACIÓN.....	84

3.2.3.2.2.2.	MODIFICACIÓN DE PROGRAMAS DE INVESTIGACIÓN .....	84
3.2.3.2.2.3.	ELIMINACIÓN DE PROGRAMAS DE INVESTIGACIÓN .....	85
3.2.3.2.2.4.	VISUALIZACIÓN DE PROGRAMAS DE INVESTIGACIÓN .....	85
3.2.3.2.3.	MÓDULO: PROYECTOS DE INVESTIGACIÓN .....	86
3.2.3.2.3.1.	CREACIÓN DE PROYECTOS DE INVESTIGACIÓN.....	86
3.2.3.2.3.2.	MODIFICACIÓN DE PROYECTOS DE INVESTIGACIÓN .....	86
3.2.3.2.3.3.	ELIMINACIÓN DE PROYECTOS DE INVESTIGACIÓN .....	87
3.2.3.2.3.4.	VISUALIZACIÓN DE PROYECTOS DE INVESTIGACIÓN .....	87
3.2.3.2.4.	MÓDULO: RESULTADOS DE INVESTIGACIÓN .....	87
3.2.3.2.4.1.	CREACIÓN DE RESULTADOS DE INVESTIGACIÓN.....	87
3.2.3.2.4.2.	MODIFICACIÓN DE RESULTADOS DE INVESTIGACIÓN .....	88
3.2.3.2.4.3.	ELIMINACIÓN DE RESULTADOS DE INVESTIGACIÓN .....	88
3.2.3.2.4.4.	VISUALIZACIÓN DE RESULTADOS DE INVESTIGACIÓN .....	88
3.2.3.2.5.	MÓDULO: OTRAS ACTIVIDADES RELACIONADAS A LA GESTIÓN DEL CONOCIMIENTO .....	89
3.2.3.2.5.1.	CREACIÓN DE ACTIVIDADES .....	89
3.2.3.2.5.2.	MODIFICACIÓN DE ACTIVIDADES .....	89
3.2.3.2.5.3.	ELIMINACIÓN DE ACTIVIDADES.....	89
3.2.3.2.5.4.	VISUALIZACIÓN DE ACTIVIDADES.....	90
3.2.3.2.6.	MÓDULO: DEPARTAMENTOS .....	90
3.2.3.2.6.1.	VISUALIZACIÓN DE UN DEPARTAMENTO.....	90
3.2.3.2.6.2.	LISTA DE PROGRAMAS DE UN DEPARTAMENTO POR TIPO DE PROGRAMA .....	90
3.2.3.2.7.	MÓDULO: ADMINISTRACIÓN DEL SISTEMA .....	91
3.2.3.2.7.1.	ADMINISTRACIÓN DE DEPARTAMENTOS .....	91

3.2.3.2.7.2.	ADMINISTRACIÓN DE ÁREAS DE CONOCIMIENTO.....	91
3.2.3.2.7.3.	ADMINISTRACIÓN DE LÍNEAS DE INVESTIGACIÓN .....	91
3.2.3.2.7.4.	ADMINISTRACIÓN DE SUB-LÍNEAS DE INVESTIGACIÓN.....	92
3.2.3.2.7.5.	ADMINISTRACIÓN DE TIPOS DE PROGRAMA .....	92
3.2.3.2.7.6.	ADMINISTRACIÓN DE TIPOS DE RESULTADO .....	92
3.2.3.2.7.7.	ADMINISTRACIÓN DE USUARIOS EXTERNOS .....	93
3.2.3.2.8.	MÓDULO: MISCELÁNEOS.....	93
3.2.3.2.8.1.	PÁGINA DE INICIO .....	93
3.2.3.2.8.2.	PÁGINA “EXPLORAR” .....	93
3.2.3.2.8.3.	PÁGINA “ACERCA DE” .....	94
3.2.3.2.8.4.	PÁGINA “AYUDA” .....	94
3.2.3.2.8.5.	BÚSQUEDAS DENTRO DEL SISTEMA.....	94
3.2.3.3.	REQUISITOS NO FUNCIONALES .....	95
3.2.3.3.1.	REQUISITOS DE RENDIMIENTO .....	95
3.2.3.3.2.	SEGURIDAD.....	95
<b>CAPÍTULO 4:</b>	<b>ELABORACIÓN Y CONSTRUCCIÓN.....</b>	<b>96</b>
4.1.	DOCUMENTO DE ARQUITECTURA DE SOFTWARE .....	96
4.1.1.	INTRODUCCIÓN .....	96
4.1.1.1.	PROPÓSITO.....	96
4.1.1.2.	ALCANCE .....	96
4.1.1.3.	ORGANIZACIÓN DEL DOCUMENTO .....	96
4.1.2.	REPRESENTACIÓN DE LA ARQUITECTURA .....	97
4.1.3.	OBJETIVOS Y RESTRICCIONES DE LA ARQUITECTURA .....	98
4.1.4.	VISTA DE CASOS DE USO .....	99
4.1.4.1.	INTRODUCCIÓN .....	99

4.1.4.2.	IDENTIFICACIÓN DE LOS CASOS DE USO RELEVANTES PARA LA ARQUITECTURA.....	99
4.1.4.2.1.	ADMINISTRACIÓN DE PROGRAMAS DE INVESTIGACIÓN .....	100
4.1.4.2.2.	ADMINISTRACIÓN DE PROYECTOS DE INVESTIGACIÓN .....	100
4.1.4.2.3.	ADMINISTRACIÓN DE RESULTADOS DE INVESTIGACIÓN .....	101
4.1.4.2.4.	ADMINISTRACIÓN DE OTRAS ACTIVIDADES RELACIONADAS A LA GESTIÓN DEL CONOCIMIENTO.....	101
4.1.4.2.5.	ADMINISTRACIÓN DE GRUPOS DE INVESTIGACIÓN .....	102
4.1.1.1.1.	CUENTAS DE USUARIO .....	102
4.1.1.1.2.	ADMINISTRACIÓN GENERAL DEL SISTEMA.....	103
4.1.1.2.	DESCRIPCIÓN DE LOS CASOS DE USO RELEVANTES PARA LA ARQUITECTURA.....	103
4.1.2.	VISTA LÓGICA .....	103
4.1.2.1.	INTRODUCCIÓN .....	103
4.1.2.2.	DESCOMPOSICIÓN EN SUBSISTEMAS.....	104
4.1.2.3.	DESCRIPCIÓN DE LOS SUBSISTEMAS .....	104
4.1.2.4.	DISEÑO DE SUBSISTEMAS .....	105
4.1.2.4.1.	DISEÑO DETALLADO DE CADA SUBSISTEMA.....	105
4.1.2.5.	REALIZACIÓN DE LOS CASOS DE USO RELEVANTES PARA LA ARQUITECTURA.....	106
4.1.2.5.1.	ADMINISTRACIÓN DE TIPOS DE PROGRAMA .....	106
4.1.2.5.2.	ADMINISTRACIÓN DE TIPOS DE RESULTADO .....	107
4.1.2.5.3.	ADMINISTRACIÓN DE ROLES .....	107
4.1.2.5.4.	ADMINISTRACIÓN DE DEPARTAMENTOS .....	107
4.1.2.5.5.	ADMINISTRACIÓN DE ÁREAS DE CONOCIMIENTO .....	107

4.1.2.5.6.	ADMINISTRACIÓN DE LÍNEAS DE INVESTIGACIÓN .....	108
4.1.2.5.7.	ADMINISTRACIÓN DE SUB-LÍNEAS DE INVESTIGACIÓN .....	108
4.1.2.5.8.	ADMINISTRACIÓN DE PROGRAMAS .....	108
4.1.2.5.9.	ADMINISTRACIÓN DE PROYECTOS .....	108
4.1.2.5.10.	ADMINISTRACIÓN DE RESULTADOS .....	109
4.1.2.5.11.	ADMINISTRACIÓN DE ACTIVIDADES .....	109
4.1.2.5.12.	ADMINISTRACIÓN DE GRUPOS DE INVESTIGACIÓN .....	109
4.1.2.5.13.	CUENTAS DE USUARIO .....	109
4.1.2.6.	DIAGRAMA DE CLASES .....	110
4.1.2.7.	DIAGRAMA DE ENTIDAD/RELACIÓN .....	111
4.1.2.8.	DIAGRAMA DE NAVEGACIÓN .....	112
4.1.3.	VISTA DE DESPLIEGUE .....	113
4.1.3.1.	INTRODUCCIÓN .....	113
4.1.3.2.	DISTRIBUCIÓN Y DESPLIEGUE .....	113
4.1.3.2.1.	ESCENARIO DEL ENTORNO .....	114
4.1.4.	VISTA DE DATOS .....	114
4.2.	PLAN DE PRUEBAS DE SOFTWARE .....	115
4.2.1.	INTRODUCCIÓN .....	115
4.2.1.1.	OBJETIVOS .....	115
4.2.1.2.	ESTRATEGIA DE PRUEBAS .....	115
4.2.1.3.	ALCANCE .....	116
4.2.1.4.	REFERENCIAS .....	117
4.2.2.	INSTRUMENTOS DE PRUEBA .....	117
4.2.2.1.	MÓDULOS DEL PROGRAMA .....	117
4.2.3.	CARACTERÍSTICAS A SER PROBADAS .....	118

4.2.4.APROXIMACION .....	118
4.2.4.1. PRUEBAS FUNCIONALES.....	119
4.2.4.2. PRUEBAS DE COMPORTAMIENTO.....	119
4.2.5.PROCESO DE PRUEBAS .....	120
4.3. DESARROLLO DE BOCETOS PARA INTERFACES DEL SISTEMA.....	134
4.3.1.MÓDULO: MISCELÁNEOS .....	134
4.3.1.1. PÁGINA DE INICIO .....	134
4.3.1.2. INICIO DE SESIÓN.....	134
4.3.1.3. PÁGINA “EXPLORAR” .....	135
4.3.2.MÓDULO: USUARIOS.....	135
4.3.2.1. ACTUALIZACIÓN DE DATOS DE UNA CUENTA.....	135
4.3.2.2. ACTUALIZACIÓN DE DATOS DE PERFIL .....	136
4.3.2.3. PÁGINA DE UN USUARIO.....	136
4.3.3.MÓDULO: PROGRAMAS .....	137
4.3.3.1. AGREGAR/MODIFICAR PROGRAMA .....	137
4.3.3.2. VISUALIZAR PROGRAMA .....	137
4.3.4.MÓDULO: PROYECTOS.....	138
4.3.4.1. AGREGAR/MODIFICAR PROYECTO .....	138
4.3.4.2. VISUALIZAR PROYECTO .....	138
4.3.5.MÓDULO: RESULTADOS .....	139
4.3.5.1. AGREGAR/MODIFICAR RESULTADO .....	139
4.3.5.2. VISUALIZAR RESULTADO .....	139
4.3.6.MÓDULO: ACTIVIDADES .....	140
4.3.6.1. AGREGAR/MODIFICAR ACTIVIDADES .....	140
4.3.6.2. VISUALIZAR ACTIVIDAD .....	140

4.4.	DESARROLLO DE INTERFACES DEL SISTEMA USANDO HTML .....	141
4.4.1.	MÓDULO: MISCELÁNEOS .....	141
4.4.1.1.	PÁGINA DE INICIO .....	141
4.4.1.2.	INICIO DE SESIÓN .....	141
4.4.1.3.	PÁGINA “EXPLORAR” .....	142
4.4.2.	MÓDULO: USUARIOS .....	143
4.4.2.1.	ACTUALIZACIÓN DE DATOS DE UNA CUENTA.....	143
4.4.2.2.	ACTUALIZACIÓN DE DATOS DE PERFIL .....	143
4.4.2.3.	PÁGINA DE UN USUARIO.....	144
4.4.3.	MÓDULO: PROGRAMAS .....	145
4.4.3.1.	AGREGAR/MODIFICAR PROGRAMA .....	145
4.4.3.2.	VISUALIZAR PROGRAMA .....	146
4.4.4.	MÓDULO: PROYECTOS.....	147
4.4.4.1.	AGREGAR/MODIFICAR PROYECTO .....	147
4.4.4.2.	VISUALIZAR PROYECTO .....	148
4.4.5.	MÓDULO: RESULTADOS .....	149
4.4.5.1.	AGREGAR/MODIFICAR RESULTADO .....	149
4.4.5.2.	VISUALIZAR RESULTADO .....	150
4.4.6.	MÓDULO: ACTIVIDADES .....	151
4.4.6.1.	AGREGAR/MODIFICAR ACTIVIDAD .....	151
4.4.6.2.	VISUALIZAR ACTIVIDAD .....	152
<b>CAPÍTULO 5:</b>	<b>TRANSICIÓN .....</b>	<b>153</b>
5.1.	INSTALACIÓN DEL SISTEMA.....	153
5.1.1.	REQUERIMIENTOS PREVIOS PARA INSTALACIÓN DEL SISTEMA... 153	
5.1.1.4.	SERVIDOR WEB / ALMACENAMIENTO .....	153

5.1.1.4.1.	HARDWARE .....	153
5.1.1.4.2.	SOFTWARE.....	153
5.1.1.5.	SERVIDOR DE BASE DE DATOS .....	154
5.1.1.5.1.	HARDWARE .....	154
5.1.1.5.2.	SOFTWARE.....	154
5.1.2.	MANUAL DE INSTALACIÓN DEL SISTEMA .....	154
5.1.2.4.	SERVIDOR DE BASE DE DATOS .....	154
5.1.2.4.1.	INSTALAR MYSQL .....	154
5.1.2.4.2.	INSTALAR MYSQL .....	154
5.1.2.4.3.	CREAR BASE DE DATOS .....	155
5.1.1.4.	SERVIDOR WEB / ALMACENAMIENTO .....	155
5.1.1.4.1.	INSTALAR APACHE .....	155
5.1.1.4.2.	INSTALAR RVM Y RAILS .....	155
5.1.1.4.3.	INSTALAR PASSENGER .....	156
5.1.1.4.4.	CONFIGURAR APACHE .....	156
5.1.1.4.5.	CONFIGURAR APLICACIÓN .....	156
5.1.1.4.5.1.	COPIAR EL PROYECTO DESDE GITHUB .....	156
5.1.1.4.5.2.	CONFIGURAR EL VIRTUAL HOST DE APACHE .....	157
5.1.1.4.5.3.	CONFIGURAR EL ACCESO A BASE DE DATOS.....	157
5.1.1.4.5.4.	CONFIGURAR EL ACCESO AL SERVIDOR DE CORREO .....	158
5.1.1.4.5.5.	MIGRACIÓN DE LA ESTRUCTURA DE LA BASE DE DATOS ..	158
5.1.1.4.6.	FINALIZAR INSTALACIÓN .....	158
5.2.	EVALUACIÓN DEL SISTEMA .....	159
5.2.1.	GOOGLE ANALYTICS.....	159
5.2.1.1.	VISIÓN GENERAL .....	159

5.2.1.1.1.	VISITAS DESDE DICIEMBRE 2012 A FEBRERO 2013 .....	159
5.2.1.1.2.	VISITANTES EXCLUSIVOS DESDE DIC. 2012 A FEB. 2013.....	159
5.2.1.1.3.	PÁGINAS VISTAS DESDE DIC. 2012 A FEB. 2013.....	160
5.2.1.1.4.	PROMEDIO DE PÁGINAS VISTAS DURANTE UNA VISITA.....	160
5.2.1.2.	DATOS DEMOGRÁFICOS .....	160
5.2.1.2.1.	IDIOMA .....	160
5.2.1.2.2.	UBICACIÓN .....	161
5.2.1.3.	TECNOLOGÍA.....	161
5.2.1.3.1.	NAVEGADOR Y SISTEMA OPERATIVO.....	161
5.2.1.3.2.	PROVEEDOR DE INTERNET.....	161
5.2.1.4.	FLUJO DE VISITANTES .....	162
5.2.2.	NEW RELIC .....	163
5.2.2.1.	VISIÓN GENERAL .....	163
5.2.2.1.1.	TIEMPO DE CARGA DEL BROWSER.....	163
5.2.2.1.2.	TIEMPO DE RESPUESTA DEL SERVIDOR.....	164
5.2.2.1.3.	THROUGHPUT .....	164
5.2.2.1.4.	PUNTAJE APDEX.....	165
5.2.2.1.5.	MAPA DE TIEMPO.....	165
5.2.2.2.	BASE DE DATOS.....	166
5.2.2.2.1.	THROUGHPUT DE LA BASE DE DATOS .....	166
5.2.2.2.2.	TIEMPO DE RESPUESTA DE LA BASE DE DATOS.....	166
5.2.2.2.3.	OPERACIONES MÁS COMUNES.....	167
<b>CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>168</b>
6.1.	CONCLUSIONES .....	168
6.2.	RECOMENDACIONES.....	174

**REFERENCIAS BIBLIOGRÁFICAS ..... 176**

## ÍNDICE DE FIGURAS

Figura 2.1. Ciclo de vida de OpenUP .....	24
Figura 2.2. Servicio con Estado.....	45
Figura 2.3. Servicio sin Estado .....	46
Figura 4.1. Casos de Uso: Administración de programas de investigación .....	100
Figura 4.2. Casos de Uso: Administración de proyectos de investigación .....	100
Figura 4.3. Casos de Uso: Administración de resultados de investigación .....	101
Figura 4.4. Casos de Uso: Administración de otras actividades relacionadas a la gestión del conocimiento .....	101
Figura 4.5. Casos de Uso: Administración de grupos de investigación.....	102
Figura 4.6. Casos de Uso: Cuentas de usuario.....	102
Figura 4.7. Casos de Uso: Administración general del Sistema.....	103
Figura 4.8. Descomposición en subsistemas .....	104
Figura 4.9. Subsistema de Administración General .....	105
Figura 4.10. Subsistema de creación y administración de contenido.....	106
Figura 4.11. Realización de Casos de Uso: Administración tipos de programa..	106
Figura 4.12. Realización de Casos de Uso: Administración tipos de resultado ..	107
Figura 4.13. Realización de Casos de Uso: Administración de roles .....	107
Figura 4.14. Realización de Casos de Uso: Administración de departamentos..	107
Figura 4.15. Realización de Casos de Uso: Administración áreas conocimiento	107
Figura 4.16. Realización de Casos de Uso: Administración líneas investigación	108

Figura 4.17. Realización de Casos de Uso: Admin. sub-líneas de investigación	108
Figura 4.18. Realización de Casos de Uso: Administración de Programas	108
Figura 4.19. Realización de Casos de Uso: Administración de proyectos	108
Figura 4.20. Realización de Casos de Uso: Administración de resultados	109
Figura 4.21. Realización de Casos de Uso: Administración de actividades	109
Figura 4.22. Realización de Casos de Uso: Admin. de grupos de investigación	109
Figura 4.23. Realización de Casos de Uso: Cuentas de usuario	109
Figura 4.24. Diagrama de clases	110
Figura 4.25. Diagrama de Entidad/Relación	111
Figura 4.26. Diagrama de navegación	112
Figura 4.27. Distribución y Despliegue	113
Figura 4.28. Descripción de la Especificación de Requisitos de Software	116
Figura 4.29. Descripción del Documento de Arquitectar de Software	116
Figura 4.30. Conceptos	117
Figura 4.31. Boceto: Página de Inicio	134
Figura 4.32. Boceto: Inicio de sesión	134
Figura 4.33. Boceto: Página "Explorar"	135
Figura 4.34. Boceto: Actualización de datos de una cuenta	135
Figura 4.35. Boceto: Actualización de datos de perfil	136
Figura 4.36. Boceto: Página de un usuario	136
Figura 4.37. Boceto: Agregar/Modificar programa	137
Figura 4.38. Boceto: Visualizar programa	137
Figura 4.39. Boceto: Agregar/Modificar Proyecto	138
Figura 4.40. Boceto: Visualizar proyecto	138
Figura 4.41. Boceto: Agregar/Modificar resultado	139

Figura 4.42. Boceto: Visualizar resultado .....	139
Figura 4.43. Boceto: Agregar/Modificar actividades .....	140
Figura 4.44. Boceto: Visualizar actividad.....	140
Figura 4.45. Interfaz HTML: Página de Inicio .....	141
Figura 4.46. Interfaz HTML: Inicio de sesión .....	141
Figura 4.47. Interfaz HTML: Página “Explorar” .....	142
Figura 4.48. Interfaz HTML: Actualización de datos de una cuenta .....	143
Figura 4.49. Interfaz HTML: Actualización de datos de perfil .....	143
Figura 4.50. Interfaz HTML: Página de un usuario.....	144
Figura 4.51. Interfaz HTML: Agregar/Modificar programa .....	145
Figura 4.52. Interfaz HTML: Visualizar programa.....	146
Figura 4.53. Interfaz HTML: Agregar/Modificar proyecto.....	147
Figura 4.54. Interfaz HTML: Visualizar proyecto .....	148
Figura 4.55. Interfaz HTML: Agregar/Modificar resultado.....	149
Figura 4.56. Interfaz HTML: Visualizar resultado .....	150
Figura 4.57. Interfaz HTML: Agregar/Modificar actividad .....	151
Figura 4.58. Interfaz HTML: Visualizar actividad .....	152
Figura 5.1. Pantalla del programa passenger-install-apache-2-module .....	156
Figura 5.2. Evaluación: Visitas desde Diciembre 2012 .....	159
Figura 5.3. Evaluación: Visitantes exclusivos desde Diciembre 2012.....	159
Figura 5.4. Evaluación: Páginas vistas desde Diciembre 2012.....	160
Figura 5.5. Evaluación: Promedio de páginas vistas durante una visita .....	160
Figura 5.6. Evaluación: Idioma de los visitantes.....	160
Figura 5.7. Evaluación: Ubicación de los visitantes.....	161
Figura 5.8. Evaluación: Navegador y Sistema Operativo de las visitas .....	161

Figura 5.9. Evaluación: Proveedor de internet de las visitas.....	161
Figura 5.10. Evaluación: Flujo de visitantes .....	162
Figura 5.11. Evaluación: Tiempo de carga del browser .....	163
Figura 5.12. Evaluación: Tiempo de respuesta del servidor.....	164
Figura 5.13. Evaluación: Throughput.....	164
Figura 5.14. Evaluación: Puntaje Apdex.....	165
Figura 5.15. Evaluación: Mapa de tiempo .....	165
Figura 5.16. Evaluación: Throughput de la base de datos .....	166
Figura 5.17. Evaluación: Tiempo de respuesta de la base de datos.....	166
Figura 5.18. Evaluación: Operaciones más comunes .....	167

## ÍNDICE DE TABLAS

Tabla 2.1. Principios OpenUP / Principios Manifiesto Agile. ....	21
Tabla 2.2. Entregables del Proyecto.....	36
Tabla 2.3. MIME-Type / Content-Type .....	50
Tabla 3.1. Referencias – Plan de desarrollo de software .....	67
Tabla 3.2. Productos del Proyecto.....	69
Tabla 3.3. Desarrollador del Proyecto de Graduación.....	70
Tabla 3.4. Directora del Proyecto de Graduación.....	70
Tabla 3.5. Co-Director del Proyecto de Graduación.....	70
Tabla 3.6. Plan de Fases.....	71
Tabla 3.7. Liberación de Prototipos .....	72
Tabla 3.8. Desarrollador del Proyecto de Graduación.....	73
Tabla 3.9. Directora del Proyecto de Graduación.....	73
Tabla 3.10. Co-Director del Proyecto de Graduación.....	73
Tabla 3.11. Referencias – Especificación de Requisitos de Software.....	76
Tabla 3.12. Características del Administrador del Sistema .....	79
Tabla 3.13. Características del Usuario del Sistema.....	79
Tabla 3.14. Requisito Funcional: Autenticación en el Sistema.....	82
Tabla 3.15. Requisito Funcional: Creación de cuentas en el Sistema .....	82
Tabla 3.16. Requisito Funcional: Cambio de contraseña de un usuario .....	82
Tabla 3.17. Requisito Funcional: Cambio de contraseña a otros usuarios .....	83
Tabla 3.18. Requisito Funcional: Restitución de contraseñas.....	83

Tabla 3.19. Requisito Funcional: Perfil de un usuario .....	83
Tabla 3.20. Requisito Funcional: Creación de programas de investigación.....	84
Tabla 3.21. Requisito Funcional: Modificación de programas de investigación ....	84
Tabla 3.22. Requisito Funcional: Eliminación de programas de investigación.....	85
Tabla 3.23. Requisito Funcional: Visualización de programas de investigación ...	85
Tabla 3.24. Requisito Funcional: Creación de proyectos de investigación .....	86
Tabla 3.25. Requisito Funcional: Modificación de proyectos de investigación.....	86
Tabla 3.26. Requisito Funcional: Eliminación de proyectos de investigación .....	87
Tabla 3.27. Requisito Funcional: Visualización de proyectos de investigación.....	87
Tabla 3.28. Requisito Funcional: Creación de resultados de investigación .....	87
Tabla 3.29. Requisito Funcional: Modificación de resultados de investigación.....	88
Tabla 3.30. Requisito Funcional: Eliminación de resultados de investigación .....	88
Tabla 3.31. Requisito Funcional: Visualización de resultados de investigación....	88
Tabla 3.32. Requisito Funcional: Creación de Actividades .....	89
Tabla 3.33. Requisito Funcional: Modificación de actividades .....	89
Tabla 3.34. Requisito Funcional: Eliminación de actividades.....	89
Tabla 3.35. Requisito Funcional: Visualización de actividades .....	90
Tabla 3.36. Requisito Funcional: Visualización de un departamento .....	90
Tabla 3.37. Requisito Funcional: Lista de programas de un departamento por tipo de programa .....	90
Tabla 3.38. Requisito Funcional: Administración de departamentos.....	91
Tabla 3.39. Requisito Funcional: Administración de áreas de conocimiento .....	91
Tabla 3.40. Requisito Funcional: Administración de líneas de investigación .....	91
Tabla 3.41. Requisito Funcional: Administración de sub-líneas de investigación .	92
Tabla 3.42. Requisito Funcional: Administración de tipos de programa.....	92

Tabla 3.43. Requisito Funcional: Administración de tipos de resultado .....	92
Tabla 3.44. Requisito Funcional: Administración de usuarios externos .....	93
Tabla 3.45. Requisito Funcional: Página de inicio.....	93
Tabla 3.46. Requisito Funcional: Página "Explorar" .....	93
Tabla 3.47. Requisito Funcional: Página "Acerca de" .....	94
Tabla 3.48. Requisito Funcional: Página "Ayuda" .....	94
Tabla 3.49. Requisito Funcional: Búsquedas dentro del Sistema .....	94
Tabla 4.1. Referencias – Plan de pruebas de software.....	117
Tabla 4.2. Descripción de los módulos del programa.....	118
Tabla 4.3. Características a ser probadas.....	118
Tabla 4.4. Pruebas Funcionales .....	119
Tabla 4.5. Formato de Resultados de Pruebas Funcionales.....	119
Tabla 4.6. Pruebas de comportamiento.....	119
Tabla 4.7. Formato de Resultados de Pruebas de Comportamiento .....	119
Tabla 4.8. Pruebas: Administrar programas .....	120
Tabla 4.9. Pruebas: Acciones usuarios anónimos relacionados a programas....	122
Tabla 4.10. Pruebas: Administrar proyectos.....	123
Tabla 4.11. Pruebas: Acciones usuarios anónimos relacionados a proyectos....	125
Tabla 4.12. Pruebas: Administrar resultados.....	127
Tabla 4.13. Pruebas: Acciones usuarios anónimos relacionados a resultados...	129
Tabla 4.14. Pruebas: Administrar actividades .....	130
Tabla 4.15. Pruebas: Acciones usuarios anónimos relacionados a actividades .	132

## RESUMEN

El presente documento muestra el fundamento teórico y el desarrollo, basado en la metodología OpenUP y el framework Ruby on Rails, del Sistema de Difusión de Gestión del Conocimiento de la ESPE.

El Sistema de Difusión de Gestión del Conocimiento de la ESPE es un sistema informático orientado a la web que pretende facilitar el registro gráfico - textual, y la socialización de la gestión del conocimiento relacionado con la docencia, investigación y vinculación. Este sistema permite que el personal docente y de investigación pueda dar a conocer su excepcional labor a la comunidad politécnica y a la sociedad en general.

La metodología de desarrollo empleada para este proyecto es OpenUP, la cual se fundamenta en la metodología RUP (Rational Unified Process), que utiliza un punto de vista pragmático y una filosofía ágil que se centraliza en la naturaleza colaborativa del proceso de desarrollo de software. Una de sus principales características es su alto grado de adaptabilidad a las necesidades de un proyecto en particular. En el caso del Sistema de Difusión de Gestión del Conocimiento de la ESPE, su desarrollo requería de una metodología que fácilmente adapte los requerimientos cambiantes, brinde una guía de la documentación del desarrollo, y se constituya en un aporte substancial para la planificación total del proyecto, cualidades que dispone íntegramente OpenUP.

El proyecto ha sido desarrollado utilizando Ruby on Rails, un framework de aplicaciones web de código abierto, escrito en el lenguaje de programación Ruby, que sigue el paradigma de la arquitectura Modelo Vista Controlador, que trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones reales, escribiendo menos código y con un mínimo de configuración. Particularmente, para este desarrollo permitió que la etapa de construcción sea rápida y sus iteraciones cortas, logrando así una retroalimentación casi inmediata.

## **CAPÍTULO 1:**

### **INTRODUCCIÓN**

#### **1.1. TEMA**

Desarrollo de un Sistema de Difusión de Gestión del Conocimiento de la ESPE, aplicando la Metodología OpenUP y el Framework Ruby on Rails.

#### **1.2. ANTECEDENTES**

La Escuela Politécnica del Ejército acreditada a nivel nacional como universidad Clase A, desde Enero de 2010, se encuentra inmersa en procesos continuos de autoevaluación con fines de mejoramiento y acreditación internacional, con la proyección de ingresar dentro del importante grupo de las 1000 universidades más reconocidas a nivel mundial.

#### **1.3. SITUACIÓN ACTUAL**

La Institución se encuentra actualmente sumida en el proceso denominado “Preparación a la Acreditación de las Carreras”, que tiene como objetivo central reflexionar críticamente sobre el accionar educativo, para reconocer las debilidades, las fortalezas y priorizar mejoras de los procesos que apoyan al desarrollo de las carreras y programas.

Este importante proceso requiere mecanismos modernos para anunciar, evidenciar y difundir la Gestión del Conocimiento que la Escuela Politécnica del

Ejército está realizando a través de sus docentes, investigadores y estudiantes, a nivel de Pregrado y Postgrado.

#### **1.4. FORMULACIÓN DEL PROBLEMA**

Tomando en cuenta al entorno actual, la Escuela Politécnica del Ejército no dispone de un sistema informático que facilite el registro gráfico y textual de la gestión del conocimiento relacionado con la docencia, investigación y vinculación, y que además cuente con las capacidades para difundir las mismas. Lo que incide en que el personal docente y de investigación no pueda dar a conocer su excepcional labor a la comunidad politécnica y a la sociedad en general.

Frente a esta necesidad, se ha propuesto desarrollar un sistema web, al que se lo ha denominado “Sistema de Difusión de Gestión del Conocimiento de la ESPE”, que permite el registro, la administración y la identificación del estado de las diferentes actividades que realiza el cuerpo docente, investigadores y desarrolladores de tesis, de pregrado y postgrado, el mismo que provee una interfaz de uso fácil e intuitivo, y es accesible a través del Internet, por los distintos estamentos de la organización y la comunidad.

#### **1.5. CAUSAS QUE MOTIVAN EL DESARROLLO DEL SISTEMA**

- La Escuela Politécnica del Ejército requiere de una forma eficiente y eficaz que permita difundir todas las actividades relacionadas con la gestión del conocimiento, ya sean éstas de ámbito académico, de investigación o vinculación, que se están llevando a cabo.

- El personal docente, los investigadores y desarrolladores de tesis, necesitan una herramienta de fácil e intuitivo manejo para documentar sus actividades de gestión de conocimiento dentro de la institución.
- La comunidad politécnica y la sociedad en general, necesitan un mecanismo de fácil acceso a las importantes actividades que la institución realiza en su beneficio.
- La necesidad de evidenciar el aporte de la institución al desarrollo del país con fines de “Acreditación de las Carreras” y proyectarse para alcanzar el objetivo de ser parte de las 1000 universidades más importantes del mundo.

## **1.6. OBJETIVOS**

### **1.6.1. GENERAL**

Desarrollar la aplicación web: “Sistema de Difusión de Gestión del Conocimiento de la ESPE”, usando la metodología OpenUP y el Framework de Ruby on Rails (RoR).

### **1.6.2. ESPECÍFICOS**

- Identificar el estado del arte de la metodología de desarrollo ágil OpenUP y el framework Ruby on Rails (RoR).
- Establecer los requerimientos funcionales del Sistema de Gestión de Conocimiento de la ESPE.
- Definir el alcance y objetivos del proyecto de acuerdo con la metodología OpenUP.
- Crear una arquitectura ejecutable para los escenarios clave y demandas de calidad del Sistema a desarrollar.

- Construir la funcionalidad del sistema mediante la técnica Behavior-Driven Development y el uso del framework Ruby on Rails (RoR).

### **1.7. JUSTIFICACIÓN**

A continuación se detallan las razones que justifican este proyecto:

- La Escuela Politécnica del Ejército no cuenta con herramientas modernas de registro y difusión de sus principales actividades académicas, científicas y de vinculación, con características de sencillez y facilidad de manejo.
- La Carrera de Ingeniería de Sistemas e Informática de la ESPE, en particular las áreas de Ingeniería de Software y Desarrollo Web, dispone de información limitada con relación a las actuales metodologías, técnicas y tecnologías disponibles para el desarrollo de aplicaciones web.
- Los actores científicos de la ESPE no tienen una herramienta informática que permita anunciar, registrar y difundir su gran labor docente, de investigación y vinculación con la colectividad,
- El personal docente no posee una forma rápida y sencilla para obtener retroalimentación a través de comentarios, que puedan ser enviados desde cualquier parte del mundo.

### **1.8. ALCANCE**

El Sistema de Difusión de Gestión del Conocimiento de la ESPE basó su desarrollo en la visión y requerimientos funcionales que demanda el personal dedicado a la docencia, la investigación y la vinculación con la colectividad de la ESPE.

Para el desarrollo del sistema se siguieron todos los lineamientos que la metodología OpenUP, la técnica BDD y el framework Ruby on Rails (RoR) dictan para el desarrollo de una aplicación web de uso fácil e intuitivo, que es accesible a través del Internet.

Se realizaron las pruebas necesarias en un ambiente real, a través de las cuales se verificó su funcionamiento y puesta a punto.

## **CAPÍTULO 2:**

### **MARCO TEÓRICO DE REFERENCIA**

Tomando en cuenta la complejidad del proyecto y la necesidad de definir de manera clara el objetivo final del sistema, producto del mismo, es necesario abarcar conceptos básicos que aportan como fundamento teórico de todo el desarrollo y razón del mismo.

#### **2.1. DATO, INFORMACIÓN Y CONOCIMIENTO**

La información presentada a continuación se ha tomado del documento llamado “Diferencia Entre Dato, Información Y Conocimiento” desarrollado por Juan Carrión (2011):

##### **2.1.1. DATO**

El dato (del latín datum), es una representación simbólica (numérica, alfabética, algorítmica etc.), atributo o característica de una entidad. Un dato, registro o huella que por si solo no tiene significado para los humanos pero que puede ser sujeto de interpretación.

En el ámbito de las ciencias de la información y la bibliotecología, se considera que un dato es una expresión mínima de contenido sobre un tema. Ejemplos de datos son: la altura de una montaña, la fecha de nacimiento de un personaje

histórico, el peso específico de una sustancia, el número de habitantes de un país, etc.

El dato es de empleo muy común en el ámbito informático, más específicamente de los lenguajes de programación, aunque también se encuentra relacionado con nociones similares de las matemáticas y la lógica. En programación un dato es la expresión general que describe las características de las entidades sobre las cuales opera un algoritmo. En lenguajes de programación un dato es un atributo que indica al ordenador (y/o al programador) algo sobre la clase de registros que se va a procesar. Esto incluye imponer restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar. Tipos de datos comunes son: enteros, números de coma flotante (decimales), cadenas alfanuméricas, fechas, horas, colores, coches o cualquier cosa.

El dato no tiene valor semántico (sentido) en sí mismo, pero convenientemente tratado (procesado) se puede utilizar en la realización de cálculos o toma de decisiones. (Carrión, 2011)

### **2.1.2. INFORMACIÓN**

La información representa un conjunto de datos relacionados que constituyen una estructura de mayor complejidad con sentido para un lector o usuario (por ejemplo, un capítulo de un libro de ciencias). Como han hecho muchos investigadores que han estudiado el concepto de información, se describe como un mensaje, normalmente bajo la forma de un documento o algún tipo de comunicación audible o visible.

Como cualquier mensaje, tiene un emisor y un receptor. La información es capaz de cambiar la forma en que el receptor percibe algo, es capaz de impactar sobre sus juicios de valor y comportamientos. La palabra “informar” significa originalmente “dar forma a” y la información es capaz de formar a la persona que la consigue, proporcionando ciertas diferencias en su interior o exterior. Por lo tanto, estrictamente hablando, es el receptor, y no el emisor, el que decide si el mensaje que ha recibido es realmente información, es decir, si realmente le informa. Un informe lleno de tablas inconexas, puede ser considerado información por el que lo escribe, pero a su vez puede ser juzgado como “ruido” por el que lo recibe.

La información se mueve entorno a las organizaciones a través de redes formales e informales. Las redes formales tienen una infraestructura visible y definida: cables, buzones de correo electrónico, direcciones, etc. Los mensajes que estas redes proporcionan incluyen e-mail, servicio de entrega de paquetes, y transmisiones a través de Internet. Las redes informales son invisibles.

A diferencia de los datos, la información tiene significado (relevancia y propósito). No sólo puede formar potencialmente al que la recibe, sino que esta organizada para algún propósito. Los datos se convierten en información cuando su creador les añade significado. Se transforman datos en información añadiéndoles valor en varios sentidos. Hay varios métodos:

- **Contextualizando:** se conoce para qué propósito se generaron los datos.
- **Categorizando:** se conoce las unidades de análisis de los componentes principales de los datos.

- **Calculando:** los datos pueden haber sido analizados matemática o estadísticamente.
- **Corrigiendo:** los errores se han eliminado de los datos.
- **Condensando:** los datos se han podido resumir de forma más concisa.

Los ordenadores pueden ayudar a añadir valor y transformar datos en información, pero es muy difícil que puedan ayudar a analizar el contexto de dicha información. Un problema muy común es confundir la información (o el conocimiento) con la tecnología que la soporta. Desde la televisión a Internet, es importante tener en cuenta que el medio no es el mensaje. Lo que se intercambia es más importante que el medio que se usa para hacerlo. Muchas veces se comenta que tener un teléfono no garantiza mantener conversaciones brillantes. En definitiva, que actualmente se tenga acceso a más tecnologías de la información no implica que se haya mejorado el nivel de información. (Carrión, 2011)

### **2.1.3. ARQUITECTURA DE LA INFORMACIÓN**

La arquitectura de la información es la disciplina y arte encargada del estudio, análisis, organización, disposición y estructuración de la información en espacios de información; y de la selección y presentación de los datos en los sistemas de información interactivos y no interactivos. La arquitectura de la información no trata del establecimiento de un conjunto de pasos o guías predefinidas, sino del diseño inteligente que subyace detrás de una interfaz o espacio de información. Trata de maximizar las sinergias que se producen entre interactividad, navegación y contenido con el objetivo de alcanzar una integración sistémica con el cerebro del usuario logrando un fenómeno de persuasión, conocimiento o información

simbiótica que se traspa de un sistema de información a otro. (Jose & Alfonso, 2012)

En relación con la World Wide Web, el Information Architecture Institute define la Arquitectura de la Información como:

1. El diseño estructural en entornos de información compartida.
2. El arte y la ciencia de organizar y rotular sitios web, intranets, comunidades en línea y software para promover la usabilidad y la facilidad de ubicación.
3. Una comunidad emergente orientada a aplicar al entorno digital los principios del diseño y la arquitectura.

La arquitectura de la información trata indistintamente del diseño de: sitios web, interfaces de dispositivos móviles o gadgets (como los lectores de mp3), CD interactivos, videoclips digitales, relojes, tableros de instrumentos de aviones de combate o civiles, interfaces de máquinas dispensadoras, interfaces de juegos electrónicos, etc.

Su principal objetivo es facilitar al máximo los procesos de comprensión y asimilación de la información, así como las tareas que ejecutan los usuarios en un espacio de información definido. (Alonso, 2009)

#### **2.1.3.1. ARQUITECTURA DE LA INFORMACIÓN COMO PROCESO**

José Luis Alonso define a la Arquitectura de la Información en su artículo llamado “La gestión de contenidos digitales: Acceso a la información en el ámbito ciudadano” (2009) de la siguiente manera:

La arquitectura de la información es un proceso iterativo, transversal, que se da a lo largo de todo el diseño del sitio y en cada una de sus fases, para asegurarse de que los objetivos de su producción y del desarrollo de la interfaz se cumplan de manera efectiva.

La arquitectura de la información como disciplina no busca definir una metodología de diseño universal sino articular un conjunto de técnicas para ayudar al desarrollo y producción de espacios de información como los sitios web.

Con el fin de que la asimilación de contenidos por parte del usuario sea eficiente y efectiva, y para que el sitio sea accesible y usable, la arquitectura de la información como proceso en general se encarga, durante el desarrollo, de definir:

- El objeto, propósito y fines del sistema de información o sitio
- La definición del público objetivo y los estudios de la audiencia
- El diseño de la interacción
- El diseño de la navegación, esquemas de organización de los contenidos
- El etiquetado o rotulado de los contenidos para acceder a la información
- La planificación, gestión y desarrollo de contenidos
- La facilidad de búsqueda y el diseño de la interfaz de búsqueda
- La usabilidad
- La accesibilidad
- La retroalimentación del resultado y los procesos de reingeniería del sitio

#### **2.1.4. CONOCIMIENTO**

El conocimiento es una mezcla de experiencia, valores, información y “saber hacer” que sirve como marco para la incorporación de nuevas experiencias e

información, y es útil para la acción. Se origina y aplica en la mente de los conocedores. En las organizaciones con frecuencia no sólo se encuentra dentro de documentos o almacenes de datos, sino que también esta en rutinas organizativas, procesos, prácticas, y normas.

Lo que inmediatamente deja claro la definición es que ese conocimiento no es simple. Es una mezcla de varios elementos; es un flujo al mismo tiempo que tiene una estructura formalizada; es intuitivo y difícil de captar en palabras o de entender plenamente de forma lógica. El conocimiento existe dentro de las personas, como parte de la complejidad humana y de nuestra impredecibilidad<sup>1</sup>. Aunque se suele pensar en activos definibles y concretos, los activos de conocimiento son mucho más difíciles de manejar.

El conocimiento se deriva de la información, así como la información se deriva de los datos. Para que la información se convierta en conocimiento, las personas debe hacer un proceso cognitivo individual y grupal para lograr apropiarse de la información a través de acciones intelectuales o habilidades del pensamiento:

- Comparación.
- Consecuencias.
- Conexiones.
- Conversación.

Estas actividades de creación de conocimiento tienen lugar dentro y entre personas. Al igual que se encuentran datos en registros, e información en

---

<sup>1</sup> Impredecibilidad: cualidad de impredecible

mensajes, se pueden obtener conocimiento de individuos, grupos de conocimiento, o incluso en rutinas organizativas. (Carrión, 2011)

### **2.1.5. GESTIÓN DE CONOCIMIENTO**

La nueva era del trabajo implica la aplicación del conocimiento como la nueva fuente de creación de valor y riqueza.

La generación de valor agregado, vía conocimiento, es la innovación y la mejora de los productos y servicios de la empresa que provienen de la inteligencia y la creatividad de las personas. (Malvicino & Serra, 2006)

En el entorno actual de alta competitividad, globalización, desarrollo tecnológico, reducción de vida de los productos y de crisis mundial, resulta clave la gestión del conocimiento para lograr el éxito sostenible de una empresa. En este contexto resulta realmente crítico desarrollar las habilidades de la empresa para adquirir información, transformarla en conocimiento, incorporarlo a la empresa como aprendizaje, compartirlo rápidamente y ponerlo en práctica.

Esto es muy importante ya que existe una correlación entre gestión del conocimiento y éxito empresarial, entre gestión del conocimiento y adaptabilidad de la empresa al entorno cambiante y desafiante, donde las amenazas se puedan convertir en oportunidades. (Ruiz de Arriaga, 2011)

Es así que, la gestión del conocimiento tiene perspectivas tácticas y operativas, es más importante que la gestión del capital intelectual y se centra en la forma de dar a conocer y administrar las actividades relacionadas con el conocimiento como su creación, captura, transformación y uso. Su función es planificar, implementar y controlar todas las actividades relacionadas con el

conocimiento y los programas requeridos para la administración efectiva del capital intelectual. (Wiig, 1997)

Muchas de las grandes compañías y organizaciones tienen recursos dedicados a la gestión del conocimiento, comúnmente como parte de sus estrategias de negocio, sistemas de información o departamentos de administración de recursos humanos.

#### **2.1.5.1. DIFUSIÓN DE LA GESTIÓN DE CONOCIMIENTO**

Miguel Laufer explica el porqué es importante difundir el conocimiento y su gestión a través de su artículo “La difusión del conocimiento” (2008):

La importancia de diseminar el conocimiento científico y tecnológico se hace cada vez mayor. La apropiación de ese conocimiento por parte de la sociedad es una necesidad obvia para la población en general, pero resulta mucho más obvia y natural para quienes se dedican de forma profesional a la investigación, la enseñanza y la comunicación en las diversas áreas de la ciencia y la tecnología.

Son cada vez más numerosos los aspectos en los cuales la sociedad acciona con base en las nociones que tiene en ese momento acerca de un determinado fenómeno, producto o hecho. Las opiniones sobre la conveniencia o no de adoptar una conducta, de aceptar o no un cierto tipo de pautas o reglas, de procedimientos y hasta de ropas o alimentos, influyen poderosamente en muy numerosas y variadas materias de interés común, determinando la orientación de la participación ciudadana. Tales opiniones modulan de manera decisiva tanto la estrategia de legisladores y grupos de interés que buscan la sanción de medidas

legislativas como la estrategia de los productores y comercializadores de bienes. Los políticos en busca de apoyo las estudian y hacen uso de ellas.

El concepto actual de ciudadanía implica, además de la participación, la capacitación para la toma de decisiones. La capacidad para arribar de manera racional a decisiones acerca de asuntos personales o comunitarios requiere manejar toda la información disponible en el momento adecuado.

Es responsabilidad de toda la sociedad, pero más aun de los comunicadores sociales como difusores del conocimiento, de los docentes como formadores de conocedores, y de los investigadores, como creadores del conocimiento, asegurar que la información necesaria, suficiente y correcta llegue a la mayor cantidad posible de ciudadanos. Se trata de una responsabilidad compartida, consciente de que los saberes no habrán llegado a su destino último mientras no sean apropiados por la sociedad, y las sociedades que a lo largo del tiempo han generado y acumulado saberes han de hacerlos conocer de todos, sin distinciones.

Llamar la atención acerca de este último aspecto, la difusión de los saberes tradicionales, es de particular importancia en los momentos actuales, dada la marginalidad en la cual han estado sumidos por largo tiempo sectores menos favorecidos o culturalmente diferentes del grueso de la sociedad. Tal es el caso de muchos de los pobladores indígenas de nuestro continente, quienes por siglos han vivido en armonía con la naturaleza y aprendido de ella.

Cierto es que la educación adquirida, la formación y la capacidad de cada persona determina sus posibilidades de comprensión y manejo de información.

Tales posibilidades pueden cubrir una gama extensa, pero la disponibilidad de la información ha de estar al alcance de todos, sobre todo en los tiempos actuales, cuando la información y los contenidos en medios electrónicos ya predominan sobre los de medios impresos. Para alcanzar una cobertura suficiente es necesario, pero no suficiente, alfabetizar y educar. También hay que hacer fácilmente accesible el conocimiento.

La cobertura por parte de la prensa escrita, radial o televisiva es de gran alcance, pero es fugaz y sus públicos tienden a estar divididos por sectores de la población. Por otra parte, de estar ampliamente disponibles, los centros de información electrónica y la comunicación presencial en exposiciones y museos, sean éstos estables o ambulantes, hacen factible hacer llegar la información a grandes masas, permitir que su profundidad sea modulada por el usuario y favorecer el aprendizaje interactivo de personas de todas las edades por igual.

Una buena parte de la acción de los organismos educativos y de fomento científico y tecnológico debería estar encaminada a cubrir la necesidad de diseminar en toda la población el conocimiento disponible, a través de programas tanto de formación como de información de largo alcance.

## **2.2. METODOLOGÍA OPENUP**

En el documento de Néstor Díaz y David Vaquero llamado “Metodologías Ágiles” (2010), se definen y comparan las metodologías ágiles Extreme Programming (XP), Scrum, y OpenUP. Adicionalmente, para el desarrollo de este apartado se ha considerado el libro “Agility and Discipline Made Easy: Practices

from OpenUP and RUP” (Kroll & MacIsaac, 2006). En estos documentos se define OpenUP de la siguiente manera:

OpenUP es una metodología de desarrollo basada en la metodología RUP (Rational Unified Process). Es el subconjunto de esta última que contiene el conjunto mínimo de prácticas que ayudan a un equipo de desarrollo software a realizar un producto de alta calidad y de una forma más eficiente. OpenUP utiliza un punto de vista pragmático y una filosofía ágil que se centraliza en la naturaleza colaborativa del proceso de desarrollo del software. Una de sus principales características es su alto grado de adaptabilidad a las necesidades de un proyecto en particular.

OpenUP intenta incluir dentro de su proceso de desarrollo únicamente el contenido imprescindible para garantizar un proceso de desarrollo de calidad y eficiente. Por este motivo no incluye guías sobre aspectos que suelen aparecer en la gestión de proyectos como pueden ser equipos de desarrollo grandes, negociaciones contractuales, uso de tecnologías específicas, definición de aplicaciones de seguridad o críticas, etc. Esto no quiere decir que OpenUP no contemple estos aspectos, simplemente los deja sin especificar, pero en cualquier momento se puede extender la metodología para cubrir alguno de estos aspectos.

OpenUP tiene las características de un Unified Process al cual se aplican enfoques iterativos e incrementales dentro de un ciclo de vida estructurado. El proceso de desarrollo en OpenUP utiliza casos de uso, escenarios, gestión del riesgo y un enfoque centralizado en la arquitectura.

### **2.2.1. PRINCIPIOS DE OPENUP**

Los principios básicos en los que se fundamenta OpenUP son los siguientes:

- Promover la colaboración para alinear los intereses comunes y difundir el conocimiento: este principio promueve la colaboración, desarrollo y compartición de un conocimiento sobre el proyecto en toda la organización, así como un entorno de trabajo saludable.
- Balanceo de las prioridades competitivas para maximizar el valor de los participantes del proyecto: se intentan promover prácticas que permitan a los participantes del proyecto el desarrollo de una solución que maximice sus beneficios aportando el máximo valor de negocio.
- Desarrollo de una arquitectura al inicio con la finalidad de minimizar riesgos y planificar el desarrollo: ejercer practicas que permitan al equipo focalizarse en la arquitectura del sistema para reducir riesgos y planificar la estrategia de desarrollo.
- Evolucionar para obtener retroalimentación continua en aras de una mejora constante: con este principio se intentan ejecutar prácticas que permitan obtener retroalimentación desde el inicio del proyecto y de forma continua, así como mostrar el valor de negocio a los participantes del proyecto.

Cada uno de estos principios se puede alinear con principios del Manifiesto Agile, como se muestra en la siguiente tabla:

Tabla 2.1. Principios OpenUP / Principios Manifiesto Agile. Fuente: (Díaz & Vaquero, 2010)

<b>Principio OpenUP</b>	<b>Principio Manifiesto Agile</b>
Colaborar para alinear intereses y difundir el conocimiento	Las personas y sus interacciones prevalecen sobre los procesos y herramientas
Balancear las prioridades competitivas para maximizar el valor de negocio	La colaboración con el cliente es mas importante que la negociación del contrato
Desarrollar una arquitectura al principio para minimizar el riesgo y planificar una estrategia de desarrollo	El software operativo y funcional aporta más que una documentación extensa y exhaustiva
Evolucionar para obtener retroalimentación continua y realizar mejoras de forma constante	Hay que ser capaces de responder al cambio en vez de estar dirigidos por una planificación cerrada

OpenUP se organiza en dos dimensiones: Contenido metodológico y contenido procedimental. El contenido metodológico es el que define elementos metodológicos tales como roles, tareas, artefactos y guías, independientemente de como se usen estos o se combinen. El contenido procedimental, por el contrario, es donde se aplican todos estos elementos metodológicos dentro de una dimensión temporal, de hecho, se pueden crear multitud de ciclos de vida diferentes a partir del mismo subconjunto de elementos metodológicos.

Los elementos que forman OpenUP se pueden enumerar en:

- **Disciplinas:** OpenUP se centra en las siguientes disciplinas: requisitos, arquitectura, desarrollo, test, gestión de proyecto, gestión de la configuración y del cambio.

Como se puede observar, hay ciertas disciplinas de RUP que han sido omitidas, como modelización del negocio, definición del entorno, gestión de requisitos avanzada y configuración de herramientas para la gestión de la configuración. Estas disciplinas no se incluyeron en OpenUP porque fueron consideradas no necesarias para proyectos de pequeña magnitud o que serían manejadas por organizaciones ajenas al equipo de desarrollo.

- **Tareas:** en OpenUP se define tarea como la unidad de trabajo que debe ser realizada por algún rol. En OpenUP se cuenta con un total de 18 tareas que son ejecutadas de forma principal por un rol (el encargado de realizar la tarea) y de forma secundaria por roles que apoyan en la ejecución de dicha tarea. La naturaleza colaborativa de OpenUP queda demostrada al observarse que ninguna tarea es ejecutada de forma aislada por un único rol sin el apoyo de ningún otro.
- **Artefactos:** un artefacto en OpenUP se considera a todo aquello que una tarea necesita para realizar su función, o bien la produce o modifica. Los distintos roles existentes son los encargados de crear y actualizar los artefactos.

En OpenUP existen un total de 17 artefactos predefinidos considerados como esenciales, pero lógicamente la organización puede decidir añadir tantos artefactos propios como considere necesario, incluso se puede optar por ignorar estos artefactos predefinidos y utilizar cualquier tipo de artefactos definidos por la organización, siempre y cuando estos cumplan el

objetivo para el cual fueron creados: almacenar información. Generalmente los artefactos suelen estar regulados por un sistema de control de versiones.

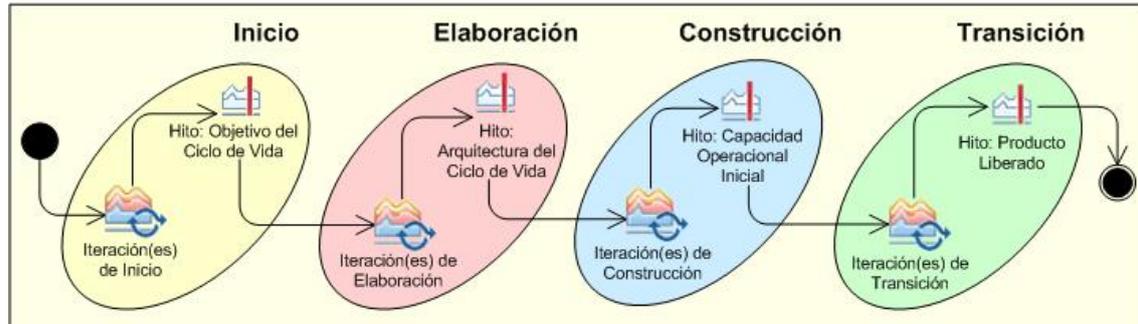
- **Procesos:** antes se mencionó que el contenido metodológico provee de explicaciones detalladas que describen como obtener ciertos objetivos independientemente de como se coloquen los elementos metodológicos dentro de un ciclo de vida. Los procesos toman los elementos metodológicos y los relacionan entre si dentro de secuencias temporales que satisfacen las necesidades de distintos tipos de proyecto. Aquellos elementos metodológicos con un alto grado de reutilización se agrupan en patrones de capacitación, aportando un enfoque de desarrollo consistente para la mayoría de proyectos de desarrollo. Estos patrones se realizan organizando las tareas en actividades, agrupándolas dentro de una secuencia que tiene sentido para un área en particular en la que el patrón es aplicado. Algunos de estos procesos pueden usarse como plantillas para alguna de las iteraciones del proyecto.

### **2.2.2. CICLO DE VIDA DE OPENUP**

En la Figura 2.1. Ciclo de vida de OpenUP. Fuente: se muestra el ciclo de vida de un proyecto en la metodología OpenUP en cada una de sus dimensiones. Los integrantes del equipo contribuyen aportando micro-incrementos que puede ser el resultado del trabajo de unas pocas horas o unos pocos días. El progreso se puede visualizar diariamente, ya que la aplicación va evolucionando en función de estos micro-incrementos.

El objetivo de OpenUP es ayudar al equipo de desarrollo a través de todo el ciclo de vida de las iteraciones, de modo que este sea capaz de añadir valor de negocio para los clientes de una forma predecible: con la entrega de un software operativo y funcional al final de cada iteración. El ciclo de vida del proyecto provee a los clientes de una visión del proyecto, transparencia y les dota de los medios para que les permitan controlar la financiación, el riesgo, el ámbito, el valor de retorno esperado, etc.

Todo proyecto en OpenUP consta de cuatro fases: inicio, elaboración, construcción y transición. Cada una de estas fases se divide a su vez en iteraciones cada una de las cuales tiene como objetivo la entrega de un software operativo y funcional. En la Figura 2.1. Ciclo de vida de OpenUP. Fuente: se puede observar con detalle estas fases:



**Figura 2.1. Ciclo de vida de OpenUP. Fuente: (Díaz & Vaquero, 2010)**

- **Fase de inicio:** las necesidades de cada participante del proyecto son tomadas en cuenta y plasmadas en objetivos del proyecto. Se deben definir: el ámbito del proyecto, los límites del mismo y el criterio de aceptación del proyecto. Los casos de uso críticos, aquellos que dirigen la funcionalidad del sistema, son definidos en esta fase, así como una estimación inicial del coste del proyecto y un boceto de la planificación.

- **Fase de elaboración:** en esta fase se realizan tareas de análisis del dominio y definición de la arquitectura del sistema. Si se decide continuar con el proyecto se debe elaborar un plan de proyecto en esta fase, para lo cual se deben establecer unos requisitos y una arquitectura estables. Por otro lado el proceso de desarrollo, las herramientas, la infraestructura a utilizar y el entorno de desarrollo también se especifican en detalle en esta fase. Al final de la fase se debe tener una definición clara y precisa de los casos de uso, los actores, la arquitectura del sistema y un prototipo ejecutable de la misma.
- **Fase de construcción:** todos los componentes y funcionalidades del sistema que faltan por implementar son realizados, testeados e integrados en esta fase. Los resultados obtenidos en forma de incrementos ejecutables deben ser desarrollados de la forma más rápida posible sin dejar de lado la calidad de lo desarrollado.
- **Fase de transición:** cuando el producto está lo suficientemente maduro (algo que es establecido por ejemplo en función del número de peticiones de cambio por parte de los clientes) como para ser introducido en la comunidad de usuarios, el proyecto se encuentra en esta fase. La fase de la transición consta de las sub-fases de testeo de versiones beta, pilotaje y capacitación de los usuarios finales y de los encargados del mantenimiento del sistema. En función de la respuesta obtenida por los usuarios puede ser necesario realizar cambios en las entregas finales o implementar alguna funcionalidad más.

### 2.2.3. PRÁCTICAS DE OPENUP

OpenUP es una metodología basada en RUP, y por lo tanto, comparte las mismas prácticas que subyacen por debajo del flujo de trabajo y los roles de OpenUP. Estas prácticas son:

- **Desarrollar software de manera iterativa:** todo desarrollo debe realizarse de modo iterativo y en pequeños incrementos. Esto permite identificar riesgos de forma prematura, y actuar en consecuencia.
- **Gestionar los requisitos:** la identificación de los requisitos clave y los que cambian continuamente es un proceso que debe ejecutarse continuamente. Deben articularse los medios que permitan priorizar, filtrar y hacer un seguimiento de los requisitos, ya que, la comunicación será mucho mas eficaz si está basada en unos requisitos claramente definidos.
- **Utilizar una arquitectura basada en componentes:** las arquitecturas basadas en componentes son más flexibles, y dado que OpenUP es una metodología ágil, y por lo tanto flexible, la arquitectura definida debe estar basada en componentes. Por otro lado, la construcción de componentes con un alto grado de reutilización puede ahorrar considerable esfuerzo de futuros desarrollos.
- **Modelar software de forma visual:** se deben aplicar métodos de visualización de modelos que permitan entender la complejidad del sistema sin dar cabida a la ambigüedad.
- **Verificar la calidad del software:** si se realizan verificaciones en cada iteración de cada una de las fases, se reduce considerablemente el costo de resolver cualquier defecto, ya que estos se identifican prematuramente.

- **Controlar los cambios aplicados al software:** todo cambio que se solicite sobre los requisitos establecidos se debe gestionar y realizar un seguimiento sobre el mismo, de este modo se puede determinar el grado de madurez del software desarrollado.

#### 2.2.4. ROLES DE OPENUP

Los roles de OpenUP que recogen el conjunto esencial de habilidades necesarias para realizar los desarrollos son los siguientes:

- **Cliente:** representan el conjunto de individuos o entidades cuyas necesidades se verán cubiertas por el proyecto. Cualquier individuo que se vea afectado por el resultado del proyecto puede constituir un rol.
- **Analista:** dentro del equipo es el encargado de representar al cliente y sus necesidades. Es el encargado de recoger las impresiones de los clientes con el fin de entender el problema y establecer las prioridades de los requisitos identificados.
- **Arquitecto:** es el encargado de realizar la arquitectura general del sistema y tomar las decisiones técnicas claves que influirán en gran medida el futuro diseño e implementación del proyecto.
- **Desarrollador:** el diseño, implementación, testeo unitario e integración de una parte del sistema es una tarea llevada a cabo por este rol.
- **Tester:** como su nombre indica, es el rol encargado de identificar, diseñar e implementar y ejecutar las pruebas necesarias, además de dejar registro de los resultados obtenidos en las mismas.

- **Gestor del proyecto:** es el encargado de dirigir la planificación del proyecto en colaboración con el equipo y los clientes. Asimismo debe conseguir que el equipo se mantenga focalizado en cumplir los objetivos planificados.
- **Cualquier otro rol:** es un rol que indica cualquier integrante del equipo que sea capaz de realizar tareas generales.

### **2.3. DESARROLLO GUIADO POR COMPORTAMIENTO**

Para el desarrollo de la codificación y pruebas del sistema se ha utilizado la técnica de desarrollo guiado por comportamiento (Behavior-Driven Development).

Esta técnica de desarrollo se ha usado como complemento a la metodología OpenUP, la misma que contempla aspectos generales de cómo manejar y dar un seguimiento apropiado al desarrollo de un proyecto de software. BDD a su vez, contempla aspectos más técnicos y enfocados únicamente en el desarrollo del sistema y pruebas al mismo, al mismo tiempo que se alinea con metodologías ágiles de desarrollo de software.

La información que se presenta a continuación es producto de la investigación bibliográfica basada en los documentos: The RSpec Book: Behaviour Driven Development with Rspec, Cucumber, and Friends (Chelimsky, Astels, Helmkamp, North, Dennis, & Hellesoy, 2012), The Cucumber Book: Behaviour-Driven Development for Testers and Developers (Wynne & Hellesoy, 2012), A Study of the Characteristics of Behaviour Driven Development (Solís & Wang, 2011) y Mapping Business Process Modeling constructs to Behavior Driven Development Ubiquitous Language (de Carvalho, de Carvalho e Silva, & Soares Manhaes, 2010).

Desarrollo guiado por comportamiento, o Behavior Driven Development (BDD) es una técnica de desarrollo ágil de software que fomenta la colaboración entre desarrolladores, testers<sup>2</sup>, analistas y personas del negocio en un proyecto de software. Originalmente fue concebido por Dan North en 2003 como respuesta y evolución en el pensamiento detrás de Test Driven Development, uniendo éste con conceptos de Domain Driven Design.

El foco del BDD es el lenguaje e interacciones usadas en el proceso de desarrollo de software. Los desarrolladores guiados por comportamiento usan su lenguaje nativo en combinación con el lenguaje ubicuo de Domain Driven Design para describir el propósito y beneficios de su código. Esto permite a los desarrolladores enfocarse en la razón por la cual el código debe ser creado, antes que en los detalles técnicos, y minimiza la traducción entre el lenguaje técnico en el cuál se escribe el código y el lenguaje de dominio hablado por las personas del negocio, usuarios, interesados, gerentes de proyecto, etc.

BDD permite un entendimiento más global del sistema a crear, eliminando el efecto túnel de los desarrolladores que trabajan sólo con TDD. También crea un entendimiento común de todos los involucrados y ayuda a eliminar malos entendidos sobre lo que el sistema debe hacer.

De la misma forma que en TDD se escriben primero las pruebas unitarias, en BDD se escriben primero el comportamiento esperado de la aplicación, haciéndolo demostrable y automatizándolo en pruebas de aceptación, lo que se denomina especificación activa.

---

<sup>2</sup> Tester: es una persona que realiza pruebas a un producto software.

BDD y TDD no son excluyentes, sino que se pueden aplicar ambas prácticas, complementándose, comenzando el ciclo con las especificaciones con BDD y utilizando TDD en el ciclo de la codificación.

### **2.3.1. HISTORIA**

Behavior-driven development ha sido desarrollado por Dan North como una respuesta a los problemas surgidos de test-driven development:

- Dónde empezar el proceso.
- Qué se debe probar y qué no se debe probar.
- Cuánto debe abarcar una sola prueba
- Cómo nombrar a las pruebas
- Cómo determinar la causa por la que una prueba falla.

BDD es un replanteamiento del enfoque para las pruebas unitarias y pruebas de aceptación que Dan North desarrollo mientras trataba con estos asuntos. Por ejemplo, se encontró que los nombres de las pruebas unitarias deben ser frases enteras que comienzan con la palabra "debería" y tienen que estar ordenados de acuerdo al valor que aportan al negocio. Las pruebas de aceptación deben ser escritas con el marco normativo ágil de una historia de usuario: "Como [rol] Quiero [función] para que [los beneficios]". Los criterios de aceptación deben ser escritos en términos de escenarios y la implementan como clases: Teniendo en cuenta [contexto inicial], cuando [evento] y [garantizar algunos resultados].

A partir de este punto, North y otros desarrollaron BDD en un período de varios años, y finalmente lo encuadraron como un framework de colaboración y comunicación para los desarrolladores, QA y los participantes no técnicos o de

negocios en un proyecto de software. Durante el "Agile specifications, BDD and Testing eXchange", en noviembre de 2009 en Londres, Dan North dio la siguiente descripción de BDD:

BDD es una metodología ágil de segunda generación, que trabaja de afuera hacia adentro, basada en múltiples partes interesadas (stakeholders<sup>3</sup>), de múltiple escala y alta automatización. Describe un ciclo de interacciones con productos bien definidos, lo que resulta en la entrega de software funcional, probado y que aporta al negocio.

En 2008, Chris Matts, quien participó en las primeras discusiones alrededor de BDD, tuvo la idea de "Feature Injection"<sup>4</sup>, lo que permitió que BDD cubra el espacio de análisis y proporcione un tratamiento completo del ciclo de vida del software desde la visión, aprobación del código, hasta la liberación del mismo.

### **2.3.2. PRINCIPIOS DE BDD**

En esencia, el desarrollo basado en comportamiento (BDD) es una versión especializada de desarrollo basado en pruebas (TDD) que se centra en la especificación del comportamiento de las unidades de software.

El desarrollo basado en pruebas es una metodología de desarrollo de software que esencialmente establece que por cada unidad de software, un desarrollador de software tiene que:

- Definir un conjunto de pruebas para la primera unidad

---

<sup>3</sup> Stakeholder: Es aquella persona o entidad que está interesada en la realización de un proyecto.

<sup>4</sup> Feature Injection: Es un framework de procesos de análisis de negocio, que permite a los equipos explotar el valor de las técnicas tradicionales en proyectos que involucran entregas iterativas continuas, tales como las que se usan agile y los procesos lean.

- Luego implementar la unidad
- Finalmente comprobar que la unidad implementada satisface las pruebas.

La definición de las pruebas es bastante general, se permiten pruebas en términos de requisitos de software de alto nivel, detalles técnicos de bajo nivel, o cualquier otra especificación en el medio.

El desarrollo guiado por comportamiento indica que las pruebas de cualquier unidad de software debe especificarse en términos de la conducta deseada de la unidad. Tomando prestado del desarrollo ágil de software, el “comportamiento deseado” en este caso consiste en los requerimientos definidos por el negocio.

#### **2.3.2.1. ESPECIFICACIONES DE COMPORTAMIENTO**

BDD establece cómo el comportamiento deseado debe ser especificado. Al respecto BDD elige utilizar un formato semi-formal para la especificación del comportamiento que se toma de las especificaciones de las historias de usuario en el campo de análisis orientado a objetos y al diseño. BDD especifica que los analistas de negocio y desarrolladores deben colaborar en este campo y debe especificar el comportamiento en términos de casos de uso, especificando cada uno de forma explícita, por escrito, en un documento específico. Cada historia de usuario debe, de alguna manera, seguir la estructura siguiente:

##### **Título**

La historia debe tener un título claro y explícito.

##### **Narrativa**

Una sección de resumen, introducción que especifica:

- Quién (qué rol de negocio o de proyecto) es el conductor o actor principal de la historia (el actor que se beneficia de negocios de la historia)
- Qué resultado desea el interesado que la historia tenga
- Cuál es el valor de negocio que el stakeholder derivará de este efecto

### **Escenarios o criterios de aceptación**

Escenario es una descripción de cada caso concreto de la narración. Un escenario tiene la estructura siguiente:

- Se inicia mediante la especificación de la condición inicial que se supone que es cierto al principio del escenario. Esto puede consistir en una sola cláusula, o varias.
- A continuación, afirma el evento que desencadena el inicio de la situación.
- Por último, se establece el resultado esperado, en una o más cláusulas.

BDD no tiene ningún requisito formal sobre estas historias de usuario, pero insiste en que cada equipo que utiliza BDD debe tener un formato simple estandarizado para escribir las historias de usuario que incluye los elementos antes mencionados. Sin embargo, en 2007 Dan North sugiere un modelo de formato de texto que ha encontrado amplio seguimiento en las diferentes herramientas de software BDD. Un ejemplo muy breve de este formato podría tener este aspecto:

**Story:** *Returns go to stock*

**In order to** *keep track of stock*

**As a** *store owner*

*I want to add items back to stock when they're returned*

**Scenario 1:** *Refunded items should be returned to stock*

**Given** *a customer previously bought a black sweater from me*

**And** *I currently have three black sweaters left in stock*

**When** *he returns the sweater for a refund*

**Then** *I should have four black sweaters in stock*

**Scenario 2:** *Replaced items should be returned to stock*

**Given** *that a customer buys a blue garment*

**And** *I have two blue garments in stock*

**And** *three black garments in stock.*

**When** *he returns the garment for a replacement in black,*

**Then** *I should have three blue garments in stock*

**And** *two black garments in stock*

Los escenarios están muy bien redactados, de forma declarativa y no imperativa, en el lenguaje del negocio, sin hacer referencia a los elementos de la interfaz de usuario, a través de las cuales las interacciones tienen lugar.

### **2.3.2.2. ESPECIFICACIÓN COMO UN LENGUAJE UBICUO**

BDD toma prestado el concepto de lenguaje ubicuo del Domain Driven Design. Un lenguaje ubicuo es semi-formal y es compartido por todos los miembros de un equipo de desarrollo de software, tanto los desarrolladores de software y el personal no técnico. El lenguaje en cuestión es a la vez utilizado y desarrollado por todos los miembros del equipo como un medio común de debatir el dominio

del software en cuestión. En esta manera, BDD se convierte en un vehículo para la comunicación entre todos los roles en un proyecto de software.

BDD utiliza la especificación del comportamiento deseado como un lenguaje omnipresente para los miembros del equipo del proyecto. Esta es la razón por la que BDD insiste en un lenguaje semi-formal para la especificación del comportamiento: cierta formalidad es un requisito para ser un lenguaje ubicuo. Además, tener un lenguaje tan ubicuo crea un modelo de dominio de especificaciones, por lo que las especificaciones pueden ser razonadas formalmente.

### **2.3.3. PRÁCTICAS RECOMENDADAS POR BDD**

- Involucrar a los interesados en el proceso
- Usar ejemplos para describir el comportamiento de la aplicación, o de unidades de código
- Automatizar estos ejemplos para proveer una rápida retroalimentación y pruebas de regresión

### **2.3.4. CICLO DE DESARROLLO BDD**

1. **Discusión:** Desarrolladores, testers, expertos de dominio y el dueño de producto se reúnen y discuten la historia, gradualmente descomponiendo y destilando el comportamiento en un conjunto de especificaciones simples.
2. **Decisión:** El dueño de producto decide cuando la especificación cumple a satisfacción el comportamiento esperado y está de acuerdo con los casos de ejemplo y demostración de la historia y cierra el alcance de la misma.

3. **Desarrollo:** Los testers refinan los ejemplos de la especificación y los desarrolladores instrumentan las especificaciones creando primero pruebas que fallan y después implementando la funcionalidad.
4. **Demostración:** Una vez que todas las pruebas pasan, la historia puede ser demostrada al dueño de producto.

## 2.4. ENTREGABLES DEL PROYECTO

Tabla 2.2. Entregables del Proyecto

Fase	Actividad	Documento
<b>Inicio</b>	Iniciación del proyecto	Visión del sistema
	Planeación del proyecto	Plan de desarrollo de software
	Identificación de requerimientos	Especificación de requisitos de software
<b>Elaboración</b>	Desarrollo de la arquitectura	Documento de arquitectura del sistema
<b>Construcción</b>	Definición de pruebas de la solución	Plan de pruebas de software
	Construcción de la solución utilizando BDD: <ul style="list-style-type: none"> <li>• Traducción de pruebas a código</li> <li>• Desarrollo del sistema</li> </ul>	<ul style="list-style-type: none"> <li>• Creación de archivos con los escenarios de prueba.</li> <li>• Codificación de la solución basándose en el comportamiento definido en las pruebas.</li> </ul>
<b>Transición</b>	Despliegue de la solución	Manual de instalación

## 2.5. RUBY ON RAILS

La plataforma de desarrollo escogido para este proyecto es Ruby on Rails, el mismo que ha marcado y definido muchas tendencias en el desarrollo de software

en los últimos años. Algunas de las grandes empresas que han usado Ruby on Rails como base para sus plataformas incluyen a Twitter y Tumbler.

Se han tomado como referencia los siguientes documentos para el desarrollo de esta sección: “The Rails 3 Way” (Fernandez, 2010), “Crafting Rails Applications: Expert Practices for Everyday Rails Development” (Valim, 2011), “Rails Recipes: Rails 3 Edition” (Fowler, 2012), “Ruby on Rails Guides: Getting Started with Rails” (Dev, Noria, Zimdars, Bigg, Cheung, & Darell, 2012), “Agile Web Development with Rails” (Ruby, Thomas, & Heinemeier Hansson, 2011), y “Beginning Rails 3” (Carneiro & Al Barazi, 2010).

Ruby on Rails, también conocido como RoR o Rails es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración. El lenguaje de programación Ruby permite la meta-programación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. Rails se distribuye a través de RubyGems, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones Ruby. (Arancibia, 2010)

### **2.5.1. FILOSOFÍA DE RUBY ON RAILS**

Los principios fundamentales de Ruby on Rails incluyen “don't repeat yourself” (DRY), convención sobre configuración y el patrón REST.

DRY significa que las definiciones deberían hacerse una sola vez. Dado que Ruby on Rails es un framework de pila completa, los componentes están integrados de manera que no hace falta establecer puentes entre ellos. Por ejemplo, en ActiveRecord, las definiciones de las clases no necesitan especificar los nombres de las columnas; Ruby puede averiguarlos a partir de la propia base de datos, de forma que definirlos tanto en el código como en el programa sería redundante. (Sanchez & Campos, 2013)

Convención sobre configuración significa que el programador sólo necesita definir aquella configuración que no es convencional. Por ejemplo, si hay una clase "Historia" en el modelo, la tabla correspondiente de la base de datos es historias, pero si la tabla no sigue la convención (por ejemplo "blogposts") debe ser especificada manualmente (`set_table_name "blogposts"`). Así, cuando se diseña una aplicación partiendo de cero sin una base de datos preexistente, el seguir las convenciones de Rails significa usar menos código (aunque el comportamiento puede ser configurado si el sistema debe ser compatible con un sistema heredado anterior). (Cerezo & Centeno, 2008)

REST es una técnica de arquitectura software para sistemas hipermedia distribuidos tal como la World Wide Web. (Mena & Bernal, 2011)

### **2.5.2. LA ARQUITECTURA MVC**

En el núcleo de Rails se tiene la arquitectura Modelo Vista Controlador, usualmente llamada MVC.

Esta arquitectura presenta los siguientes beneficios:

- Aislamiento entre la lógica de negocio y la interfaz de usuario

- Facilidad de mantenimiento de código (DRY)
- Indica claramente donde los diferentes tipos de código deben ubicarse para facilitar el mantenimiento de los mismos.

#### **2.5.2.1. MODELO**

Un modelo representa los datos de la aplicación y las reglas para manipularlos. En el caso de Rails, los modelos se utilizan principalmente para la gestión de las reglas de la interacción con la tabla correspondiente en la base de datos. En la mayoría de los casos, cada tabla en la base de datos se corresponde con un modelo en la aplicación. La mayor parte de la lógica de negocio de la aplicación se concentrará en los modelos.

#### **2.5.2.2. VISTA**

La vista representa la interfaz de usuario de la aplicación. En Rails, las vistas son con frecuencia archivos HTML con código Ruby embebido que realizan tareas relacionadas únicamente con la presentación de los datos. Las vistas se encargan de proporcionar datos al navegador web u otra herramienta que se utiliza para hacer las peticiones de la aplicación.

#### **2.5.2.3. CONTROLADOR**

En MVC, las clases del controlador responden a la interacción del usuario e invocan a la lógica de la aplicación, que a su vez manipula los datos de las clases del Modelo y muestra los resultados usando las Vistas. En las aplicaciones web basadas en MVC, los métodos del controlador son invocados por el usuario usando el navegador web. (Dev, Noria, Zimdars, Bigg, Cheung, & Darell, 2012)

### **2.5.3. COMPONENTES DE RAILS**

Rails posee varios componentes, los mismos que se describen a continuación:

- Action Pack
  - Action Controller
  - Action Dispatch
  - Action View
- Action Mailer
- Active Model
- Active Record
- Active Resource
- Active Support
- Railties

#### **2.5.3.1. ACTION PACK**

Action Pack es una sola gema<sup>5</sup> que contiene Action Controller, Action View y Action Dispatch. Es decir la vista y controlador (VC) de MVC.

##### **2.5.3.1.1. ACTION CONTROLLER**

Action Controller es el componente que maneja los controladores en una aplicación Rails. El marco de acción del controlador procesa las solicitudes entrantes a una aplicación Rails, la extracción de parámetros, y el envío de los mismos a la acción prevista. Los servicios prestados por Action Controller incluyen la gestión de sesiones, la representación de plantillas, y la gestión de redireccionamiento.

---

<sup>5</sup> Gema: Es un plug-in y/o código añadido a un proyecto Rails.

### **2.5.3.1.2. ACTION VIEW**

Action View gestiona las vistas de una aplicación Rails. Permite crear salidas HTML, XML y JSON por defecto. Además, maneja el renderizado<sup>6</sup> de plantillas, incluyendo plantillas anidadas y parciales, e incluye soporte integrado AJAX.

### **2.5.3.1.3. ACTION DISPATCH**

Action Dispatch controla el enrutamiento de peticiones web y las envía como se desee, ya sea para la misma aplicación o cualquier otra aplicación Rack.

### **2.5.3.2. ACTION MAILER**

Action Mailer es un framework para la creación de servicios de correo electrónico. Se puede utilizar Action Mailer para recibir y procesar el correo entrante, así como enviar mensajes de texto con formato sencillo o correos electrónicos más complejos, basados en plantillas flexibles.

### **2.5.3.3. ACTIVE MODEL**

Proporciona una interfaz definida entre los servicios de la gema Action Pack y gemas de Object Relationship Mapping<sup>7</sup>, como Active Record. Active Model permite a Rails utilizar otros frameworks ORM en lugar de Active Record en el caso que la aplicación así lo necesite.

---

<sup>6</sup> Renderizar: es la acción de asignar y calcular todas las propiedades de un objeto antes de mostrarlo en pantalla.

<sup>7</sup> ORM: es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

#### **2.5.3.4. ACTIVE RECORD**

Es la base de los modelos en una aplicación Rails. Proporciona independencia a nivel de la base de datos, además de la funcionalidad básica CRUD<sup>8</sup>, capacidades avanzadas de búsqueda, y la capacidad de relacionar modelos, entre otros servicios.

#### **2.5.3.5. ACTIVE RESOURCE**

Proporciona un framework para manejar la conexión entre los objetos de negocio y servicios web RESTful. Implementa una forma de asignar los recursos web para objetos locales con la semántica CRUD.

#### **2.5.3.6. ACTIVE SUPPORT**

Es una extensa colección de clases utilitarias y extensiones de la biblioteca estándar de Ruby que se utilizan en Rails, tanto por el código del núcleo así como por la aplicación.

#### **2.5.3.7. RAILTIES**

Es el núcleo de Rails que permite construir nuevas aplicaciones en base al mismo y une los diversos frameworks y plug-ins<sup>9</sup> en cualquier aplicación. (Dev, Noria, Zimdars, Bigg, Cheung, & Darell, 2012)

### **2.5.4. REPRESENTATIONAL STATE TRANSFER (REST)**

Leonardo de Seta en su documento “Introducción a los servicios RESTful” (2008), indica que:

---

<sup>8</sup> CRUD: es el acrónimo de Crear, Obtener, Actualizar y Borrar (del original en inglés: Create, Read, Update and Delete).

<sup>9</sup> Plug-in: es un conjunto de componentes de software que agrega capacidades específicas a un sistema más grande.

REST define un set de principios arquitectónicos por los cuales se diseñan servicios web focalizándose en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP hacia clientes escritos en diversos lenguajes. REST emergió en los últimos años como el modelo predominante para el diseño de servicios. De hecho, REST logró un impacto tan grande en la web que prácticamente logró desplazar a SOAP y las interfaces basadas en WSDL por tener un estilo bastante más simple de usar.

#### **2.5.4.1. PRINCIPIOS DE REST**

Una implementación concreta de un servicio web REST sigue cuatro principios de diseño fundamentales:

- Utiliza los métodos HTTP de manera explícita
- No mantiene estado
- Expone URIs con forma de directorios
- Transfiere XML, JavaScript Object Notation (JSON), o ambos

##### **2.5.4.1.1. REST UTILIZA LOS MÉTODOS HTTP DE MANERA EXPLÍCITA**

Una de las características claves de los servicios web REST es el uso explícito de los métodos HTTP, siguiendo el protocolo definido por el RFC 2616. Por ejemplo, HTTP GET se define como un método productor de datos, cuyo uso está pensado para que las aplicaciones cliente obtengan recursos, busquen datos de un servidor web, o ejecuten una consulta esperando que el servidor web la realice y devuelva un conjunto de recursos.

REST hace que los desarrolladores usen los métodos HTTP explícitamente de manera que resulte consistente con la definición del protocolo. Este principio de

diseño básico establece una asociación uno-a-uno entre las operaciones de crear, leer, actualizar y borrar y los métodos HTTP. De acuerdo a esta asociación, se usa:

- POST para crear un recurso en el servidor
- GET para obtener un recurso
- PUT para cambiar el estado de un recurso o actualizarlo
- DELETE para eliminar un recurso

#### **2.5.4.1.2. REST NO MANTIENE ESTADO**

Los servicios web REST necesitan escalar para satisfacer una demanda en constante crecimiento. Se usan clústeres<sup>10</sup> de servidores con balanceadores de carga y alta disponibilidad, proxies<sup>11</sup>, y gateways<sup>12</sup>, que permiten conformar una topología de servicio, con el fin de transferir peticiones de un equipo a otro para disminuir el tiempo total de respuesta de una invocación al servicio web. El uso de servidores intermedios para mejorar la escalabilidad hace necesario que los clientes de servicios web REST envíen peticiones completas e independientes; es decir, se deben enviar peticiones que incluyan todos los datos necesarios para cumplir el pedido, de manera que los componentes en los servidores intermedios puedan redireccionar y gestionar la carga sin mantener el estado localmente entre las peticiones.

---

<sup>10</sup> Es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio.

<sup>11</sup> Proxy: Es un programa o dispositivo que realiza una acción en representación de otro

<sup>12</sup> Gateway: Es un dispositivo que permite interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación

Una petición completa e independiente hace que el servidor no tenga que recuperar ninguna información de contexto o estado al procesar la petición. Una aplicación o cliente de servicio web REST debe incluir dentro del encabezado y del cuerpo HTTP de la petición todos los parámetros, contexto y datos que necesita el servidor para generar la respuesta. De esta manera, el no mantener estado mejora el rendimiento de los servicios web y simplifica el diseño e implementación de los componentes del servidor, ya que la ausencia de estado en el servidor elimina la necesidad de sincronizar los datos de la sesión con una aplicación externa.

#### 2.5.4.1.2.1. SERVICIOS CON ESTADO VS. SIN ESTADO

La Figura 2.2. Servicio con Estado. muestra un servicio con estado, del cual una aplicación realiza peticiones para la página siguiente en un conjunto de resultados multi-página, asumiendo que el servicio mantiene información sobre la última página que pidió el cliente. En un diseño con estado, el servicio incrementa y almacena en algún lugar una variable `paginaAnterior` para poder responder a las peticiones siguientes.

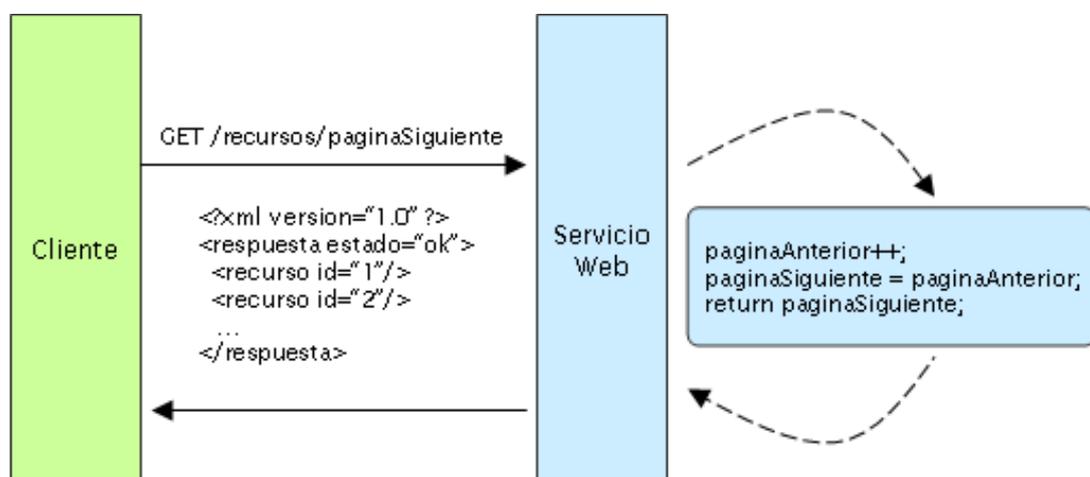


Figura 2.2. Servicio con Estado. Fuente: (De Seta, 2008)

Los servicios con estado tienden a volverse complicados. Un entorno de servicios con estado necesita bastante análisis y diseño desde el inicio para almacenar los datos eficientemente y sincronizar la sesión del cliente dentro de un clúster de servidores, siendo esta operación costosa en procesamiento, lo que impacta negativamente en el rendimiento general del servidor.

Por otro lado, los servicios sin estado son mucho más simples de diseñar, escribir y distribuir a través de múltiples servidores. Un servicio sin estado no sólo funciona mejor, sino que además mueve la responsabilidad de mantener el estado al cliente de la aplicación. En un servicio web REST, el servidor es responsable de generar las respuestas y proveer una interfaz que le permita al cliente mantener el estado de la aplicación por su cuenta. Por ejemplo, en el mismo ejemplo de una petición de datos en múltiples páginas, el cliente debería incluir el número de página a recuperar en vez de pedir "la siguiente", tal como se muestra en la Figura 2.3. Servicio sin Estado. Fuente: , a continuación:

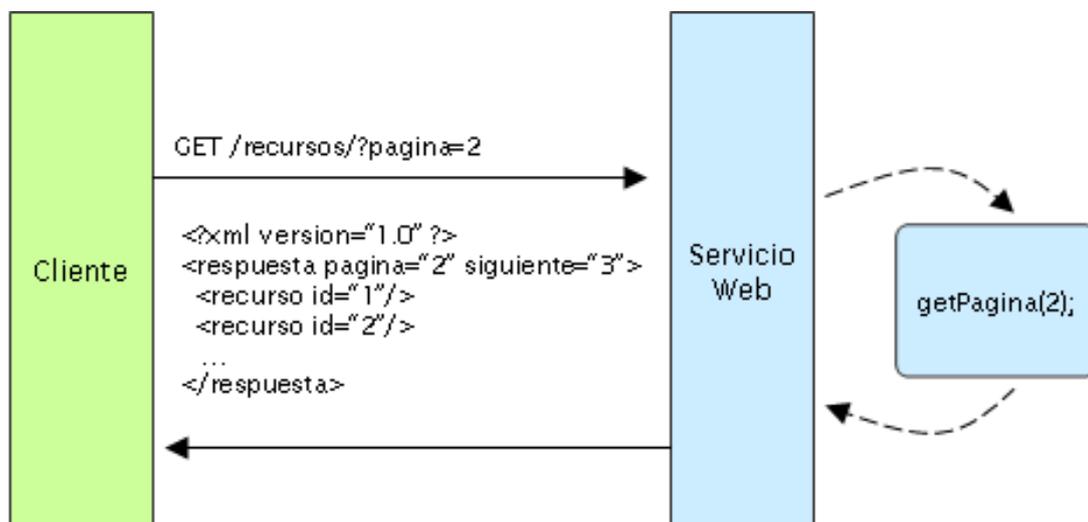


Figura 2.3. Servicio sin Estado. Fuente: (De Seta, 2008)

Un servicio web sin estado genera una respuesta que se enlaza a la siguiente página del conjunto y le permite al cliente hacer todo lo que necesita para almacenar la página actual.

#### **2.5.4.1.3. REST EXPONE URIS CON FORMA DE DIRECTORIOS**

Desde el punto de vista del cliente de la aplicación que accede a un recurso, la URI determina qué tan intuitivo va a ser el servicio web REST, y si el servicio va a ser utilizado tal como fue pensado al momento de diseñarlo. La tercera característica de los servicios web REST es justamente sobre las URIs.

Las URI de los servicios web REST deben ser intuitivas, hasta el punto de que sea fácil adivinarlas. Se considera a las URI como una interfaz auto-documentada que necesita de muy poca o ninguna explicación o referencia para que un desarrollador pueda comprender a lo que apunta, y a los recursos derivados relacionados.

Una forma de lograr este nivel de usabilidad es definir URIs con una estructura de directorios. Este tipo de URIs es jerárquica, con una única ruta raíz, y va abriendo ramas a través de las sub-rutas para exponer las áreas principales del servicio. De acuerdo a esta definición, una URI no es solamente una cadena de caracteres delimitada por barras, sino más bien un árbol con subordinados y padres organizados como nodos. Por ejemplo, en un servicio de hilos de discusiones que tiene temas varios, se podría definir una estructura de como esta:

```
http://www.miservicio.org/discusion/temas/{tema}
```

La raíz, /discusion, tiene un nodo /temas como hijo. Bajo este nodo hay un conjunto de nombres de temas (como pueden ser tecnología, actualidad, y más),

cada uno de los cuales apunta a un hilo de discusión. Dentro de esta estructura, resulta fácil recuperar hilos de discusión al agregar algo después de /temas/.

Se puede también enumerar algunas guías generales más para el momento de crear URIs para un servicio web REST:

- Ocultar la tecnología usada en el servidor que aparecería como extensión de archivos (.jsp, .php, .asp), de manera de poder portar la solución a otra tecnología sin cambiar las URI.
- Mantener todo en minúsculas.
- Sustituir los espacios con guiones o guiones bajos (uno u otro).
- Evitar el uso de cadenas de consulta.
- En vez de usar un 404 Not Found<sup>13</sup> si la petición es una URI parcial, devolver una página o un recurso predeterminado como respuesta.

Las URI deberían ser estáticas de manera que cuando cambie el recurso o cambie la implementación del servicio, el enlace se mantenga igual. Esto permite que el cliente pueda generar "favoritos". También es importante que la relación entre los recursos que está explícita en las URI se mantenga independiente de las relaciones que existen en el medio de almacenamiento del recurso.

#### **2.5.4.1.4. REST TRANSFIERE XML, JSON, O AMBOS**

La última restricción al momento de diseñar un servicio web REST tiene que ver con el formato de los datos que la aplicación y el servicio intercambian en las

---

<sup>13</sup> 404 Not found: Es un código de estado HTTP que indica que el host ha sido capaz de comunicarse con el servidor, pero no existe el fichero que ha sido solicitado.

peticiones/respuestas. Aquí es donde realmente vale la pena mantener las cosas simples, legibles por humanos y computadoras.

Los objetos del modelo de datos generalmente se relacionan de alguna manera, y las relaciones entre los objetos del modelo de datos (los recursos) deben reflejarse en la forma en la que se representan al momento de transferir los datos al cliente. En el servicio de hilos de discusión que se tomó como ejemplo, una representación de un recurso conectado podría ser un tema de discusión raíz con todos sus atributos, y links embebidos a las respuestas al tema.

```
<discusion fecha="{fecha}" tema="{tema}">
  <comentario>{comentario}</comentario>
  <respuestas>
    <respuesta de="foo@mail.com"
href="/discusion/temas/{tema}/foo"/>
    <respuesta de="bar@mail.com"
href="/discusion/temas/{tema}/bar"/>
  </respuestas>
</discusion>
```

Por último, es bueno construir los servicios de manera que usen el atributo HTTP Accept del encabezado, en donde el valor de este campo es un tipo MIME. De esta manera, los clientes pueden pedir por un contenido en particular que mejor pueden analizar. Algunos de los tipos MIME más usados para los servicios web REST son:

Tabla 2.3. MIME-Type / Content-Type. Fuente: (De Seta, 2008)

<b>MIME-Type<sup>14</sup></b>	<b>Content-Type<sup>15</sup></b>
<b>JSON</b>	application/json
<b>XML</b>	application/xml
<b>XHTML</b>	application/xhtml+xml

Esto permite que el servicio sea utilizado por distintos clientes escritos en diferentes lenguajes, corriendo en diversas plataformas y dispositivos. El uso de los tipos MIME y del encabezado HTTP Accept es un mecanismo conocido como negociación de contenido, el cual le permite a los clientes elegir qué formato de datos puedan leer, y minimiza el acoplamiento de datos entre el servicio y las aplicaciones que lo consumen.

## **2.6. HERRAMIENTAS PARA EL DESARROLLO DEL PROYECTO**

### **2.6.1. SISTEMA DE CONTROL DE VERSIONES DISTRIBUIDO**

Las buenas prácticas de ingeniería de software modernas incluyen la utilización de sistemas de control de versiones (VCS por sus siglas en inglés). Estas son parte clave de los entornos de desarrollo modernos, donde interactúan con servidores de integración, soluciones de gestión de requerimientos, etc. Este tipo de herramientas ha evolucionado en funcionalidad y arquitectura. Actualmente, se tiene como alternativas utilizarlas ya sea en la red local o en la nube.

---

<sup>14</sup> Mime-Type: Multipurpose Internet Mail Extensions son una serie de convenciones dirigidas al intercambio a través de Internet de todo tipo de archivos de forma transparente para el usuario.

<sup>15</sup> Content-Type: indica el tipo de medio que representa el contenido del mensaje, consiste en un tipo: type y un subtipo: subtype

Por lo tanto, es indispensable que todo miembro de un equipo de desarrollo de software utilice y domine una herramienta de este tipo para poder aplicar las demás buenas prácticas de la ingeniería de software moderna.

Con el software de control de versiones se puede seguir la pista al historial del proyecto y revertir cualquier edición cuando se cambie de idea. Además, se puede crear y guardar evidencia de múltiples versiones y ramas del trabajo ejecutado. Todos los proyectos de software usan invariablemente algún tipo de control de versiones.

#### **2.6.1.1. DEFINICIÓN**

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. El Control de Versiones es una manera ordenada de mantener un rastro de modificaciones y almacenar versiones y revisiones específicas.

Un sistema de control de versiones (o sistema de control de revisiones) es una combinación de tecnologías y prácticas para seguir y controlar los cambios realizados en los archivos de un proyecto.

#### **2.6.1.2. IMPORTANCIA DEL CONTROL DE VERSIONES**

- La razón por la cual el control de versiones es universal es porque ayuda virtualmente en todos los aspectos al dirigir un proyecto: comunicación entre los desarrolladores, manejo de los lanzamientos, administración de fallos, estabilidad entre el código y los esfuerzos de desarrollo experimental y, atribución y autorización en los cambios de los desarrolladores. El sistema

de control de versiones permite a una fuerza coordinadora central abarcar todas estas áreas.

- El núcleo del sistema es la gestión de cambios: identificar cada cambio a los archivos del proyecto, anotar cada cambio con meta-data como la fecha y el autor de la modificación y disponer esta información en cualquier momento.
- Es un mecanismo de comunicación donde el cambio es la unidad básica de información.
- Permite revertir archivos a un estado anterior, revertir un proyecto completo a un estado anterior, comparar cambios a lo largo del tiempo, revisar quién realizó la última modificación, quién introdujo un error y cuándo, entre otros.
- También significa que se pueden recuperar fácilmente archivos perdidos.
- Además, la obtención de todos esos beneficios resultan a un coste muy bajo.

### 2.6.1.3. TERMINOLOGÍA BÁSICA

- **Repositorio:** Es el lugar donde se almacena el código fuente por lo general este es accesible a todos los usuarios que colaboran o que siguen el desarrollo.
- **CheckOut:** es la acción de desbloquear una copia de trabajo por lo general, esta copia se almacena en el equipo local.
- **Tag:** es una copia exacta de una versión específica, es decir, esta versión esta liberada y no sufrirá más cambios.
- **Trunk/master:** esta es la rama principal de desarrollo por lo general todo el desarrollo de un proyecto se lleva a cabo sobre esta rama.

- **Branch, o ramas:** son bifurcaciones del proyecto, estas pueden llevar un nuevo feature, o características totalmente inestables destinadas a pruebas de concepto, o un desarrollo específico con fines específicos.
- **Commit, o push:** es la acción de enviar cambios locales al servidor.

#### 2.6.1.4. GIT

Git es un sistema de control de versiones distribuido, de código abierto, y gratuito destinado a manejar desde pequeños a grandes proyectos con velocidad y eficiencia. Y la palabra clave es “distribuido”, esto significa que repositorio central no existe, y el repositorio está distribuido por todos los participantes. Es decir todos los participantes tienen en local todo el histórico, etiquetas, ramas, etc. La gran ventaja de esto es que no se necesita estar conectado a la red para hacer cualquier operación contra el repositorio, por lo que el trabajo es mucho más rápido y con menos dependencias.

El acceso a cada repositorio se realiza a través del protocolo de Git, montado sobre ssh, o usando HTTP aunque no es necesario ningún servidor web para poder publicar el repositorio en la red.

Git está especialmente indicado cuando:

- No se dispone de conexión continua o la conexión es muy lenta.
- Para equipos muy grandes, con poca comunicación o en distintas localizaciones geográficas.
- Cuando miembros del equipo tienen que viajar mucho y no pueden dejar el desarrollo.

- Cuando se quiere que cualquier persona colabore con el proyecto, sin necesidad de que pertenezca a la organización o tenga permisos de acceso (en estos casos para subir las colaboraciones, se suele hacer a base de parches).
- Cuando los cambios sobre el código son muy grandes (con Git es muy fácil trabajar con branches y merge).
- Para trabajar con nuevos modelos descentralizados (se podría hacer, por ejemplo, una jerarquía de repositorios donde los cambios fueran subiendo de las hojas del árbol a la raíz en base de aprobaciones y revisiones por los nodos intermedios).

#### **2.6.1.4.1.OBJETOS DE GIT**

Hay 4 tipos de objetos en git (el más importante a entender es el commit).

- **Blob:** se usa para almacenar datos de archivos, es generalmente un archivo.
- **Tree:** es, básicamente, como un directorio, hace referencia a un conjunto de otros trees y/o blobs (por ej. archivos y subdirectorios).
- **Commit:** apunta a un determinado tree, marcando como era en un momento determinado (quien no haya entendido lo que es un tree, sustituya la palabra tree por archivo). Contiene información sobre ese momento determinado, los cambios del autor desde el último commit, el commit anterior (conocido como parent), etc. También se puede entender un commit, de una forma más imprecisa y coloquial, como la modificación o el conjunto de modificaciones a uno o varios archivos del repositorio. Otra

forma de entenderlo también sería, como una "foto" de uno o varios archivos del repositorio en un momento determinado.

- **Tag:** es una forma de marcar un commit como específico de alguna forma. Se usa normalmente para marcar algunos commits como releases específicos o algo destacable en esas líneas.

#### **2.6.1.5. GITHUB**

Github es una plataforma de desarrollo colaborativo de software para alojar proyectos usando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

Github no sólo ofrece alojamiento del código si no muchas más posibilidades asociadas a los repositorios como son: forks, issues, pull requests, diffs, etc.

#### **2.6.2. COFFEESCRIPT**

Iván Jaimes menciona en su documento electrónico "CoffeeScript" (2012) que:

CoffeeScript es un lenguaje de programación que compila a JavaScript. Dicho lenguaje añade "azúcar sintáctico" (syntactic sugar) inspirado en Ruby, Python and Haskell para mejorar la legibilidad y brevedad de JavaScript, agregando también características especiales como "Lista de Comprensión" (List Comprehension) y "Coincidencia de Patrones" (Pattern Matching).

CoffeeScript compila de forma predictiva el código de JavaScript haciendo que los programas se generen con menos líneas de código, sin afectar el desempeño o rendimiento de la aplicación.

### **2.6.3. TWITTER BOOTSTRAP**

Txema Rodríguez (Bootstrap, 2012) menciona que: Bootstrap es un framework que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. Ha sido desarrollado por Twitter que recientemente liberó su versión 2.0. La mayor ventaja es que se puede crear interfaces que se adapten a los distintos navegadores (responsive design), al mismo tiempo que se apoya en un framework potente con numerosos componentes webs que ahorran mucho esfuerzo y tiempo.

#### **2.6.3.1. CARACTERÍSTICAS PRINCIPALES**

Bootstrap ofrece una serie de plantillas CSS y archivos Javascript que permiten integrar el framework de forma sencilla en los proyectos webs:

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tablets y móviles a distintas escalas y resoluciones.
- Se integra con las principales librerías Javascript, por ejemplo JQuery.
- Ofrece un diseño sólido usando LESS y estándares como CSS3/HTML5.
- Es un framework ligero que se integra de forma limpia.
- Funciona con todos los navegadores, incluido Internet Explorer usando HTML Shim para que reconozca los tags HTML5.
- Dispone de distintos layouts predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.

### 2.6.3.2. COMPONENTES

Bootstrap ofrece distintos componentes para usar con estilos predefinidos y fáciles de configurar. Además de los distintos plugins integrados con jQuery que se disponen en la versión 2.0 de Bootstrap.

- **Botones:** se pueden aplicar tanto a enlaces como a etiquetas button/input simplemente usando la clase btn. Así se puede distinguir el propósito de cada botón con los distintos estilos prefijados o variar su tamaño.
- **Dropdown:** con ellos se puede asociar un menú desplegable a un botón de forma sencilla agrupando distintas opciones.
- **Formularios:** Bootstrap cuenta con distintas disposiciones que se pueden adaptar a diferentes necesidades. Además de incluir distintos elementos para remarcar distintas acciones sobre los formularios.
- **Plug-ins de jQuery:** se cuenta con distintos plug-ins que permiten crear modales, o agregar un tooltip sobre algún elemento.

### 2.6.4. RESPONSIVE DESIGN

El diseño web adaptativo o adaptable (en inglés, Responsive Web Design) es una técnica de diseño y desarrollo web que mediante el uso de estructuras e imágenes fluidas, así como de media-queries en la hoja de estilo CSS, consigue adaptar el sitio web al entorno del usuario. (Labrada & Salgado, 2013)

El diseñador y autor norteamericano Ethan Marcotte creó y difundió esta técnica a partir de una serie de artículos en “A List Apart”, una publicación en línea especializada en diseño y desarrollo web.

#### **2.6.4.1. ORIGEN**

La idea y el propósito del diseño web adaptativo fue previamente discutida y descrita por el consorcio W3C<sup>16</sup> en julio de 2008 en su recomendación "Mobile Web Best Practices" bajo el subtítulo "One Web".

Dicha recomendación, aunque específica para dispositivos móviles, puntualiza que está hecha en el contexto de "One Web", y que por lo tanto engloba no solo la experiencia de navegación en dispositivos móviles sino también en dispositivos de mayor resolución de pantalla como dispositivos de sobremesa.

El concepto de "One Web" hace referencia a la idea de construir una Web para todos (Web for All) y accesible desde cualquier tipo de dispositivo (Web on Everything).

Hoy en día, la variedad de dispositivos existente en el mercado ha provocado que la información disponible no sea accesible desde todos los dispositivos, o bien es accesible pero la experiencia de navegación es muy pobre.

#### **2.6.4.2. VENTAJAS**

Con una sola versión en HTML y CSS se cubren todas las resoluciones de pantalla, es decir, el sitio web creado estará optimizado para todo tipo de dispositivos: PCs, tabletas, teléfonos móviles, etc. Esto mejora la experiencia de usuario a diferencia de lo que ocurre, por ejemplo, con sitios web de ancho fijo cuando se acceden desde dispositivos móviles.

---

<sup>16</sup> W3C: es una comunidad internacional donde las organizaciones Miembro, personal [ingles] a tiempo completo y el público en general trabajan conjuntamente para desarrollar estándares Web.

De esta forma se reducen los costes de creación y mantenimiento, pues se evita tener que desarrollar aplicaciones específicas para versiones móviles, por ejemplo, una aplicación específica para móviles Android, otra para iPhone, etc.

Desde el punto de vista de la optimización de motores de búsqueda, sólo aparecería una URL en los resultados de búsqueda, con lo cual se ahorran redirecciones y los fallos que se derivan de éstas. También se evitarían errores al acceder al sitio web en concreto desde los llamados "social links", es decir, desde enlaces que los usuarios comparten en medios sociales tales como Facebook, Twitter, etc y que pueden acabar en error dependiendo de qué enlace se copió (desde qué dispositivo se copió) y desde qué dispositivo se accede. (Fernández, 2012)

#### **2.6.5. DISQUS**

Disqus es un servicio en línea que proporciona una plataforma informática centralizada para discusiones en sitios web, ofrece las siguientes ventajas:

- Moderación de comentarios unificada.
- Capacidades de anti-spam.
- Moderación de comentarios por email.
- Sistema de comentarios anidados
- Posibilidad de comentar anónimamente o identificándose mediante una cuenta de redes sociales o plataformas como Twitter, Facebook, Google, Yahoo! o Open ID.

### **2.6.6. TEAMWORKPM.NET**

Es un software de administración de proyectos basado en web que permite llevar el control de los proyectos desde cualquier lugar y que mantiene tanto a clientes, equipo y demás involucrados del proyecto, al tanto de lo que sucede.

El panel de control que brinda esta herramienta ofrece la posibilidad de observar todos los proyectos que se estén manejando.

De igual forma ofrece características muy poderosas y fundamentales para administrar un proyecto con éxito, tales como: administración de las tareas, seguimiento de los hitos establecidos, contacto directo mediante mensajería, administración de documentos, control y seguimiento del tiempo, además de otras atractivas funciones para mantener un riguroso monitoreo del proyecto.

### **2.6.7. OMNIGRAFFLE**

OmniGraffle es una aplicación para diagramación creada por The Omni Group. Puede ser usado para crear diagramas, diagramas de flujo, cuadros organizacionales e ilustraciones. Dispone de una interface de drag-and-drop y WYSIWYG (Lo que ve es lo que obtiene por sus siglas en inglés). Dispone de extensiones que permiten expandir sus funcionalidades básicas.

Permite utilizar técnicas de mapeo-mental para organizar información, así como también un manejador de estilos, que permite el diseño de prototipos de páginas web y documentos PDF, así como también crear bosquejos o documentos con lluvias de ideas.

### **2.6.8. OMNIPLAN**

OmniPlan es un software para planificación y gestión de proyectos de The Omni Group. Está diseñado para ayudar a visualizar, mantener y simplificar el manejo de proyectos. Permite dividir las tareas, optimizar los recursos necesarios, controlar los costos y tener una visión general de todo el proyecto.

OmniPlan ofrece funcionalidades tales como: diagramas de Gantt, calendarios, resúmenes, hitos y resalta las rutas críticas para tomar las mejores decisiones. Desde vistas personalizables a la entrada rápida de datos, OmniPlan le ayuda a administrar los proyectos que son tan simples o complejos como sea necesario que sean, sin complejidad.

### **2.6.9. SUBLIME TEXT 2**

Sublime Text 2 es un editor multiplataforma de texto y código hecho en c++. El programa fue originalmente diseñado como una extensión de Vim con funcionalidades extras.

Algunas de las funcionalidades que ofrece:

- Mini mapa: una vista previa de todo el código fuente
- Posibilidad de seleccionar múltiples secciones de código.
- Edición con varios paneles.
- Marcadores dentro de diferentes archivos.
- Soporte nativo para 27 lenguajes de programación.
- Búsqueda y reemplazo basado en Regular Expressions (regex).

### **2.6.10. RUBYMINE**

Rubymine es un editor inteligente de Ruby con fragmentos de rápida adición y refactorización automática. Provee diagramas de modelos, vistas generales del proyecto y otras vistas especializadas para una fácil e intuitiva navegación así como también ofrecer una visión global del proyecto

Dispone de integración con herramientas orientadas a BDD como: Rspec, Cucumber, Shoulda, MiniTest y Test::Unit

Integración con lenguajes como HTML, CSS, JavaScript además de nuevas tecnologías como CoffeeScript, HAML, Sass y Less.

## **2.7. HERRAMIENTAS DE ANÁLISIS**

### **2.7.1. GOOGLE ANALYTICS**

Google Analytics es un servicio gratuito de estadísticas de sitios web. Ofrece información agrupada según los intereses de tres tipos distintos de personas involucradas en el funcionamiento de una página: ejecutivos, técnicos de marketing y webmasters.

Se pueden obtener informes como el seguimiento de usuarios exclusivos, el rendimiento del segmento de usuarios, los resultados de la campaña de marketing, el marketing de motores de búsqueda, las pruebas de versión de anuncios, el rendimiento del contenido, el análisis de navegación, los objetivos y proceso de redireccionamiento o los parámetros de diseño web. Este producto se desarrolló en base a la compra de Urchin (hasta entonces la mayor compañía de análisis estadístico de páginas web) por parte de Google. (Bermeo & Ortiz, 2012)

## 2.7.2. MIXPANEL

Max Sainz en su documento llamado "Las métricas de siempre están muertas" (2012) indica que muchas personas miden el éxito una web por el número de visitas únicas y páginas vistas. Estas son las métricas que se han venido utilizando durante mucho tiempo y que para la mayoría siguen midiendo el éxito de algo, pero Mixpanel tiene en mente cambiar esta concepción.

Es necesario cambiar el estándar para decidir el éxito, ya que métricas tales como los visitantes únicos y las páginas vistas son cosa del pasado. En la nueva web, el éxito se mide por el engagement<sup>17</sup> y el número de usuarios retenidos, métricas que pueden ser medidas por medio de Mixpanel.

La idea de Mixpanel es que la industria cambie, para que dejen de tenerse tanto en cuenta estas métricas que se han quedado anticuadas y son un tanto engañosas, porque la web ha pasado de ser estática a dinámica. Por ello Suhail Doshi, fundadora de Mixpanel propone medir:

- **El engagement:** no importa cómo se mida ese engagement (se mide diferente dependiendo del tipo de plataforma y lo que se ofrece), pero éste es mucho más importante que el simple hecho de que la gente haya visto tus páginas.
- **Retención:** no importa el número de consumidores/usuarios que se tenga, lo importante es que les guste lo que se ofrece y que regresen.
- **One Key Metric:** esta métrica será la clave para una empresa, y con ella se puede saber cómo está evolucionando el startup. Por ejemplo en Instagram

---

<sup>17</sup> Es el arte de crear una relación incondicional hacia una marca. En informática se mide a través del tiempo que un usuario permanece en la aplicación.

no es el número de usuarios registrados ni el número de veces que se bajan la aplicación, es el número de fotos que se suben a diario.

El Internet evoluciona rápido y para muchos las antiguas métricas siguen siendo las que cuentan, pero con el paso del tiempo será cada vez más común ver cómo lo realmente importante es el nivel de interacción y de retorno de los clientes a un sitio.

### **2.7.3. NEW RELIC**

New Relic es una aplicación web todo-en-uno, una herramienta de rendimiento que le permite ver el rendimiento de la experiencia del usuario final, a través de servidores, y hasta la línea de código de aplicación.

New Relic es la única herramienta APM que ofrece capacidades de clase empresarial, soporte multi-idioma, control de usuarios, monitoreo de servidor, profundo análisis electrónico de transacciones en una sola herramienta, ofrecida como un servicio.

#### **2.7.3.1. CAPACIDADES TÉCNICAS**

- Análisis de rendimiento: New Relic soporta los siguientes lenguajes y frameworks de aplicaciones: Ruby, Rails, PHP, Java, .Net, Python.
- Monitoreo real de usuarios: Monitorea los eventos ready y load, críticos para el desempeño de la página y la percepción del usuario.
- Monitoreo de servidores: New Relic monitorea los recursos del servidor con un agente residente en el sistema operativo. Los recursos que pueden ser monitoreados son: CPU, Memoria, espacio en disco y estado de la red.
- Monitoreo de rendimiento de SQL/NoSQL

- Trazado de transacciones de aplicaciones web
- Umbrales y alertas de notificación.
- Historial de despliegue
- Monitoreo de disponibilidad
- Mapas de la arquitectura de la aplicación
- Análisis de escalabilidad.
- Reportes a nivel de servicio.
- Perfilamiento de la aplicación.
- Instrumentación y descubrimiento de aplicaciones automático

## **CAPÍTULO 3:**

### **INICIO**

#### **3.1. PLAN DE DESARROLLO DE SOFTWARE**

##### **3.1.1. INTRODUCCIÓN**

###### **3.1.1.1. PROPÓSITO**

El propósito de este documento es describir las actividades que se desarrollarán en la fase de elaboración y la fase de construcción del Sistema de Difusión de Gestión del Conocimiento de la ESPE.

###### **3.1.1.2. ALCANCE**

En este documento se especifica, en detalle, las actividades a realizar para alcanzar lo descrito en el documento de visión del Sistema de Difusión de Gestión del Conocimiento de la ESPE. Para ello, se describe el flujo de trabajo a seguir para la implementación de los casos de uso.

###### **3.1.1.3. DEFINICIONES, SIGLAS Y ABREVIACIONES**

- **Caso de Uso:** formato que proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.
- **Cronograma:** calendario de trabajo.

- **Documento Visión:** informe que ofrece una visión general de los requerimientos del proyecto, características clave y restricciones principales.
- **ESPE:** siglas de la Escuela Politécnica del Ejército.
- **Iteración:** una fase en el proceso de desarrollo del proyecto según la metodología OpenUP.
- **OpenUP:** es un método y un proceso de desarrollo de software.
- **Recurso:** conjunto de elementos disponibles para el desarrollo del proyecto.
- **Usuario:** la persona que utilizará el software.

#### 3.1.1.4. REFERENCIAS

Tabla 3.1. Referencias – Plan de desarrollo de software

Ref.	Título
1	Visión del sistema
2	Documento de especificación de requerimientos del sistema
3	Documento de arquitectura de software
4	Plan de pruebas de software

#### 3.1.1.5. DESCRIPCIÓN

El documento de plan de desarrollo de software detalla la planificación del desarrollo del proyecto, haciendo las distribuciones de tiempo necesarias para cada una de las fases.

### **3.1.2. PLAN DE PROYECTO**

Se desarrollarán los distintos documentos que involucran las fases de OpenUP, estas son: inicio, elaboración, construcción y transición. En la etapa de inicio, se desarrollarán los documentos de visión, especificación de requerimientos y el plan de desarrollo de software.

En las etapas de elaboración y construcción se generará el documento de arquitectura y serán presentados varios prototipos con sus respectivas pruebas.

#### **3.1.2.1. PROPÓSITO, ALCANCE Y OBJETIVOS DEL PROYECTO**

Desarrollar un sistema basado en la web, que permita la difusión de la gestión del conocimiento realizada dentro de la Escuela Politécnica del Ejército, facilitando así su divulgación en la comunidad politécnica y la sociedad en general.

Las funciones principales del sistema incluyen: el registro de las programas de investigación, registro de proyectos de investigación, registro de resultados de investigación, registro de otras actividades relacionadas a la gestión del conocimiento, creación de usuarios dentro del sistema, y la presentación de todos estos de una manera lógica.

#### **3.1.2.2. SUPOSICIONES Y RESTRICCIONES**

Para el desarrollo de este proyecto se cuenta con un equipo de trabajo de 3 personas.

El desarrollo de la aplicación considera las siguientes restricciones:

- El sistema está basado y propuesto para la Escuela Politécnica del Ejército, el mismo debe estar alineado con la forma de trabajo definida por la Institución.
- Considerando que la Escuela Politécnica del Ejército es una Institución del Estado se deberá utilizar tecnología de tipo Open Source.

### 3.1.2.3. PRODUCTOS DEL PROYECTO

Tabla 3.2. Productos del Proyecto

<b>Actividad</b>	<b>Iteración</b>
Documento de visión del sistema.	1
Especificación de requisitos de software	1
Plan de desarrollo de software	1
Documento de arquitectura de software	1
Bocetos de las interfaces del sistema	1
Prototipos en HTML	2
Desarrollo de pruebas	2
Primer prototipo funcional	2
Refinamiento de los requerimientos funcionales	3
Refinamiento de las pruebas	3
Segundo prototipo funcional / sistema completo	3

### 3.1.3. ORGANIZACIÓN DEL PROYECTO

#### 3.1.3.1. ESTRUCTURA ORGANIZACIONAL

La estructura organizacional incluye todos los participantes que se estimen convenientes para proporcionar los requisitos y validar el sistema. El personal del proyecto estará formado por los siguientes puestos de trabajo y personal asociado:

Tabla 3.3. Desarrollador del Proyecto de Graduación

Nombre	Santiago Ríos
Rol	Desarrollador del Proyecto de Graduación
Información de contacto	santiago.rios9@gmail.com

Tabla 3.4. Directora del Proyecto de Graduación

Nombre	Ing. Cecilia Hinojosa
Rol	Directora del Proyecto de Graduación
Información de contacto	cmhinojosa@espe.edu.ec

Tabla 3.5. Co-Director del Proyecto de Graduación

Nombre	Ing. Ramiro Delgado
Rol	Co-Director del Proyecto de Graduación
Información de contacto	rndelgado@espe.edu.ec

### **3.1.4. GERENCIA DE PROYECTOS**

#### **3.1.4.1. ESTIMACIÓN DEL PROYECTO**

Los costos generados por el proyecto serán absorbidos por el desarrollador del proyecto, y corresponden a aquellos relacionados con costos de licencias de herramientas que servirán para el desarrollo del proyecto.

Se estima una entrega de 50% de funcionalidad dentro de la primera iteración. El 100% de funcionalidad será entregada en la segunda iteración.

### 3.1.4.2. PLAN DE PROYECTO

#### 3.1.4.2.1. PLAN DE FASES

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La Tabla 3.6. Plan de Fases muestra la distribución de tiempo y el número de iteraciones de las fases de elaboración y construcción.

Tabla 3.6. Plan de Fases

<b>Fase</b>	<b>No. Iteraciones</b>	<b>Duración</b>
Fase de inicio	1	2 Semanas
Fase de elaboración	1	6 Semanas
Fase de construcción	2	12 Semanas
Fase de transición	1	2 Semanas

#### 3.1.4.2.2. OBJETIVOS DE LAS ITERACIONES

Al culminar la iteración 1 de la fase de elaboración se habrán concretado los siguientes documentos:

- Documento de visión del sistema
- Documento de especificación de requisitos de software
- Plan de desarrollo de software
- Documento de arquitectura de software
- Bocetos de las interfaces del sistema

Al culminar la iteración 1 de la fase de construcción se habrán cumplido las siguientes actividades:

- Prototipos en HTML.
- Desarrollo de pruebas.
- Primer prototipo funcional

Al culminar la iteración 2 de la fase de construcción se habrán cumplido las siguientes actividades:

- Refinamiento de los requisitos funcionales, de acuerdo a la retroalimentación obtenida por medio de la presentación del primer prototipo.
- Refinamiento de pruebas.
- Segundo prototipo funcional / sistema completo

### **3.1.4.2.3. LIBERACIÓN DE PROTOTIPOS**

Se tiene prevista la liberación de 2 prototipos antes de la presentación final del sistema:

Tabla 3.7. Liberación de Prototipos

<b>Prototipo</b>	<b>Funcionalidad</b>	<b>Liberación</b>
Prototipo 1	50% de la funcionalidad total : Permite la gestión (registrar, modificar, eliminar y visualizar) de programas, proyectos y otras actividades relacionadas.	Semana 6
Prototipo 2	100% de la funcionalidad total: Se incluyen la gestión de resultados y documentos adjuntos, de igual manera se agrega la funcionalidad de administración general del sistema.	Semana 12

## **3.2. ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE**

### **3.2.1. INTRODUCCIÓN**

#### **3.2.1.1. PROPÓSITO**

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales y del sistema para la implementación de la aplicación

web Sistema de Difusión de Gestión del Conocimiento de la ESPE, que será utilizada por estudiantes, profesores, directivos y público en general.

Este documento está estructurado en base al estándar IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998.

### 3.2.1.2. ALCANCE

El Sistema de Difusión de Gestión del Conocimiento de la ESPE basará su desarrollo en la visión y requerimientos funcionales que demanden el personal dedicado a la docencia, la investigación y la vinculación con la colectividad de la ESPE.

### 3.2.1.3. PERSONAL INVOLUCRADO

Tabla 3.8. Desarrollador del Proyecto de Graduación

Nombre	Santiago Ríos
Rol	Desarrollador del Proyecto de Graduación
Información de contacto	santiago.rios9@gmail.com

Tabla 3.9. Directora del Proyecto de Graduación

Nombre	Ing. Cecilia Hinojosa
Rol	Directora del Proyecto de Graduación
Información de contacto	cmhinojosa@espe.edu.ec

Tabla 3.10. Co-Director del Proyecto de Graduación

Nombre	Ing. Ramiro Delgado
Rol	Co-Director del Proyecto de Graduación
Información de contacto	rndelgado@espe.edu.ec

### 3.2.1.4. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

- **Programa de investigación:** Se refiere al compromiso que asume un grupo de investigadores de conducir el trabajo en una determinada línea,

perspectiva, temática científica o de desarrollo y en sostener esa dirección durante un período de tiempo más o menos prolongado. Un programa, entonces, supone establecer límites y concentrar esfuerzos e implica una labor solidaria y colectiva en tanto que lo asume un equipo, cuyos integrantes podrán trabajar en distintos proyectos pero mancomunados en los objetivos comunes que caracterizan al programa.

- **Proyecto de investigación:** es un procedimiento que siguiendo el método científico recaba todo tipo de información y formula hipótesis acerca de cierto fenómeno social o científico, empleando las diferentes formas de investigación.
- **Resultado de investigación:** son los documentos, archivos o imágenes que son alcanzados durante un proyecto de investigación.
- **Dominio:** nombre base que agrupa a un conjunto de equipos o dispositivos y que permite proporcionar nombres de equipo más fácilmente recordables en lugar de una dirección IP numérica.
- **Servidor web:** es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que se llaman hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.
- **URL:** significa localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.
- **HTTP:** el protocolo de transferencia de hipertexto (HyperText Transfer Protocol) es el protocolo usado en cada transacción de la web (WWW).

- **Enlace:** una referencia en un documento de hipertexto a otro recurso.
- **Sistema de gestión de base de datos:** es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.
- **Base de datos:** es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.
- **Aplicación:** es un programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo.
- **MySQL:** sistema de gestión de base de datos relacional. Comúnmente utilizados en aplicaciones web.
- **Apache:** servidor de web de código abierto.
- **Dirección IP (Internet Protocol):** número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP.
- **Protocolo IP:** el Protocolo de Internet (Internet Protocol) es un protocolo no orientado a la conexión, usado tanto por el origen como por el destino para la comunicación de estos a través de una red (Internet).
- **Browser o motor de navegación:** es una aplicación que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet
- **DNS:** es una base de datos distribuida y jerárquica que almacena información asociada de una dirección IP a nombres de dominio en redes como Internet.

- **HTML (HyperText Markup Language):** es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.
- **Ajax (Asynchronous JavaScript And XML):** es una técnica de desarrollo web para crear aplicaciones interactivas.
- **RIA (Rich Internet Applications):** es un nuevo tipo de aplicación con más ventajas que las tradicionales aplicaciones web. Esta surge como una combinación de las ventajas que ofrecen las aplicaciones web y las aplicaciones tradicionales.
- **Javascript:** es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web.
- **Internet:** es un método de interconexión descentralizada de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP.
- **Encriptar:** ocultar el contenido de un mensaje mediante un proceso de codificación de manera tal que el acceso al contenido del mensaje sea restringido.

### 3.2.1.5. REFERENCIAS

Tabla 3.11. Referencias – Especificación de Requisitos de Software

Ref.	Título
1	Visión del sistema
2	Plan de desarrollo de software
3	Documento de arquitectura de software
4	Plan de pruebas de software

### **3.2.1.6. RESUMEN**

En las siguientes secciones se encontrará la descripción del sistema, así como el detalle de lo que es necesario desarrollar para alcanzar el resultado requerido.

### **3.2.2. DESCRIPCIÓN GENERAL**

#### **3.2.2.1. PERSPECTIVA DEL PRODUCTO**

El Sistema de Difusión de Gestión del Conocimiento de la ESPE será un producto diseñado para trabajar en entornos web, para que sea accedido desde cualquier parte del mundo con una conexión a internet.

El sistema además, será un sistema independiente previsto para funcionar de manera distribuida sobre la infraestructura del Departamento de Ciencias de la Computación (DCC) de la Escuela Politécnica del Ejército.

El sistema se lo desarrollará tomando en cuenta las mejores prácticas, con el fin de que sea escalable, es decir que conforme aumente la demanda del uso del mismo sea posible el aumentar la infraestructura sobre el que corre.

#### **3.2.2.2. FUNCIONALIDAD DEL PRODUCTO**

El Sistema de Difusión de Gestión del Conocimiento de la ESPE permitirá realizar las siguientes funciones:

- **Creación y autenticación de usuarios:** el sistema permitirá la creación de cuentas para personas que cuenten con correos electrónicos de la Escuela Politécnica del Ejército, es decir que termine con @espe.edu.ec. De igual manera permitirá que los usuarios se identifiquen por medio de su correo electrónico y una contraseña.

- **Administración de programas:** el sistema permitirá crear, actualizar, eliminar y visualizar programas que hayan sido creados dentro del aplicativo. Los programas constarán de un título, descripción y de un grupo que participará en su desarrollo.
- **Administración de proyectos:** el sistema permitirá crear, actualizar, eliminar y visualizar proyectos que hayan sido creados dentro del sistema. Los proyectos deberán estar relacionados con un programa previamente creado.
- **Administración de resultados:** el sistema permitirá crear, actualizar, eliminar y visualizar resultados que hayan sido creados dentro de la aplicación. Los resultados deberán estar relacionados con un proyecto previamente creado.
- **Administración de otras actividades:** el sistema permitirá crear, actualizar, eliminar y visualizar actividades que hayan sido creadas dentro del sistema. Las actividades deberán estar relacionados con un programa previamente creado.
- **Administración de grupos de investigación:** el sistema permitirá crear, actualizar, eliminar y visualizar los grupos de investigación que se han creado dentro de la aplicación. Los que estarán compuestos por usuarios con sus respectivos roles.

### **3.2.2.3. CARACTERÍSTICAS DE LOS USUARIOS**

El Sistema de Difusión de Gestión del Conocimiento de la ESPE contará únicamente con los tipos de usuario descritos en las siguientes tablas:

Tabla 3.12. Características del Administrador del Sistema

Tipo de usuario	Administrador del sistema
Formación	Persona con formación en Ing. en Sistemas y conocimientos en investigación.
Habilidades	Manejo de sistemas web
Actividades	Persona encargada de administrar y mantener el sistema por medio de tareas tales como la creación, modificación o eliminación de departamentos, líneas de investigación, sub-líneas de investigación y áreas de conocimiento.

Tabla 3.13. Características del Usuario del Sistema

Tipo de usuario	Usuario del sistema
Formación	Catedrático o investigador de la Escuela Politécnica del Ejército
Habilidades	Manejo de sistemas web
Actividades	Docencia Investigación Vinculación con la colectividad

#### **3.2.2.4. RESTRICCIONES**

##### **3.2.2.4.1. POLÍTICAS REGULATORIAS**

La aplicación se soportará en software de licencia abierta: servidor web (Apache), servidor de base de datos (MySQL) y el lenguaje de programación (Ruby), y la utilización de estos programas estará regida por las políticas establecidas por este tipo de licenciamiento.

##### **3.2.2.4.2. LIMITACIONES DE HARDWARE**

El sistema correrá sobre la infraestructura que asigne el Departamento de Ciencias de Computación de la Escuela Politécnica del Ejército.

#### **3.2.2.4.3. INTERFACES CON OTRAS APLICACIONES**

Debido a que el sistema no interactúa con otros sistemas y es autónomo no se desarrollarán interfaces entre aplicaciones, sin embargo contará con integración para compartir fotografías en Facebook y Twitter.

Por otra parte, se realizarán las conexiones necesarias para la utilización del servidor Web, servidor de base de datos (MySQL) y Ruby.

#### **3.2.2.4.4. CONSIDERACIONES DE SEGURIDAD**

Cada usuario deberá autenticarse y su acceso será verificado por el servidor para su respectiva labor de acuerdo a lo que su rol especifique. Todas las contraseñas deberán constar de por lo menos 6 caracteres y estar encriptadas en la base de datos para dar una buena seguridad al sistema y su información.

#### **3.2.2.5. SUPOSICIONES Y DEPENDENCIAS**

- Los servidores correrán el sistema operativo Linux, distribución Ubuntu. Los mismos deberán estar configurados de tal manera que puedan interactuar entre sí, es decir contar con interfaces de red y estar conectados de manera lógica, con el fin de proveer la infraestructura necesaria para correr el sistema a desarrollar.
- La versión de Ruby on Rails que se utilizará es v3.1, y correrá sobre la versión de Ruby 1.9.3.
- El servidor web tendrá una dirección IP (Internet Protocol) pública, que podrá ser apuntada desde un servidor DNS externo.

### **3.2.2.6. EVOLUCIÓN PREVISIBLE DEL SISTEMA**

El sistema será diseñado y desarrollado teniendo en cuenta el crecimiento a futuro que el mismo pudiera tener.

Por otro lado, el Sistema de Difusión de Gestión del Conocimiento de la ESPE, podría a futuro integrarse con los sistemas y aplicativos que se usan dentro de la Escuela Politécnica del Ejército.

### **3.2.3. REQUISITOS ESPECÍFICOS**

#### **3.2.3.1. REQUISITOS COMUNES DE LAS INTERFACES**

En esta sección se describen con detalle todas las entradas y salidas del sistema.

##### **3.2.3.1.1. INTERFACES DE USUARIO**

Las interfaces de usuario serán creadas utilizando como base al framework Twitter Bootstrap.

##### **3.2.3.1.2. INTERFACES DE HARDWARE**

- Servidor web y de almacenamiento – Características mínimas
  - Procesador doble núcleo de 2GHz
  - Memoria RAM: 4 GB
  - Disco Duro: 100 GB
  
- Servidor de base de datos – Características mínimas
  - Procesador doble núcleo de 2GHz
  - Memoria RAM: 2 GB
  - Disco Duro: 60 GB

### 3.2.3.2. REQUISITOS FUNCIONALES

#### 3.2.3.2.1. MÓDULO: USUARIOS

##### 3.2.3.2.1.1. AUTENTICACIÓN EN EL SISTEMA

Tabla 3.14. Requisito Funcional: Autenticación en el Sistema

Número de requisito	RF323211
Nombre de requisito	Autenticación en el sistema
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema deberá permitir la autenticación de usuarios por medio de un correo electrónico que pertenezca a la Escuela Politécnica del Ejército (@espe.edu.ec) y una contraseña.

##### 3.2.3.2.1.2. CREACIÓN DE CUENTAS EN EL SISTEMA

Tabla 3.15. Requisito Funcional: Creación de cuentas en el Sistema

Número de requisito	RF323212
Nombre de requisito	Creación de cuentas en el sistema
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema deberá permitir la creación de usuarios por medio de un correo electrónico institucional de la Escuela Politécnica del Ejército (@espe.edu.ec) y una contraseña. Se deberá enviar un correo de verificación a la dirección de correo electrónico que haya sido provista por el usuario.

##### 3.2.3.2.1.3. CAMBIO DE CONTRASEÑA DE UN USUARIO

Tabla 3.16. Requisito Funcional: Cambio de contraseña de un usuario

Número de requisito	RF323213
Nombre de requisito	Cambio de contraseña de un usuario
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema permitirá el cambio de contraseña a cualquier usuario que se encuentre firmado en el sistema.

### 3.2.3.2.1.4. CAMBIO DE CONTRASEÑA A OTROS USUARIOS

Tabla 3.17. Requisito Funcional: Cambio de contraseña a otros usuarios

Número de requisito	RF323214
Nombre de requisito	Cambio de contraseña a otros usuarios
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Actor	N/A
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema no permitirá que un usuario cambie la contraseña de otro usuario.

### 3.2.3.2.1.5. RESTITUCIÓN DE CONTRASEÑAS

Tabla 3.18. Requisito Funcional: Restitución de contraseñas

Número de requisito	RF323215
Nombre de requisito	Restitución de contraseñas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema permitirá que un usuario solicite restituir su contraseña, para lo cual se enviará un enlace válido hasta el ingreso de una nueva contraseña por parte del solicitante.

### 3.2.3.2.1.6. PERFIL DE UN USUARIO

Tabla 3.19. Requisito Funcional: Perfil de un usuario

Número de requisito	RF323216
Nombre de requisito	Perfil de un usuario
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema permitirá que un usuario tenga un perfil, el mismo constará de nombre completo, fecha de nacimiento, número de teléfono de contacto, y una autobiografía.

### 3.2.3.2.2. MÓDULO: PROGRAMAS DE INVESTIGACIÓN

#### 3.2.3.2.2.1. CREACIÓN DE PROGRAMAS DE INVESTIGACIÓN

Tabla 3.20. Requisito Funcional: Creación de programas de investigación

Número de requisito	RF323221
Nombre de requisito	Creación de programas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>El sistema permitirá la creación de programas, los mismos que tendrán un nombre, una fecha de inicio, una duración, un tipo de programa, pertenecerán a un grupo de investigación, una o varias sub-líneas de investigación, y tendrán una descripción.</p> <p>Adicionalmente, estos podrán estar marcados como publicado (visibles para todos los visitantes, caso contrario visible solo para su propietario), y también como finalizado (cuando un proyecto ha sido completado).</p> <p>Al mismo tiempo, los programas tendrán varios Proyectos, los mismos que serán creados de forma independiente.</p>

#### 3.2.3.2.2.2. MODIFICACIÓN DE PROGRAMAS DE INVESTIGACIÓN

Tabla 3.21. Requisito Funcional: Modificación de programas de investigación

Número de requisito	RF323222
Nombre de requisito	Modificación de programas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>El sistema permitirá la actualización de los datos de un programa al usuario que lo creó, o a los usuarios que constan dentro del grupo de investigación que el programa pertenece.</p> <p>Los datos que podrán ser modificados son: el título, fecha de inicio, duración, tipo de programa, grupo de investigación, sub-línea de investigación, estado de publicado, estado de finalizado y su descripción.</p>

### 3.2.3.2.2.3. ELIMINACIÓN DE PROGRAMAS DE INVESTIGACIÓN

Tabla 3.22. Requisito Funcional: Eliminación de programas de investigación

Número de requisito	RF323223
Nombre de requisito	Eliminación de programas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema permitirá la eliminación de un programa, siempre y cuando el usuario firmado en el sistema, sea parte del grupo de investigación que está asociado al programa o el usuario que creo originalmente el programa.

### 3.2.3.2.2.4. VISUALIZACIÓN DE PROGRAMAS DE INVESTIGACIÓN

Tabla 3.23. Requisito Funcional: Visualización de programas de investigación

Número de requisito	RF323224
Nombre de requisito	Visualización de programas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>El sistema permitirá mostrar el programa de una manera lógica, y visualmente atractiva.</p> <p>La página de despliegue contará con: el título del programa, el grupo de investigación al que pertenece o el usuario que presenta el programa, tipo de programa, departamento/s a los que está asociado y la/s áreas de conocimiento, la/s líneas de investigación y las sub-líneas de investigación, la descripción y una lista con todos los proyectos componentes del programa.</p>

### 3.2.3.2.3. MÓDULO: PROYECTOS DE INVESTIGACIÓN

#### 3.2.3.2.3.1. CREACIÓN DE PROYECTOS DE INVESTIGACIÓN

Tabla 3.24. Requisito Funcional: Creación de proyectos de investigación

Número de requisito	RF323231
Nombre de requisito	Creación de proyectos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>El sistema permitirá la creación de proyectos, los mismos que tendrán un título, una fecha de inicio, una duración, un programa asociado, sub-líneas de investigación a las que se aporta, usuarios asociados al proyecto con sus respectivos roles, estado de publicado, marcado como finalizado, y una descripción.</p> <p>Un proyecto solamente podrá aportar a un programa. Además, cada proyecto podrá tener uno o más resultados.</p>

#### 3.2.3.2.3.2. MODIFICACIÓN DE PROYECTOS DE INVESTIGACIÓN

Tabla 3.25. Requisito Funcional: Modificación de proyectos de investigación

Número de requisito	RF323232
Nombre de requisito	Modificación de proyectos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>El sistema permitirá la actualización de los datos de un proyecto al usuario que lo creó o a los usuarios que están aportando al mismo.</p> <p>Los datos que podrán ser modificados son: el título, la fecha de inicio, la duración, el programa al que aporta, la/s sub-línea/s de conocimiento a las que está asociado, el estado de publicado, el estado de finalizado, los usuarios que están asociados al proyecto, así como sus roles, y finalmente la descripción del proyecto.</p>

### 3.2.3.2.3.3. ELIMINACIÓN DE PROYECTOS DE INVESTIGACIÓN

Tabla 3.26. Requisito Funcional: Eliminación de proyectos de investigación

Número de requisito	RF323233
Nombre de requisito	Eliminación de proyectos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema permitirá la eliminación de un proyecto, siempre y cuando el usuario firmado en el sistema, sea el que originalmente creó el proyecto o uno de los usuarios que está asociado al proyecto.

### 3.2.3.2.3.4. VISUALIZACIÓN DE PROYECTOS DE INVESTIGACIÓN

Tabla 3.27. Requisito Funcional: Visualización de proyectos de investigación

Número de requisito	RF323234
Nombre de requisito	Visualización de un proyecto
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>El sistema permitirá mostrar el proyecto de una manera lógica, y visualmente atractiva.</p> <p>La página de despliegue contará con: el título, la fecha de inicio, la duración, el estado (que será calculado a partir de la fecha de inicio y la duración), el programa al que aporta, los departamentos involucrados, las líneas de investigación que están relacionadas, la descripción, y una lista de los resultados que están asociados al proyecto.</p>

### 3.2.3.2.4. MÓDULO: RESULTADOS DE INVESTIGACIÓN

#### 3.2.3.2.4.1. CREACIÓN DE RESULTADOS DE INVESTIGACIÓN

Tabla 3.28. Requisito Funcional: Creación de resultados de investigación

Número de requisito	RF323241
Nombre de requisito	Creación de resultados
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema permitirá la creación de resultados de investigación, los mismos que contarán con: un título, un tipo de resultado, un proyecto al que aportan, pueden estar publicados, tener uno o varios archivos, una o varias imágenes y una descripción.

### 3.2.3.2.4.2. MODIFICACIÓN DE RESULTADOS DE INVESTIGACIÓN

Tabla 3.29. Requisito Funcional: Modificación de resultados de investigación

Número de requisito	RF323242
Nombre de requisito	Modificación de resultados
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>El sistema permitirá la actualización de los datos de un resultado al autor, es decir al usuario que lo creó originalmente.</p> <p>Los datos que podrán ser modificados son: el título, el tipo de resultado, el proyecto al que pertenece, el estado de publicado, sus archivos, sus imágenes, y su descripción.</p>

### 3.2.3.2.4.3. ELIMINACIÓN DE RESULTADOS DE INVESTIGACIÓN

Tabla 3.30. Requisito Funcional: Eliminación de resultados de investigación

Número de requisito	RF323243
Nombre de requisito	Eliminación de resultados
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>El sistema permitirá la eliminación de un resultado, siempre y cuando el usuario firmado en el sistema, sea el que originalmente creó el resultado.</p>

### 3.2.3.2.4.4. VISUALIZACIÓN DE RESULTADOS DE INVESTIGACIÓN

Tabla 3.31. Requisito Funcional: Visualización de resultados de investigación

Número de requisito	RF323244
Nombre de requisito	Visualización de un resultado
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>El sistema permitirá mostrar el resultado de una manera lógica, y visualmente atractiva.</p> <p>La página de despliegue contará con: el título, el tipo de resultado, el programa al que aporta, links de descarga para los archivos, las imágenes y la descripción.</p>

### 3.2.3.2.5. MÓDULO: OTRAS ACTIVIDADES RELACIONADAS A LA GESTIÓN DEL CONOCIMIENTO

#### 3.2.3.2.5.1. CREACIÓN DE ACTIVIDADES

Tabla 3.32. Requisito Funcional: Creación de Actividades

Número de requisito	RF323251
Nombre de requisito	Creación de actividades
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema permitirá la creación de otras actividades relacionadas a la gestión del conocimiento, las mismos que contarán con: un título, un programa relacionado, una fecha, una o varias sub-líneas de investigación, estado de publicado, uno o varios archivos, una o varias imágenes, y una descripción.

#### 3.2.3.2.5.2. MODIFICACIÓN DE ACTIVIDADES

Tabla 3.33. Requisito Funcional: Modificación de actividades

Número de requisito	RF323252
Nombre de requisito	Modificación de actividades
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema permitirá la actualización de los datos de una actividad al autor, es decir al usuario que lo creó originalmente.  Los datos que podrán ser modificados son: el título, el programa relacionado, la fecha, la/s sub-líneas de investigación, estado de publicado, el/los archivos, la/las imágenes, y su descripción.

#### 3.2.3.2.5.3. ELIMINACIÓN DE ACTIVIDADES

Tabla 3.34. Requisito Funcional: Eliminación de actividades

Número de requisito	RF323253
Nombre de requisito	Eliminación de actividades
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema permitirá la eliminación de una actividad, siempre y cuando el usuario firmado en el sistema, sea el que originalmente creó la actividad.

### 3.2.3.2.5.4. VISUALIZACIÓN DE ACTIVIDADES

Tabla 3.35. Requisito Funcional: Visualización de actividades

Número de requisito	RF323254
Nombre de requisito	Visualización de una actividad
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>El sistema permitirá mostrar una actividad de una manera lógica, y visualmente atractiva.</p> <p>La página de despliegue contará con: el título, el programa al que pertenece, la fecha, los departamentos a los que pertenece, las líneas de investigación a las que se aporta, links de descarga a los archivos adjuntos, las imágenes que se han asociado a la actividad y su descripción.</p>

### 3.2.3.2.6. MÓDULO: DEPARTAMENTOS

#### 3.2.3.2.6.1. VISUALIZACIÓN DE UN DEPARTAMENTO

Tabla 3.36. Requisito Funcional: Visualización de un departamento

Número de requisito	RF323261
Nombre de requisito	Visualización de un departamento
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>La página de Departamento contará con el nombre del departamento, los programas y proyectos que han sido desarrollados dentro del mencionado departamento, así como las líneas de investigación a las que han aportado cada uno de dichos programas.</p>

#### 3.2.3.2.6.2. LISTA DE PROGRAMAS DE UN DEPARTAMENTO POR TIPO DE PROGRAMA

Tabla 3.37. Requisito Funcional: Lista de programas de un departamento por tipo de programa

Número de requisito	RF323262
Nombre de requisito	Lista de programas dentro de un departamento por tipo de programa
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>Esta vista filtrará los programas de acuerdo a un tipo de programa que se pasará como parámetro dentro del URL.</p>

### 3.2.3.2.7. MÓDULO: ADMINISTRACIÓN DEL SISTEMA

#### 3.2.3.2.7.1. ADMINISTRACIÓN DE DEPARTAMENTOS

Tabla 3.38. Requisito Funcional: Administración de departamentos

Número de requisito	RF323271
Nombre de requisito	Departamentos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Administrador del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>El sistema se apoyará en una entidad llamada Departamento, la misma que contará con un nombre.</p> <p>Esta entidad solamente podrá ser creada, modificada y eliminada por el administrador del sistema</p>

#### 3.2.3.2.7.2. ADMINISTRACIÓN DE ÁREAS DE CONOCIMIENTO

Tabla 3.39. Requisito Funcional: Administración de áreas de conocimiento

Número de requisito	RF323272
Nombre de requisito	Áreas de conocimiento
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Actor	Administrador del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>Como parte de los Departamentos se contará con áreas de conocimiento, las que constarán de un nombre.</p> <p>Las áreas de conocimiento solamente podrán ser creadas, modificadas y eliminadas por el administrador, quien se encargará de relacionarlas con el departamento correspondiente.</p>

#### 3.2.3.2.7.3. ADMINISTRACIÓN DE LÍNEAS DE INVESTIGACIÓN

Tabla 3.40. Requisito Funcional: Administración de líneas de investigación

Número de requisito	RF323273
Nombre de requisito	Líneas de investigación
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Actor	Administrador del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	<p>Otra entidad que tendrá un papel muy importante en el funcionamiento del sistema serán las líneas de investigación.</p> <p>Las líneas de investigación solamente podrán ser creadas, modificadas y eliminadas por el administrador del sistema.</p>

### 3.2.3.2.7.4. ADMINISTRACIÓN DE SUB-LÍNEAS DE INVESTIGACIÓN

Tabla 3.41. Requisito Funcional: Administración de sub-líneas de investigación

Número de requisito	RF323274
Nombre de requisito	Sub-líneas de investigación
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Actor	Administrador del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	Las sub-líneas de investigación constarán de un nombre y formarán parte de una línea de investigación. Esta tipo de entidad solamente podrá ser creada, modificada y eliminada por el administrador del sistema.

### 3.2.3.2.7.5. ADMINISTRACIÓN DE TIPOS DE PROGRAMA

Tabla 3.42. Requisito Funcional: Administración de tipos de programa

Número de requisito	RF323275
Nombre de requisito	Tipos de programa
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Actor	Administrador del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	Los programas podrán ser de varios tipos, tales como: Investigación o Vinculación con la colectividad. Con el fin de proveer con la suficiente versatilidad y futuro crecimiento al sistema, se ha creado esta entidad que podrá únicamente ser creada, modificada o eliminada por el administrador. Esta entidad constará solamente de un nombre.

### 3.2.3.2.7.6. ADMINISTRACIÓN DE TIPOS DE RESULTADO

Tabla 3.43. Requisito Funcional: Administración de tipos de resultado

Número de requisito	RF323276
Nombre de requisito	Tipos de resultado
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Actor	Administrador del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	Los resultados podrán ser de varios tipos, tales como: Paper científico, Póster o Capítulo de libro. Con el fin de proveer con la suficiente versatilidad y futuro crecimiento al sistema, se ha creado esta entidad que podrá únicamente ser creada, modificada o eliminada por el administrador. Esta entidad constará solamente de un nombre.

### 3.2.3.2.7.7. ADMINISTRACIÓN DE USUARIOS EXTERNOS

Tabla 3.44. Requisito Funcional: Administración de usuarios externos

Número de requisito	RF323274
Nombre de requisito	Usuarios externos
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Actor	Administrador del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema permitirá la creación de usuarios que no tengan correos electrónicos dentro del dominio de la Escuela Politécnica del Ejército (@espe.edu.ec) únicamente por parte del administrador, quien además podrá modificar o eliminar cuentas de dichos usuarios.

### 3.2.3.2.8. MÓDULO: MISCELÁNEOS

#### 3.2.3.2.8.1. PÁGINA DE INICIO

Tabla 3.45. Requisito Funcional: Página de inicio

Número de requisito	RF323281
Nombre de requisito	Página de inicio
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	La página de inicio corresponderá a una lista de actividades, junto con sus imágenes, a manera de un carrusel que cambiará de actividad cada 5 segundos.  Al final de la página habrá accesos directos a los últimos programas creados, los últimos grupos de investigación creados, los departamentos y las distintas líneas de investigación.

#### 3.2.3.2.8.2. PÁGINA "EXPLORAR"

Tabla 3.46. Requisito Funcional: Página "Explorar"

Número de requisito	RF323282
Nombre de requisito	Página "Explorar"
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	La página llamada "Explorar", mostrará los departamentos, áreas de conocimiento y líneas de investigación que cuenten por lo menos con un programa o proyecto de investigación creado y publicado.

### 3.2.3.2.8.3. PÁGINA “ACERCA DE”

Tabla 3.47. Requisito Funcional: Página "Acerca de"

Número de requisito	RF323283
Nombre de requisito	Página “Acerca de”
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	La página llamada “Acerca de” tendrá información sobre el presente proyecto de graduación, el objetivo del sistema y detalles sobre los autores del mismo.

### 3.2.3.2.8.4. PÁGINA “AYUDA”

Tabla 3.48. Requisito Funcional: Página "Ayuda"

Número de requisito	RF323284
Nombre de requisito	Página “Ayuda”
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	La página llamada “Ayuda” tendrá información general sobre cómo hacer uso del sistema y aprovecharlo al máximo.

### 3.2.3.2.8.5. BÚSQUEDAS DENTRO DEL SISTEMA

Tabla 3.49. Requisito Funcional: Búsquedas dentro del Sistema

Número de requisito	RF323285
Nombre de requisito	Búsquedas dentro del sistema
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Usuario del sistema
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Descripción	El sistema contará con la opción de realizar búsquedas por medio de palabras claves. Los resultados de las búsquedas incluirán Departamentos, áreas de conocimiento, líneas de investigación, sub-líneas de investigación, programas, proyectos y resultados.

### **3.2.3.3. REQUISITOS NO FUNCIONALES**

#### **3.2.3.3.1. REQUISITOS DE RENDIMIENTO**

El sistema deberá tener tiempos de respuesta menores a 6 segundos en el 90% de las peticiones.

El sistema deberá estar diseñado para mantener conexiones con varios usuarios simultáneos.

#### **3.2.3.3.2. SEGURIDAD**

El sistema mantendrá encriptadas todas las contraseñas de los usuarios. Esta medida permitirá que las contraseñas permanezcan secretas aún si algún perpetrador tiene acceso a las bases de datos del sistema.

## **CAPÍTULO 4:**

### **ELABORACIÓN Y CONSTRUCCIÓN**

#### **4.1. DOCUMENTO DE ARQUITECTURA DE SOFTWARE**

##### **4.1.1. INTRODUCCIÓN**

###### **4.1.1.1. PROPÓSITO**

El presente documento de arquitectura, tiene como finalidad presentar la arquitectura del Sistema de Difusión de Gestión del Conocimiento de la ESPE, el cual detalla en diferentes vistas el comportamiento del sistema desde distintos contextos.

###### **4.1.1.2. ALCANCE**

La descripción del documento se encuentra basado en diferentes perspectivas lógicas del Sistema de Difusión de Gestión del Conocimiento de la ESPE, por tal motivo se incluye aspectos relevantes de cada una de las mismas, pero, no se detallan aspectos relacionados con el lenguaje de programación.

###### **4.1.1.3. ORGANIZACIÓN DEL DOCUMENTO**

El documento se desarrolla y organiza en base a la plantilla Software Architecture Document del proceso de desarrollo de software elaborado por OpenUP, adaptada a las características particulares del tipo de proyecto en desarrollo.

La sección 4.1.2 realiza una introducción a la representación de la arquitectura de forma de asegurar una comprensión cabal del documento.

Las siguientes secciones abarcan la descripción de la arquitectura del Sistema de Difusión de Gestión del Conocimiento.

Luego de una descripción inicial de los objetivos y restricciones influyentes, se desarrolla cada una de las vistas.

En las secciones finales, y sobre la base de lo desarrollado anteriormente, se incluye la descripción de la arquitectura del presente sistema.

#### **4.1.2. REPRESENTACIÓN DE LA ARQUITECTURA**

El modelo propuesto utiliza el siguiente conjunto de vistas

- **Vista de casos de uso:** lista los casos de uso o escenarios del modelo de casos de uso que representen funcionalidades centrales del sistema final, que requieran una gran cobertura arquitectónica o aquellos que implican algún punto especialmente delicado de la arquitectura.
- **Vista lógica:** describe las partes arquitectónicamente significativas del modelo de diseño, como son la descomposición en capas, subsistemas o paquetes. Una vez presentadas estas unidades lógicas principales, se profundiza en ellas hasta el nivel adecuado.
- **Vista de despliegue:** describe uno o más escenarios de distribución física del sistema sobre los cuales se ejecutará y hará el despliegue del mismo. Muestra la comunicación entre los diferentes nodos que componen los escenarios antes mencionados, así como el mapeo de los elementos de la Vista de Procesos en dichos nodos.

- **Vista de implementación:** describe la estructura general del Modelo de Implementación y el mapeo de los subsistemas, paquetes y clases de la Vista Lógica a subsistemas y componentes de implementación.
- **Vista de datos:** describe los elementos principales del Modelo de Datos, brindando un panorama general de dicho modelo en términos de tablas.

#### **4.1.3. OBJETIVOS Y RESTRICCIONES DE LA ARQUITECTURA**

El desarrollo de Sistema de Difusión de Gestión del Conocimiento de la ESPE está orientado a la elaboración de un software de alta calidad, que sea escalable, sostenible y que aporte a las necesidades de la Institución.

Las principales metas a nivel de arquitectura son las siguientes:

- Diseño basado en componentes de propósito claro y concreto y con alto grado de cohesión,
- Desacoplamiento entre componentes que permita el fácil reemplazo de los mismos,
- Componentes altamente reutilizables.

En la planificación de este proceso se han encontrado las siguientes restricciones:

- Restricciones de diseño: debido a que el sistema está basado y propuesto para la Escuela Politécnica del Ejército, el mismo debe estar alineado con la forma de trabajo definida por la Institución.
- Restricciones de tecnología: debido a que la Escuela Politécnica del Ejército es una Institución del Estado se deberá utilizar tecnología de tipo Open Source.

#### **4.1.4. VISTA DE CASOS DE USO**

##### **4.1.4.1. INTRODUCCIÓN**

La Vista de Casos de Uso presenta un subconjunto del Modelo de Casos de Uso. Describe los casos de uso o escenarios que representen funcionalidades centrales del sistema final, que requieran una gran cobertura arquitectónica o aquellos que impliquen algún punto especialmente delicado de la arquitectura. Estos casos de uso, en conjunto con los requerimientos no funcionales, permiten descubrir y diseñar la arquitectura del sistema.

##### **4.1.4.2. IDENTIFICACIÓN DE LOS CASOS DE USO RELEVANTES PARA LA ARQUITECTURA**

Para el diseño del Sistema de Difusión de Gestión del Conocimiento de la ESPE, se identifican como casos de uso relevantes, desde el punto de vista de la arquitectura, los siguientes:

- Administración de Programas
- Administración de Proyectos
- Administración de Resultados
- Administración de Actividades
- Administración de Grupos de Investigación
- Cuentas de Usuario
- Administración General del Sistema

#### 4.1.4.2.1. ADMINISTRACIÓN DE PROGRAMAS DE INVESTIGACIÓN

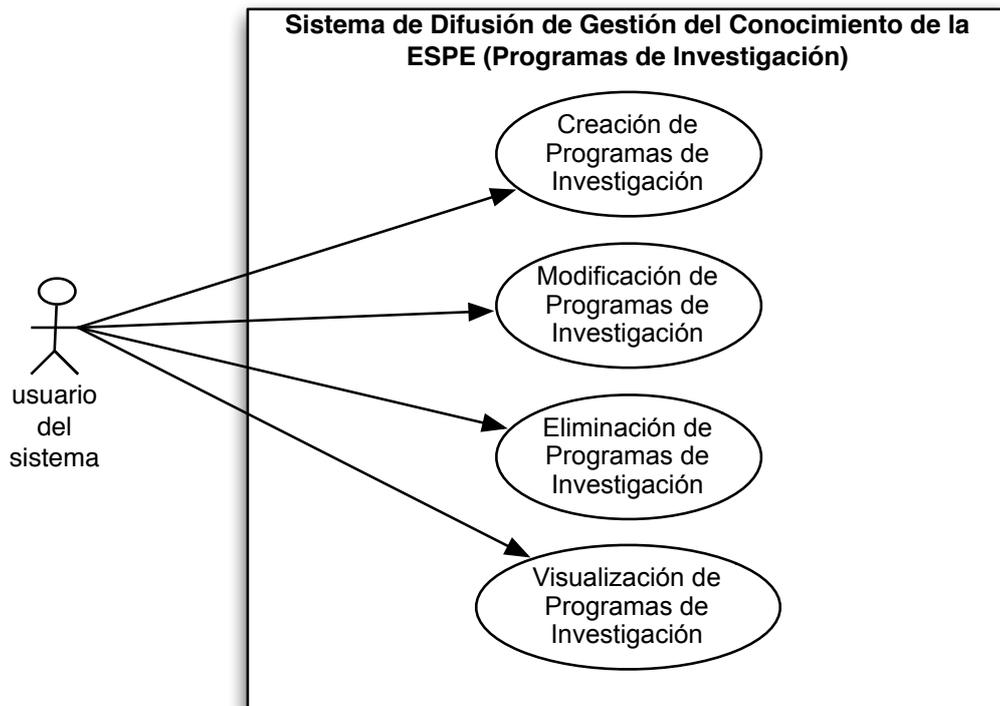


Figura 4.1. Casos de Uso: Administración de programas de investigación

#### 4.1.4.2.2. ADMINISTRACIÓN DE PROYECTOS DE INVESTIGACIÓN

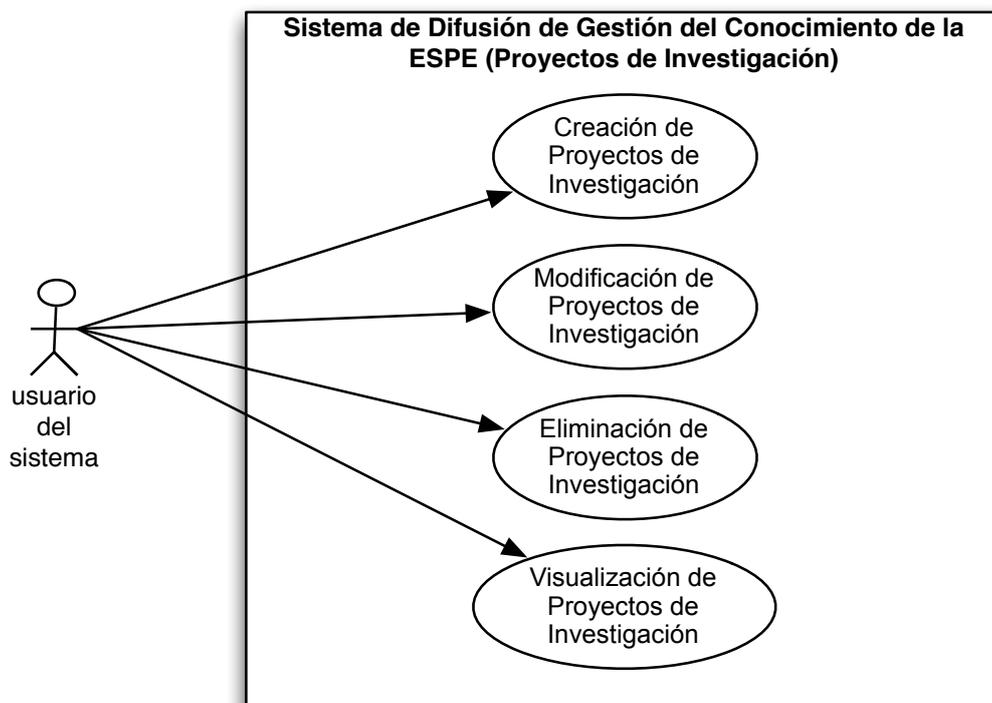


Figura 4.2. Casos de Uso: Administración de proyectos de investigación

#### 4.1.4.2.3. ADMINISTRACIÓN DE RESULTADOS DE INVESTIGACIÓN



Figura 4.3. Casos de Uso: Administración de resultados de investigación

#### 4.1.4.2.4. ADMINISTRACIÓN DE OTRAS ACTIVIDADES RELACIONADAS A LA GESTIÓN DEL CONOCIMIENTO

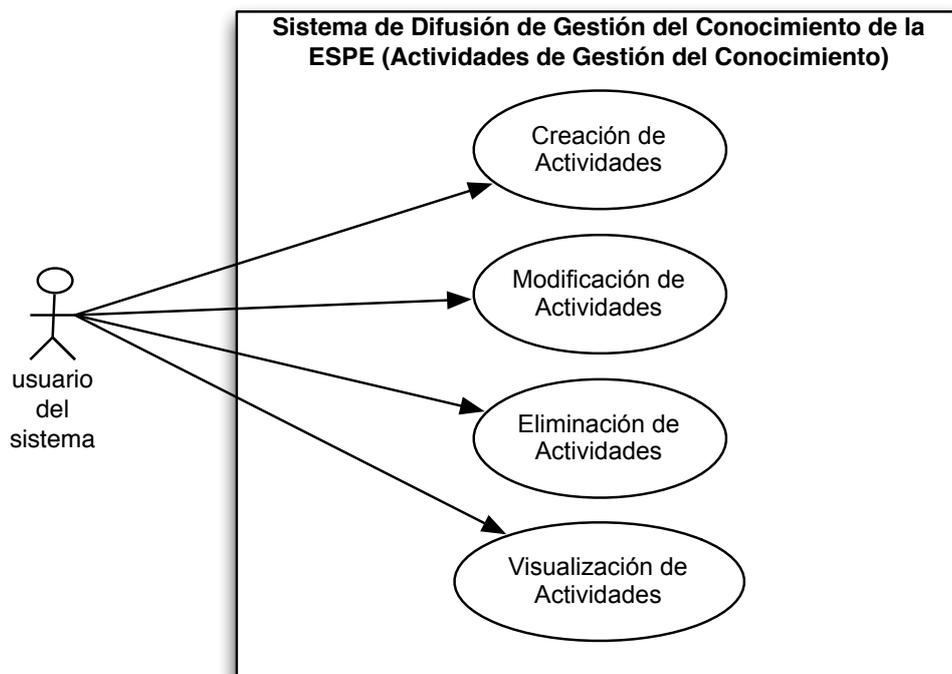


Figura 4.4. Casos de Uso: Administración de otras actividades relacionadas a la gestión del conocimiento

#### 4.1.4.2.5. ADMINISTRACIÓN DE GRUPOS DE INVESTIGACIÓN

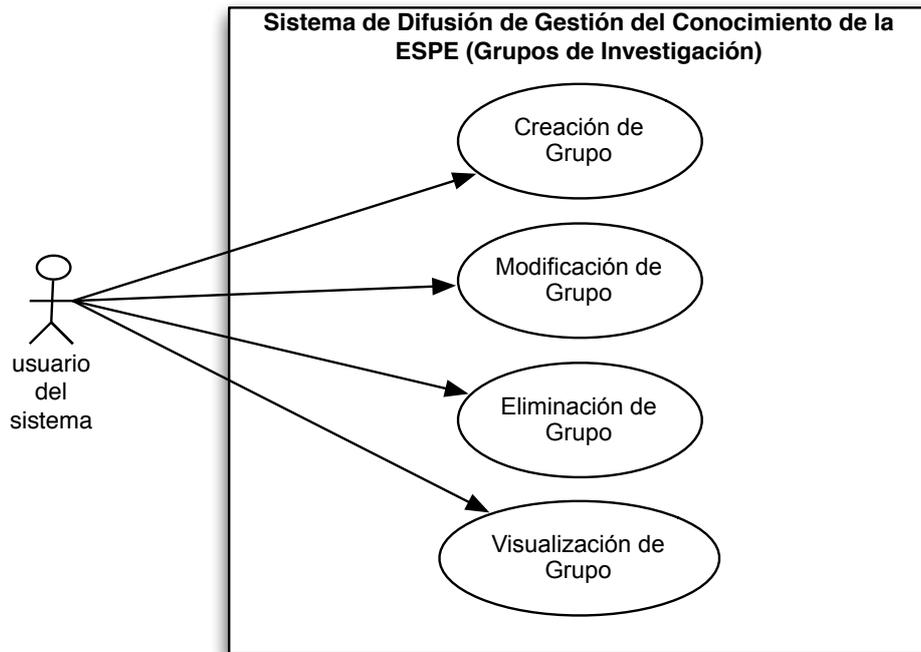


Figura 4.5. Casos de Uso: Administración de grupos de investigación

#### 4.1.1.1.1. CUENTAS DE USUARIO

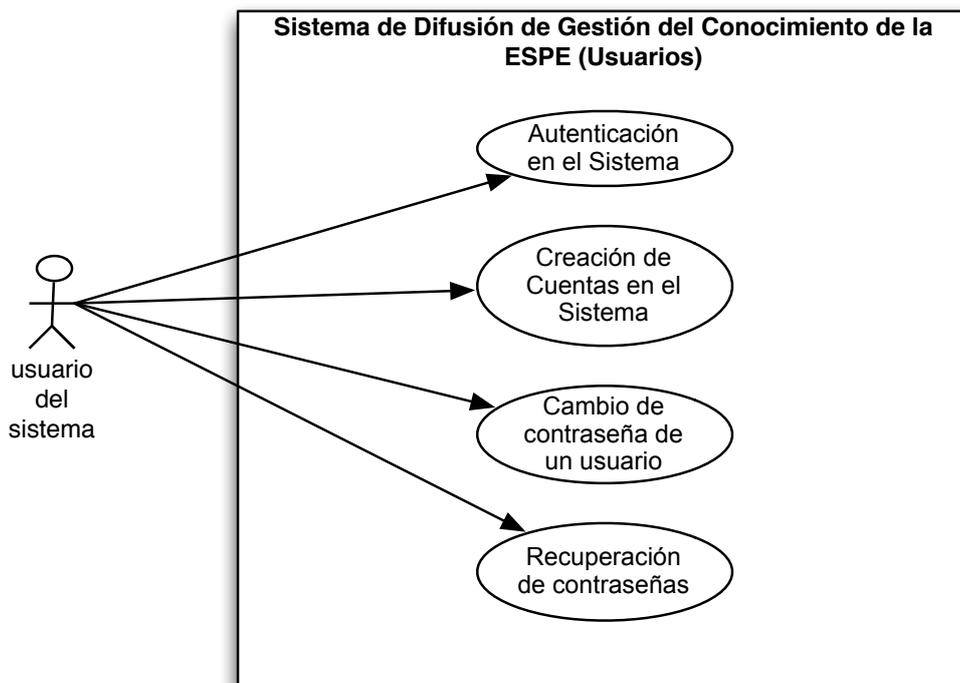


Figura 4.6. Casos de Uso: Cuentas de usuario

#### 4.1.1.1.2. ADMINISTRACIÓN GENERAL DEL SISTEMA

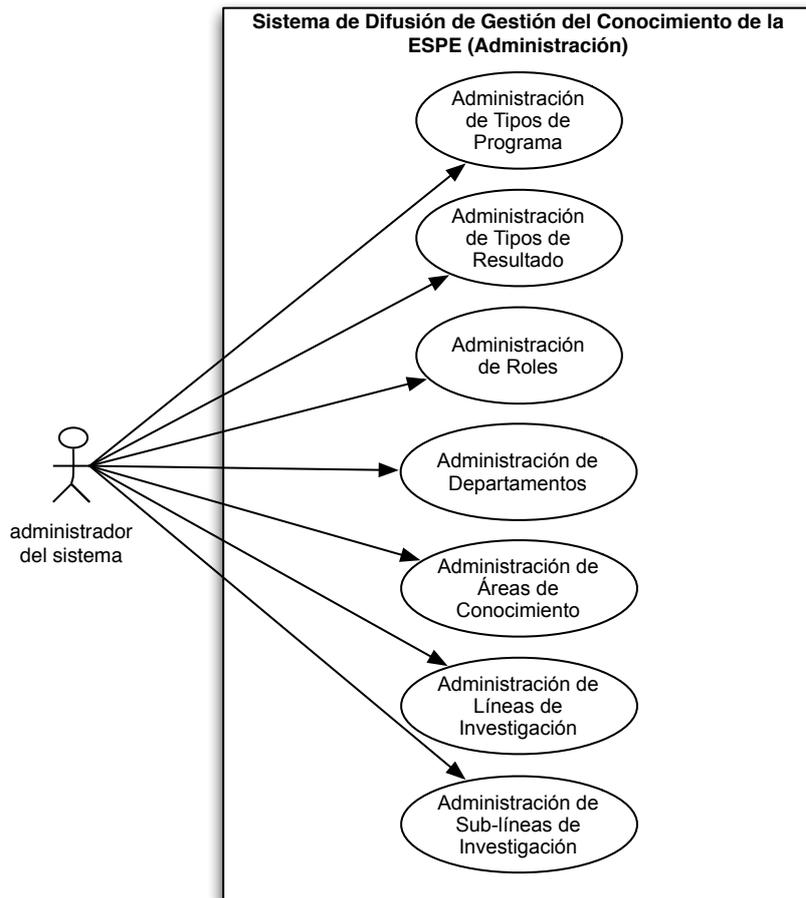


Figura 4.7. Casos de Uso: Administración general del Sistema

#### 4.1.1.2. DESCRIPCIÓN DE LOS CASOS DE USO RELEVANTES PARA LA ARQUITECTURA

Remitirse a la sección 3.2.3.2 Requisitos Funcionales del documento Especificación de Requisitos de Software.

#### 4.1.2. VISTA LÓGICA

##### 4.1.2.1. INTRODUCCIÓN

Se presentan en este punto los sucesivos refinamientos que definen las diferentes unidades lógicas que componen la arquitectura del sistema.

El primer refinamiento realizado consiste en la descomposición en subsistemas. Los subsistemas representan cortes verticales al diseño del sistema. Cada subsistema consiste en el agrupamiento de diferentes funcionalidades relacionadas entre sí y posee la capacidad de funcionar como un sistema en sí mismo.

Posteriormente se explora la composición de cada uno de los subsistemas. Y, finalmente se incluye la realización de los casos de uso descritos en la sección anterior mediante los componentes arquitectónicos definidos.

#### 4.1.2.2. DESCOMPOSICIÓN EN SUBSISTEMAS

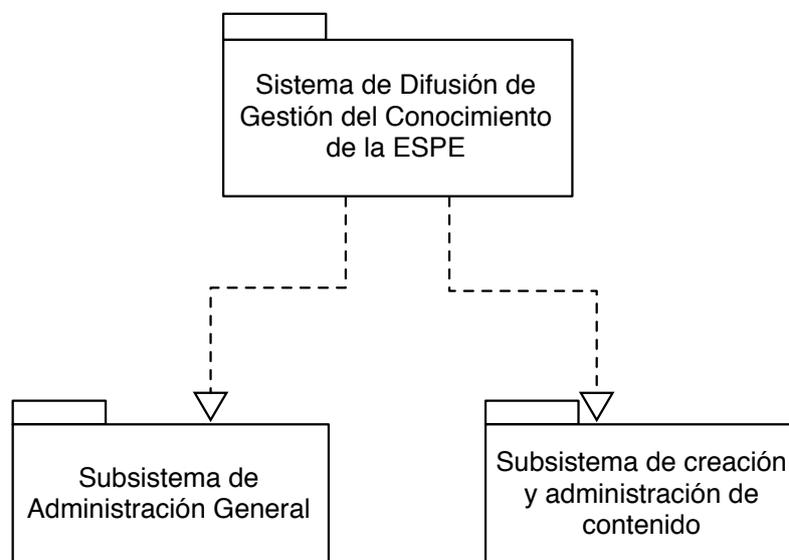


Figura 4.8. Descomposición en subsistemas

#### 4.1.2.3. DESCRIPCIÓN DE LOS SUBSISTEMAS

- **Subsistema de Administración General:** Este subsistema es el que permitirá la creación, modificación y eliminación de entidades permanentes del sistema.

- **Subsistema de creación y administración de contenido:** Este subsistema es el encargado de la creación, modificación y eliminación de programas, proyectos, resultados y actividades inherentes a la Institución.

#### 4.1.2.4. DISEÑO DE SUBSISTEMAS

##### 4.1.2.4.1. DISEÑO DETALLADO DE CADA SUBSISTEMA

- **Subsistema de Administración General**

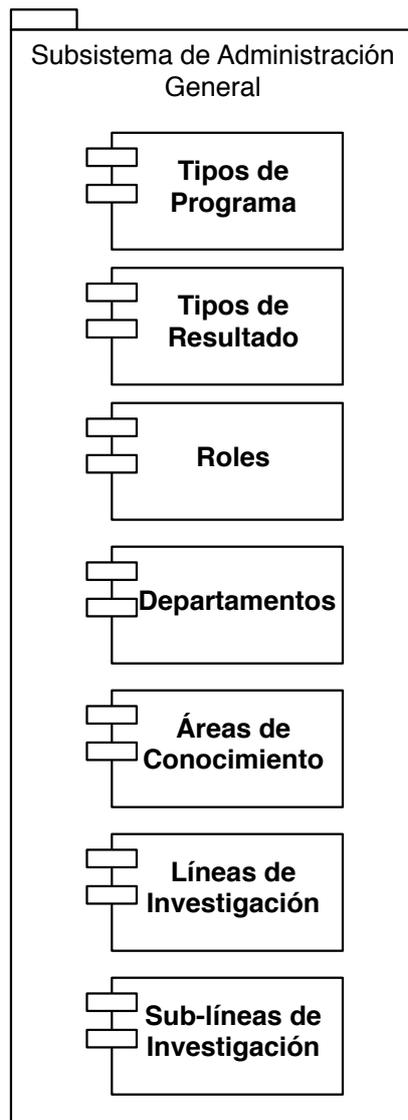


Figura 4.9. Subsistema de Administración General

- **Subsistema de creación y administración de contenido**

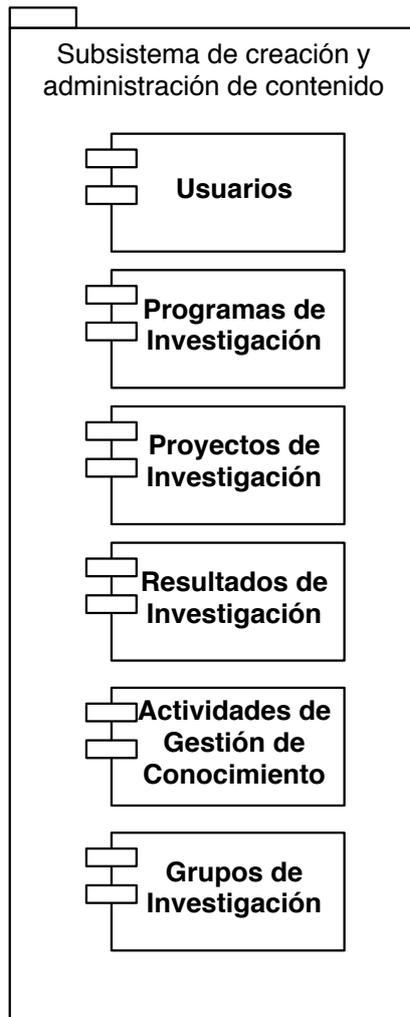


Figura 4.10. Subsistema de creación y administración de contenido

#### 4.1.2.5. REALIZACIÓN DE LOS CASOS DE USO RELEVANTES PARA LA ARQUITECTURA

##### 4.1.2.5.1. ADMINISTRACIÓN DE TIPOS DE PROGRAMA

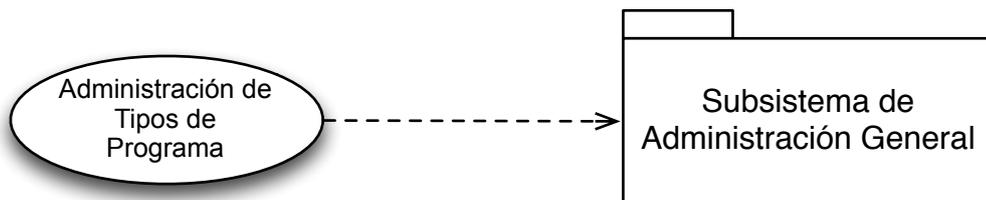


Figura 4.11. Realización de Casos de Uso: Administración de tipos de programa

#### 4.1.2.5.2. ADMINISTRACIÓN DE TIPOS DE RESULTADO

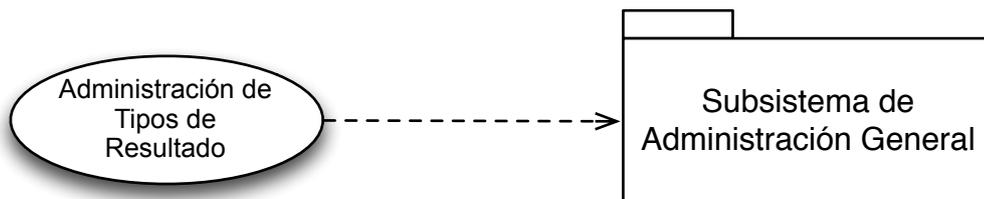


Figura 4.12. Realización de Casos de Uso: Administración de tipos de resultado

#### 4.1.2.5.3. ADMINISTRACIÓN DE ROLES

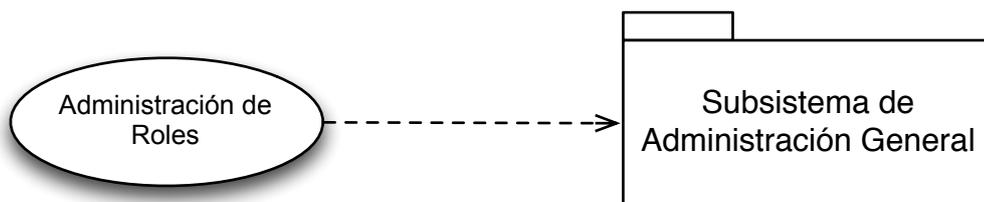


Figura 4.13. Realización de Casos de Uso: Administración de roles

#### 4.1.2.5.4. ADMINISTRACIÓN DE DEPARTAMENTOS

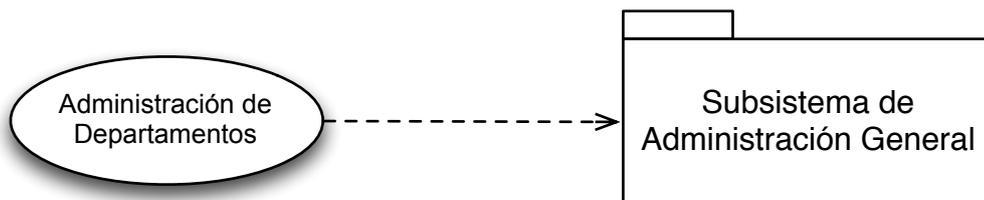


Figura 4.14. Realización de Casos de Uso: Administración de departamentos

#### 4.1.2.5.5. ADMINISTRACIÓN DE ÁREAS DE CONOCIMIENTO



Figura 4.15. Realización de Casos de Uso: Administración de áreas de conocimiento

#### 4.1.2.5.6. ADMINISTRACIÓN DE LÍNEAS DE INVESTIGACIÓN

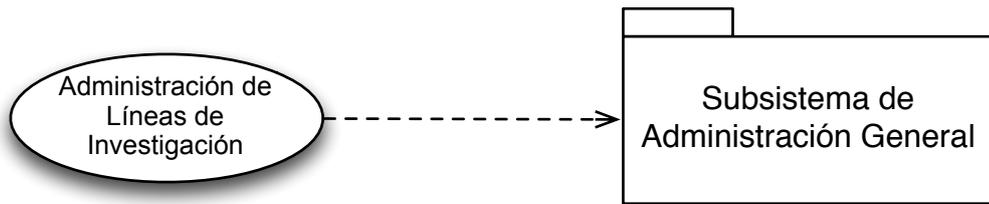


Figura 4.16. Realización de Casos de Uso: Administración de líneas de investigación

#### 4.1.2.5.7. ADMINISTRACIÓN DE SUB-LÍNEAS DE INVESTIGACIÓN

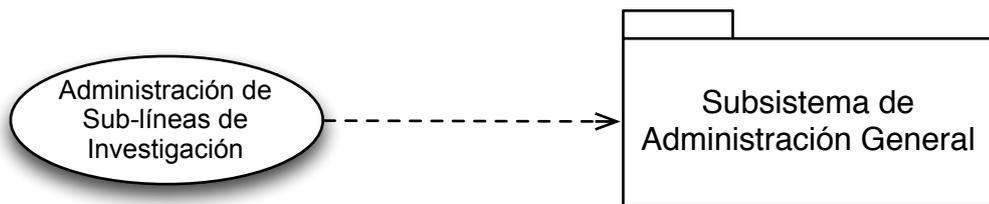


Figura 4.17. Realización de Casos de Uso: Administración de sub-líneas de investigación

#### 4.1.2.5.8. ADMINISTRACIÓN DE PROGRAMAS

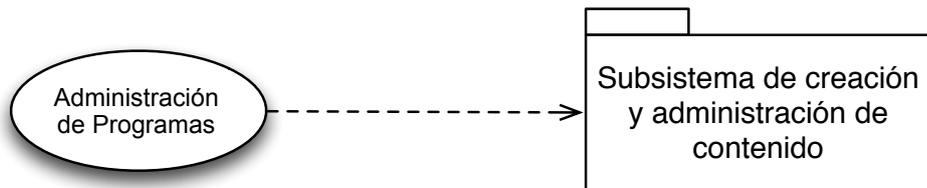


Figura 4.18. Realización de Casos de Uso: Administración de Programas

#### 4.1.2.5.9. ADMINISTRACIÓN DE PROYECTOS



Figura 4.19. Realización de Casos de Uso: Administración de proyectos

#### 4.1.2.5.10. ADMINISTRACIÓN DE RESULTADOS

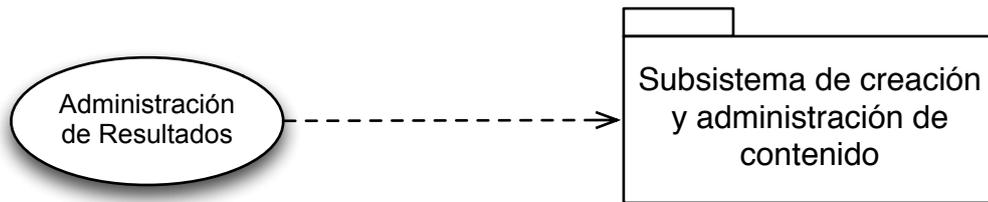


Figura 4.20. Realización de Casos de Uso: Administración de resultados

#### 4.1.2.5.11. ADMINISTRACIÓN DE ACTIVIDADES



Figura 4.21. Realización de Casos de Uso: Administración de actividades

#### 4.1.2.5.12. ADMINISTRACIÓN DE GRUPOS DE INVESTIGACIÓN

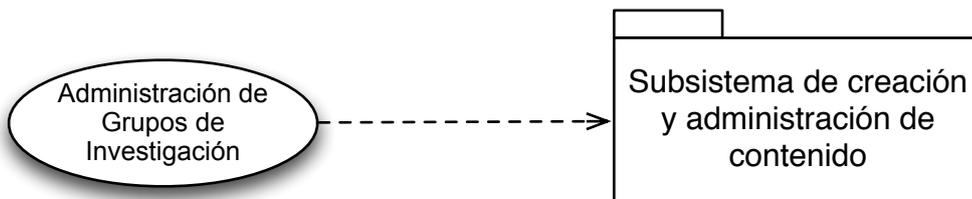


Figura 4.22. Realización de Casos de Uso: Administración de grupos de investigación

#### 4.1.2.5.13. CUENTAS DE USUARIO



Figura 4.23. Realización de Casos de Uso: Cuentas de usuario

#### 4.1.2.6. DIAGRAMA DE CLASES

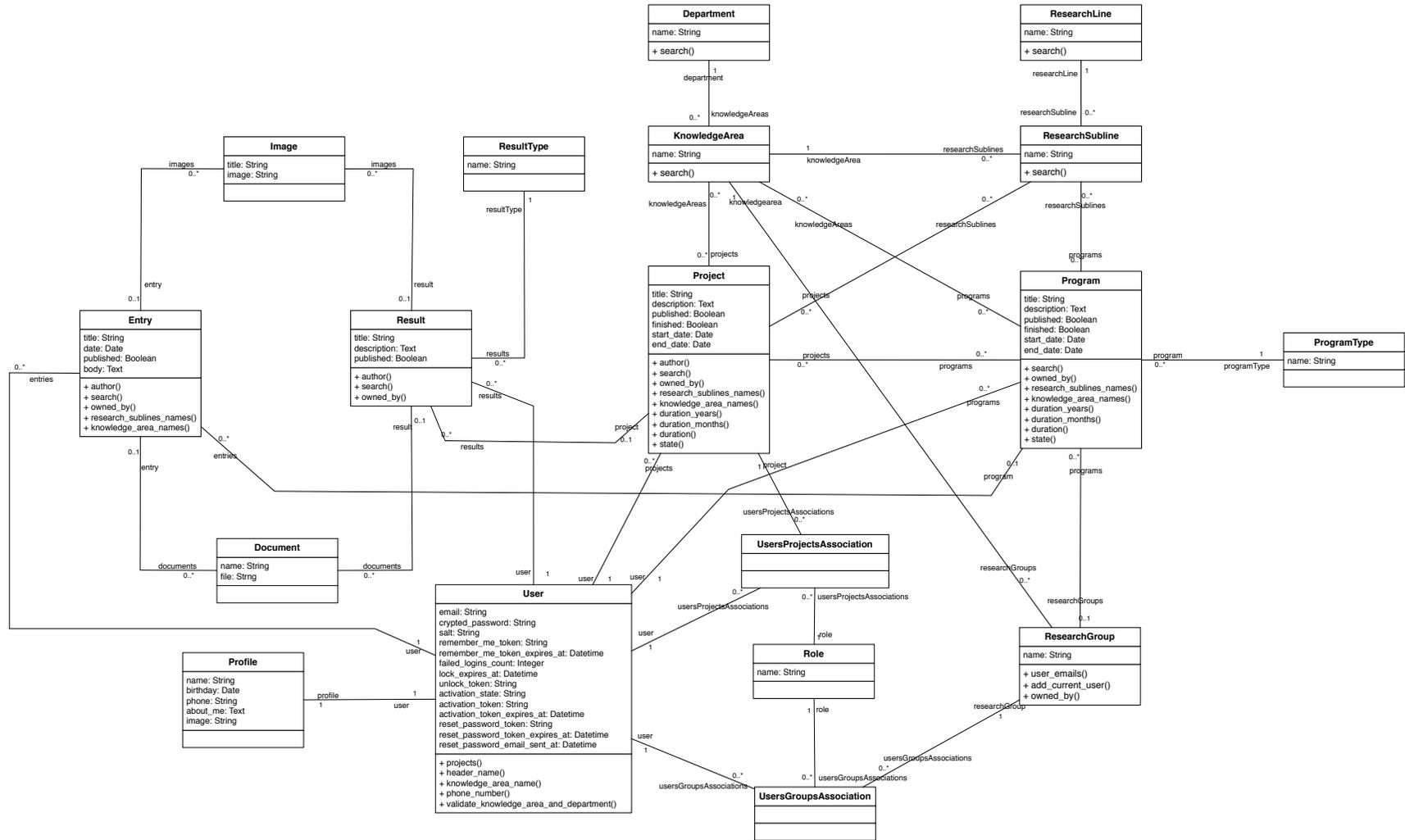


Figura 4.24. Diagrama de clases

## 4.1.2.7. DIAGRAMA DE ENTIDAD/RELACIÓN

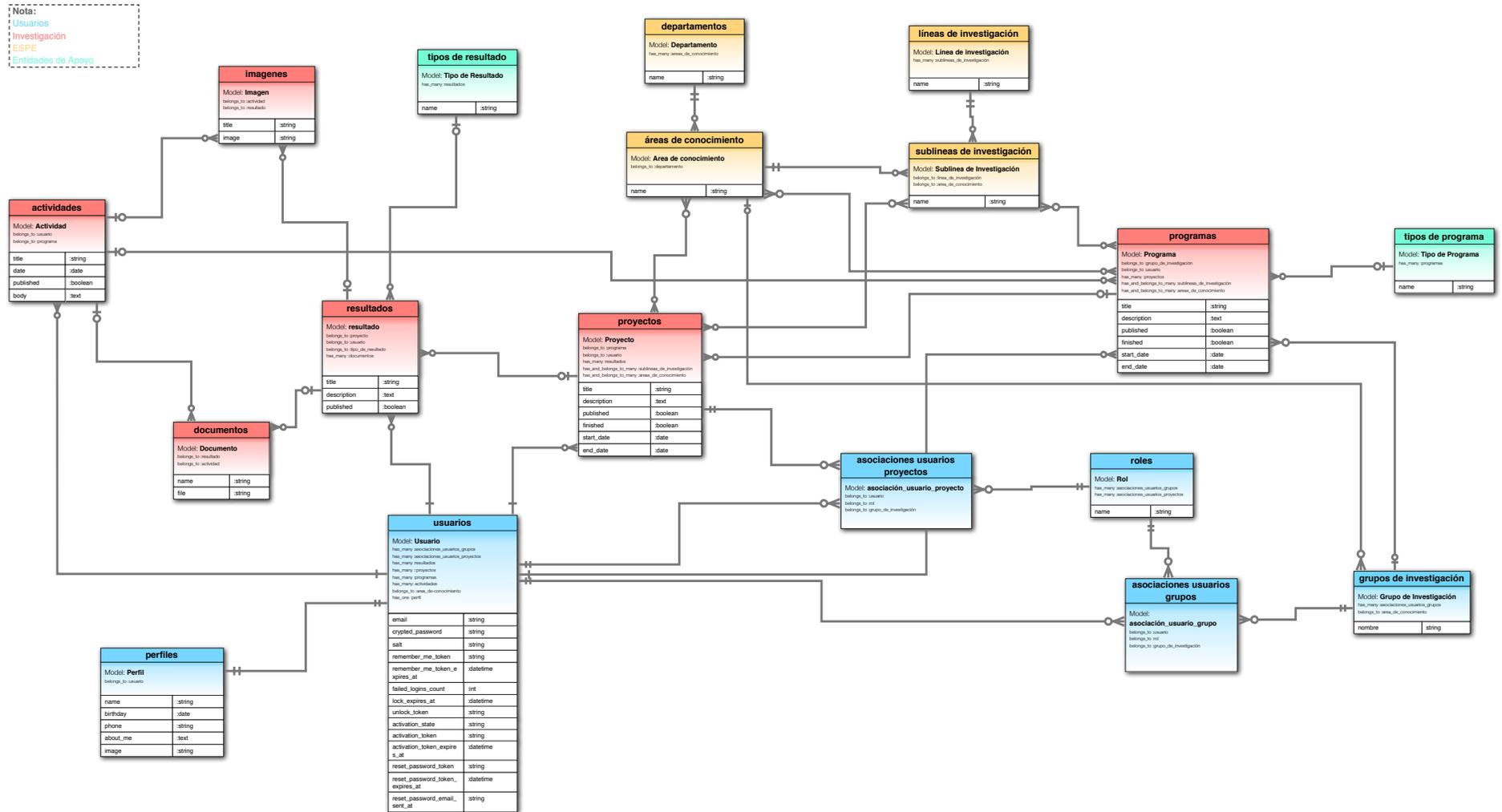


Figura 4.25. Diagrama de Entidad/Relación

#### 4.1.2.8. DIAGRAMA DE NAVEGACIÓN

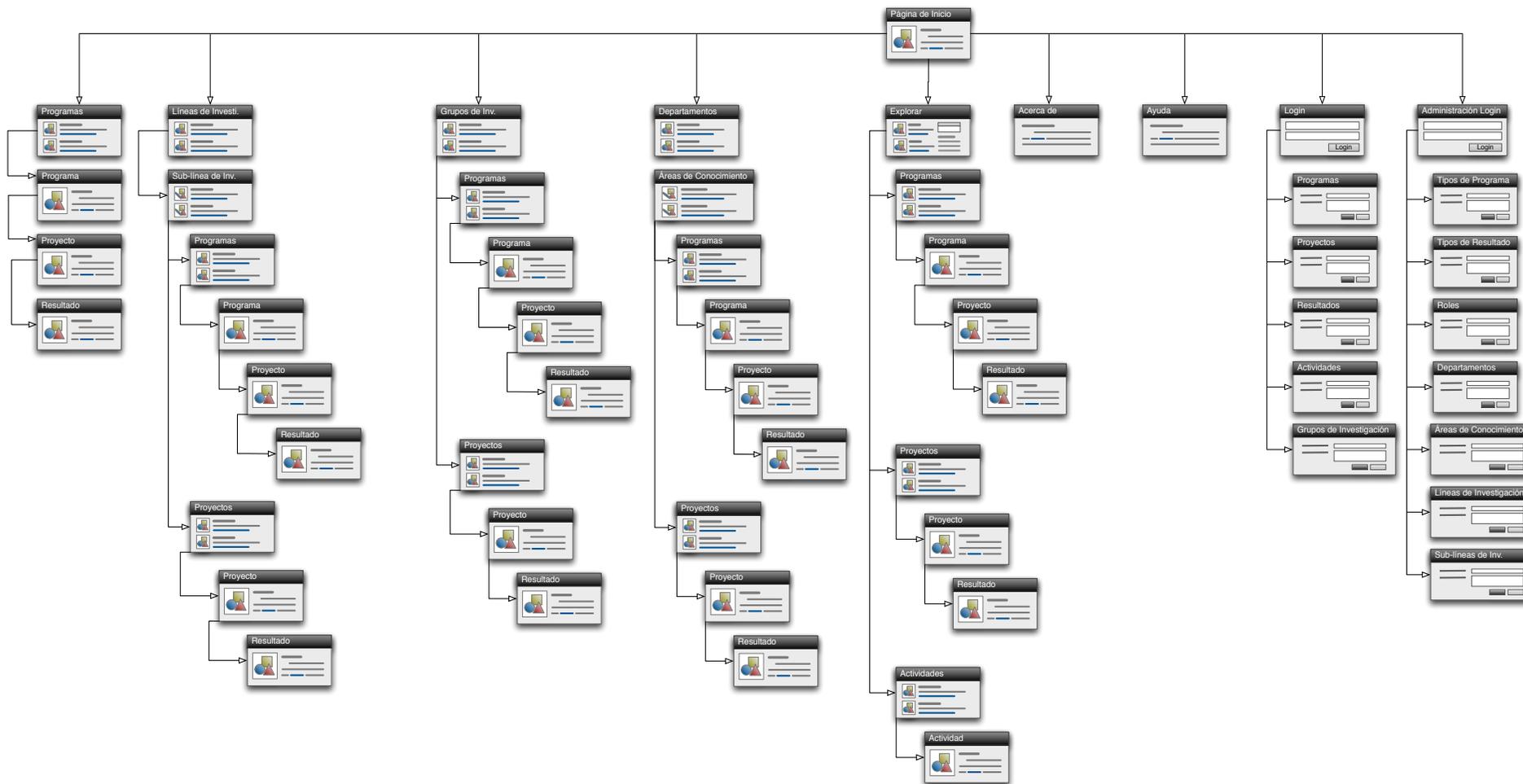


Figura 4.26. Diagrama de navegación

### 4.1.3. VISTA DE DESPLIEGUE

#### 4.1.3.1. INTRODUCCIÓN

Esta sección describe una o más configuraciones físicas sobre las cuales se realiza el despliegue del software y se lo ejecuta, así como la infraestructura necesaria para su instalación.

Para el caso del Sistema de Difusión de Gestión de Conocimiento de la ESPE se describe el escenario general de distribución esperado para los componentes de software antes descritos, las características de los nodos presentados y la comunicación entre los mismos.

#### 4.1.3.2. DISTRIBUCIÓN Y DESPLIEGUE

La siguiente figura presenta el escenario de distribución esperado para la instalación del sistema:

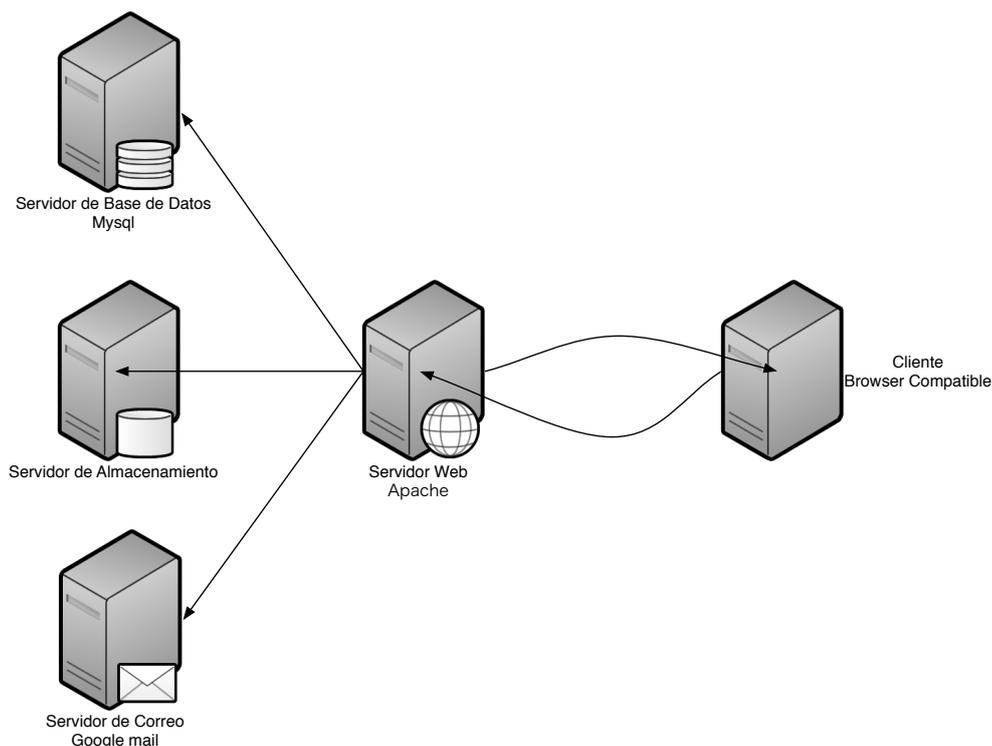


Figura 4.27. Distribución y Despliegue

#### 4.1.3.2.1. ESCENARIO DEL ENTORNO

A continuación se describen los nodos presentes en la figura:

- **Servidor de base de datos (MySQL):** es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.
- **Servidor de almacenamiento:** proporciona una sencilla interfaz de servicios web que puede utilizarse para almacenar y recuperar la cantidad de datos que se desee, cuando se desee, y desde cualquier parte de la web.
- **Servidor de correo (Google mail):** es un servicio de correo electrónico con posibilidades POP3 e IMAP gratuito proporcionado por la empresa estadounidense Google.
- **Servidor web (Apache):** es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual.
- **Cliente (navegador compatible):** representa la aplicación que se ejecuta en la estación de trabajo de un usuario, mediante el cual accederá al sistema. En este computador se deberá tener instalado un navegador compatible con la aplicación desarrollada, capaz de ejecutar Javascript (Chrome, Firefox, Internet Explorer, etc).

#### 4.1.4. VISTA DE DATOS

Remitirse al **Error! Reference source not found.** Vista de Datos.

## **4.2. PLAN DE PRUEBAS DE SOFTWARE**

### **4.2.1. INTRODUCCIÓN**

#### **4.2.1.1. OBJETIVOS**

El plan de pruebas de software se elabora con el fin de especificar qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el proceso de validación y verificación de los requerimientos funcionales y no funcionales del Sistema de Difusión de Gestión del Conocimiento de la ESPE. Además, a través del plan de pruebas se puede continuar con la trazabilidad de los requerimientos, con lo cual el grupo de trabajo, identifica el porcentaje de avance que se ha logrado hasta cierto momento.

Al desarrollar el plan de pruebas, se puede obtener información sobre los errores, defectos o fallas que tiene el prototipo, así se realizan las correcciones pertinentes, según el caso y se asegura la calidad del producto que se está entregando al cliente. El plan de pruebas se aplica sobre el producto, es decir, el código fuente, los resultados de las pruebas son registrados en un formato que se encuentra en el reporte de pruebas.

#### **4.2.1.2. ESTRATEGIA DE PRUEBAS**

A través de los diferentes documentos que se han elaborado, se pretende registrar información directamente relacionada con las pruebas, para asegurar la calidad de estas y del producto. Además le permite al responsable de las pruebas saber exactamente los criterios que se deben tener en cuenta para probar cada elemento del sistema. A continuación se explica brevemente el aporte de cada documento con respecto al plan de pruebas.

## Especificación de Requisitos de Software

- **Priorización:** se escogen los requerimientos de mayor priorización para aplicar las pruebas correspondientes.
- **Dependencias:** permite una visualización clara sobre los diferentes grupos de requerimientos que deben ser evaluados dentro de las pruebas de integración.
- **Trazabilidad:** le permite al responsable de la prueba saber el estado de requerimiento.

Figura 4.28. Descripción de la Especificación de Requisitos de Software

## Documento de Arquitectura de Software

- **Diagrama de Componentes:** La separación del sistema por componentes permite la clasificación de las pruebas según la funcionalidad del módulo.
- **Diagrama de Casos de Uso:** Permite una mejor visualización de los diferentes escenarios para realizar las pruebas de sistema.
- **Vista Física:** Prueba de los componentes de Hardware que tiene la aplicación.

Figura 4.29. Descripción del Documento de Arquitecutar de Software

Con esta estrategia se asegura llevar el seguimiento de la trazabilidad que se ha manejado desde la especificación de requerimientos (documento ERS), además de mantener la consistencia entre la aplicación y su respectiva documentación.

### 4.2.1.3. ALCANCE

Teniendo en cuenta los documentos hechos anteriormente, el grupo de trabajo pretende realizar las pruebas, de manera incremental, por módulo.

Las pruebas funcionales se realizarán inmediatamente después de haber implementado la aplicación, esto quiere decir, que el orden corresponde al descrito en la prioridad de requerimientos.

#### 4.2.1.4. REFERENCIAS

Tabla 4.1. Referencias – Plan de pruebas de software

Ref.	Título
1	Visión del sistema
2	Documento de especificación de requerimientos del sistema
3	Documento de arquitectura de software
4	Plan de desarrollo de software

#### 4.2.2. INSTRUMENTOS DE PRUEBA

##### 4.2.2.1. MÓDULOS DEL PROGRAMA

En esta sección se muestran los módulos que componen el Sistema de Difusión de Gestión del Conocimiento de la ESPE, además de las especificaciones de las pruebas a realizar en cada uno. Cabe anotar, que cada módulo representa un componente del sistema.

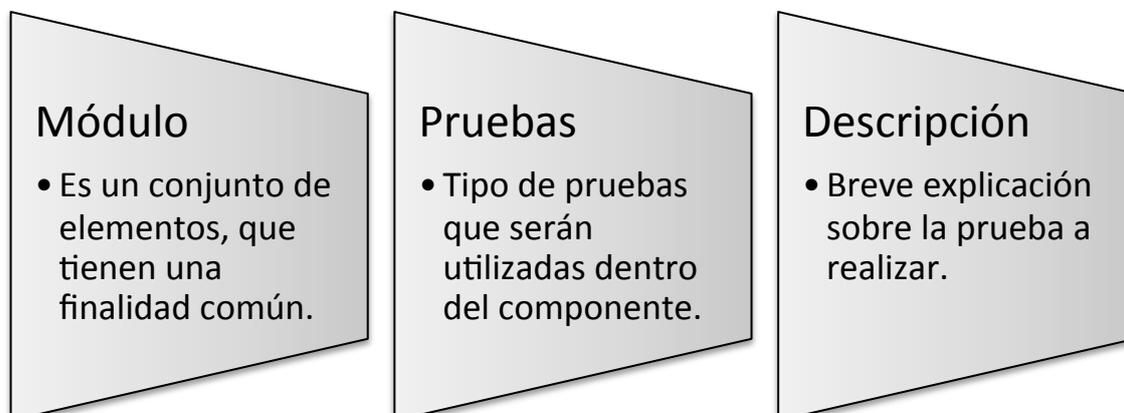


Figura 4.30. Conceptos

Tabla 4.2. Descripción de los módulos del programa

Modulo	Pruebas	Descripción
<b>Interface Gráfica de Usuario</b>	Facilidad de uso	La facilidad de uso permite evaluar si el usuario, en cada momento, tiene el conocimiento sobre qué puede o qué debería hacer y cómo hacerlo.
	Look & feel	Look & feel se refiere a la apariencia que se proporciona al usuario.
<b>Lógica de Negocio</b>	Funcionalidad	El sistema debe realizar todo los requerimientos establecidos con el cliente, este módulo será guiado por los diferentes tipos de requerimientos que se han manejado durante el proyecto.
<b>Objetos de Acceso a Datos</b>	Persistencia	El sistema debe ser capaz de guardar datos para ser usados en otro momento, además de tener acceso a ellos sin tener ningún problema de consistencia e integridad.
<b>NO funcionales</b>	No funcionales	El sistema debe cumplir con los requerimientos no funcionales que se han especificado en el SRS teniendo en cuenta el diseño.

#### 4.2.3. CARACTERISTICAS A SER PROBADAS

En esta sección se encuentran las características del sistema a ser probadas con un caso de estudio específico, conforme al alcance del Proyecto.

Tabla 4.3. Características a ser probadas

Característica	Descripción	Módulo
Requerimientos Funcionales	Se debe tener en cuenta el criterio de aceptación y dependencias, para realizar pruebas en los módulos. Además se debe utilizar el documento de casos de uso para tener claro los casos de éxito y fallo, y si la herramienta cumple con ellos.	Los módulos que permiten probar esta característica son: Lógica de Negocio Objetos de Acceso a Datos

#### 4.2.4. APROXIMACION

En esta sección se exponen los tipos de pruebas a utilizar en el Sistema de Difusión de Gestión del Conocimiento de la ESPE, para cada una de ellas se presenta un formato para el registro de los resultados.

#### 4.2.4.1. PRUEBAS FUNCIONALES

Las pruebas funcionales son pruebas aplicadas al conjunto de requerimientos que se definieron en el documento ERS.

Tabla 4.4. Pruebas Funcionales

<b>NOMBRE: Pruebas de Integración</b>		<b>IDENTIFICADOR: IT01</b>
<b>ACTIVIDADES</b>	Validación de requerimientos	
<b>TIEMPO ESTIMADO</b>	30 minutos por prueba	
<b>MÉTODOS O HERRAMIENTAS</b>	Rspec	
<b>ENTREGABLES</b>	Informe generado en donde se indica el resultado, conforme al Formato de Resultados de Pruebas Funcionales de la Tabla 4.5	

Tabla 4.5. Formato de Resultados de Pruebas Funcionales

<b>ID</b>	<b>GRUPO DE REQUERIMIENTOS</b>	<b>RESULTADOS DE LA PRUEBA</b>
<b>Ord.</b>	Grupo de requerimientos que son probados.	Resultado de la prueba.

#### 4.2.4.2. PRUEBAS DE COMPORTAMIENTO

Las pruebas de comportamiento son pruebas realizadas al sistema en su conjunto, que identifican la manera en que los casos de uso se cumplen, con rutas de éxito y fallo, que han sido definidas en las tablas Tabla 4.8 - Tabla 4.15.

Tabla 4.6. Pruebas de comportamiento

<b>NOMBRE: Pruebas de comportamiento</b>		<b>IDENTIFICADOR: CT01</b>
<b>ACTIVIDADES</b>	Se realizarán pruebas de comportamiento	
<b>TIEMPO ESTIMADO</b>	30 minutos por prueba	
<b>MÉTODOS O HERRAMIENTAS</b>	Cucumber	
<b>ENTREGABLES</b>	Informe generado en donde se indica el resultado, conforme al Formato de Resultados de Pruebas de Comportamiento de la Tabla 4.7.	

Tabla 4.7. Formato de Resultados de Pruebas de Comportamiento

<b>ID</b>	<b>CASO DE USO</b>	<b>RESULTADOS DE LA PRUEBA</b>
<b>Ord.</b>	Grupo de requerimientos a ser probados	Resultado de la prueba.

#### 4.2.5. PROCESO DE PRUEBAS

En esta sección se presentan los casos de pruebas generales para usarlos con el Sistema de Difusión de Gestión del Conocimiento de la ESPE. Cada cuadro está asociado a un caso de uso, desde ahí se desglosa en los diferentes módulos involucrados para el funcionamiento y se evalúa el resultado obtenido.

En las siguientes tablas, se muestran los casos de pruebas a realizar:

Tabla 4.8. Pruebas: Administrar programas

NOMBRE	Administrar Programas	PRUEBAS	P1
<b>PROPÓSITO</b>	Verificar si es posible crear, eliminar, modificar y visualizar programas dentro del sistema		
<b>PRERREQUISITOS</b>	Estar firmado en el sistema		
<b>UBICACIÓN</b>	Base de datos MySQL Navegador		
<b>ENTRADA</b>	Selección de la acción que se desea hacer: <ul style="list-style-type: none"> <li>• Listar Programas</li> <li>• Crear programa</li> <li>• Modificar programa</li> <li>• Eliminar programa</li> <li>• Visualizar programa</li> </ul>		
<b>ORÁCULO</b>	<ol style="list-style-type: none"> <li>1. LISTAR PROGRAMAS PROPIOS: Mostrar todos los programas creados por el usuario actual.</li> <li>2. CREAR PROGRAMA: La base de datos se encuentra actualizada con el nuevo programa</li> <li>3. ELIMINAR PROGRAMA: El programa especificado ha sido eliminado de la base de datos</li> <li>4. MODIFICAR PROGRAMA: La base de datos se encuentra actualizada con los cambios del programa seleccionado.</li> <li>5. VISUALIZAR PROGRAMA PUBLICADO: El navegador mostrará el programa, que se encuentra almacenado en la base de datos.</li> <li>6. VISUALIZAR PROGRAMA PROPIO NO PUBLICADO: El navegador mostrará el programa, que se encuentra almacenado en la base de datos.</li> <li>7. VISUALIZAR PROGRAMA NO PUBLICADO: El navegador será redirigido a la página de inicio.</li> </ol>		
<b>PASOS</b>	<ol style="list-style-type: none"> <li>1. LISTAR PROGRAMAS PROPIOS               <ul style="list-style-type: none"> <li>• Visitar el perfil del usuario actual.</li> <li>• Visualizar todos los programas, estén marcados como</li> </ul> </li> </ol>		

	<p>publicados o no publicados, que pertenezcan al usuario actual.</p> <ol style="list-style-type: none"> <li>2. CREAR PROGRAMA <ul style="list-style-type: none"> <li>• Visitar la página de inicio.</li> <li>• Seleccionar el Menú</li> <li>• Seleccionar “Crear Programa”.</li> <li>• Llenar el título del programa</li> <li>• Llenar la descripción del programa</li> <li>• Seleccionar un tipo de programa de la lista respectiva.</li> <li>• Pulsar el botón Crear Programa.</li> <li>• Visualizar programa.</li> </ul> </li> <li>3. MODIFICAR PROGRAMA <ul style="list-style-type: none"> <li>• Visitar la página de un programa creado previamente.</li> <li>• Dar click en el link Editar.</li> <li>• Modificar los datos necesarios</li> <li>• Pulsar el botón Modificar Programa</li> </ul> </li> <li>4. ELIMINAR PROGRAMA <ul style="list-style-type: none"> <li>• Visitar la página de un programa creado previamente por el usuario que está firmado en el sistema</li> <li>• Pulsar el link para Borrar</li> <li>• Ver mensaje de confirmación.</li> </ul> </li> <li>5. VISUALIZAR PROGRAMA PUBLICADO <ul style="list-style-type: none"> <li>• Visitar la página de un programa que ha sido marcado como publicado.</li> <li>• Visualizar todos los datos pertinentes al programa</li> </ul> </li> <li>6. VISUALIZAR PROGRAMA PROPIO NO PUBLICADO <ul style="list-style-type: none"> <li>• Visitar la página de un programa que ha sido marcado como NO publicado, pero ha sido creado por el usuario actual.</li> <li>• Visualizar todos los datos pertinentes al programa</li> </ul> </li> <li>7. VISUALIZAR PROGRAMA NO PUBLICADO <ul style="list-style-type: none"> <li>• Visitar la página de un programa que ha sido marcado como NO publicado y que NO ha sido creado por el usuario actual.</li> <li>• Redirigir a la página de inicio.</li> </ul> </li> </ol>
<b>Módulos Asociados</b>	<ul style="list-style-type: none"> <li>• Programas</li> <li>• Proyectos</li> <li>• Usuarios</li> <li>• Sub-líneas de Investigación</li> </ul>

Tabla 4.9. Pruebas: Acciones de usuarios anónimos relacionados a programas

<b>NOMBRE</b>	<b>Acciones de Usuarios Anónimos Relacionados a Programas</b>	<b>PRUEBAS</b>	<b>P2</b>
<b>PROPÓSITO</b>	Verificar si es posible crear, eliminar, modificar y visualizar programas dentro del sistema		
<b>PRERREQUISITOS</b>	NO estar firmado en el sistema		
<b>UBICACIÓN</b>	Navegador		
<b>ENTRADA</b>	Seleccionar un requerimiento Selección de la acción que desea hacer <ul style="list-style-type: none"> <li>• Listar Programas</li> <li>• Crear programa</li> <li>• Modificar programa</li> <li>• Eliminar programa</li> <li>• Visualizar programa</li> </ul>		
<b>ORÁCULO</b>	<ol style="list-style-type: none"> <li>1. LISTAR PROGRAMAS DE UN USUARIO: Mostrar todos los programas marcados como publicados que pertenecen a un usuario</li> <li>2. CREAR PROGRAMA: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el sistema.</li> <li>3. MODIFICAR PROGRAMA: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el sistema cuando se intenta modificar un programa existente.</li> <li>4. ELIMINAR PROGRAMA: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el sistema cuando se intenta borrar un programa existente.</li> <li>5. VISUALIZAR PROGRAMA PUBLICADO: El navegador mostrará el programa, que se encuentra almacenado en la base de datos.</li> <li>6. VISUALIZAR PROGRAMA NO PUBLICADO: El navegador será redirigido a la página de inicio.</li> </ol>		
<b>PASOS</b>	<ol style="list-style-type: none"> <li>1. LISTAR PROGRAMAS DE UN USUARIO                             <ul style="list-style-type: none"> <li>• Visitar el perfil del usuario actual.</li> <li>• Visualizar todos los programas marcados como publicados, que pertenezcan al usuario indicado. Los programas marcados como NO publicados no deberán ser desplegados.</li> </ul> </li> <li>2. CREAR PROGRAMA                             <ul style="list-style-type: none"> <li>• Visitar la página de crear programa</li> <li>• Desplegar un mensaje solicitando al usuario iniciar</li> </ul> </li> </ol>		

	<p>sesión.</p> <p>3. MODIFICAR PROGRAMA</p> <ul style="list-style-type: none"> <li>• Visitar la página de edición de un programa creado previamente.</li> <li>• Desplegar un mensaje solicitando al usuario iniciar sesión.</li> </ul> <p>4. ELIMINAR PROGRAMA</p> <ul style="list-style-type: none"> <li>• Visitar la página de un programa creado previamente por un usuario registrado.</li> <li>• Verificar que el link borrar no exista.</li> </ul> <p>5. VISUALIZAR PROGRAMA PUBLICADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un programa que ha sido marcado como publicado.</li> <li>• Visualizar todos los datos pertinentes al programa</li> </ul> <p>6. VISUALIZAR PROGRAMA NO PUBLICADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un programa que ha sido marcado como NO publicado.</li> <li>• Desplegar un mensaje solicitando al usuario iniciar sesión.</li> </ul>
<b>Módulos Asociados</b>	<ul style="list-style-type: none"> <li>• Programas</li> <li>• Proyectos</li> <li>• Usuarios</li> <li>• Sub-líneas de Investigación</li> </ul>

Tabla 4.10. Pruebas: Administrar proyectos

<b>NOMBRE</b>	<b>Administrar Proyectos</b>	<b>PRUEBAS</b>	<b>P3</b>
<b>PROPÓSITO</b>	Verificar si es posible crear, eliminar, modificar y visualizar programas dentro del sistema		
<b>PRERREQUISITOS</b>	Estar firmado en el sistema		
<b>UBICACIÓN</b>	Base de datos MySQL Navegador		
<b>ENTRADA</b>	Selección de la acción que desea hacer: <ol style="list-style-type: none"> <li>1. Listar proyectos</li> <li>2. Crear proyecto</li> <li>3. Modificar proyecto</li> <li>4. Eliminar proyecto</li> <li>5. Visualizar proyecto</li> </ol>		
<b>ORÁCULO</b>	1. LISTAR PROYECTOS PROPIOS: Mostrar todos los		

	<p>proyectos creados por el usuario actual.</p> <ol style="list-style-type: none"> <li>2. CREAR PROYECTO: La base de datos se encuentra actualizada con el nuevo proyecto</li> <li>3. ELIMINAR PROYECTO: El proyecto especificado ha sido eliminado de la base de datos</li> <li>4. MODIFICAR PROYECTO: La base de datos se encuentra actualizada con los cambios del proyecto seleccionado.</li> <li>5. VISUALIZAR PROYECTO PUBLICADO: El navegador mostrará el proyecto que se encuentra almacenado en la base de datos.</li> <li>6. VISUALIZAR PROYECTO PROPIO NO PUBLICADO: El navegador mostrará el proyecto que se encuentra almacenado en la base de datos.</li> <li>7. VISUALIZAR PROYECTO NO PUBLICADO: El navegador será redirigido a la página de inicio.</li> </ol>
<p><b>PASOS</b></p>	<ol style="list-style-type: none"> <li>1. LISTAR PROYECTOS PROPIOS <ul style="list-style-type: none"> <li>• Visitar el perfil del usuario actual.</li> <li>• Visualizar todos los proyectos marcados como publicados o no publicados, que pertenezcan al usuario actual.</li> </ul> </li> <li>2. CREAR PROYECTO <ul style="list-style-type: none"> <li>• Visitar la página de inicio.</li> <li>• Seleccionar el Menú</li> <li>• Seleccionar “Crear Proyecto”.</li> <li>• Llenar el título del proyecto</li> <li>• Llenar la descripción del proyecto</li> <li>• Seleccionar un programa de la lista respectiva.</li> <li>• Pulsar el botón Crear Proyecto.</li> <li>• Visualizar proyecto.</li> </ul> </li> <li>3. MODIFICAR PROYECTO <ul style="list-style-type: none"> <li>• Visitar la página de un proyecto creado previamente.</li> <li>• Dar click en el link Editar.</li> <li>• Modificar los datos necesarios</li> <li>• Pulsar el botón Modificar Proyecto</li> </ul> </li> <li>4. ELIMINAR PROYECTO <ul style="list-style-type: none"> <li>• Visitar la página de un proyecto creado previamente por el usuario que está firmado en el sistema</li> <li>• Pulsar el link Borrar</li> <li>• Ver mensaje de confirmación.</li> </ul> </li> </ol>

	<p>5. VISUALIZAR PROYECTO PUBLICADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un proyecto que ha sido marcado como publicado.</li> <li>• Visualizar todos los datos pertinentes al proyecto</li> </ul> <p>6. VISUALIZAR PROYECTO PROPIO NO PUBLICADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un proyecto que ha sido marcado como NO publicado, pero ha sido creado por el usuario actual.</li> <li>• Visualizar todos los datos pertinentes al proyecto</li> </ul> <p>7. VISUALIZAR PROYECTO NO PUBLICADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un proyecto que ha sido marcado como NO publicado y que NO ha sido creado por el usuario actual.</li> <li>• Redirigir a la página de inicio.</li> </ul>
<b>Módulos Asociados</b>	<ul style="list-style-type: none"> <li>• Programas</li> <li>• Proyectos</li> <li>• Resultados</li> <li>• Usuarios</li> <li>• Sub-líneas de Investigación</li> </ul>

Tabla 4.11. Pruebas: Acciones de usuarios anónimos relacionados a proyectos

<b>NOMBRE</b>	<b>Acciones de Usuarios Anónimos Relacionados a Proyectos</b>	<b>PRUEBAS</b>	<b>P4</b>
<b>PROPÓSITO</b>	Verificar si es posible crear, eliminar, modificar y visualizar proyectos dentro del sistema		
<b>PRERREQUISITOS</b>	NO estar firmado en el sistema		
<b>UBICACIÓN</b>	Navegador		
<b>ENTRADA</b>	Seleccionar un requerimiento Selección de la acción que desea hacer <ol style="list-style-type: none"> <li>1. Listar proyectos</li> <li>2. Crear proyecto</li> <li>3. Modificar proyecto</li> <li>4. Eliminar proyecto</li> <li>5. Visualizar proyecto</li> </ol>		
<b>ORÁCULO</b>	<ol style="list-style-type: none"> <li>1. LISTAR PROYECTOS DE UN USUARIO: Mostrar todos los proyectos marcados como publicados que pertenecen a un</li> </ol>		

	<p>usuario</p> <ol style="list-style-type: none"> <li>2. CREAR PROYECTO: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el sistema.</li> <li>3. MODIFICAR PROYECTO: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el sistema cuando se intenta modificar un proyecto existente.</li> <li>4. ELIMINAR PROYECTO: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el sistema cuando se intenta borrar un proyecto existente.</li> <li>5. VISUALIZAR PROYECTO PUBLICADO: El navegador mostrará el proyecto que se encuentra almacenado en la base de datos.</li> <li>6. VISUALIZAR PROYECTO NO PUBLICADO: El navegador será redirigido a la página de inicio.</li> </ol>
<p><b>PASOS</b></p>	<ol style="list-style-type: none"> <li>1. LISTAR PROYECTOS DE UN USUARIO <ul style="list-style-type: none"> <li>• Visitar el perfil del usuario actual.</li> <li>• Visualizar todos los proyectos marcados como publicados, que pertenezcan al usuario indicado. Los proyectos marcados como NO publicados no deberán ser desplegados.</li> </ul> </li> <li>2. CREAR PROYECTO <ul style="list-style-type: none"> <li>• Visitar la página de crear proyecto</li> <li>• Desplegar un mensaje solicitando al usuario iniciar sesión.</li> </ul> </li> <li>3. MODIFICAR PROYECTO <ul style="list-style-type: none"> <li>• Visitar la página de edición un proyecto creado previamente.</li> <li>• Desplegar un mensaje solicitando al usuario iniciar sesión.</li> </ul> </li> <li>4. ELIMINAR PROYECTO <ul style="list-style-type: none"> <li>• Visitar la página de un proyecto creado previamente por un usuario registrado.</li> <li>• Verificar que el link borrar no exista.</li> </ul> </li> <li>5. VISUALIZAR PROYECTO PUBLICADO <ul style="list-style-type: none"> <li>• Visitar la página de un proyecto que ha sido marcado como publicado.</li> <li>• Visualizar todos los datos pertinentes al proyecto.</li> </ul> </li> <li>6. VISUALIZAR PROYECTO NO PUBLICADO <ul style="list-style-type: none"> <li>• Visitar la página de un proyecto que ha sido marcado como NO publicado.</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>• Desplegar un mensaje solicitando al usuario iniciar sesión.</li> </ul>
<b>Módulos Asociados</b>	<ul style="list-style-type: none"> <li>• Programas</li> <li>• Proyectos</li> <li>• Resultados</li> <li>• Usuarios</li> <li>• Sub-líneas de Investigación</li> </ul>

Tabla 4.12. Pruebas: Administrar resultados

<b>NOMBRE</b>	<b>Administrar Resultados</b>	<b>PRUEBAS</b>	<b>P5</b>
<b>PROPÓSITO</b>	Verificar si es posible crear, eliminar, modificar y visualizar resultados dentro del sistema		
<b>PRERREQUISITOS</b>	Estar firmado en el sistema		
<b>UBICACIÓN</b>	Base de datos MySQL Navegador		
<b>ENTRADA</b>	Selección de la acción que desea hacer: <ol style="list-style-type: none"> <li>1. Listar resultados</li> <li>2. Crear resultado</li> <li>3. Modificar resultado</li> <li>4. Eliminar resultado</li> <li>5. Visualizar resultado</li> </ol>		
<b>ORÁCULO</b>	<ol style="list-style-type: none"> <li>1. LISTAR RESULTADOS PROPIOS DE UN PROYECTO: Mostrar todos los resultados pertenecientes a un proyecto, creados por el usuario actual.</li> <li>2. CREAR RESULTADO: La base de datos se encuentra actualizada con el nuevo resultado</li> <li>3. ELIMINAR RESULTADO: El resultado especificado ha sido eliminado de la base de datos</li> <li>4. MODIFICAR RESULTADO: La base de datos se encuentra actualizada con los cambios del resultado seleccionado.</li> <li>5. VISUALIZAR RESULTADO PUBLICADO: El navegador mostrará el resultado que se encuentra almacenado en la base de datos.</li> <li>6. VISUALIZAR RESULTADO PROPIO NO PUBLICADO: El navegador mostrará el resultado que se encuentra almacenado en la base de datos.</li> <li>7. VISUALIZAR RESULTADO NO PUBLICADO: El navegador será redirigido a la página de inicio.</li> </ol>		
<b>PASOS</b>	<ol style="list-style-type: none"> <li>1. LISTAR RESULTADOS DE UN PROYECTO PROPIO               <ul style="list-style-type: none"> <li>• Visitar un proyecto propio</li> </ul> </li> </ol>		

	<ul style="list-style-type: none"> <li>• Visualizar todos los resultados, estén marcados como publicados o no publicados, que pertenezcan al usuario actual.</li> </ul> <p>2. CREAR RESULTADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de inicio.</li> <li>• Seleccionar el Menú</li> <li>• Seleccionar “Crear Resultado”.</li> <li>• Llenar el título del resultado</li> <li>• Llenar la descripción del resultado</li> <li>• Seleccionar un tipo de resultado de la lista respectiva.</li> <li>• Seleccionar un proyecto al que pertenece</li> <li>• Pulsar el botón Crear Resultado.</li> <li>• Visualizar resultado.</li> </ul> <p>3. MODIFICAR RESULTADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un resultado creado previamente.</li> <li>• Dar click en el link Editar.</li> <li>• Modificar los datos necesarios</li> <li>• Pulsar el botón Modificar Resultado</li> </ul> <p>4. ELIMINAR RESULTADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un resultado creado previamente por el usuario que está firmado en el sistema</li> <li>• Pulsar el link Borrar</li> <li>• Ver mensaje de confirmación.</li> </ul> <p>5. VISUALIZAR RESULTADO PUBLICADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un resultado que ha sido marcado como publicado.</li> <li>• Visualizar todos los datos pertinentes al resultado</li> </ul> <p>6. VISUALIZAR RESULTADO PROPIO NO PUBLICADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un resultado que ha sido marcado como NO publicado, pero ha sido creado por el usuario actual.</li> <li>• Visualizar todos los datos pertinentes al resultado</li> </ul> <p>7. VISUALIZAR RESULTADO NO PUBLICADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un resultado que ha sido marcado como NO publicado y que NO ha sido creado por el usuario actual.</li> <li>• Redirigir a la página de inicio.</li> </ul>
<b>Módulos Asociados</b>	<ul style="list-style-type: none"> <li>• Programas</li> <li>• Proyectos</li> <li>• Resultados</li> <li>• Documentos</li> <li>• Usuarios</li> </ul>

Tabla 4.13. Pruebas: Acciones de usuarios anónimos relacionados a resultados

<b>NOMBRE</b>	<b>Acciones de Usuarios Anónimos Relacionados a Resultados</b>	<b>PRUEBAS</b>	<b>P6</b>
<b>PROPÓSITO</b>	Verificar si es posible crear, eliminar, modificar y visualizar resultados dentro del sistema		
<b>PRERREQUISITOS</b>	NO estar firmado en el sistema		
<b>UBICACIÓN</b>	Navegador		
<b>ENTRADA</b>	Seleccionar un requerimiento Selección de la acción que desea hacer <ol style="list-style-type: none"> <li>1. Listar resultados</li> <li>2. Crear resultado</li> <li>3. Modificar resultado</li> <li>4. Eliminar resultado</li> <li>5. Visualizar resultado</li> </ol>		
<b>ORÁCULO</b>	<ol style="list-style-type: none"> <li>1. VISUALIZAR RESULTADOS DE UN PROYECTO: Mostrar todos los resultados marcados como publicados que pertenecen a un proyecto.</li> <li>2. CREAR RESULTADO: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el sistema.</li> <li>3. MODIFICAR RESULTADO: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el sistema cuando se intenta modificar un resultado existente.</li> <li>4. ELIMINAR RESULTADO: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el sistema cuando se intenta borrar un resultado existente.</li> <li>5. VISUALIZAR RESULTADO PUBLICADO: El navegador mostrará el resultado que se encuentra almacenado en la base de datos.</li> <li>6. VISUALIZAR RESULTADO NO PUBLICADO: El navegador será redirigido a la página de inicio.</li> </ol>		
<b>PASOS</b>	<ol style="list-style-type: none"> <li>1. VISUALIZAR RESULTADOS DE UN PROYECTO:                             <ul style="list-style-type: none"> <li>• Visitar la página de un proyecto existente</li> <li>• Visualizar todos los resultados que estén marcados como publicados. Los resultados marcados como NO publicados no deberán ser desplegados.</li> </ul> </li> <li>2. CREAR RESULTADO                             <ul style="list-style-type: none"> <li>• Visitar la página de crear resultado</li> <li>• Desplegar un mensaje solicitando al usuario iniciar</li> </ul> </li> </ol>		

	<p>sesión.</p> <p>3. MODIFICAR RESULTADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de edición de un resultado creado previamente.</li> <li>• Desplegar un mensaje solicitando al usuario iniciar sesión.</li> </ul> <p>4. ELIMINAR RESULTADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un resultado creado previamente por un usuario registrado.</li> <li>• Verificar que el link borrar no exista.</li> </ul> <p>5. VISUALIZAR RESULTADO PUBLICADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un resultado que ha sido marcado como publicado.</li> <li>• Visualizar todos los datos pertinentes al programa</li> </ul> <p>6. VISUALIZAR RESULTADO NO PUBLICADO</p> <ul style="list-style-type: none"> <li>• Visitar la página de un resultado que ha sido marcado como NO publicado.</li> <li>• Desplegar un mensaje solicitando al usuario iniciar sesión.</li> </ul>
<b>Módulos Asociados</b>	<ul style="list-style-type: none"> <li>• Programas</li> <li>• Proyectos</li> <li>• Resultados</li> <li>• Usuarios</li> </ul>

Tabla 4.14. Pruebas: Administrar actividades

<b>NOMBRE</b>	<b>Administrar Actividades</b>	<b>PRUEBAS</b>	<b>P7</b>
<b>PROPÓSITO</b>	Verificar si es posible crear, eliminar, modificar y visualizar actividades dentro del sistema		
<b>PRERREQUISITOS</b>	Estar firmado en el sistema		
<b>UBICACIÓN</b>	Base de datos MySQL Navegador		
<b>ENTRADA</b>	Selección de la acción que desea hacer: <ol style="list-style-type: none"> <li>1. Listar actividades</li> <li>2. Crear actividad</li> <li>3. Modificar actividad</li> <li>4. Eliminar actividad</li> <li>5. Visualizar actividad</li> </ol>		
<b>ORÁCULO</b>	<ol style="list-style-type: none"> <li>1. LISTAR ACTIVIDADES PROPIAS: Mostrar todas las actividades creadas por el usuario actual.</li> <li>2. CREAR ACTIVIDAD: La base de datos se encuentra</li> </ol>		

	<p>actualizada con la nueva actividad</p> <ol style="list-style-type: none"> <li>3. ELIMINAR ACTIVIDAD: La actividad especificada ha sido eliminada de la base de datos</li> <li>4. MODIFICAR ACTIVIDAD: La base de datos se encuentra actualizada con los cambios de la actividad seleccionada.</li> <li>5. VISUALIZAR ACTIVIDAD PUBLICADA: El navegador mostrará la actividad que se encuentra almacenada en la base de datos.</li> <li>6. VISUALIZAR ACTIVIDAD PROPIA NO PUBLICADA: El navegador mostrará la actividad que se encuentra almacenada en la base de datos.</li> <li>7. VISUALIZAR ACTIVIDAD NO PUBLICADA: El navegador será redirigido a la página de inicio.</li> </ol>
<p><b>PASOS</b></p>	<ol style="list-style-type: none"> <li>1. LISTAR ACTIVIDADES PROPIAS <ul style="list-style-type: none"> <li>• Visitar la página “Mis Actividades”</li> <li>• Visualizar todas las actividades marcadas como publicadas o no publicadas, que pertenezcan al usuario actual.</li> </ul> </li> <li>2. CREAR ACTIVIDAD <ul style="list-style-type: none"> <li>• Visitar la página de inicio.</li> <li>• Seleccionar el Menú</li> <li>• Seleccionar “Crear Actividad”.</li> <li>• Llenar el título de la actividad</li> <li>• Llenar la descripción de la actividad</li> <li>• Seleccionar un programa al que pertenece</li> <li>• Pulsar el botón Crear Actividad.</li> <li>• Visualizar actividad.</li> </ul> </li> <li>3. MODIFICAR ACTIVIDAD <ul style="list-style-type: none"> <li>• Visitar la página de una actividad creada previamente.</li> <li>• Dar click en el link Editar.</li> <li>• Modificar los datos necesarios</li> <li>• Pulsar el botón Modificar Actividad</li> </ul> </li> <li>4. ELIMINAR ACTIVIDAD <ul style="list-style-type: none"> <li>• Visitar la página de una actividad creada previamente por el usuario que está firmado en el sistema</li> <li>• Pulsar el link para Borrar</li> <li>• Ver mensaje de confirmación.</li> </ul> </li> <li>5. VISUALIZAR ACTIVIDAD PUBLICADA <ul style="list-style-type: none"> <li>• Visitar la página de una actividad que ha sido marcada como publicada.</li> <li>• Visualizar todos los datos pertinentes a la actividad.</li> </ul> </li> <li>6. VISUALIZAR ACTIVIDAD PROPIA NO PUBLICADA</li> </ol>

	<ul style="list-style-type: none"> <li>• Visitar la página de una actividad que ha sido marcada como NO publicada, pero ha sido creada por el usuario actual.</li> <li>• Visualizar todos los datos pertinentes a la actividad</li> </ul> <b>7. VISUALIZAR ACTIVIDAD NO PUBLICADA</b> <ul style="list-style-type: none"> <li>• Visitar la página de una actividad que ha sido marcada como NO publicada y que NO ha sido creada por el usuario actual.</li> <li>• Redirigir a la página de inicio.</li> </ul>
<b>Módulos Asociados</b>	<ul style="list-style-type: none"> <li>• Programas</li> <li>• Actividades</li> <li>• Documentos</li> <li>• Usuarios</li> </ul>

Tabla 4.15. Pruebas: Acciones de usuarios anónimos relacionados a actividades

<b>NOMBRE</b>	<b>Acciones de Usuarios Anónimos Relacionados a Actividades</b>	<b>PRUEBAS</b>	<b>P8</b>
<b>PROPÓSITO</b>	Verificar si es posible crear, eliminar, modificar y visualizar actividades dentro del sistema		
<b>PRERREQUISITOS</b>	NO estar firmado en el sistema		
<b>UBICACIÓN</b>	Navegador		
<b>ENTRADA</b>	Seleccionar un requerimiento Selección de la acción que desea hacer <ol style="list-style-type: none"> <li>1. Listar actividades</li> <li>2. Crear actividad</li> <li>3. Modificar actividad</li> <li>4. Eliminar actividad</li> <li>5. Visualizar actividad</li> </ol>		
<b>ORÁCULO</b>	<ol style="list-style-type: none"> <li>1. VISUALIZAR ACTIVIDADES: Mostrar todos las actividades marcadas como publicadas del sistema.</li> <li>2. CREAR ACTIVIDAD: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el sistema.</li> <li>3. MODIFICAR ACTIVIDAD: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el sistema cuando se intenta modificar una actividad existente.</li> <li>4. ELIMINAR ACTIVIDAD: El sistema debe mostrar un mensaje que solicite al usuario iniciar sesión en el</li> </ol>		

	<p>sistema cuando se intenta borrar una actividad existente.</p> <p>5. VISUALIZAR ACTIVIDAD PUBLICADA: El navegador mostrará la actividad que se encuentra almacenada en la base de datos.</p> <p>6. VISUALIZAR ACTIVIDAD NO PUBLICADA: El navegador será redirigido a la página de inicio.</p>
<b>PASOS</b>	<p>1. VISUALIZAR ACTIVIDADES DE UN PROGRAMA:</p> <ul style="list-style-type: none"> <li>• Visitar la página de la lista de Actividades.</li> <li>• Visualizar todas las actividades que estén marcadas como publicadas. Las actividades marcadas como NO publicadas no deberán ser desplegadas.</li> </ul> <p>2. CREAR ACTIVIDAD</p> <ul style="list-style-type: none"> <li>• Visitar la página de crear actividad</li> <li>• Desplegar un mensaje solicitando al usuario iniciar sesión.</li> </ul> <p>3. MODIFICAR ACTIVIDAD</p> <ul style="list-style-type: none"> <li>• Visitar la página de edición una actividad creada previamente.</li> <li>• Desplegar un mensaje solicitando al usuario iniciar sesión.</li> </ul> <p>4. ELIMINAR ACTIVIDAD</p> <ul style="list-style-type: none"> <li>• Visitar la página de una actividad creada previamente por un usuario registrado.</li> <li>• Verificar que el link borrar no exista.</li> </ul> <p>5. VISUALIZAR ACTIVIDAD PUBLICADA</p> <ul style="list-style-type: none"> <li>• Visitar la página de una actividad que ha sido marcada como publicada.</li> <li>• Visualizar todos los datos pertinentes a la actividad.</li> </ul> <p>6. VISUALIZAR ACTIVIDAD NO PUBLICADA</p> <ul style="list-style-type: none"> <li>• Visitar la página de una actividad que ha sido marcada como NO publicada.</li> <li>• Desplegar un mensaje solicitando al usuario iniciar sesión.</li> </ul>
<b>Módulos Asociados</b>	<ul style="list-style-type: none"> <li>• Programas</li> <li>• Actividades</li> <li>• Usuarios</li> </ul>

### 4.3. DESARROLLO DE BOCETOS PARA LAS INTERFACES DEL SISTEMA

#### 4.3.1. MÓDULO: MISCELÁNEOS

##### 4.3.1.1. PÁGINA DE INICIO

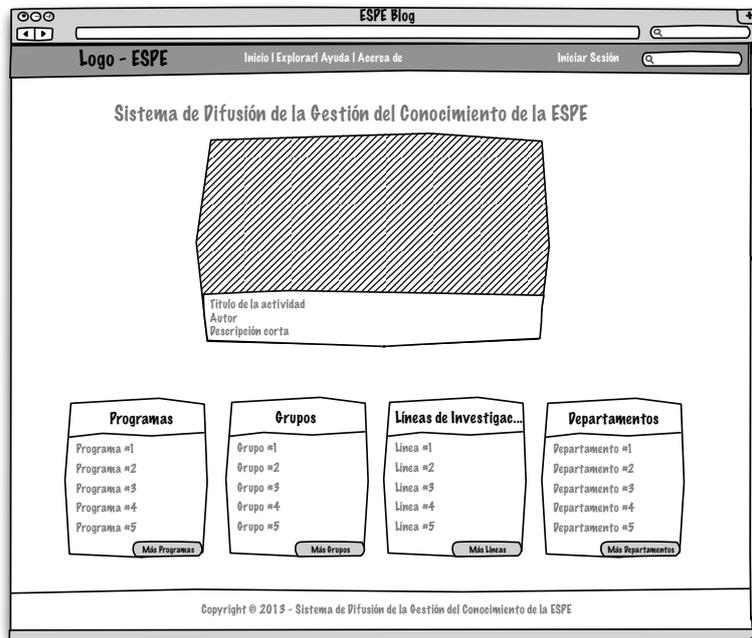


Figura 4.31. Boceto: Página de Inicio

##### 4.3.1.2. INICIO DE SESIÓN

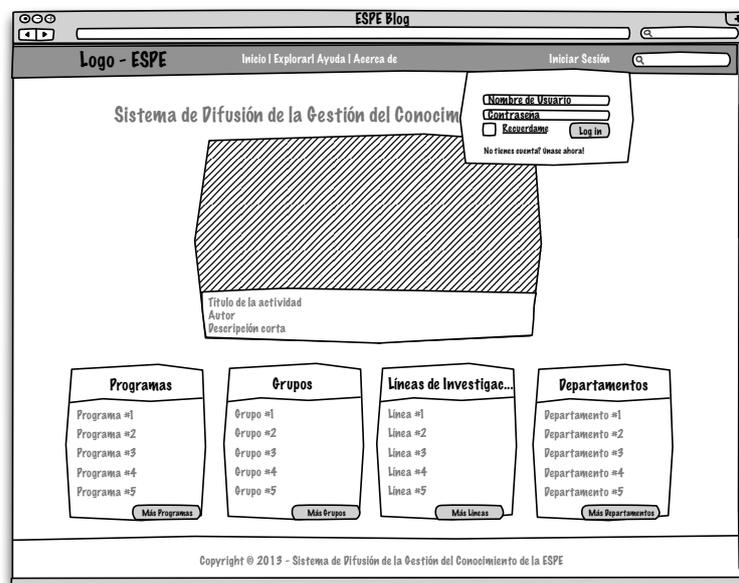


Figura 4.32. Boceto: Inicio de sesión

### 4.3.1.3. PÁGINA “EXPLORAR”

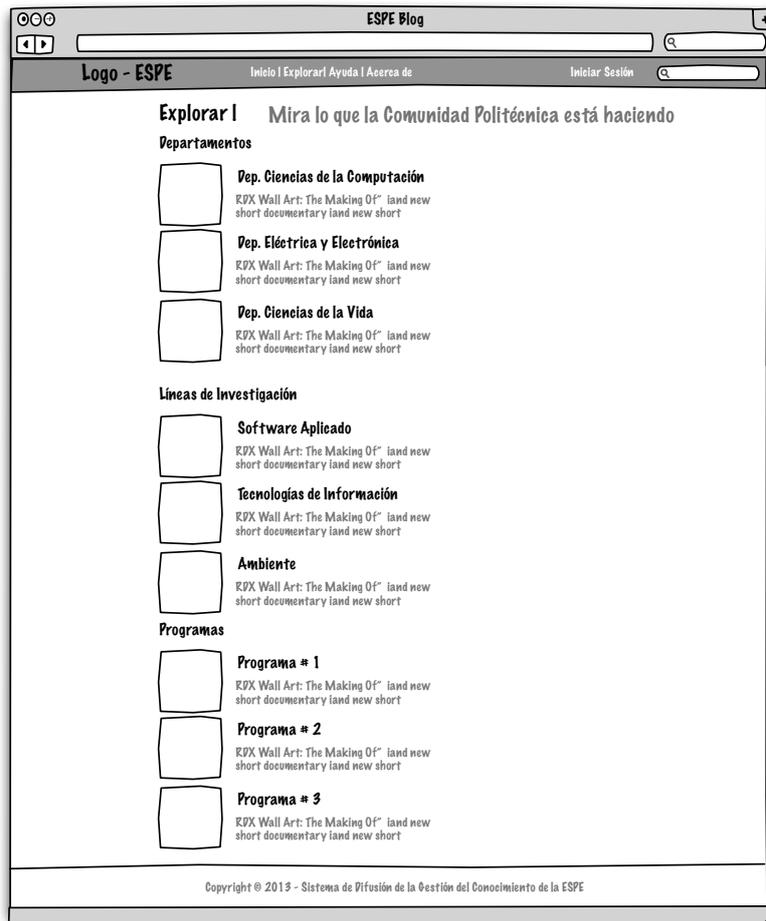


Figura 4.33. Boceto: Página “Explorar”

### 4.3.2. MÓDULO: USUARIOS

#### 4.3.2.1. ACTUALIZACIÓN DE DATOS DE UNA CUENTA

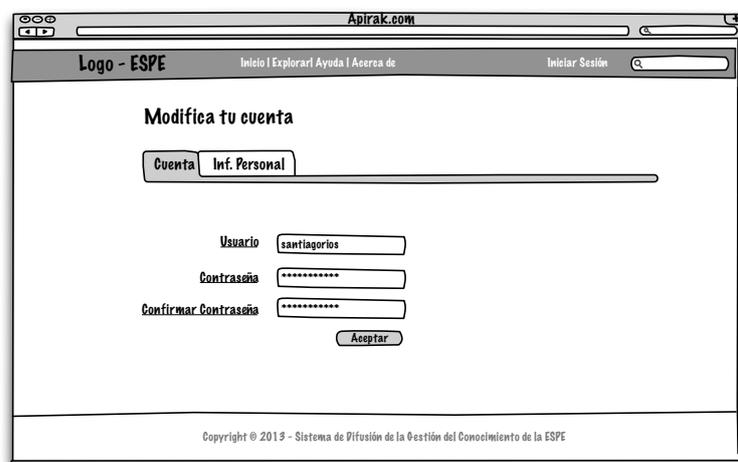


Figura 4.34. Boceto: Actualización de datos de una cuenta

### 4.3.2.2. ACTUALIZACIÓN DE DATOS DE PERFIL

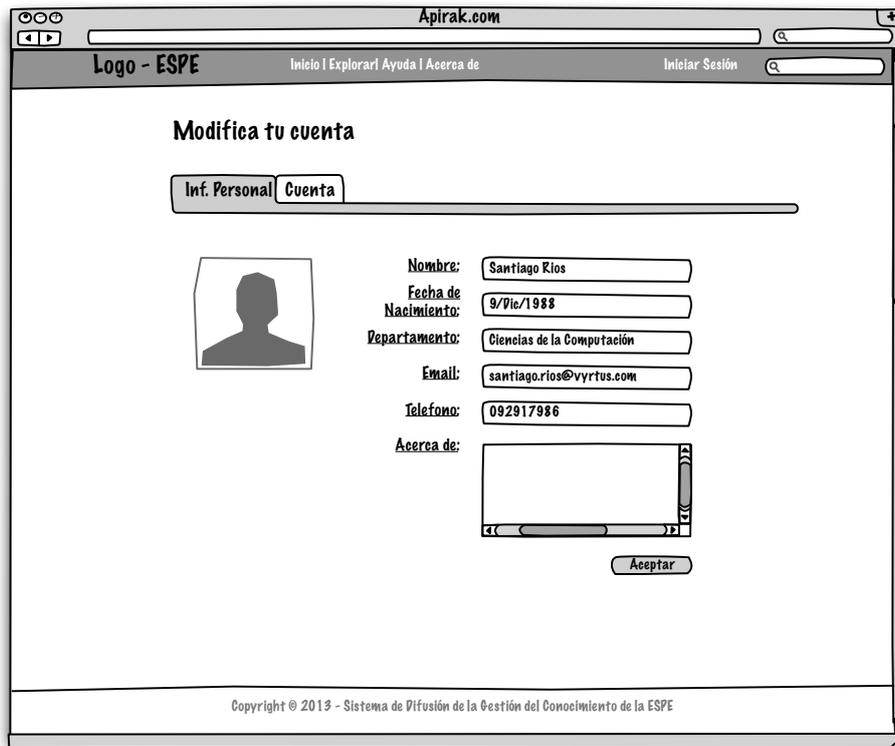


Figura 4.35. Boceto: Actualización de datos de perfil

### 4.3.2.3. PÁGINA DE UN USUARIO

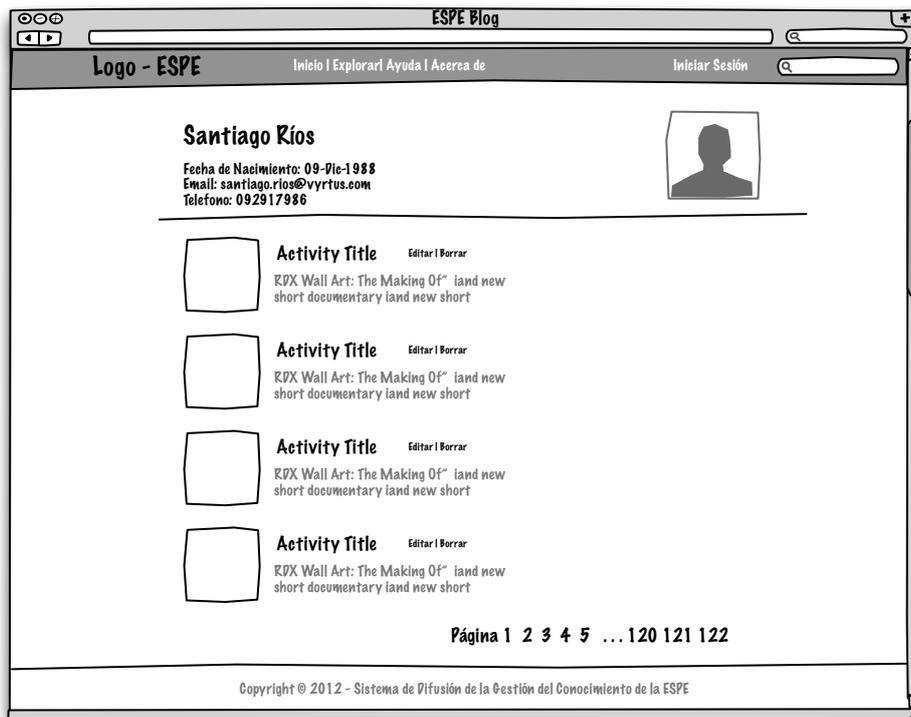


Figura 4.36. Boceto: Página de un usuario

### 4.3.3. MÓDULO: PROGRAMAS

#### 4.3.3.1. AGREGAR/MODIFICAR PROGRAMA

The screenshot shows a web browser window titled 'ESPE Blog'. The page header includes 'Logo - ESPE', navigation links ('Inicio | Explorar | Ayuda | Acerca de'), and a search bar with 'Iniciar Sesión'. The main content area is titled 'Agregar Programa de Investigación' and contains the following form fields:

- Título:** Programa # 1
- Fecha de Inicio:** 01/12/2012
- Duración:** 2 años (dropdown) and 6 meses (dropdown)
- Tipo de programa:** Investigación (dropdown)
- Grupo de Investigación:** Grupo #1 (dropdown)
- Sub-Líneas de Investigación:** Ingeniería de Software
- Publicado?:**
- Finalizado?:**
- Descripción:** A large text area with a scrollbar.

An 'Aceptar' button is located at the bottom right of the form. The footer contains the text: 'Copyright © 2013 - Sistema de Difusión de la Gestión del Conocimiento de la ESPE'.

Figura 4.37. Boceto: Agregar/Modificar programa

#### 4.3.3.2. VISUALIZAR PROGRAMA

The screenshot shows a web browser window titled 'ESPE Blog'. The page header is identical to the previous figure. The main content area is titled 'Programa # 1' and includes the following information:

- Propuesto por:** Santiago Kios
- Tipo de programa:** Investigación
- Fecha:** 27 de Abril del 2012
- Duración:** 2 años y 6 meses
- Estado:** En ejecución
- Departamentos:**
  - Ciencias de la Computación (Ingeniería de Software)
- Líneas de Investigación:**
  - Software Aplicado (Base de datos)
- Descripción:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc euismod mattis faucibus. Aliquam erat volutpat. Mauris id orci eros. Quisque a odio ac magna ornare pellentesque a eget massa. Mauris sodales, turpis quis scelerisque consequat, diam eros egestas enim, at pulvinar lorem mi vel lorem. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Suspendisse potenti. Donec et nulla purus. In magna urna....
- Proyectos componentes:**
  - Proyecto # 1
  - Proyecto # 2
  - Proyecto # 3
  - Proyecto # 4
  - Proyecto # 5

An 'Editar | Borrar' link is located in the top right corner of the content area. The footer contains the text: 'Copyright © 2013 - Sistema de Difusión de la Gestión del Conocimiento de la ESPE'.

Figura 4.38. Boceto: Visualizar programa

## 4.3.4. MÓDULO: PROYECTOS

### 4.3.4.1. AGREGAR/MODIFICAR PROYECTO

The wireframe shows a web browser window titled 'ESPE Blog'. The page header includes 'Logo - ESPE', navigation links ('Inicio | Explorar! Ayuda | Acerca de'), and a search bar. The main content area is titled 'Agregar Proyecto de Investigación'. It contains several input fields: 'Titulo:' with the value 'Proyecto # 1', 'Fecha de Inicio:' with '01/12/2012', 'Duración:' with '2 años' and '6 meses' options, 'Programa:' with 'Programa #1', and 'Sub-Líneas de Investigación:' with 'Ingeniería de Software'. Below these is a 'Usuario' section with 'Usuario:' (Santiago Ríos) and 'Rol:' (Tesisista) fields. There are checkboxes for 'Publicado?' and 'Finalizado?'. A large text area for 'Descripción' is at the bottom, followed by an 'Aceptar' button. The footer contains the copyright notice: 'Copyright © 2013 - Sistema de Difusión de la Gestión del Conocimiento de la ESPE'.

Figura 4.39. Boceto: Agregar/Modificar Proyecto

### 4.3.4.2. VISUALIZAR PROYECTO

The wireframe shows a web browser window titled 'ESPE Blog'. The page header is identical to the previous figure. The main content area is titled 'Proyecto # 1' and includes an 'Editar | Borrar' link. The project details are: 'Propuesto por: Santiago Ríos', 'Fecha: 27 de Abril del 2012', 'Duración: 2 años y 6 meses', 'Estado: En ejecución', and 'Programa: Programa #1'. The 'Departamentos:' section lists 'Ciencias de la Computación (Ingeniería de Software)'. The 'Líneas de Investigación:' section lists 'Software Aplicado (Base de datos)'. The 'Descripción:' section contains a paragraph of Lorem Ipsum text. The 'Resultados' section lists five items: 'Resultado # 1', 'Resultado # 2', 'Resultado # 3', 'Resultado # 4', and 'Resultado # 5'. The footer contains the copyright notice: 'Copyright © 2013 - Sistema de Difusión de la Gestión del Conocimiento de la ESPE'.

Figura 4.40. Boceto: Visualizar proyecto

## 4.3.5. MÓDULO: RESULTADOS

### 4.3.5.1. AGREGAR/MODIFICAR RESULTADO

The wireframe shows a web browser window titled 'ESPE Blog'. The page header includes 'Logo - ESPE', navigation links ('Inicio | Explorar | Ayuda | Acerca de'), and a search bar. The main content area is titled 'Agregar Resultado de Investigación' and contains the following elements:

- Nombre:** A text input field containing 'Resultado # 1'.
- Tipo de resultado:** A dropdown menu with 'Capítulo de libro' selected.
- Proyecto:** A dropdown menu with 'Proyecto #1' selected.
- Publicado:** An unchecked checkbox.
- Archivo:** A section with a text input for 'Nombre' containing 'Archivo #1' and a 'Choose file' button.
- Agregar Archivo:** A section with a text input for 'Titulo' containing 'Imagen # 1' and a placeholder image with diagonal hatching.
- Agregar una Imagen:** A section with a large empty text area for 'Descripción' and an 'Aceptar' button at the bottom right.

At the bottom of the page, there is a copyright notice: 'Copyright © 2013 - Sistema de Difusión de la Gestión del Conocimiento de la ESPE'.

Figura 4.41. Boceto: Agregar/Modificar resultado

### 4.3.5.2. VISUALIZAR RESULTADO

The wireframe shows a web browser window titled 'ESPE Blog'. The page header includes 'Logo - ESPE', navigation links ('Inicio | Explorar | Ayuda | Acerca de'), and a search bar. The main content area displays the details of a result:

- Resultado # 1** (Title)
- Presentado por:** Santiago Rios
- Proyecto:** Proyecto #1
- Descargar Archivo (Titulo del archivo)** (Link)
- Descargar Archivo (Titulo del archivo #2)** (Link)

Below the text, there is a large empty rectangular box on the left and a grid of six smaller placeholder images with diagonal hatching on the right. Underneath is a section for 'Descripción:' followed by a paragraph of placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc euismod mattis faucibus. Aliquam erat volutpat. Mauris id orci eros. Quisque a odio ac magna ornare pellentesque a eget massa. Mauris sodales, turpis quis scelerisque consequat, diam eros egestas enim, at pulvinar lorem mi vel lorem. Class aptent taciti sociosq ad litora torquent per conubia nostra...'.

At the bottom of the page, there is a copyright notice: 'Copyright © 2013 - Sistema de Difusión de la Gestión del Conocimiento de la ESPE'.

Figura 4.42. Boceto: Visualizar resultado

## 4.3.6. MÓDULO: ACTIVIDADES

### 4.3.6.1. AGREGAR/MODIFICAR ACTIVIDADES

The wireframe shows a web browser window titled 'ESPE Blog'. The page header includes 'Logo - ESPE', navigation links ('Inicio | Explorar | Ayuda | Acerca de'), and a search bar ('Iniciar Sesión'). The main content area is titled 'Agregar Actividad' and contains the following form elements:

- Nombre:** Text input field with 'Actividad # 1'.
- Programa:** Dropdown menu with 'Programa #1' selected.
- Fecha:** Text input field with '12/12/2012'.
- Sub-Líneas de Investigación:** Text input field with 'Ingeniería de Software'.
- Publicado:** A checkbox.
- Archivo:** A section with a 'Nombre:' text input field containing 'Archivo #1' and a 'Choose file' button.
- Agregar Archivo:** A section with an 'Imagen:' label and a 'Titulo:' text input field containing 'Imagen # 1'. Below the title is a placeholder image represented by a square with diagonal hatching.
- Agregar una Imagen:** A section with a 'Descripción:' label and a large text area with a vertical scrollbar.
- Acceptar:** A button at the bottom right of the form.

At the bottom of the page, there is a copyright notice: 'Copyright © 2012 - Sistema de Gestión de Conocimiento de la ESPE'.

Figura 4.43. Boceto: Agregar/Modificar actividades

### 4.3.6.2. VISUALIZAR ACTIVIDAD

The wireframe shows a web browser window titled 'ESPE Blog'. The page header includes 'Logo - ESPE', navigation links ('Inicio | Explorar | Ayuda | Acerca de'), and a search bar ('Iniciar Sesión'). The main content area is titled 'Actividad # 1' and contains the following elements:

- Presentado por:** Santiago Kios.
- Programa:** Programa #1.
- Departamentos:** A list containing 'Ciencias de la Computación (Ingeniería de Software)'.
- Líneas de Investigación:** A list containing 'Software Aplicado (Base de datos)'.
- Descargar Archivo:** Two links: '(título del archivo)' and '(título del archivo #2)'. Below these links is a grid of six square placeholders: a large one on the left, and five smaller ones on the right (two in the top row, three in the bottom row). Some of these squares contain diagonal hatching.
- Descripción:** A text area containing a placeholder Lorem Ipsum text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc euismod mattis faucibus. Aliquam erat volutpat. Mauris id orci eros. Quisque a odio ac magna ornare pellentesque a eget massa. Mauris sodales, turpis quis scelerisque consequat, diam eros egestas enim, at pulvinar lorem mi vel lorem. Class aptent taciti sociosqu ad litora torquent per conubia nostra...'.

At the bottom of the page, there is a copyright notice: 'Copyright © 2013 - Sistema de Difusión de la Gestión del Conocimiento de la ESPE'.

Figura 4.44. Boceto: Visualizar actividad

## 4.4. DESARROLLO DE LAS INTERFACES DEL SISTEMA USANDO HTML

### 4.4.1. MÓDULO: MISCELÁNEOS

#### 4.4.1.1. PÁGINA DE INICIO



Figura 4.45. Interfaz HTML: Página de Inicio

#### 4.4.1.2. INICIO DE SESIÓN



Figura 4.46. Interfaz HTML: Inicio de sesión

### 4.4.1.3. PÁGINA “EXPLORAR”

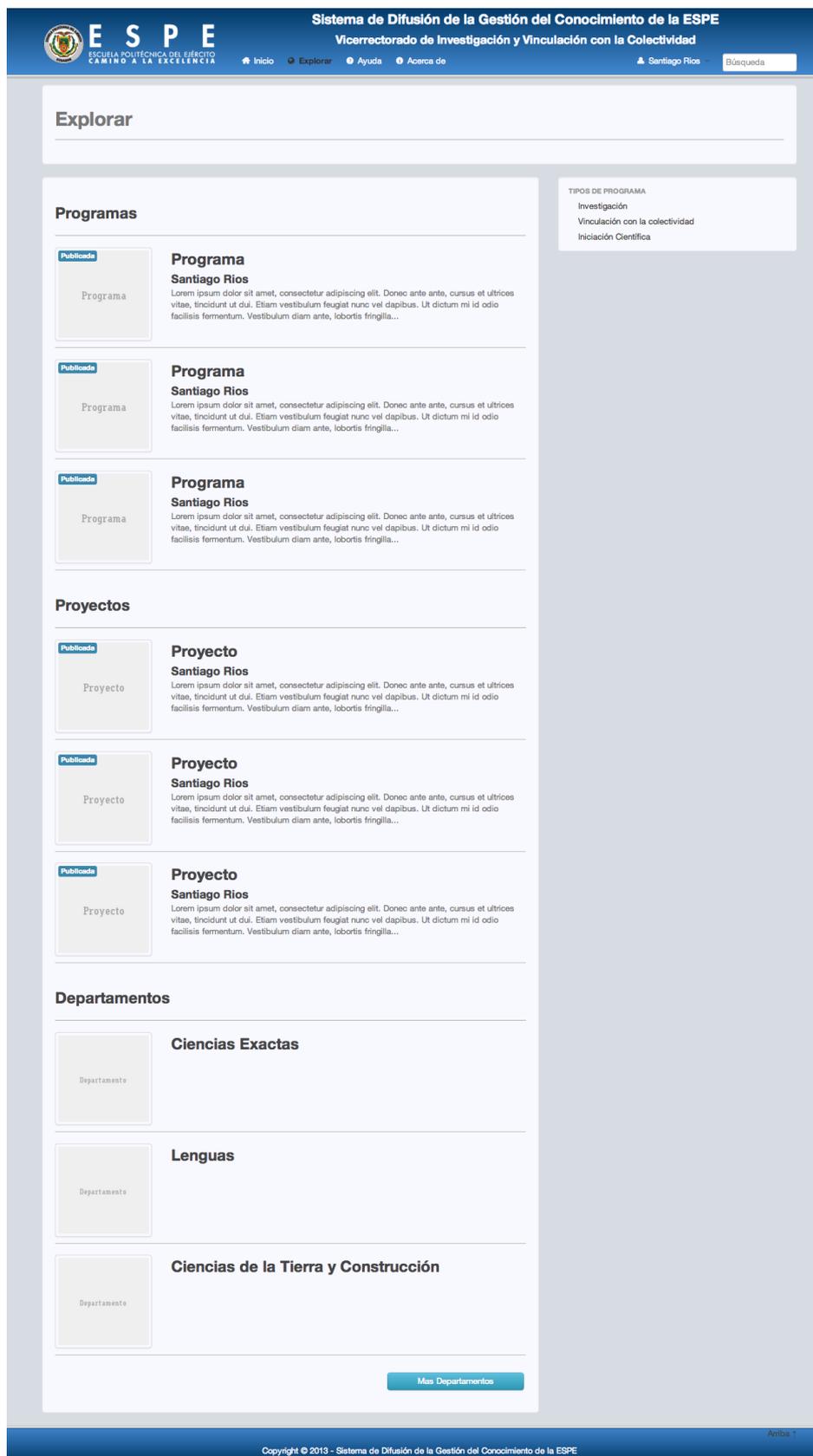


Figura 4.47. Interfaz HTML: Página “Explorar”

## 4.4.2. MÓDULO: USUARIOS

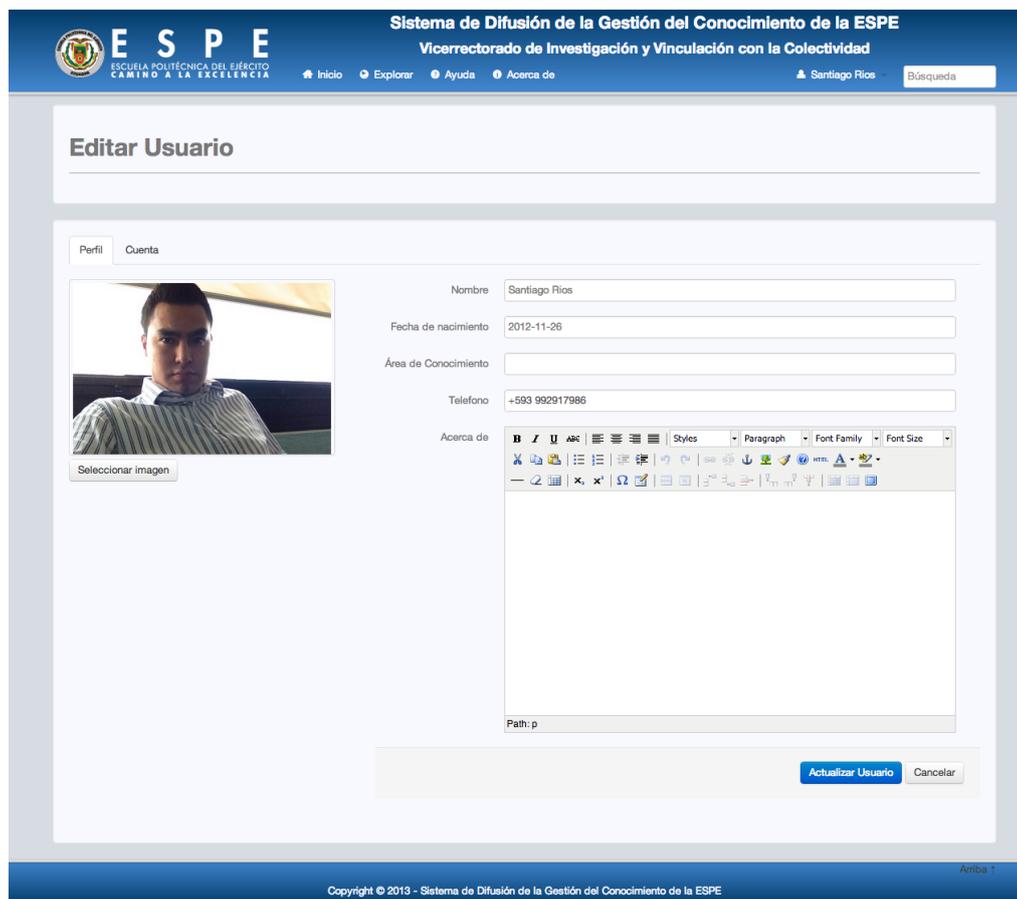
### 4.4.2.1. ACTUALIZACIÓN DE DATOS DE UNA CUENTA



The screenshot shows the 'Edit Usuario' interface. At the top, there is a header with the ESPE logo and navigation links. The main content area has a tabbed interface with 'Perfil' and 'Cuenta' tabs. The 'Cuenta' tab is active, displaying the following fields: 'Email' (srios@espe.edu.ec), 'Contraseña', and 'Confirmar Contraseña'. There are 'Actualizar Usuario' and 'Cancelar' buttons at the bottom. The footer contains copyright information for 2012.

Figura 4.48. Interfaz HTML: Actualización de datos de una cuenta

### 4.4.2.2. ACTUALIZACIÓN DE DATOS DE PERFIL



The screenshot shows the 'Editar Usuario' interface. At the top, there is a header with the ESPE logo and navigation links. The main content area has a tabbed interface with 'Perfil' and 'Cuenta' tabs. The 'Perfil' tab is active, displaying a profile picture of a man and the following fields: 'Nombre' (Santiago Rios), 'Fecha de nacimiento' (2012-11-26), 'Área de Conocimiento', 'Telefono' (+593 992917986), and 'Acercas de'. There is a rich text editor for the 'Acercas de' field with a toolbar. There are 'Actualizar Usuario' and 'Cancelar' buttons at the bottom. The footer contains copyright information for 2013.

Figura 4.49. Interfaz HTML: Actualización de datos de perfil

### 4.4.2.3. PÁGINA DE UN USUARIO

The screenshot shows the user profile page for Santiago Rios. The header includes the ESPE logo and navigation links. The profile section displays the user's name, email, and a profile picture. Below this are three sections: 'Programas', 'Proyectos', and 'Actividades', each containing a list of items with status indicators (Publicada, No publicada) and placeholder text. A sidebar on the right lists 'TIPOS DE PROGRAMA' and 'USUARIOS'.

**ESPE**  
ESCUELA POLITÉCNICA DEL EJÉRCITO  
CAMINO A LA EXCELENCIA

Sistema de Difusión de la Gestión del Conocimiento de la ESPE  
Vicerrectorado de Investigación y Vinculación con la Colectividad

Inicio Explorar Ayuda Acerca de Santiago Rios Búsqueda

### Santiago Rios

Email: srrios@espe.edu.ec

**Programas**

- Programa**  
Santiago Rios  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ante ante, cursus et ultrices vitae, tincidunt ut dui. Etiam vestibulum feugiat nunc vel dapibus. Ut dictum mi id odio facilisis fermentum. Vestibulum diam ante, lobortis fringilla...
- Programa**  
Santiago Rios  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ante ante, cursus et ultrices vitae, tincidunt ut dui. Etiam vestibulum feugiat nunc vel dapibus. Ut dictum mi id odio facilisis fermentum. Vestibulum diam ante, lobortis fringilla...
- Programa**  
Santiago Rios  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ante ante, cursus et ultrices vitae, tincidunt ut dui. Etiam vestibulum feugiat nunc vel dapibus. Ut dictum mi id odio facilisis fermentum. Vestibulum diam ante, lobortis fringilla...

**Proyectos**

- Test**  
Varios autores  
dasda
- Proyecto**  
Santiago Rios  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ante ante, cursus et ultrices vitae, tincidunt ut dui. Etiam vestibulum feugiat nunc vel dapibus. Ut dictum mi id odio facilisis fermentum. Vestibulum diam ante, lobortis fringilla...

**Actividades**

- Test**  
Santiago Rios  
sadsada
- Actividad**  
Santiago Rios Febrero 01, 2013  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ante ante, cursus et ultrices vitae, tincidunt ut dui. Etiam vestibulum feugiat nunc vel dapibus. Ut dictum mi id odio facilisis fermentum. Vestibulum diam ante, lobortis fringilla...

TIPOS DE PROGRAMA  
Investigación  
Vinculación con la colectividad  
Iniciación Científica

USUARIOS  
Programas de los grupos  
Proyectos de los grupos

Copyright © 2013 - Sistema de Difusión de la Gestión del Conocimiento de la ESPE Arriba ↑

Figura 4.50. Interfaz HTML: Página de un usuario

### 4.4.3. MÓDULO: PROGRAMAS

#### 4.4.3.1. AGREGAR/MODIFICAR PROGRAMA

The screenshot shows the 'Nuevo Programa' (New Program) form within the ESPE system. The header includes the ESPE logo and the text 'Sistema de Difusión de la Gestión del Conocimiento de la ESPE' and 'Vicerrectorado de Investigación y Vinculación con la Colectividad'. The form contains the following fields and controls:

- Título:** A text input field.
- Fecha de inicio:** A date selection field.
- Duración:** Two dropdown menus for 'años' (years) and 'meses' (months).
- Tipo de programa:** A dropdown menu.
- Grupo de investigación:** A dropdown menu.
- Sub-líneas de Investigación:** A text input field.
- Publicado?:** A checkbox.
- Finalizado?:** A checkbox.
- Descripción:** A rich text editor with a toolbar containing various formatting options like bold, italic, underline, bulleted list, numbered list, link, unlink, image, and text color.
- Path:** A small text field with the value 'p'.
- Buttons:** 'Crear Programa' (Create Program) and 'Cancelar' (Cancel).

At the bottom of the page, there is a copyright notice: 'Copyright © 2012 - Sistema de Gestión de Conocimiento de la ESPE' and a user name 'Ámba'.

Figura 4.51. Interfaz HTML: Agregar/Modificar programa

## 4.4.3.2. VISUALIZAR PROGRAMA

**Sistema de Difusión de la Gestión del Conocimiento de la ESPE**  
Vicerrectorado de Investigación y Vinculación con la Colectividad

Inicio Explorar Ayuda Acerca de Santiago Ríos Búsqueda

### Programa

**Propuesto por:** Santiago Ríos  
**Tipo de programa:** Investigación  
**Fecha de inicio:** 2013-02-01  
**Duración:** 1 año 2 meses  
**State:** En ejecución  
**Descripción:**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ante ante, cursus et ultrices vitae, tincidunt ut dui. Etiam vestibulum feugiat nunc vel dapibus. Ut dictum mi id odio facilisis fermentum. Vestibulum diam ante, lobortis fringilla ullamcorper quis, pulvinar quis felis. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nunc eu ullamcorper ante. Donec euismod convallis purus, eget mollis risus dapibus in. Integer enim magna, posuere non facilisis sit amet, interdum eu lacus. Mauris mollis massa vitae diam tristique venenatis. Mauris aliquet pulvinar tempus. Vivamus quis justo sed metus egestas porta vel vitae lectus. Integer vestibulum malesuada mi a rhoncus.

Quisque vitae odio ut lacus dignissim aliquam vel ac elit. Sed pulvinar pellentesque lacus, sit amet ultrices dui rutrum vitae. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Mauris nec libero sem, quis aliquet velit. Nam consequat, mauris a sodales commodo, ipsum magna malesuada dolor, et placerat massa orci nec nisi. Ut dignissim pulvinar est, eu porttitor sem consequat nec. Donec sapien arcu, blandit non ultricies pretium, convallis quis mi. Vestibulum a odio mi. Sed ultricies quam a nulla posuere in porttitor metus malesuada. Curabitur porta porttitor vehicula.

Morbi in rutrum diam. In nulla mi, semper non blandit a, pharetra quis ante. Vestibulum posuere velit sit amet lorem accumsan quis gravida mi pulvinar. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Phasellus consequat eros a lorem sagittis bibendum. Aenean vitae uma ut metus elementum laoreet et ac est. Nunc malesuada congue magna, at tempus nunc fringilla eu. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec scelerisque diam at dui adipiscing elementum. Vestibulum risus eros, rhoncus et tristique vitae, posuere eget eros. Pellentesque odio augue, dapibus vitae tempus ut, molestie sed uma. Quisque varius rhoncus placerat.

Sed commodo lobortis lacinia. Duis scelerisque mauris et turpis pharetra facilisis. Cras eu auctor nisi. Fusce ac nulla in enim facilisis cursus quis ut massa. Curabitur vehicula tempus lorem, non semper elit porttitor in. Proin et cursus orci. Phasellus consectetur volutpat iaculis. Ut dapibus tempor ligula, ut adipiscing lectus luctus ut. Ut in ornare purus. Integer aliquet tellus sit amet ante pretium dapibus. Mauris molestie hendrerit risus, ac posuere neque interdum vitae. Vivamus id facilisis arcu.

Donec magna ante, cursus vel imperdiet suscipit, auctor sit amet enim. Donec mattis sapien sit amet quam feugiat sed eleifend augue feugiat. Duis risus sapien, tincidunt eu mattis a, fermentum dictum quam. Donec dignissim vestibulum pulvinar. Sed auctor mi in nisi consequat lobortis nec nec odio. Praesent auctor, eros vel semper tincidunt, eros quam suscipit tortor, at hendrerit nisi risus eu est. Sed vel imperdiet leo. Integer tempor neque sed sem fermentum ultrices. Nam egestas fringilla aliquet. Quisque varius, risus ut facilisis porttitor, elit diam fringilla purus, quis commodo ante uma eget nisi. Nulla a feugiat ipsum. Ut id orci lectus. Quisque ac nisi dui.

**Proyectos componentes:**

- Proyecto (Pendiente)
- Proyecto (Pendiente)
- Proyecto (Pendiente)

0 comments

Leave a message...

Discussion Community Share

No one has commented yet.

Comment feed Subscribe via email DISQUS

EXPLORAR

PROGRAMAS

- Programa
- Programa
- Programa

DEPARTAMENTOS

- Ciencias Humanas y Sociales
- Ciencias de la Vida
- Ciencias Económicas, Administrativas y de Come...
- Ciencias de la Energía y Mecánica
- Ciencias Exactas

LÍNEAS DE INVESTIGACIÓN

- Estudios Geoespaciales
- Sistemas Eléctricos, Electrónicos Y Computaciona...
- Energía
- Seguridad y Defensa
- Automática y Control

Copyright © 2013 - Sistema de Difusión de la Gestión del Conocimiento de la ESPE

Ariba ↑

Figura 4.52. Interfaz HTML: Visualizar programa

## 4.4.4. MÓDULO: PROYECTOS

### 4.4.4.1. AGREGAR/MODIFICAR PROYECTO

**ESPE**  
ESCUELA POLITÉCNICA DEL EJÉRCITO  
CAMINO A LA EXCELENCIA

Sistema de Difusión de la Gestión del Conocimiento de la ESPE  
Vicerrectorado de Investigación y Vinculación con la Colectividad

Inicio Explorar Ayuda Acerca de Santiago Ríos Búsqueda

### Nuevo Proyecto

\* Título

Fecha de inicio

Duración años meses

\* Programa

Sub-líneas de Investigación

Publicado?

Finalizado?

**Usuario**

Usuario

Rol

Quitar este usuario

Agregar usuario

Descripción

Path: p

Crear Proyecto Cancelar

Copyright © 2012 - Sistema de Gestión de Conocimiento de la ESPE

Arriba ↑

Figura 4.53. Interfaz HTML: Agregar/Modificar proyecto

## 4.4.4.2. VISUALIZAR PROYECTO

**Sistema de Difusión de la Gestión del Conocimiento de la ESPE**  
Vicerrectorado de Investigación y Vinculación con la Colectividad

Inicio Explorar Ayuda Acerca de Santiago Rios Búsqueda

### Proyecto

**Propuesto por:** Santiago Rios  
**Fecha de inicio:** 2013-02-06  
**Duración:** 2 años  
**Estado:** Pendiente  
**Programa:** Programa  
**Descripción:**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ante ante, cursus et ultrices vitae, tincidunt ut du. Etiam vestibulum feugiat nunc vel dapibus. Ut dictum mi et odio facilisis fermentum. Vestibulum diam ante, lobortis fringilla ullamcorper quis, pulvinar quis fela. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nunc eu ullamcorper ante. Donec euismod convallis purus, eget mollis risus dapibus in. Integer enim magna, posuere non facilisis sit amet, interdum eu lacus. Mauris mollis massa vitae diam tristique venenatis. Mauris aliquet pulvinar tempus. Vivamus quis justo sed metus egestas porta vel vitae lectus. Integer vestibulum malesuada mi a rhoncus.

Quisque vitae odio ut lacus dignissim aliquam vel ac elit. Sed pulvinar pellentesque lacus, sit amet ultricies du rutrum vitae. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Mauris nec libero sem, quis aliquet velit. Nam consequat, mauris a sodales commodo, ipsum magna malesuada dolor, et placerat massa orci nec nisi. Ut dignissim pulvinar est, eu porttitor sem consequat nec. Donec sapien arcu, blandit non ultrices pretium, convallis quis mi. Vestibulum a odio mi. Sed ultrices quam a nulla posuere in porttitor metus malesuada. Curabitur porta porttitor vehicula.

Morbi in rutrum diam. In nulla mi, semper non blandit a, pharetra quis ante. Vestibulum posuere velit sit amet lorem accumsan quis gravida mi pulvinar. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Phasellus consequat eros a lorem sagittis bibendum. Aenean vitae urna ut metus elementum laoreet et ac est. Nunc malesuada congue magna, at tempus nunc fringilla eu. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec scelerisque diam at dui adipiscing elementum. Vestibulum risus eros, rhoncus et tristique vitae, posuere eget eros. Pellentesque odio augue, dapibus vitae tempus ut, molestie sed urna. Quisque varius rhoncus placerat.

Sed commodo lobortis laecina. Duis scelerisque mauris et turpis pharetra facilisis. Cras eu auctor nisi. Fusce ac nulla in enim facilisis cursus quis ut massa. Curabitur vehicula tempus lorem, non semper elit porttitor in. Proin et cursus orci. Phasellus consectetur volutpat laculis. Ut dapibus tempor ligula, ut adipiscing lectus luctus ut. Ut in ornare purus. Integer aliquet tellus sit amet ante pretium dapibus. Mauris molestie hendrerit risus, ac posuere neque interdum vitae. Vivamus id facilisis arcu.

Donec magna ante, cursus vel imperdiet suscipit, auctor sit amet enim. Donec mattis sapien sit amet quam feugiat sed eleifend augue feugiat. Duis risus sapien, tincidunt eu mattis a, fermentum dictum quam. Donec dignissim vestibulum pulvinar. Sed auctor mi in nisi consequat lobortis nec nec odio. Praesent auctor, eros vel semper tincidunt, eros quam suscipit tortor, at hendrerit nisi risus eu est. Sed vel imperdiet leo. Integer tempor neque sed sem fermentum ultrices. Nam egestas fringilla aliquet. Quisque varius, risus ut facilisis porttitor, elit diam fringilla purus, quis commodo ante urna eget nisi. Nulla a feugiat ipsum. Ut et orci lectus. Quisque ac nisi du.

### Resultados:

- Resultado (Paper científico)
- Resultado (Paper científico)

0 comments

Leave a message...

Discussion | Community | Share

No one has commented yet.

Comment feed | Subscribe via email | DISQUS

EXPLORAR

PROGRAMAS

- Programa
- Programa
- Programa

DEPARTAMENTOS

- Ciencias de la Computación
- Ciencias Exactas
- Ciencias Económicas, Administrativas y de Come...
- Lenguas
- Ciencias de la Tierra y Construcción

LÍNEAS DE INVESTIGACIÓN

- Ambiente
- Nanotecnología
- Emprendedorismo Y Estrategia Organizacional
- Estudios Geoespaciales
- Automática y Control

Arriba 1

Copyright © 2013 - Sistema de Difusión de la Gestión del Conocimiento de la ESPE

Figura 4.54. Interfaz HTML: Visualizar proyecto

## 4.4.5. MÓDULO: RESULTADOS

### 4.4.5.1. AGREGAR/MODIFICAR RESULTADO

The screenshot shows a web interface for adding or modifying a result. At the top, there is a blue header with the ESPE logo and the text 'Sistema de Difusión de la Gestión del Conocimiento de la ESPE' and 'Vicerrectorado de Investigación y Vinculación con la Colectividad'. Below the header, there are navigation links: 'Inicio', 'Explorar', 'Ayuda', and 'Acerca de'. On the right side of the header, there is a user profile 'Santiago Rios' and a search box labeled 'Búsqueda'.

The main content area is titled 'Nuevo Resultado' and contains the following form elements:

- Título:** A text input field.
- Tipo de resultado:** A dropdown menu.
- Proyecto:** A dropdown menu.
- Publicado?:** A checkbox.
- Archivo:** A section with a 'Nombre' text input, a 'Choose File' button (with 'No file chosen' text), and a 'Quitar este archivo' button.
- Agregar archivo:** A section with a 'Titulo' text input, a placeholder image with the text 'no image', a 'Seleccionar imagen' button, and a 'Quitar esta imagen' button.
- Agregar una imagen:** A section with a rich text editor (WYSIWYG) containing various formatting tools (bold, italic, underline, text color, background color, link, unlink, list, indent, outdent, undo, redo, etc.) and a 'Path: p' label.

At the bottom of the form, there are two buttons: 'Crear Resultado' and 'Cancelar'.

At the bottom of the page, there is a footer with the text 'Copyright © 2012 - Sistema de Gestión de Conocimiento de la ESPE' and a small 'Arriba ↑' link.

Figura 4.55. Interfaz HTML: Agregar/Modificar resultado

## 4.4.5.2. VISUALIZAR RESULTADO

**ESPE**  
ESCUELA POLITÉCNICA DEL EJÉRCITO  
CAMINO A LA EXCELENCIA

Sistema de Difusión de la Gestión del Conocimiento de la ESPE  
Vicerrectorado de Investigación y Vinculación con la Colectividad

[Inicio](#) [Explorar](#) [Ayuda](#) [Acerca de](#)

Santiago Rios

### Resultado

Presentado por: Santiago Rios  
Proyecto: Proyecto



**Descripción:**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ante ante, cursus et ultrices vitae, tincidunt ut dui. Etiam vestibulum feugiat nunc vel dapibus. Ut dictum mi id odio facilisis fermentum. Vestibulum diam ante, lobortis fringilla ullamcorper quis, pulvinar quis felis. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nunc eu ullamcorper ante. Donec euismod convallis purus, eget mollis risus dapibus in. Integer enim magna, posuere non facilisis sit amet, interdum eu lacus. Mauris mollis massa vitae diam tristique venenatis. Mauris aliquet pulvinar tempus. Vivamus quis justo sed metus egestas porta vel vitae lectus. Integer vestibulum malesuada mi a rhoncus.

Quisque vitae odio ut lacus dignissim aliquam vel ac elit. Sed pulvinar pellentesque lacus, sit amet ultrices dui rutrum vitae. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Mauris nec libero sem, quis aliquet velit. Nam consequat, mauris a sodales commodo, ipsum magna malesuada dolor, et placerat massa orci nec nisi. Ut dignissim pulvinar est, eu porttitor sem consequat nec. Donec sapien arcu, blandit non ultricies pretium, convallis quis mi. Vestibulum a odio mi. Sed ultricies quam a nulla posuere in porttitor metus malesuada. Curabitur porta porttitor vehicula.

Morbi in rutrum diam. In nulla mi, semper non blandit a, pharetra quis ante. Vestibulum posuere velit sit amet lorem accumsan quis gravida mi pulvinar. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Phasellus consequat eros a lorem sagittis bibendum. Aenean vitae uma ut metus elementum laoreet et ac est. Nunc malesuada congue magna, at tempus nunc fringilla eu. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec scelerisque diam at dui adipiscing elementum. Vestibulum risus eros, rhoncus et tristique vitae, posuere eget eros. Pellentesque odio augue, dapibus vitae tempus ut, molestie sed uma. Quisque varius rhoncus placerat.

Sed commodo lobortis lacinia. Duis scelerisque mauris et turpis pharetra facilisis. Cras eu auctor nisi. Fusce ac nulla in enim facilisis cursus quis ut massa. Curabitur vehicula tempus lorem, non semper elit porttitor in. Proin et cursus orci. Phasellus consectetur volutpat iaculis. Ut dapibus tempor ligula, ut adipiscing lectus luctus ut. Ut in ornare purus. Integer aliquet tellus sit amet ante pretium dapibus. Mauris molestie hendrerit risus, ac posuere neque interdum vitae. Vivamus id facilisis arcu.

Donec magna ante, cursus vel imperdiet suscipit, auctor sit amet enim. Donec mattis sapien sit amet quam feugiat sed eleifend augue feugiat. Duis risus sapien, tincidunt eu mattis a, fermentum dictum quam. Donec dignissim vestibulum pulvinar. Sed auctor mi in nisi consequat lobortis nec nec odio. Praesent auctor, eros vel semper tincidunt, eros quam suscipit tortor, at hendrerit nisi risus eu est. Sed vel imperdiet leo. Integer tempor neque sed sem fermentum ultrices. Nam egestas fringilla aliquet. Quisque varius, risus ut facilisis porttitor, elit diam fringilla purus, quis commodo ante uma eget nisi. Nulla a feugiat ipsum. Ut id orci lectus. Quisque ac nisi dui.

0 comments ★ 0

Discussion | Community | Share 

No one has commented yet.

[Comment feed](#) [Subscribe via email](#) **DISQUS**

EXPLORAR

PROGRAMAS

- Programa
- Programa
- Programa

DEPARTAMENTOS

- Seguridad
- Eléctrica y Electrónica
- Ciencias Humanas y Sociales
- Ciencias de la Tierra y Construcción
- Ciencias de la Energía y Mecánica

LÍNEAS DE INVESTIGACIÓN

- Seguridad Informática
- Curriculum
- Sistemas Agroproductivos Y Biotecnológicos Sost...
- Tiempo Libre, Deporte Y, Calidad De Vida
- Nanotecnología

Copyright © 2013 - Sistema de Difusión de la Gestión del Conocimiento de la ESPE Amiba 1

Figura 4.56. Interfaz HTML: Visualizar resultado

## 4.4.6. MÓDULO: ACTIVIDADES

### 4.4.6.1. AGREGAR/MODIFICAR ACTIVIDAD

The screenshot shows the 'Nueva Actividad' (New Activity) form within the 'Sistema de Difusión de la Gestión del Conocimiento de la ESPE' (Knowledge Management System of the ESPE). The header includes the ESPE logo and navigation links: Inicio, Explorar, Ayuda, and Acerca de. The user is logged in as Santiago Ríos, and there is a search bar.

The form fields include:

- Título:** A text input field.
- Programa:** A dropdown menu.
- Fecha:** A text input field containing '2012-12-08'.
- Archivo:** A section for uploading a file, including a 'Nombre' field, a 'Choose File' button (with 'No file chosen' text), and a 'Quitar este archivo' button.
- Sub-líneas de Investigación:** A text input field.
- Publicado?:** A checkbox that is checked.
- Imagen:** A section for adding an image, including a 'Titulo' field, a placeholder box with 'no image' text, a 'Seleccionar imagen' button, and a 'Quitar esta imagen' button.
- Descripción:** A rich text editor with a toolbar containing various formatting and editing tools.
- Path:** A text input field containing 'p'.

At the bottom of the form are two buttons: 'Crear Actividad' (Create Activity) and 'Cancelar' (Cancel).

Copyright © 2012 - Sistema de Gestión de Conocimiento de la ESPE

Figura 4.57. Interfaz HTML: Agregar/Modificar actividad

## 4.4.6.2. VISUALIZAR ACTIVIDAD

The screenshot displays the website for the Escuela Politécnica del Ejército (ESPE). The header features the ESPE logo and the text 'Sistema de Difusión de la Gestión del Conocimiento de la ESPE' and 'Vicerrectorado de Investigación y Vinculación con la Colectividad'. Navigation links include 'Inicio', 'Explorar', 'Ayuda', and 'Acercas de'. A search bar is located in the top right corner.

### Escuela Politécnica del Ejército

**Autor:** Santiago Rios  
**Fecha:** 2012-11-27

**Descripción:**

La ESPE - Escuela Superior Politécnica del Ejército es un centro de educación superior ubicado en Sangotquí (Quito - Ecuador). Tiene su origen en la Escuela de Oficiales Ingenieros, creada el 16 de junio de 1922. Actualmente cuenta con sedes en Sangotquí (principal), Latacunga, Santo Domingo de los Tsáchilas.

Constituye uno de los institutos de educación superior más prestigiosos del Ecuador, el Consejo Nacional de Evaluación y Acreditación de la Educación Superior del Ecuador (CONEA), en el 2009 la ubicó en la categoría "A", la máxima calificación otorgada a 11 instituciones en el país y sólo por detrás de dos escuelas politécnicas. Adicionalmente, el CONEA extendió la carta de Acreditación a la Escuela Politécnica del Ejército el 7 de Enero del 2010. Desde 2012 pertenece a la Red Ecuatoriana de Universidades para Investigación y Postgrados.

**0 comments** ★ 0

Leave a message...

Discussion | Community | Share

No one has commented yet.

Comment feed | Subscribe via email

DISQUS

EXPLORAR  
PROGRAMAS  
Nano Satelites  
GRUPOS DE INVESTIGACIÓN  
Grupo #1  
DEPARTAMENTOS  
Ciencias de la Energía y Mecánica  
Ciencias Exactas  
Lenguas  
Ciencias de la Tierra y Construcción  
Ciencias de la Computación  
LÍNEAS DE INVESTIGACIÓN  
Seguridad Informática  
Estudios Geospaciales  
Modelamiento Matemático  
Sistemas Agroproductivos Y Biotecnológicos Sost...  
Estructuras y Construcciones

Arriba ↑

Copyright © 2012 - Sistema de Gestión de Conocimiento de la ESPE

Figura 4.58. Interfaz HTML: Visualizar actividad

## **CAPÍTULO 5:**

### **TRANSICIÓN**

#### **5.1. INSTALACIÓN DEL SISTEMA**

##### **5.1.1. REQUERIMIENTOS PREVIOS PARA LA INSTALACIÓN DEL SISTEMA**

###### **5.1.1.4. SERVIDOR WEB / ALMACENAMIENTO**

###### **5.1.1.4.1. HARDWARE**

- Procesador doble núcleo de 2GHz
- Memoria RAM: 4 GB
- Disco Duro: 100 GB

###### **5.1.1.4.2. SOFTWARE**

- Ubuntu Server 12.10 o superior
- Git (versión 1.7.9.6 o superior)
- Ruby (versión 1.9.3p194 o superior)
- Rails (versión 3.1.3 o superior)
- Apache2

### **5.1.1.5. SERVIDOR DE BASE DE DATOS**

#### **5.1.1.5.1. HARDWARE**

- Procesador doble núcleo de 2GHz
- Memoria RAM: 2 GB
- Disco Duro: 60 GB

#### **5.1.1.5.2. SOFTWARE**

- Ubuntu Server 12.10 o superior
- Git (versión 1.7.9.6 o superior)
- MySQL (versión 5.5 o superior)

### **5.1.2. MANUAL DE INSTALACIÓN DEL SISTEMA**

#### **5.1.2.4. SERVIDOR DE BASE DE DATOS**

##### **5.1.2.4.1. INSTALAR MYSQL**

La instalación de MySQL se la realizado al ejecutar el siguiente comando:

```
sudo apt-get install mysql-server
```

##### **5.1.2.4.2. INSTALAR MYSQL**

Se deberá editar el archivo `/etc/mysql/my.cnf` para configurar los ajustes básicos tales como: ubicación del log, puerto, etc.

Después de realizar los cambios en el archivo, se deberá reiniciar el servicio:

```
sudo /etc/init.d/mysql start
```

#### 5.1.2.4.3. CREAR BASE DE DATOS

Para crear una base de datos en el servidor, se deberá realizar una conexión por medio del siguiente comando:

```
mysql -h=<Dirección IP> -u=<usuario> -p=<contraseña>
```

Una vez que se ha realizado correctamente la conexión, se deberá correr la siguiente sentencia:

```
create database <Nombre de la base de datos>
```

Sin embargo no es necesario reproducir la estructura de la misma manualmente, ya que Rails se encargará de realizar esta tarea posteriormente.

#### 5.1.1.4. SERVIDOR WEB / ALMACENAMIENTO

##### 5.1.1.4.1. INSTALAR APACHE

La instalación de apache se la realizado al ejecutar el siguiente comando:

```
sudo aptitude install apache2
```

##### 5.1.1.4.2. INSTALAR RVM Y RAILS

RVM servirá para aislar las gemas de la aplicación, además de instar Ruby en caso que aun no esté instalado. Para su instalación se deberá ejecutar el siguiente comando:

```
curl -L https://get.rvm.io | bash -s stable --rails
```

#### 5.1.1.4.3. INSTALAR PASSENGER

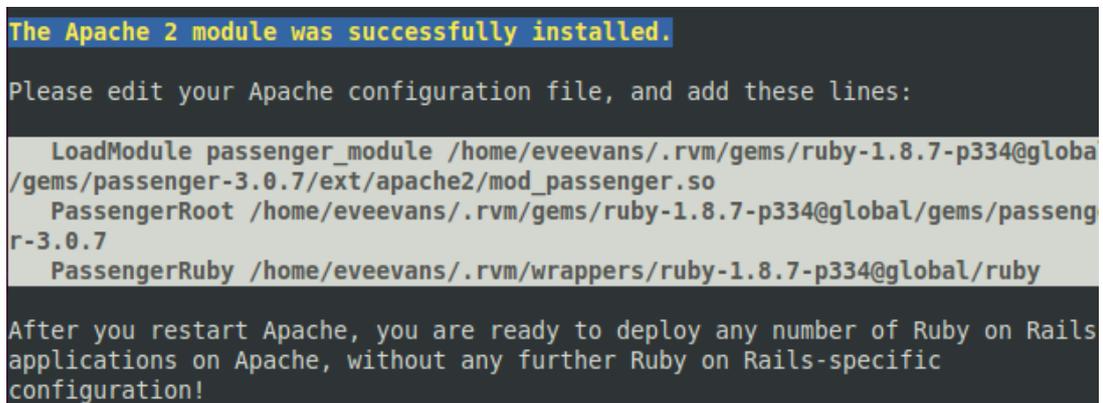
Para instalar passenger basta ejecutar los siguientes comandos:

```
sudo gem install passenger
```

```
sudo passenger-install-apache2-module
```

#### 5.1.1.4.4. CONFIGURAR APACHE

Una vez finalizada la instalación anterior (passenger-install-apache2-module), se indicaran las líneas que se deben agregar al archivo de configuración de Apache:



```
The Apache 2 module was successfully installed.  
Please edit your Apache configuration file, and add these lines:  
  
LoadModule passenger_module /home/eveevans/.rvm/gems/ruby-1.8.7-p334@global/gems/passenger-3.0.7/ext/apache2/mod_passenger.so  
PassengerRoot /home/eveevans/.rvm/gems/ruby-1.8.7-p334@global/gems/passenger-3.0.7  
PassengerRuby /home/eveevans/.rvm/wrappers/ruby-1.8.7-p334@global/ruby  
  
After you restart Apache, you are ready to deploy any number of Ruby on Rails applications on Apache, without any further Ruby on Rails-specific configuration!
```

Figura 5.1. Pantalla del programa passenger-install-apache-2-module

El archivo de configuración comúnmente se encuentra en `/etc/apache2/apache2.conf`

#### 5.1.1.4.5. CONFIGURAR APLICACIÓN

##### 5.1.1.4.5.1. COPIAR EL PROYECTO DESDE GITHUB

Se deberá acceder al path:

```
/var/www/
```

Desde esta ruta, se puede obtener el proyecto en su última versión por medio del siguiente comando:

```
git clone git@github.com:santiagorioss/Tesis.git
```

#### **5.1.1.4.5.2. CONFIGURAR EL VIRTUAL HOST DE APACHE**

Se deberá crear un nuevo virtual host con los siguientes datos:

```
<VirtualHost *:80>  
  ServerName www.nombredelsitio.com  
  DocumentRoot /var/www/tesis/public/  
  <Directory /var/www/tesis/public/>  
    AllowOverride all  
    Options -MultiViews  
  </Directory>  
</VirtualHost>
```

#### **5.1.1.4.5.3. CONFIGURAR EL ACCESO A BASE DE DATOS**

Es necesario indicar a la aplicación cómo se va a comunicar con la base de datos, esto se lo realiza al configurar el archivo tesis/config/database.yml, en la sección producción:

```
production:  
  adapter: mysql2  
  encoding: utf8  
  database: <base de datos>  
  username: <usuario>  
  password: <contraseña>  
  host: <dirección IP del servidor>  
  port: 3306
```

#### 5.1.1.4.5.4. CONFIGURAR EL ACCESO AL SERVIDOR DE CORREO

Es necesario indicar a la aplicación la cuenta de correo electrónico que se usará para enviar notificaciones a los usuarios. Para realizar esta tarea es necesario modificar el archivo `config/environments/production.rb`, y reemplazar la información correspondiente:

```
config.action_mailer.smtp_settings = {
  :address => "<dirección del servidor de correo>",
  :port    => <puerto>,
  :domain  => '<dominio>',
  :user_name => '<usuario>',
  :password => '<contraseña>',
  :authentication => '<tipo de autenticación>',
  :enable_starttls_auto => true
}
```

#### 5.1.1.4.5.5. MIGRACIÓN DE LA ESTRUCTURA DE LA BASE DE DATOS

Para que la aplicación pueda almacenar información en la base de datos, se deberá ejecutar el siguiente comando:

```
rake db:migrate RAILS_ENV="production"
```

#### 5.1.1.4.6. FINALIZAR INSTALACIÓN

Finalmente, para terminar la instalación del sistema es necesario reiniciar el servicio de apache2:

```
sudo /etc/init.d/apache2 restart
```

## 5.2. EVALUACIÓN DEL SISTEMA

Para la evaluación del sistema se han utilizado las herramientas Google Analytics y New Relic, las mismas que permiten tener una visión más profunda de quién está usando el sistema y cómo se está desempeñando el mismo. Esta sección pretende brindar una guía a futuro, para el seguimiento del proyecto.

### 5.2.1. GOOGLE ANALYTICS

Por medio de la recopilación de datos que se ha realizado desde diciembre del 2012, se han obtenido los siguientes cuadros demostrativos que permitirán, a futuro, revisar las estadísticas del sitio.

#### 5.2.1.1. VISIÓN GENERAL

##### 5.2.1.1.1. VISITAS DESDE DICIEMBRE 2012 A FEBRERO 2013



Figura 5.2. Evaluación: Visitas desde Diciembre 2012

##### 5.2.1.1.2. VISITANTES EXCLUSIVOS DESDE DICIEMBRE DE 2012 A FEBRERO DE 2013

Este indicador, cuenta cada usuario de forma individual, independientemente del número de accesos que realice



Figura 5.3. Evaluación: Visitantes exclusivos desde Diciembre 2012

### 5.2.1.1.3. PÁGINAS VISTAS DESDE DICIEMBRE 2012 A FEBRERO 2013

Una página vista hace referencia a la visualización de una página de su sitio web que el código de seguimiento de Google Analytics está controlando. Si un usuario vuelve a cargar la página después de que se haya cargado completamente, esto contará como una página vista adicional. Si un usuario navega a una página diferente y más tarde vuelve a la página original, también se registrará una segunda página vista.



Figura 5.4. Evaluación: Páginas vistas desde Diciembre 2012

### 5.2.1.1.4. PROMEDIO DE PÁGINAS VISTAS DURANTE UNA VISITA



Figura 5.5. Evaluación: Promedio de páginas vistas durante una visita

## 5.2.1.2. DATOS DEMOGRÁFICOS

### 5.2.1.2.1. IDIOMA

El cuadro a continuación permite conocer el idioma del navegador que utilizó un usuario para acceder al sitio web.

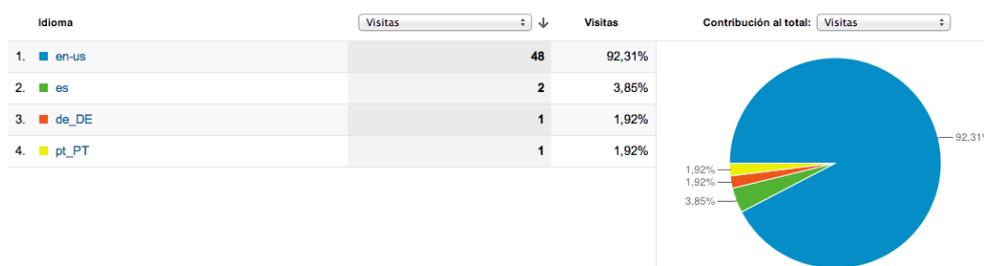


Figura 5.6. Evaluación: Idioma de los visitantes

### 5.2.1.2.2. UBICACIÓN

El siguiente gráfico muestra la ubicación desde donde se han accedido a la aplicación web:



Figura 5.7. Evaluación: Ubicación de los visitantes

### 5.2.1.3. TECNOLOGÍA

#### 5.2.1.3.1. NAVEGADOR Y SISTEMA OPERATIVO

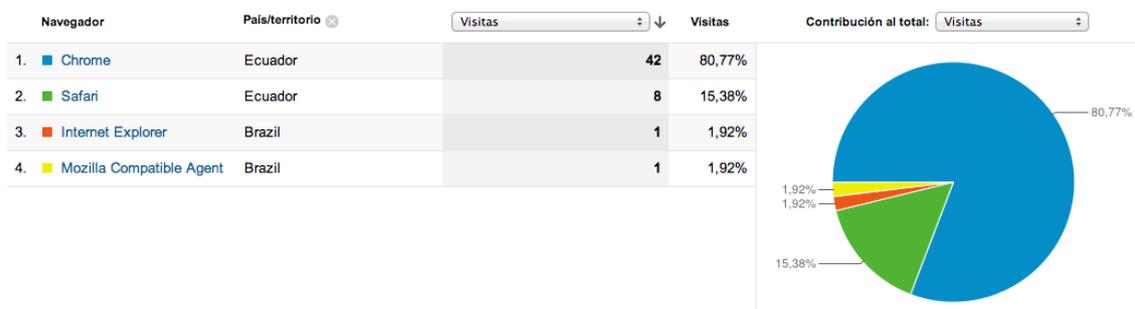


Figura 5.8. Evaluación: Navegador y Sistema Operativo de las visitas

#### 5.2.1.3.2. PROVEEDOR DE INTERNET



Figura 5.9. Evaluación: Proveedor de internet de las visitas

#### 5.2.1.4. FLUJO DE VISITANTES

El “Flujo de Visitantes” es un gráfico que representa el flujo a través de la aplicación web, empezando por un parámetro definido por el administrador (en este caso país de origen) para observar el camino, desde donde inició la visita hasta donde finalizó:

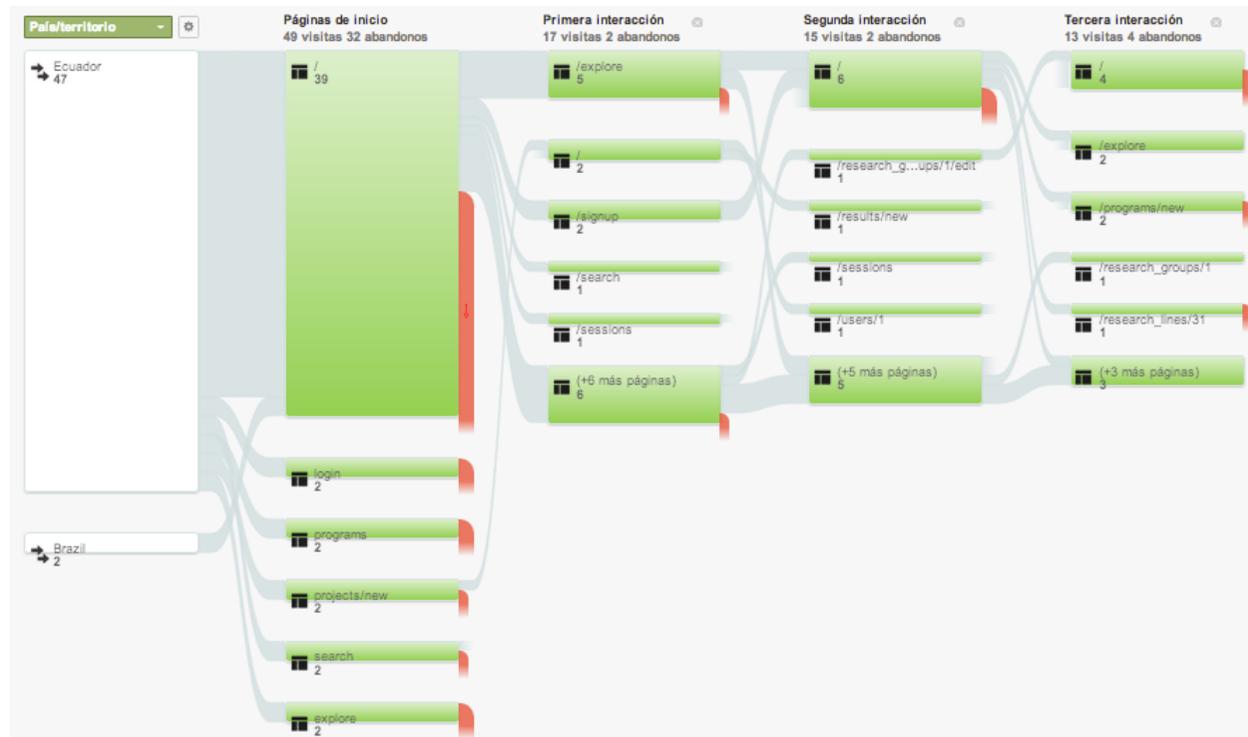


Figura 5.10. Evaluación: Flujo de visitantes

## 5.2.2. NEW RELIC

### 5.2.2.1. VISIÓN GENERAL

#### 5.2.2.1.1. TIEMPO DE CARGA DEL BROWSER

Es el tiempo promedio de carga de página dividido en los siguientes segmentos de tiempo:

- **Cola de pedidos:** El tiempo de espera entre el servidor web y el código de aplicación. Números grandes indican un servidor de aplicaciones de ocupado.
- **Aplicación web:** El tiempo pasado en el código de aplicación.
- **Red:** La latencia de la red, o el tiempo que tarda una petición a través del Internet.
- **Procesamiento del DOM:** En el navegador, analizar e interpretar el HTML. Medido por el navegador "DOMContent" del evento.
- **Renderizado de la página:** En el navegador, mostrando el código HTML, Javascript que se ejecuta en línea y carga imágenes. Medido por el navegador "Load" del evento.

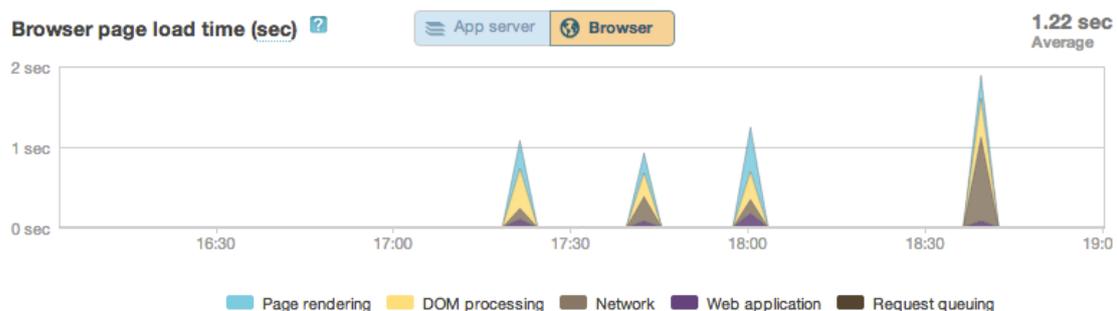


Figura 5.11. Evaluación: Tiempo de carga del browser

### 5.2.2.1.2. TIEMPO DE RESPUESTA DEL SERVIDOR

El tiempo de respuesta del servidor depende de varios factores, como son la base de datos, la programación en sí (en este gráfico se lo refiere como Ruby), y finalmente el encolamiento de las peticiones. Estos datos permiten valorar la calidad del software que se ha desarrollado como parte del Sistema de Difusión de Gestión del Conocimiento de la ESPE:

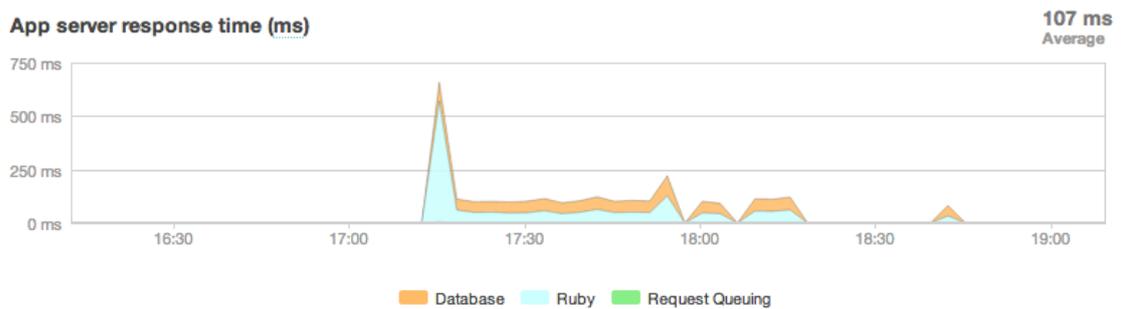


Figura 5.12. Evaluación: Tiempo de respuesta del servidor

### 5.2.2.1.3. THROUGHPUT

New Relic mide el throughput utiliza los siguientes indicadores: PPM (o páginas por minuto) y RPM (o peticiones por minuto):

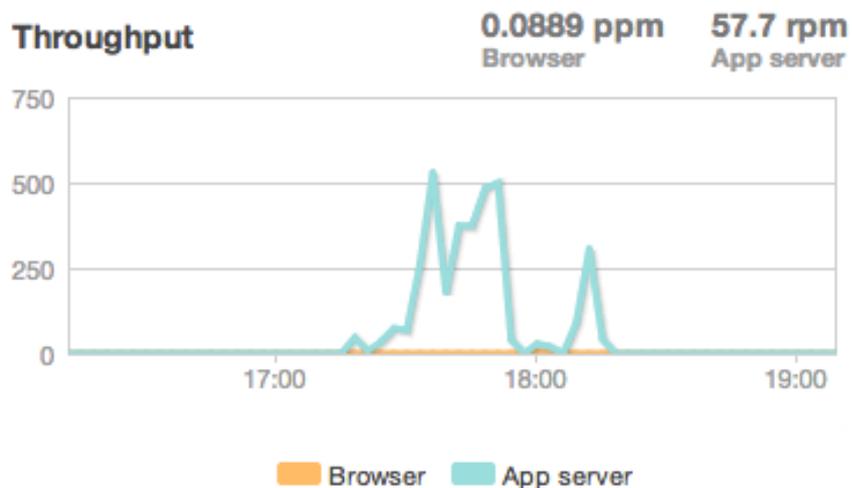


Figura 5.13. Evaluación: Throughput

#### 5.2.2.1.4. PUNTAJE APDEX

El puntaje Apdex está configurado como 0,5 segundos. Eso significa que las peticiones que respondieron en menos de 0,5 segundos son satisfactorios (s, en color celeste), las que respondiendo entre 0,5 segundos y 2,0 segundos están marcadas como tolerantes (t, en colores verde y amarillo), y las respuestas que tomaron más de 2,0 segundos son frustrantes (f, en color rojo).

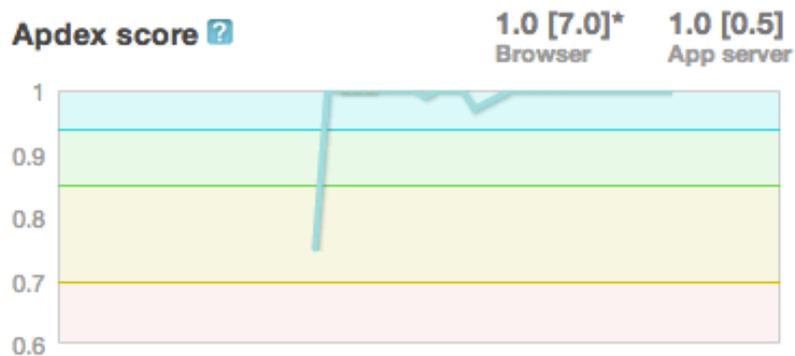


Figura 5.14. Evaluación: Puntaje Apdex

#### 5.2.2.1.5. MAPA DE TIEMPO

Permite definir el tiempo promedio que toma una página en cargarse dentro del navegador de un usuario. Este tiempo está compuesto por el tiempo de respuesta de la base de datos (database), el tiempo de ejecución de la programación del sistema para desplegar una página en especial (tesis), y finalmente el tiempo que se demora el navegador en procesar la respuesta entregada por el servidor (end user).

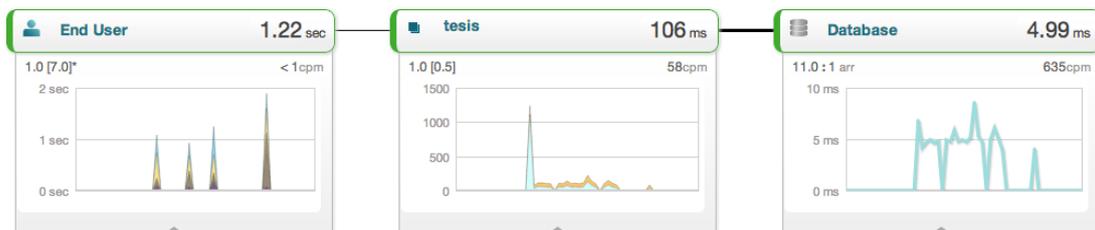


Figura 5.15. Evaluación: Mapa de tiempo

## 5.2.2.2. BASE DE DATOS

### 5.2.2.2.1. THROUGHPUT DE LA BASE DE DATOS

El throughput de la base de datos es medido en CPM o Calls per minute (Llamadas por minuto), y permite conocer la carga que debe soportar la base de datos para desempeñar un buen trabajo:

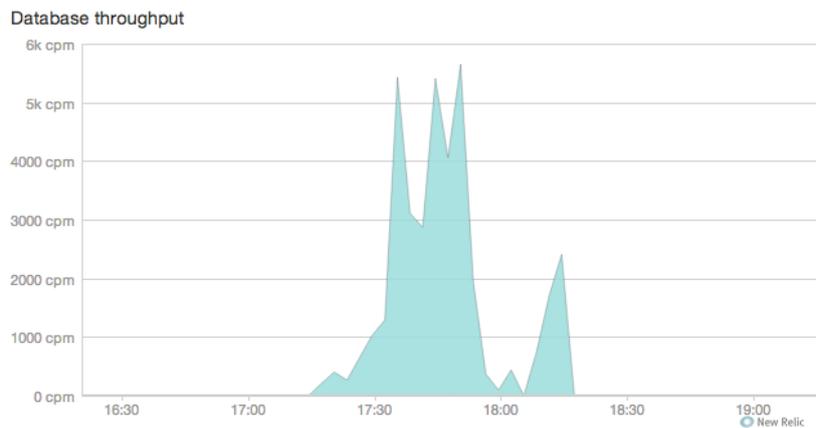


Figura 5.16. Evaluación: Throughput de la base de datos

### 5.2.2.2.2. TIEMPO DE RESPUESTA DE LA BASE DE DATOS

El tiempo de respuesta de la base de datos es importante, ya que afecta directamente a la experiencia del usuario, en la gráfica a continuación se indica el tiempo que demoró la base de datos en entregar el resultado solicitado por un cliente:



Figura 5.17. Evaluación: Tiempo de respuesta de la base de datos

### 5.2.2.2.3. OPERACIONES MÁS COMUNES

El gráfico a continuación, permite identificar las operaciones más comunes que se realizan dentro del sistema:

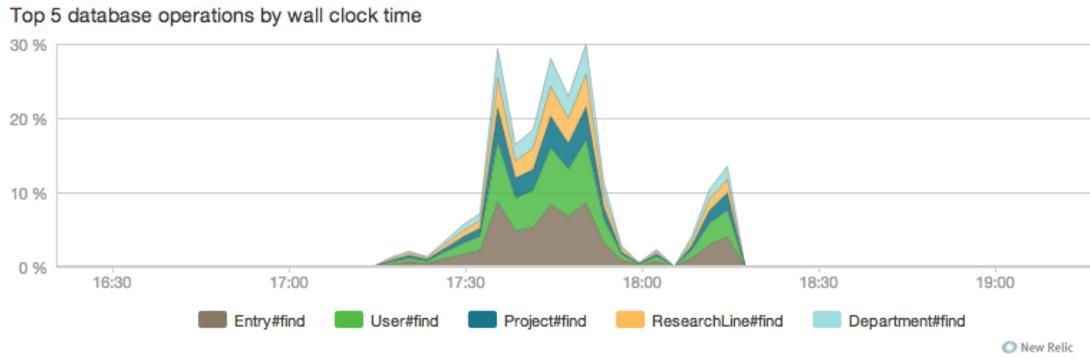


Figura 5.18. Evaluación: Operaciones más comunes

## **CAPÍTULO 6:**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **6.1. CONCLUSIONES**

1. Los objetivos general y específicos planteados en el plan de tesis lograron ser totalmente cumplidos, gracias al aporte de los Señores Directores del Proyecto, la colaboración de los Señores Docentes de la Escuela Politécnica del Ejército, y el compromiso puesto por parte del desarrollador del sistema en la aplicación de modernas tecnologías, técnicas y metodologías de desarrollo de software que requirieron en gran parte de un autoaprendizaje.
2. Por medio del sistema desarrollado como parte de este proyecto, la Escuela Politécnica del Ejército estará en la capacidad de socializar su quehacer investigativo y proyectar los resultados que provengan de estas actividades para enaltecer su nombre y reconocimiento a nivel nacional e internacional.
3. Los resultados obtenidos de las pruebas del sistema demuestran fehacientemente el cumplimiento de todas las especificaciones funcionales y de rendimiento establecidas para el Sistema. El Sistema de Difusión de Gestión del Conocimiento de la ESPE tiene las características que todo buen software debe poseer:

- Eficiencia: evidenciada por los resultados obtenidos en la medición del desempeño del software, que demuestran un buen uso de los recursos que se manipulan.
  - Flexibilidad: por su facilidad de configuración, parametrización, adaptación y escalamiento a los cambios que puedan darse en relación a la gestión del conocimiento dentro de la Escuela Politécnica del Ejército.
  - Portabilidad: ya que el aplicativo puede ser fácilmente migrado a diferentes plataformas de hardware o software.
  - Integridad: debido a que el sistema es capaz de proteger sus diferentes componentes contra los procesos o elementos que no tengan derecho de acceso a los mismos.
4. El uso de una metodología ágil como OpenUP favoreció el desarrollo del proyecto por las siguientes razones:
- Permitió que el Sistema de Difusión de Gestión del Conocimiento de la ESPE, sea fácilmente adaptable a los requerimientos cambiantes que se dieron durante el desarrollo del proyecto.
  - Proporcionó un marco de referencia para las etapas en que se dividió el proyecto.
  - Contribuyó como guía en la documentación que se generó del proyecto.
  - Permitió que el desarrollo del sistema se oriente únicamente en tareas y procesos que agregaron valor al producto final.
  - Permitió realizar una planificación real y facilitó la comunicación dentro del equipo de trabajo.

5. Tradicionalmente, iniciar una nueva aplicación web es un proceso bastante complejo, ya que se necesita estudiar y elegir los varios componentes de software para solucionar los problemas comunes de arquitectura, tales como: persistencia, logs, scripts de creación, configuración de aplicaciones, componentes de la capa web y flujos de trabajo. La selección de Ruby on Rails resultó un acierto, por cuanto proporcionó un enfoque coherente para la creación del sistema con una arquitectura lista para usarse, debido a que con Rails toda la base necesaria se encuentra disponible, permitiendo así empezar inmediatamente a indagar en la comprensión del problema de negocio y crear rápidamente un sistema funcional. Lo que habilita al equipo de desarrollo ser productivo en minutos, y no en semanas o meses.
6. Resultó importante la participación del personal docente e investigador de la Escuela Politécnica del Ejército, en la definición de los requerimientos funcionales del Sistema de Difusión de Gestión del Conocimiento de la ESPE, pues a través de las reuniones mantenidas, fue posible captar el sentido de esos requerimientos, los mismos que se resumen en: facilidad de acceso, sencilla actualización de los datos, libertad y responsabilidad sobre el contenido difundido, despliegue lógico de datos, y todo ajustado a la estructura institucional con relación a los procesos de investigación y vinculación con la colectividad.
7. La relación desarrollador, directores del proyecto, y los interesados en su uso, descrita anteriormente, fue encausada a través de la técnica de desarrollo guiado por comportamiento, o Behavior Driven Development (BDD), habiéndose comprobado que al ser una técnica de desarrollo ágil

de software fomentó la colaboración entre desarrolladores, testers, analistas y personas del negocio en el proyecto de software motivo del Proyecto de Graduación; lo que permitió al desarrollador enfocarse en las razones por las cuales el código debía ser creado, antes que en los detalles técnicos. Adicionalmente, minimizó la traducción entre el lenguaje técnico en el cuál se escribe el código y el lenguaje de dominio hablado por los involucrados en el proyecto.

8. Por otra parte, el uso del desarrollo guiado por comportamiento permitió obtener una rápida traducción de los requerimientos en pruebas, que facilitó la medición del avance del sistema y al mismo tiempo aseguró que se cumplan las expectativas de todos los interesados (stakeholders).
9. En el presente proyecto se han cumplido todos los aspectos que contempla como alcance la metodología OpenUP para proyectos de software, estos son: Inicio, Elaboración, Construcción y Transición; lo que evitó la existencia de errores en el sistema, requerimientos mal definidos, e incumplimiento de la visión concebida para el sistema
10. El sistema ha sido desarrollado considerando las tendencias actuales del desarrollo web, las mismas que se manifiestan en las tecnologías de back-end y front-end utilizadas, y que corresponden a aquellas empleadas en aplicaciones muy reconocidas tales como Twitter o Tumblr.
11. El uso de Javascript y AJAX jugó un papel importante en el desarrollo para que la experiencia de usuario del sistema resulte satisfactoria, considerando que éstos permiten agregar elementos de manera dinámica, mostrar mensajes de retroalimentación inmediatamente, y en

general, facilidad de uso para la difusión del contenido propio de cada usuario.

12. La administración del sistema ha sido pensada para ofrecer una interfaz intuitiva, de fácil y rápido acceso, que permita al administrador realizar todas las actividades necesarias, con el menor trabajo posible.
13. Los diferentes componentes que conforman la infraestructura de la aplicación han sido cuidadosamente escogidos aplicando criterios válidos de ingeniería, los mismos que exigen cubrir la demanda actual del sistema y poseer perspectivas de escalamiento que permita asegurar su funcionamiento continuo y mejorar el tiempo de respuesta del sistema.
14. Resultó importante la aplicación del sistema de control de versiones, git, pues gracias a él, quien ejecutó el proceso de desarrollo del sistema, adquirió la confianza suficiente para avanzar en la modificación e implantación del código fuente y la consecuente realización de los requerimientos con la certeza de tener la capacidad de volver a una versión anterior si algún inconveniente se hubiese presentado en el progreso hacia una nueva versión.
15. Para la evaluación práctica del sistema se contempló la utilización de varias herramientas de uso público, gracias a las cuales se pudieron tener valores reales sobre el alto desempeño de la aplicación desarrollada, se identificaron las solicitudes más comunes, se determinó el promedio de duración de las visitas al sistema y en general un mejor entendimiento de cómo se usa y cómo se podría orientar futuros desarrollos relacionados.
16. Se pudo verificar el impacto que tiene el uso de la arquitectura RESTful en la aplicación, tomando en cuenta las pruebas que se han realizado a la

misma y los tiempos de respuesta que son satisfactorios con respecto a los requisitos solicitados.

17. Las bondades de las tecnologías, técnicas y metodologías que éstas poseen y que han sido probadas a través del desarrollo de este sistema serían válidas para desarrollos de otras aplicaciones o sistemas complementarios que beneficien a la Escuela Politécnica del Ejército.
18. El sistema se encuentra listo para ser instalado en la plataforma que la Escuela Politécnica del Ejército asigne para el efecto, siguiendo paso a paso el sencillo procedimiento detallado en el Manual de Instalación, por parte de personal de sistemas con conocimientos básicos de administración de servidores Linux, servidores web y servidores de base de datos.

## **6.2. RECOMENDACIONES**

### **Administrador del Sistema de Difusión de Gestión del Conocimiento de la ESPE**

1. En la etapa de explotación del sistema desarrollado, se recomienda la aplicación de políticas para la creación de respaldos regulares a nivel de la base de datos, con el fin de prevenir la pérdida de información que pueda presentarse por algún fallo a nivel de hardware.

### **Coordinadores de Proyectos de Software Institucionales - ESPE**

1. Tomando en cuenta que el sistema de control de versiones utilizado para el desarrollo del presente proyecto favoreció la confianza en el proceso de desarrollo por parte del desarrollador, de igual manera el seguimiento y la trazabilidad de los requerimientos al código fuente que se generó para satisfacerlos, se recomienda el uso de este tipo de sistema en futuras ampliaciones y mejoras del sistema.
2. Para proyectos de desarrollo de aplicaciones más complejas que la presentada en este documento, los mismos que requieran la participación de un grupo de desarrollo integrado por un administrador general del proyecto, varios líderes por cada subsistema que conforme la aplicación y el conjunto de desarrolladores de cada componente, se recomienda considerar una herramienta de administración de proyectos, tal como Teamworkpm.net, para facilitar: la comunicación entre los distintos partícipes, el seguimiento de las distintas partes, y la integración final.

3. Para valorar futuros desarrollos se recomienda utilizar los resultados obtenidos mediante el uso de herramientas analíticas similares a las actuales Mixpanel y Google analytics.
4. Se recomienda emplear las metodologías y framework utilizados en este proyecto para crear nuevos módulos aplicables a la gestión del conocimiento en la ESPE, como por ejemplo:
  - Planificación de proyectos de investigación
  - Seguimiento de proyectos de investigación
  - Evaluación de proyectos de investigación.
5. Se recomienda la adopción de herramientas tales como New Relic en las sistemas web institucionales para la detección de: errores y sus orígenes, cuellos de botella, tiempos de respuesta excesivos, y más.

### **Unidad de Gestión de la Investigación - ESPE**

1. Se recomienda unificar los criterios de desarrollo en la Institución aplicando técnicas y metodologías actuales, sin cambiarlas hasta cuando aparezcan técnicas o metodologías nuevas, mejores y de alto impacto en el desarrollo de software.

## REFERENCIAS BIBLIOGRÁFICAS

Adzic, G. (2011). *Specification by Example: How Successful Teams Deliver the Right Software* (1era ed.). Manning Publications.

Alonso, J. L. (2009). *La gestión de contenidos digitales: Acceso a la información en el ámbito ciudadano*. Recuperado el 9 de Diciembre de 2012 from Pliegos de Yuste: <http://www.pliegosdeyuste.eu/n9pliegos/pdfs/115.pdf>

Arancibia, J. (14 de Junio de 2010). *Sistema de Administración de Propiedades*. Recuperado el 20 de Diciembre de 2012 from Universitat Oberta de Catalunya: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/2626/1/Proyecto%20Wakodi.pdf>

Bermeo, C., & Ortiz, M. (Junio de 2012). *Sistema Informático de Análisis y Proyección de Indicadores Cibernéticos bajo los parámetros de Webometrics*. Recuperado el 23 de Diciembre de 2012 from Universidad Politécnica Salesiana - Ecuador: <http://dspace.ups.edu.ec/bitstream/123456789/3077/1/UPS-CT002500.pdf>

Carneiro, C. J., & Al Barazi, R. (2010). *Beginning Rails 3* (1era ed.). USA: Apress.

Carrión, J. (Octubre de 2011). *Diferencia Entre Dato, Información Y Conocimiento*. Recuperado el 4 de Diciembre de 2012 from Universidad Veracruzana de México:

<http://sapp.uv.mx/univirtual/cursosDI/CIAsemestral/Modulo1/Doctos/DiferenciaDataInfoYConoc.pdf>

Cerezo, A., & Centeno, J. (27 de Junio de 2008). *SALAS: Un Sistema de Reserva de Aulas en Ruby on Rails*. Recuperado el 9 de Diciembre de 2012 from Universidad Rey Juan Carlos - España:

<http://upcommons.upc.edu/pfc/bitstream/2099.1/5542/2/Memoria.pdf>

Chelimsky, D., Astels, D., Helmkamp, B., North, D., Dennis, Z., & Hellesoy, A. (2012). *The RSpec Book: Behaviour Driven Development with Rspec, Cucumber, and Friends* (2da ed.). USA: Pragmatic Bookshelf.

Davenport, T., & Prusak, L. (2000). *Working Knowledge* (2da ed.). Boston, USA: Harvard Business Review Press.

de Carvalho, R. A., de Carvalho e Silva, F. L., & Soares Manhaes, R. (25 de Junio de 2010). *Mapping Business Process Modeling constructs to Behavior Driven Development Ubiquitous Language*. Recuperado el 9 de Diciembre de 2012 from Cornell University Library: <http://arxiv.org/pdf/1006.4892.pdf>

De Seta, L. (13 de Noviembre de 2008). *Introducción a los servicios RESTful*.

Recuperado el 16 de Diciembre de 2012 from Dos Ideas:

<http://www.dosideas.com/noticias/java/314-introduccion-a-los-servicios-web-restful.html>

Dev, V., Noria, X., Zimdars, J., Bigg, R., Cheung, F., & Darell, T. (1 de Diciembre de 2012). *Ruby on Rails Guides: Getting Started with Rails*. Recuperado el 15 de Diciembre de 2012 from Ruby on Rails:

[http://guides.rubyonrails.org/getting\\_started.html](http://guides.rubyonrails.org/getting_started.html)

Díaz, N., & Vaquero, D. (23 de Febrero de 2010). *Metodologías Ágiles*.

Recuperado el 15 de Diciembre de 2012 from Morfeo Formación:

<http://formacion.morfeo-project.org/wiki/index.php/PT3:GPSL:UF6>

Fazal, U., & Raham, S. (2011). *Quality Assurance Techniques in OpenUP: Agile processes such as Scrum, XP, OpenUP etc. have techniques that improve software quality*. USA: LAP LAMBERT Academic Publishing.

Fernández, L. A. (5 de Diciembre de 2012). *Diseño Web Adaptivo*. Recuperado el 22 de Diciembre de 2012 from GFI Blog: <http://blog.gfi.es/responsive-web-design/>

Fernandez, O. (2010). *The Rails 3 Way* (2da ed.). Boston, USA: Addison-Wesley Professional.

Fowler, C. (2012). *Rails Recipes: Rails 3 Edition* (2da ed.). USA: Pragmatic Bookshelf.

Jaimes, I. (15 de Junio de 2012). *CoffeeScript*. Recuperado el 9 de Diciembre de 2012 from Iver's web place: <http://es.iver.mx/coffeescript/#interview>

Jose, N., & Alfonso, E. (2012). *Planificación - Arquitectura de la Información en Entornos Web*. Recuperado el 5 de Diciembre de 2012 from Universidad

Tecnológica Nacional de Argentina:

[http://www.frm.utn.edu.ar/archivos/oferta%20educativa/ing\\_sistemas/programas/Planificacion\\_Arqu\\_de\\_informacion\\_2012.pdf](http://www.frm.utn.edu.ar/archivos/oferta%20educativa/ing_sistemas/programas/Planificacion_Arqu_de_informacion_2012.pdf)

Kroll, P., & MacIsaac, B. (2006). *Agility and Discipline Made Easy: Practices from OpenUP and RUP* (1era ed.). USA: Addison-Wesley Professional.

Labrada, E., & Salgado, C. (1 de Enero de 2013). *Diseño Web Adaptativo o responsivo*. Recuperado el 5 de Enero de 2013 from Revista Digital Universitaria - UNAM México: <http://www.revista.unam.mx/vol.14/num1/art07/index.html>

Langr, J., & Ottinger, T. (2011). *Agile in a Flash: Speed-Learning Agile Software Development*. Canadá: Pragmatic Bookshelf.

Laufer, M. (Noviembre de 2008). *La difusión del conocimiento*. Recuperado el 15 de Diciembre de 2012 from SciELO:  
[http://www.scielo.org.ve/scielo.php?pid=S0378-18442008001100002&script=sci\\_arttext&tlng=en](http://www.scielo.org.ve/scielo.php?pid=S0378-18442008001100002&script=sci_arttext&tlng=en)

Leffingwell, D. (2011). *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Boston, USA: Addison-Wesley Professional.

Malvicino, S., & Serra, R. (Febrero de 2006). *La importancia de la gestión del conocimiento y el desarrollo del capital humano en las organizaciones del siglo XXI*. Recuperado el 09 de Diciembre de 2012 from Gestipolis:  
<http://www.gestipolis.com/canales6/ger/modelo-intervencion-gestion-conocimiento.htm>

Mena, E., & Bernal, I. (Febrero de 2011). *Implementación de un prototipo de Cloud Computing de modelo privado para ofrecer Infraestructura como Servicio (IaaS)*. Recuperado el 15 de Diciembre de 2012 from Escuela Politécnica Nacional - Ecuador: <http://bibdigital.epn.edu.ec/bitstream/15000/2758/1/CD-3422.pdf>

Rodríguez, T. (16 de Junio de 2012). *Bootstrap*. Recuperado el 22 de Diciembre de 2012 from Genbetadev: <http://www.genbetadev.com/frameworks/bootstrap>

Ruby, S., Thomas, D., & Heinemeier Hansson, D. (2011). *Agile Web Development with Rails* (4ta ed.). USA: Pragmatic Bookshelf.

Ruiz de Arriaga, J. M. (13 de Junio de 2011). *La importancia de la gestión del conocimiento y su impacto en la competitividad de las empresas*. Recuperado el 10 de Diciembre de 2012 from Arriaga Asociados:

<http://www.arriagaasociados.com/2011/06/13/la-importancia-de-la-gestion-del-conocimiento-y-su-impacto-en-la-competitividad-de-las-empresas/>

Sainz, M. (27 de Diciembre de 2012). *Las métricas de siempre están muertas*.

Recuperado el 30 de Diciembre de 2012 from Biz-Tec: <http://biz-tec.mx/2012/12/las-metricas-de-siempre-estan-muertas/>

Sanchez, J., & Campos, J. (13 de Enero de 2013). *Planificación y Gestión Activa de Proyectos en GNU/Linux con Redmine*. Recuperado el 26 de Enero de 2013

from Universitat Oberta de Catalunya - España:

[http://openaccess.uoc.edu/webapps/o2/bitstream/10609/19274/1/jcamposl\\_TFC\\_0113.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/19274/1/jcamposl_TFC_0113.pdf)

Solís, C., & Wang, X. (12 de Septiembre de 2011). *A Study of the Characteristics of Behaviour Driven Development*. Recuperado el 22 de Diciembre de 2012 from

University of Limerick Institutional Repository:

[http://ulir.ul.ie/bitstream/handle/10344/1256/Solis\\_2011\\_behaviour.pdf](http://ulir.ul.ie/bitstream/handle/10344/1256/Solis_2011_behaviour.pdf)

Stephens, M., & Rosenberg, D. (2010). *Design Driven Testing: Test Smarter, Not Harder* (1era ed.). New York, USA: Apress.

Valim, J. (2011). *Crafting Rails Applications: Expert Practices for Everyday Rails Development* (1era ed.). Pragmatic Bookshelf.

Wiig, K. (1997). Integrating Intellectual Capital and Knowledge Management. *Long Range Planning* , 30 (3), 399-405.

Wynne, M., & Hellesoy, A. (2012). *The Cucumber Book: Behaviour-Driven Development for Testers and Developers* . USA: Pragmatic Bookshelf.