



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS E INFORMÁTICA**

**TEMA: DISEÑO E IMPLEMENTACIÓN DEL SERVICIO INTERACTIVO
DE COMUNICACIÓN ACADÉMICA PARA LA ESPE MEDIANTE
SERVICIOS DE SMS'S, EMAIL'S Y REDES SOCIALES BAJO
TECNOLOGÍA RIA.**

AUTOR: LEONARDO MOLINA NOBOA

DIRECTOR: ING. VILLACIS, CESAR

CODIRECTOR: ING. SALVADOR, SANTIAGO

SANGOLQUÍ, MARZO 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Sangolquí, 05 de Marzo del 2015

Quienes al pie de la presente firmamos, damos fe y testimonio que el proyecto de grado, previo a la obtención del título en Ingeniería en Sistemas e Informática, titulado como: **“DISEÑO E IMPLEMENTACIÓN DEL SERVICIO INTERACTIVO DE COMUNICACIÓN ACADÉMICA PARA LA ESPE MEDIANTE SERVICIOS DE SMS’S, EMAIL’S Y REDES SOCIALES BAJO TECNOLOGÍA RIA”**, fue desarrollado íntegramente por la señor **Leonardo Edward Molina Noboa**, bajo nuestra dirección y tutela.

Certificando lo antepuesto para su uso de la manera que se creyere conveniente, nos suscribimos,

Atentamente,



Ing. César Villacis

DIRECTOR



Ing. Santiago Salvador

CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

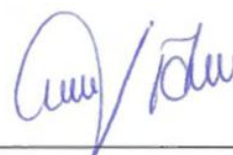
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

DECLARACIÓN

Yo, Leonardo Edward Molina Noboa, declaro que el presente trabajo es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación personal y que he consultado las referencias bibliográficas que se incluyen en el documento.

La Universidad de las Fuerzas Armadas ESPE puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual por su reglamento y por la normativa institucional vigente.

Sangolquí, 05 de Marzo de 2015



Leonardo E. Molina Noboa

C.C. 1712339017

AUTOR

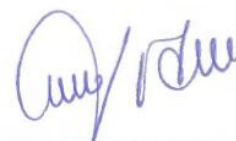
UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, Leonardo Edward Molina Noboa, autorizo a la Universidad de las Fuerzas Armadas ESPE a que publique en el repositorio digital de la biblioteca Alejandro Segovia el presente proyecto de tesis, así como también los materiales y documentos relacionados a la misma.

Sangolquí, 05 de Marzo de 2015



Leonardo E. Molina Noboa

C.C. 1712339017

AUTOR

DEDICATORIA

Quiero dedicar este trabajo de grado a Dios Todopoderoso y a la Santísima Virgen María, por haberme mantenido siempre en pie de lucha para lograr el objetivo.

A mí querido Padre, porque me enseñó a luchar por la vida y ahora que ocupa un lugar especial junto a Dios, no ha dejado de guiar mis pasos. Labor cumplida viejito lindo.

A mi Madre hermosa, mujer noble dueña de una fortaleza y fe inigualable, que siempre será mi principal fuente de lucha. Lo logramos Madre Querida.

A mi hijita Mía, quien me prestó el tiempo que le pertenecía para culminar mi tesis, te amo y te amaré por siempre. Acabe el deber mi amorcito.

A mis hermanos Jorge, Blanca, (+) Gabriel, Richard, John, Lourdes y Henry, porque siempre me enseñaron que todo lo que se quiere se puede. No llegue entre los primeros, pero acabé la carrera hermanos.

A mi esposa Karina, por su ayuda y sus ganas de que yo salga adelante, y que sea mejor cada día.

A mis sobrinos, cuñadas, maestros y amigos, por brindarme su respaldo y apoyo ante todo.

AGRADECIMIENTO

Primeramente doy infinitamente gracias a Dios y a la Virgen Santísima, por haberme dado la fuerza y valor para terminar mis estudios universitarios.

Agradezco también la confianza y el apoyo de mi madre y de toda mi familia porque han contribuido positivamente para llevar a cabo esta difícil jornada.

Le agradezco la confianza, apoyo y dedicación de tiempo a mis profesores: Ingenieros, Mauricio Campaña, César Villacis y Santiago Salvador, por haber compartido sus conocimientos y sobre todo su amistad.

TABLA DE CONTENIDOS

CERTIFICACIÓN.....	<i>i</i>
DECLARACIÓN.....	<i>ii</i>
AUTORIZACIÓN.....	<i>iii</i>
DEDICATORIA.....	<i>iv</i>
AGRADECIMIENTO.....	<i>v</i>
TABLA DE CONTENIDOS	<i>vi</i>
ÍNDICE DE FIGURAS.....	<i>ix</i>
ÍNDICE DE TABLAS.....	<i>xii</i>
RESUMEN	<i>xiii</i>
ABSTRACT.....	<i>xiv</i>
CAPÍTULO 1.....	<i>1</i>
1. INTRODUCCIÓN.....	<i>1</i>
1.1. Antecedentes.....	<i>2</i>
1.2. Situación Actual	<i>3</i>
1.3. Planteamiento del Problema	<i>4</i>
1.4. Justificación	<i>5</i>
1.5. Alcance de Prototipo	<i>6</i>
1.6. Limitaciones.....	<i>8</i>
1.7. OBJETIVOS	<i>9</i>
1.7.1 OBJETIVO GENERAL	<i>9</i>
1.7.2 OBJETIVOS ESPECÍFICOS	<i>9</i>
CAPITULO 2.....	<i>10</i>
2. MARCO TEORICO.....	<i>10</i>
2.1. Tecnología De Información Y Comunicación (TIC)	<i>10</i>
2.1.1 Tics para Mensajería Móvil Celular (SMS)	<i>12</i>
2.1.2 Tics para las Redes Sociales	<i>15</i>
2.1.3 TIC's para Correos Electrónicos.....	<i>20</i>
2.2. RIA (RICH INTERNET APPLICATION).....	<i>20</i>
2.2.1 Antecedente.....	<i>20</i>
2.2.2 Aplicaciones de Escritorio y Aplicaciones Web Convencionales	<i>21</i>
2.2.3 Aplicaciones y Tecnologías Enriquecidas para Internet RIA	<i>22</i>
2.2.4 Arquitectura de la Tecnología RIA.....	<i>24</i>
2.2.5 Características de RIA	<i>26</i>
2.2.6 Beneficios de RIA.....	<i>27</i>
2.3. APLICACIONES WEB Y CLIENTE-SERVIDOR.....	<i>27</i>
2.3.1 Antecedentes	<i>27</i>
2.3.2 Aplicaciones Web	<i>28</i>

2.3.3	Aplicaciones Web 2.0	29
2.3.4	Diferencias entre Aplicaciones WEB 2.0 y Aplicaciones WEB 1.0	30
2.3.5	Ventajas de la web 2.0.....	32
2.3.6	Arquitectura de la Tecnología Cliente-Servidor.....	33
2.3.7	Modelo Aplicaciones en 2 Capas.....	34
2.3.8	Modelo Aplicaciones de 3 Capas.....	35
2.3.9	Modelo Aplicaciones de N Capas	38
2.4.	PATRONES DE DISEÑO	39
2.4.1	Objetivos de los patrones	39
2.4.2	Categorías de patrones.....	40
2.4.3	Cómo seleccionar un patrón de diseño.....	40
2.4.4	Cómo usar un patrón de diseño.....	40
2.4.5	Patrón de la Arquitectura MVC.....	41
2.4.6	Patrón Active Record.....	42
2.5.	JPA (JAVA PERSISTENCE API).....	44
2.5.1	Consultas JPA - JPQL	47
2.5.2	Características JPA.....	48
2.5.3	Implementaciones Hibernate	49
2.6.	JAVA SERVER PAGES (JSF).....	52
2.6.1	Ciclo de Vida de JSF	54
2.6.2	Beans y páginas JSF	58
2.6.3	Beans	59
2.6.4	Eventos	60
2.7.	AJAX - TECNOLOGÍA RIA	61
2.7.1	Primefaces.....	69
2.8.	SMSLIB	70
2.8.1	Características	71
2.8.2	Varias puertas de enlace.....	71
2.8.3	Recibir síncrono / asíncrono	71
2.8.4	Síncrono / asíncrono Envío	72
2.9.	Maven	72
2.9.1	Ciclo de vida Maven.....	73
2.9.2	Artefactos y POM	74
2.9.3	Integración con IDE	75
2.9.4	Maven y Artefactos	75
2.9.5	Instrucciones básicas Maven	76
2.10.	METODOLOGÍA	77
2.10.1	RUP (Proceso Unificado de Rational).....	77
2.10.2	Las mejores prácticas	78
2.10.3	Ciclos y fases.....	78
2.10.4	Dimensiones del RUP	78
2.10.5	Fases del RUP.....	80
2.10.6	Disciplinas.....	83
2.10.7	Lenguaje de Modelamiento Unificado (UML)	84
2.10.8	Diagrama de Casos de Uso	84
CAPÍTULO 3.....	93	
3. DESARROLLO DE LA APLICACIÓN	93	
3.1. METODOLOGÍA DE DESARROLLO	93	

3.2.	FASE DE INICIACIÓN	94
3.2.1	Alcance Del Proyecto	94
3.2.2	Definición de Los Riesgos.....	98
3.2.3	Especificación de requerimientos de Software estándar IEEE 830.....	99
3.2.4	Funciones del Servicio de Comunicación.....	126
3.3.	FASE DE ELABORACIÓN	139
3.3.1	Desarrollo Del Modelado Del Negocio	139
3.3.2	Identificación del Modelo General.....	140
3.3.3	Especificación de los Requerimientos a Detalle	141
3.4.	DISEÑO DEL SISTEMA	145
3.4.1	DIAGRAMAS DE CASOS DE USO	145
3.4.2	DIAGRAMAS DE ACTIVIDADES	176
3.4.3	Diagrama de Modelo Lógico	185
3.4.4	Diagrama de Modelo Físico.	186
3.4.5	Arquitectura del Servicio de Comunicación.....	189
3.4.6	Modelo de Clases.	192
3.5.	FASE DE CONSTRUCCIÓN	194
3.5.1	ELABORACIÓN DEL SISTEMA.....	194
3.5.2	DIAGRAMAS DE ESTADOS.....	194
3.5.3	DIAGRAMA DE SECUENCIAS.....	196
3.5.4	ESTRUCTURA Y DESARROLLO DEL SISTEMA.	208
3.6.	FASE DE TRANSICIÓN.....	221
3.6.1	TESTS o PRUEBAS	222
3.6.2	Pruebas de Caja Blanca.....	230
3.6.3	Pruebas de Interfaces.....	236
3.6.4	Pruebas de Validación	236
CAPITULO 4.....	237	
4.	CONCLUSIONES Y RECOMENDACIONES.....	237
4.1.	CONCLUSIONES	237
4.2.	RECOMENDACIONES	239
4.3.	REFERENCIAS BIBLIOGRÁFICAS	240
ANEXOS	243	

ÍNDICE DE FIGURAS

Figura No. 2.1: TIC's - Procesos TIC's.....	10
Figura No. 2.2: TIC's SMS - Estructura de transferencias de mensajes cortos.....	14
Figura No. 2.3: TIC's SMS - Red estructural de SMS enviados por Módem.....	15
Figura No. 2.4: TIC's Twitter - Envío de mensaje Twitter.....	19
Figura No. 2.5: RIA - Aplicaciones de Internet Enriquecidas.....	23
Figura No. 2.6: RIA - Arquitectura (Campana, 2009).....	25
Figura No. 2.7: Aplicaciones Web - Diferencia entre Web 1.0 y Web 2.0.....	30
Figura No. 2.8: Aplicaciones Web - Modelo Cliente Servidor.....	34
Figura No. 2.9: Aplicaciones Web - Aplicaciones 2 Capa (Tolmo, 2012).....	35
Figura No. 2.10: Aplicaciones Web - Arquitectura 3 Capas (Tolmo, 2012).....	36
Figura No. 2.11: Aplicaciones Web - Arquitectura 3 Capas Vista Modelo (Tolmo, 2012).....	37
Figura No. 2.12: Aplicaciones Web - Arquitectura N Capas (Tolmo, 2012).....	38
Figura No. 2.13: Patrón Diseño - MVC.....	41
Figura No. 2.14: JPA - Creación Entidad.....	45
Figura No. 2.15: JPA - Creación Entity Factory.....	46
Figura No. 2.16: JPA - Transacción EntityManager.....	46
Figura No. 2.17: JPA - Transacción getResultList.....	47
Figura No. 2.18: JPA - Transacción getSingleResult.....	48
Figura No. 2.19: Hibernate - Arquitectura (Alvarez, 2012).....	49
Figura No. 2.20: Hibernate - Diagrama (Alvarez, 2012).....	50
Figura No. 2.21: JSF - Ciclo vida.....	55
Figura No. 2.22: JSF - Proceso del Ciclo vida.....	55
Figura No. 2.23: JSF - Creación de Bean.....	60
Figura No. 2.24: JSF - Ejemplo de Eventos.....	60
Figura No. 2.25: AJAX - Tecnologías bajo AJAX (LibrosWeb, 2013).....	63
Figura No. 2.26: AJAX - Modelo tradicional y modelo AJAX (LibrosWeb, 2013).....	63
Figura No. 2.27: AJAX - Comparación síncronas y asíncronas AJAX. (LibrosWeb, 2013).....	64
Figura No. 2.28: AJAX - Etiqueta.....	68
Figura No. 2.29: AJAX - Procesamiento Parcial.....	68
Figura No. 2.30: Primefaces - Página oficial del Showcase de Primeface.....	69
Figura No. 2.31: Maven - Configuración POM.....	74
Figura No. 2.32: Maven - Artefacto.....	75
Figura No. 2.33: Maven - Variable Artefacto.....	76
Figura No. 2.34: RUP (Wikipedia, Proceso Unificado de Rational, 2014).....	77
Figura No. 2.35: RUP - Fases (Wikipedia, Proceso Unificado de Rational, 2014).....	79
Figura No. 2.36: UML - Caso de Uso.....	85
Figura No. 2.37: UML - Actor.....	85
Figura No. 2.38: UML - Asociación en Casos de Uso.....	86
Figura No. 2.39: UML - Inclusión en Casos de Uso.....	87
Figura No. 2.40: UML - Extensión en Casos de Uso.....	88
Figura No. 2.41: UML - Límites en Casos de Uso.....	88
Figura No. 2.42: UML - Diagrama de Clases.....	89
Figura No. 2.43: UML - Diagrama de Secuencia.....	90
Figura No. 2.44: UML - Diagrama de Colaboración.....	91
Figura No. 2.45: UML - Diagrama de Estados.....	92
Figura No. 3.1: Modelo - Modelado Del Negocio.....	140
Figura No. 3.2: Modelo - General.....	140
Figura No. 3.3: Diagrama - Actores del Servicio de comunicación.....	145
Figura No. 3.4: Caso de Uso - Autentificación.....	146
Figura No. 3.5: Caso de Uso - Suscripción Académica.....	150
Figura No. 3.6: Caso de Uso - Cancelación Suscripción Académica.....	152
Figura No. 3.7: Caso de Uso - Notificación Entrante Gestión Académica.....	154

Figura No. 3.8: Caso de Uso - Notificación Saliente Gestión Académica.....	160
Figura No. 3.9: Caso de Uso - Anuncios Académicos.....	163
Figura No. 3.10: Caso de Uso - Parametrización.....	169
Figura No. 3.11: Caso de Uso - Administración Usuarios.....	170
Figura No. 3.12: Caso de Uso - Administración Periodos.....	172
Figura No. 3.13: Caso de Uso - Administración Contactos.....	173
Figura No. 3.14: Caso de Uso - Configuración Módem.....	174
Figura No. 3.15: Diagrama Actividades - Acceder TREEONE.....	176
Figura No. 3.16: Diagrama Actividades - Verificar Usuario.....	177
Figura No. 3.17: Diagrama Actividades - Suscribirse a TREEONE.....	178
Figura No. 3.18: Diagrama Actividades - Cancelar Suscripción.....	179
Figura No. 3.19: Diagrama Actividades - Obtener Petición.....	180
Figura No. 3.20: Diagrama Actividades - Enviar Mensaje SMS.....	181
Figura No. 3.21: Diagrama Actividades - Obtener Información de Notificación.....	182
Figura No. 3.22: Diagrama Actividades - Registro Anuncio Docentes.....	183
Figura No. 3.23: Diagrama Actividades - Registro Anuncio Personal.....	184
Figura No. 3.24: Diagrama - Modelo Lógico.....	185
Figura No. 3.25: Diagrama - Modelo Físico Parte 1.....	186
Figura No. 3.26: Diagrama - Modelo Físico Parte 2.....	187
Figura No. 3.27: Diagrama - Modelo Físico Parte 3.....	188
Figura No. 3.28: Arquitectura - MVC del Sistema.....	189
Figura No. 3.29: Arquitectura - Estructura General del Sistema.....	190
Figura No. 3.30: Diagrama - Modelo de Clases.....	193
Figura No. 3.31: Diagrama de Estados - Suscripción Académica.....	195
Figura No. 3.32: Diagrama de Estados - Cancelación Suscripción Académica.....	196
Figura No. 3.33: Diagrama de Secuencia - Autenticación.....	197
Figura No. 3.34: Diagrama de Secuencia - Autenticación.....	198
Figura No. 3.35: Diagrama de Secuencia - Suscripción Académica.....	199
Figura No. 3.36: Diagrama de Secuencia - Cancelación Suscripción Académica.....	200
Figura No. 3.37: Diagrama de Secuencia - Obtener Petición.....	201
Figura No. 3.38: Diagrama de Secuencia - Registrar Gestiones Académica.....	202
Figura No. 3.39: Diagrama de Secuencia - Registrar Anuncios Docentes.....	203
Figura No. 3.40: Diagrama de Secuencia - Registrar Anuncio Personal.....	204
Figura No. 3.41: Diagrama de Secuencia - Enviar mensajes SMS.....	205
Figura No. 3.42: Diagrama de Secuencia - Parametrización.....	206
Figura No. 3.43: Diagrama de Secuencia - Administración Usuarios.....	206
Figura No. 3.44: Diagrama de Secuencia - Administración Periodos.....	207
Figura No. 3.45: Diagrama de Secuencia - Administración Contactos.....	207
Figura No. 3.46: Estructura de directorios del proyecto.....	208
Figura No. 3.47: Ejemplo Entidad Pojo.....	211
Figura No. 3.48: Ejemplo Backing Bean.....	212
Figura No. 3.49: Ejemplo Controlador.....	213
Figura No. 3.50: Ejemplo Enumerador.....	215
Figura No. 3.51: Ejemplo Utilidad Mail.....	216
Figura No. 3.52: Ejemplo Utilidad SMS.....	218
Figura No. 3.53: Ejemplo Utilidad Twitter.....	219
Figura No. 3.54: Ejemplo POM.....	220
Figura No. 3.55: Estructura de directorios Configuraciones Adicionales.....	220
Figura No. 3.56: Conexión Módem.....	230
Figura No. 3.57: Pruebas - SMS Saliente.....	231
Figura No. 3.58: Pruebas - SMS Entrante y Saliente Notas.....	231
Figura No. 3.59: Pruebas - SMS Entrante y Saliente Impedimentos - Cuentas.....	232
Figura No. 3.60: Pruebas - Seguimiento Twitter.....	232
Figura No. 3.61: Pruebas - Anuncios Mensaje Twitter.....	233
Figura No. 3.62: Pruebas - Anuncios Mensaje Correo ESPE No1.....	233

Figura No. 3.63: Pruebas - Anuncios Mensaje Correo ESPE No2.....	233
Figura No. 3.64: Pruebas - Anuncios Mensaje Correo Personal No1.....	234
Figura No. 3.65: Pruebas - Anuncios Mensaje Correo Personal No2.....	234
Figura No. 3.66: Pruebas - Anuncios Mensaje Correo Facebook No1.....	235
Figura No. 3.67: Pruebas - Anuncios Mensaje Correo Facebook No2.....	235
Figura No. 3.68: Pruebas - Anuncios Mensaje Suscripción.....	235

ÍNDICE DE TABLAS

Tabla No. 2.1: Métodos Active Record.....	44
Tabla No. 3.1: Requerimientos - Acceso al Servicio de Comunicación.....	126
Tabla No. 3.2: Requerimientos - Suscripción.....	127
Tabla No. 3.3: Requerimientos - Cancelación de la Suscripción.....	127
Tabla No. 3.4: Requerimientos - Notificación Entrante Gestión Académica.....	128
Tabla No. 3.5: Requerimientos - Notificación Saliente Gestión Académica.....	128
Tabla No. 3.6: Requerimientos - Anuncios.....	129
Tabla No. 3.7: Requerimientos - Parametrización.....	129
Tabla No. 3.8: Requerimientos - Administración de Usuarios Suscriptos.....	129
Tabla No. 3.9: Requerimientos - Administración de Periodos.....	130
Tabla No. 3.10: Requerimientos - Administración de Contactos.....	130
Tabla No. 3.11: Requerimientos - Configuración y Activación del Módem.....	130
Tabla No. 3.12: Requerimientos - Otras Funcionalidades.....	131
Tabla No. 3.13: Especificaciones - Actores del Servicio de Comunicación.....	141
Tabla No. 3.14: Especificaciones - Características del Software.....	142
Tabla No. 3.15: Especificaciones - Casos de Uso del Servicio TREEONE.....	142
Tabla No. 3.16: Especificaciones - Clases del Servicio de Comunicación.....	144
Tabla No. 3.17: Escenarios - Acceder TREEONE.....	147
Tabla No. 3.18: Escenarios - Verificar Usuario.....	149
Tabla No. 3.19: Escenarios - Suscribir al Servicio TREEONE.....	150
Tabla No. 3.20: Escenarios - Cancelación Suscripción.....	153
Tabla No. 3.21: Escenarios - Consultar Petición.....	155
Tabla No. 3.22: Escenarios - Enviar Mensaje SMS.....	158
Tabla No. 3.23: Escenarios - Registrar Gestiones Académicas.....	161
Tabla No. 3.24: Escenarios - Obtener Información de Notificación.....	161
Tabla No. 3.25: Escenarios - Registrar Anuncios Docentes.....	164
Tabla No. 3.26: Escenarios - Registrar Anuncio Personal.....	167
Tabla No. 3.27: Escenarios - Configurar Parámetros.....	170
Tabla No. 3.28: Escenarios - Configurar Suscriptor Administrador.....	171
Tabla No. 3.29: Escenarios - Generar Periodo.....	173
Tabla No. 3.30: Escenarios - Editar Contactos.....	174
Tabla No. 3.31: Escenarios - Configurar Parámetros Módem.....	175
Tabla No. 3.32: Escenarios - Activar/Desactivar Módem.....	175
Tabla No. 3.33: Escenarios - Acceso al Servicio de Comunicación.....	223
Tabla No. 3.34: Escenarios - Acceso al Servicio de Comunicación 2.....	223
Tabla No. 3.35: Escenarios Suscripción - Usuario No Suscripto.....	224
Tabla No. 3.36: Escenarios Suscripción - Usuario Suscripto.....	224
Tabla No. 3.37: Escenarios Cancelar Suscripción.....	225
Tabla No. 3.38: Escenarios Suscriptores Administradores - Crear.....	225
Tabla No. 3.39: Escenarios Suscriptores Administradores - Eliminación.....	226
Tabla No. 3.40: Escenarios Gestión Parametrización.....	226
Tabla No. 3.41: Escenarios Gestión de Periodos.....	226
Tabla No. 3.42: Escenarios Gestión Módem.....	227
Tabla No. 3.43: Escenarios Cambio de Contraseñas.....	227
Tabla No. 3.44: Escenarios Notificación Entrante.....	228
Tabla No. 3.45: Escenarios Notificación Saliente.....	228
Tabla No. 3.46: Escenarios Anuncios Docente.....	229
Tabla No. 3.47: Escenarios Anuncios Personal.....	229
Tabla No. 3.48: Pruebas de Interfaces.....	236

RESUMEN

El avance tecnológico en el área de la comunicación ha logrado que se cuente con diferentes servicios de comunicación instantáneo, entre estos servicios se tiene a la mensajería de telefonía móvil, correos electrónicos y mensajería en las redes sociales. En base a estos servicios el presente proyecto tiene como objetivo principal el diseño e implementación de un Prototipo para el Servicio Interactivo de Comunicación Instantánea, permitiendo apoyar al Sistema Académico Banner de la Universidad de las Fuerzas Armadas ESPE en el envío de las gestiones académicas (Notas Académicas, Impedimentos, Valores pendientes de pago) a los estudiantes universitarios en el mismo momento que son registradas en el sistema académico y adicional podrán realizar consultas de las mismas. Se identificó como problema principal la falta de un medio de notificación instantáneo, que permita informar a los estudiantes universitarios sobre las diferentes gestiones académicas registradas en el Sistema Banner. Para el desarrollo del proyecto se utilizó la metodología RUP que se basa en las mejores prácticas para el análisis, implementación y documentación de Sistemas Web Orientados a Objetos, haciendo uso del modelo UML, además la arquitectura del proyecto se fundamenta en el patrón modelo vista controlador (MVC) bajo la tecnología RIA. En conclusión el personal universitario contará con un servicio alternativo de comunicación que le permitirá estar informado a tiempo de las diferentes gestiones académicas, permitiéndoles tomar acciones inmediatas en caso de requerirlas.

Palabras Clave:

SMS

SERVICIOS DE COMUNICACIÓN

GESTIÓN ACADÉMICA

MVC

NOTIFICACIÓN

RIA

ABSTRACT

The technological advance in the communication area has made have different services of instantaneous communication like mobile messaging, emails and social networks messaging. Based on these services, this project has the principal objective the design and Implementation of a prototype for Interactive Instantaneous Communication Service, allowing support the Academic System Banner University of ESPE Armed Forces in the shipping academic procedures (Academic Notes, Disabilities, securities pending payment) college students at the same time they are registered in the system and further may make queries of the same. Was identified as a major problem the lack of a means of instant notification, allowing inform college students about the different academic procedures recorded in the Banner system. To develop this project was used RUP methodology, based on best practices for analysis, implementation and documentation for Oriented Object of Web Systems, using the UML model, further the architecture of the project is based on the model view controller pattern (MVC) under the scheme of RIA technology. In conclusion the university staff will have as an alternative service communication it will allow you to be informed on time of the different academic procedures, enabling them to take immediate action in case of requiring them.

KeyWords:

SMS

COMMUNICATION SERVICES

ACADEMIC MANAGEMENT

MVC

NOTIFICATION

RIA

CAPÍTULO 1

1. INTRODUCCIÓN

Los crecientes avances tecnológicos y la importancia de estar informados no sólo a nivel Institucional sino también a nivel personal, nace la necesidad de desarrollar nuevos servicios de comunicación e información, para lo cual la Universidad de las Fuerzas Armadas ESPE como parte de su crecimiento tecnológico, impulsa la adopción de estos nuevos servicios para el beneficio de la comunidad universitaria.

En la actualidad la Universidad de las Fuerzas Armadas ESPE, mantiene al correo electrónico como el único medio referente de comunicación, permitiendo informar al personal universitario sobre sus diferentes acontecimientos sociales y académicos. Además otorga al estudiante universitario de un modelo de consultas electrónicas a través del Portal Académico "Mi ESPE".

Estos modelos de consultas que son muy importantes para los estudiantes, a veces demoran en los tiempos de entrega ya que los estudiantes para poder acceder al portal deben disponer de un computador o de cualquier dispositivo que tenga acceso a internet, además de esto se puede evidenciar que existen accesos innecesarios al portal debido a que la información consultada no se encuentra disponible, actualizada o registrada en el sistema, ocasionando recargas continuas de las páginas y generando mayor consumo de CPU, razones por la cuales se debe contar con un modelo alternativo de consultas instantáneas que permita agilizar el proceso de entrega de información académica.

Con este antecedente se propone como proyecto de tesis, la creación de un prototipo para el servicio interactivo de consultas y notificaciones instantáneas de las diferentes Gestiones Académicas que ofrece la ESPE.

Adicional al proceso de envíos de las gestiones académicas por medio del servicio de mensajería corta o SMS, se presenta el proceso para el registro de anuncios académicos, permitiendo al docente y al personal de la ESPE notificar noticias, comunicaciones y recordatorios de los diferentes eventos académicos.

Para el análisis del proyecto se empezó describiendo todos los requerimientos funcionales y no funcionales del Servicio de Comunicación, basándose en las normativas que indica el documento IEEE 830.

Estos requerimientos permitieron ayudar y guiar la planificación del desarrollo de software basado en las diferentes fases que maneja la metodología RUP.

A nivel de la arquitectura de software, se trabajó bajo el modelo de 3 capas, utilizando el patrón de desarrollo MVC (Modelo, Vistas y Controlador), permitiendo organizar y desarrollar aplicaciones orientadas a la WEB seguras y óptimas, además para soportar esta arquitectura se utilizó un servidor de aplicaciones JBOSS con base de datos POSTGRESQL, ya que el propósito del desarrollo es utilizar software libre.

1.1. Antecedentes

Con la tecnología que ofrece la telefonía celular, el internet y las redes sociales, proveen soluciones informáticas que permiten desarrollar servicios eficaces de comunicación en función de satisfacer las necesidades de la comunidad universitaria.

En el escenario actual, la creciente producción de información, la rapidez en el intercambio de información y la necesidad de contar con la información oportuna, la Universidad de las Fuerzas Armadas ESPE se enfoca por el desarrollo tecnológico de estos servicios interactivos de comunicaciones que permita apoyar al sistema académico de la ESPE en la gestión de entrega de información.

El sistema académico desde un inicio hace uso de los correos electrónicos y en la actualidad los servicios de aulas virtuales, permitiendo compartir su información con la Comunidad Universitaria. Estos servicios son modelos de comunicación muy importantes para la ESPE, pero con las nuevas tendencias tecnológicas en el mercado, nace la necesidad invertir en nuevos productos.

1.2. Situación Actual

La Universidad de las Fuerzas Armadas ESPE ha venido experimentando un crecimiento a nivel de sus sistemas de información, con el fin de otorgar a la comunidad universitaria servicios de accesibilidad inmediata referente a las gestiones académicas.

Estos servicios de accesibilidad que se ofrecen a través del Portal Mi ESPE, mantiene al correo electrónico y a las salas virtuales, como únicos medios referentes de comunicación e información, permitiendo notificar al personal universitario sobre los diferentes acontecimientos sociales y académicos ejecutados en la Universidad.

Actualmente, para poder acceder a la información de las gestiones académicas, el Sistema Banner otorga al estudiante de un modelo de consultas electrónicas. Este modelo genera dificultades ya que los estudiantes para poder acceder al Sistema Banner deben disponer de un computador o de cualquier dispositivo que tenga acceso a internet, contando además con un usuario de acceso al sistema, ya que por políticas de seguridad, este usuario debe estar en constante mantenimiento, motivo por el cual tiende a bloquearse, generando demoras en sus procesos de desbloqueo.

Además se puede evidenciar que existen accesos innecesarios al Sistema Banner, debido a que la información consultada de las gestiones académicas no se encuentran disponibles, actualizadas o registradas,

ocasionando recargas continuas de las páginas y generando mayor consumo de CPU en el servidor de aplicaciones.

Esta situación actual indica que el Sistema de Gestión Académica Banner no cuenta con un servicio alternativo de consultas y notificaciones inmediatas, en donde permita comunicar al estudiante sobre las gestiones académicas en el mismo momento que son registradas en el Sistema Banner y además de permitir la consulta de las mismas sin tener que acceder al Sistema.

Con lo anteriormente indicado, se exponen diversos estudios realizados que demuestran a Internet y a los teléfonos móviles como los medios de comunicación más utilizados en la actualidad, ya sea por su eficacia en recibir mensajes, como para enviar.

1.3. Planteamiento del Problema

Actualmente existe el mecanismo de consulta electrónica, que consiste en que los estudiantes deban acceder al Portal desde cualquier PC que tenga acceso al Internet, acceden con su ID banner al Sistema ingresando por diferentes filtros que se presentan, sin embargo las consultas realizadas no se encuentran registradas o actualizadas en las fecha definidas, generándole continuas consultas y recargas de la página cada qué vez en que los estudiantes acceden, de esta forma se produce un alto consumo del servidor en donde está alojada la aplicación.

Mientras que el personal docente y administrativo, para enviar o recibir un anuncio o comunicativo académico lo debe hacer directamente ingresando a su cuenta de correo electrónico de la ESPE y seleccionar los destinatarios, para lo cual lo deben hacer desde su propio PC que tenga acceso al Internet.

Con esto se puede identificar que el principal problema se refiere a la indisposición de los diferentes medios de comunicación, ya que con los

métodos anteriormente descritos, no le permiten al estudiante obtener la información de las gestiones académicas en el momento que las requiera o recibir en el mismo momento que son registradas en el Sistema Banner Mi ESPE.

Del análisis de las distintas situaciones expuestas en el contexto de esta investigación, se identifica como necesidad, mejorar los procesos de comunicación entre las diferentes gestiones académicas que ofrece la ESPE al estudiante, adicionando también la creación del servicio de registro y envío de anuncios académicos mediante el uso de nuevos servicios de comunicación.

1.4. Justificación

Los avances tecnológicos que existen a nivel de las comunicaciones, han permitido que tanto el hombre como las instituciones puedan desempeñarse de una manera más eficiente, logrando optimizar los tiempos y recursos de las actividades de trabajo y estudio que realiza.

En el caso de la telefonía móvil, esta se ha convertido en un objeto prácticamente imprescindible en nuestra vida cotidiana, personal y profesional, ya que actualmente es la tecnología de comunicación con el mayor crecimiento alcanzado en los últimos años, proporcionando de una capacidad comunicativa e interactiva independientemente de nuestra ubicación geográfica.

En cambios los correos electrónicos y los servicios que prestan las redes sociales, han tenido un crecimiento y un alto índice de usabilidad, ya que presentan diferentes valores agregados de comunicación como el de informar, publicar, mensajear y compartir información.

Para ello se propone modelos alternativos de consultas instantáneas, entre estos se tiene el uso de la mensajería móvil SMS, de los mensajes por redes Sociales y de los correos electrónicos, que permitirán resolver y

mejorar las distintas necesidades comunicativas que existe en la comunidad universitaria, permitiendo agilizar el proceso de entrega de las diferentes gestiones académica, entre ellas: Servicio de notificación y consulta de notas, impedimentos, valores pendientes de pago, servicio de anuncios Académicos.

Técnicamente el proyecto es viable ya que se cuenta con el apoyo de las Tics, quienes serán lo que proveerán del esquema de la base datos y registros del sistema de gestión académica.

1.5. Alcance de Prototipo

El alcance del proyecto tiene la concepción de potenciar con las mejores prácticas el manejo de las gestiones académicas que ofrece el sistema académico Mi ESPE, basado en la interacción entre el personal universitario y los sistemas de información.

Con lo acotado anteriormente, el alcance tiene los siguientes puntos:

- Desarrollo del Servicio de Comunicación a nivel de prototipo, para la recepción de las diferentes Gestiones Académicas, sean estas: Notas, Impedimentos y Saldos Pendiente, por medio del servicio de mensajería móvil - SMS, servicios de correos electrónicos y servicios de mensajes por redes sociales.
- Desarrollo del Servicio para las consultas de las diferentes Gestiones Académicas, sean estas: Notas, Impedimentos y Saldos Pendiente, por medio del servicio de mensajería móvil – SMS.
- Desarrollo del servicio de anuncios académicos por medio de un aplicativo web orientado solamente para el personal docentes administrativos de la Universidad. Estos anuncios serán enviados a través de los servicios de correos electrónicos y de los servicios de redes sociales, en este caso Twitter y correo Facebook.

- El Desarrollo del sitio web para la suscripción al Servicio de Comunicación instantánea, consultas de mensajes y anuncios recibidos, administración de los contactos e instructivo de uso.
- El Desarrollo del rol administrativo que permitirá gestionar la activación del módem, la gestión de usuarios administrativos del Servicio de Comunicación y de administración de los periodos académicos.
- Todos los desarrollos serán a nivel de prototipo, debido a la indisposición del acceso a los datos del Sistema Banner Mi ESPE, que por razones de las políticas de seguridad que mantienen las UTIC's, no permiten el acceso a terceros sobre la información académica en los ambientes de producción y pruebas.
- Debido a lo anteriormente expuesto, se debe crear una base de datos de pruebas en MySQL, que simule la obtención de los registros de usuarios y de las gestiones académicas del Sistema Banner Mi ESPE.
- Creación de los Servicios Web para consultar la información de los usuarios y de las gestiones académicas registradas en la base de datos de pruebas Banner.
- Generar los diagramas de casos de uso, diagrama de clases, diagramas de estados, diagramas de secuencia y los diagramas de actividades, que determinan el análisis, desarrollo y la implementación del prototipo.
- Elaborar el modelo entidad relación de la base de datos para el desarrollo del prototipo, que permita soportar toda la funcionalidad del proyecto a elaborar
- Determinar los requisitos que se necesitan para la implementación del prototipo.

Es importante recalcar, que las políticas de seguridad que mantiene la UTICs, son en beneficio del resguardo de la información académica, evitando los diferentes riesgos por daños de la información, filtros de información, por pérdida de información o por la generación de información inconsistente.

1.6. Limitaciones

- Solo se realizarán consultas de las gestiones académicas tales como: Notas, Impedimentos y valores pendientes de pago mediante mensajes de texto - SMS.
- No se realizarán consultas al sistema de gestión académica Banner, debido a las políticas seguridad que tiene el sistema.
- Se manejará una base de pruebas en MYSQL que simule el comportamiento de obtención de usuarios y de las gestiones académicas, con el fin de probar la funcionalidad de los servicios de comunicación.
- La aplicación podrá entregar la información por SMS siempre y cuando el módem tenga crédito o saldo.
- La aplicación podrá entregar la información por correos o por cuentas sociales siempre y cuando este registradas correctamente en el Servicio de Comunicación.

1.7. OBJETIVOS

1.7.1 OBJETIVO GENERAL

Diseñar e Implementar el prototipo para el Servicio Interactivo de Comunicación de las Gestiones Académicas de la ESPE, utilizando los SMSs, Correos y Redes Sociales bajo Aplicaciones de Internet Enriquecidas R.I.A.

1.7.2 OBJETIVOS ESPECÍFICOS

- Revisar y Describir la Tecnología de Información y de Comunicación con los servicios de SMS, redes sociales y correos electrónicos.
- Revisar y Describir la metodología de desarrollo de Software RUP, Modelamiento UML y la tecnología de aplicaciones RIA.
- Realizar el Análisis de Requerimientos del Sistema o Servicio de Comunicación.
- Desarrollar e Implementar a nivel de prototipo el Servicio de Comunicación orientado a la WEB 2.0 bajo la tecnología RIA
- Diseñar e Implementar la Interfaz para el Registro de Anuncios Académicos.
- Implementar bajo la arquitectura de desarrollo en 3 capas con el patrón de diseños MVC.

CAPITULO 2

2. MARCO TEORICO

2.1. Tecnología De Información Y Comunicación (TIC)

Las TIC's se basan en el conjunto de tecnologías aplicadas para proveer a las personas, de la información y comunicación (últimamente entretenimiento) a través de medios tecnológicos de última generación. (Graells, 2008).

En la última década los sistemas de comunicación han sufrido cambios radicales debido al desarrollo y a la difusión de nuevas tecnologías de información y de comunicación por medio de la telefonía celular e Internet.

En la *figura No. 2.1* se puede observar el proceso de construcción del conocimiento y Tics, en las cuales se detallan los siguientes procesos:



Figura No. 2.1. TIC's - Procesos TIC's.

INFORMACIÓN, datos que tienen significado para determinados colectivos. La información resulta fundamental para las personas, ya que a partir del proceso cognitivo de la información que obtenemos continuamente con

nuestros sentidos vamos tomando las decisiones que dan lugar a todas nuestras acciones.

TECNOLOGÍA, se refiere a la aplicación de los conocimientos científicos y técnicos, que facilita la realización de las actividades humanas. Supone la creación de productos, instrumentos, lenguajes y métodos al servicio de las personas.

COMUNICACIÓN, se refiere a la transmisión de mensajes entre personas. Como seres sociales las personas necesitan comunicarse para expresar sus pensamientos, sentimientos y deseos.

INTERACTIVIDAD, es el proceso de intercambio de información entre 2 agentes o más para conseguir un resultado previamente definido. Este se realiza a través de una interfaz y supone una interpretación de parte de los agentes

Cuando se unen la Información, la tecnología y la comunicación, se está haciendo referencia al conjunto de avances tecnológicos que proporciona la informática, las telecomunicaciones y las tecnologías, comprendiendo los desarrollos relacionados con los ordenadores, Internet, la telefonía.

Respecto a las funciones de comunicación de las TIC, se pueden resaltar las siguientes:

- Facilita la comunicación interpersonal y trabajo colaborativo entre los alumnos, tanto en el aula como fuera de ella.
- Facilita el acceso a la información, anuncios y noticias mediante la utilización de los medios de comunicación, como lo es el correo electrónico, los SMS y redes sociales.
- Tienen mayor influencia y beneficia en mayor proporción al área educativa ya que la hace más accesible y dinámica.
- Son de carácter innovador y creativo, pues dan acceso a nuevas formas de comunicación.

- Se relacionan con mayor frecuencia con el uso de la Internet y la informática.
- Resultan un gran alivio económico a largo plazo, aunque en el tiempo de adquisición resulte una fuerte inversión.

De la diversidad de los usos, se plantean los diferentes medios de comunicación para el proyecto de tesis planteado, como lo son los SMS, correos y redes sociales.

2.1.1 Tics para Mensajería Móvil Celular (SMS)

Uno de los canales de comunicación basado en las TIC que ha experimentado un mayor auge en los últimos años ha sido el servicio de mensajes cortos o SMS (Short Message Service), este servicio se encuentra disponible en los teléfonos móviles y permite el envío de mensajes cortos entre ellos.

"SMS - en inglés Short Messaging System - permite el envío de mensajes de texto a cualquier teléfono móvil / celular y debido a la popularidad y uso generalizado de los mismos por casi todo el mundo, los mensajes de texto son ideales para enviar cualquier comunicado o información de forma inmediata a un grupo". (EuropeMarketext)

Actualmente las Cooperativas y bancos son los que más utilizan los SMS para llegar a sus clientes, ya que no se usan tanto para los chats entre personas, sino para brindar servicio al cliente, como entrega de estados de cuenta o información de retiro de dinero de cajeros automáticos.

Ahora los mensajes tienen una limitación vinculada al tamaño máximo de caracteres, que no pueden superar los 150. (Aunque a ellos se solventa con el envío de más de un mensaje), y otro importante relacionada con el coste económico de envío, al depender de las tarifas que se finan con las operadoras móviles.

Importancia de los SMS

Los usuarios de teléfono móvil se han convertido en el mercado público objetivo, debido a que hoy en día existen más personas que poseen móvil, que usuarios de Internet, lectores de revistas y prensa o personas que poseen un TV.

Para lo cual se resaltan las siguientes afirmaciones sobre la importancia del uso de este medio (EuropeMarkettext):

- A nivel global, el número de teléfonos móviles supera en once veces los puntos de acceso a internet, esto significa que se puede llegar a muchas más personas vía SMS, que por correo electrónico o fax.
- La comunicación mediante SMS garantiza la confirmación de lectura y tiene mayor índice de lectura y respuesta que los otros medios.
- Los SMS también se consideran “móviles”, ya que se puede llegar a los destinatarios sin importar su ubicación. Además, a diferencia de otros medios de comunicación, un mensaje de texto es almacenado en el teléfono móvil / celular para recuperarlo fácilmente en el momento conveniente.
- Los SMS reducen tiempo y costos de comunicación, con la ventaja adicional de brindar un medio que permite realizar manual o automáticamente, envíos masivos a grandes grupos de contactos usando mínimos recursos para realizar la tarea.
- Los mensajes de texto también son excelentes para contactar personas sin interrumpirlas o perturbarlas. Usar SMS ayuda a reducir las llamadas telefónicas, interrumpir o interferir una persona en su jornada de trabajo o en cualquier otra ocasión en donde no necesite usar este medio de comunicación.

Arquitectura de los SMS

El servicio de mensajes cortos o SMS (**Short Message Service**) es un servicio disponible en todos los teléfonos móviles permitiendo el envío de mensajes cortos (también conocidos como mensajes de texto) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano.

El servicio SMS transfiere un mensaje de texto entre una estación móvil **MO-SM** (*Mobile Originated*, originado en el terminal del usuario) a un **MT-MS** (*Mobile Terminated-Short Message*, de llegada al terminal destino), estos mensajes originados llegan al centro de servicio **SMSC** centro de mensajes cortos (*Short Message Service Center*), que se encarga de almacenarlos hasta que son enviados.

El servicio SMS se divide en dos servicios Básicos como lo demuestra la figura No. 2.2.

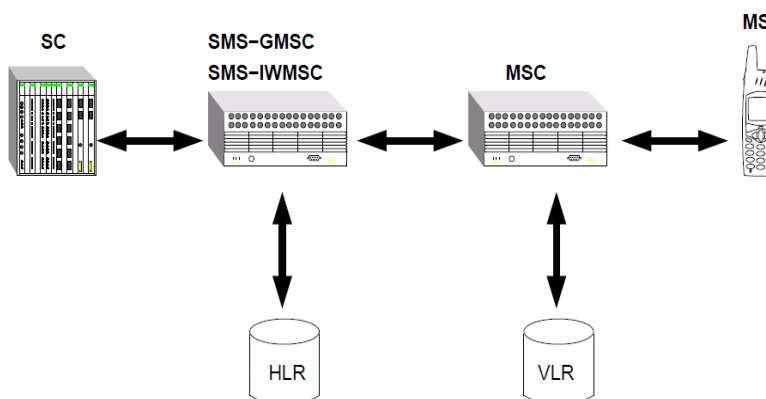


Figura No. 2.2: TIC's SMS - Estructura de transferencias de mensajes cortos.

Fuente: <http://bibing.us.es/proyectos/abreproy/11141/fichero/PFC%252F4+Red+GSM.pdf>

1. **SMMT** (Short Message Mobile Terminated Point-to-Point). Servicio de entrega de un mensaje desde el **SC** hasta una **SMSC**, obteniéndose un informe sobre lo ocurrido.

2. **SMMO** (Short Message Mobile Originated Point-to-Point). Servicio de envío de un mensaje desde una **SMSC** hasta un **SC**, obteniéndose un informe sobre lo ocurrido.

Módem GSM.

Un módem GSM puede ser un dispositivo externo o una tarjeta PC Card / PCMCIA Card. Por lo general, un módem GSM externo está conectado a un computador mediante una conexión serial o por USB, requiriendo de una SIMCARD para su funcionamiento.

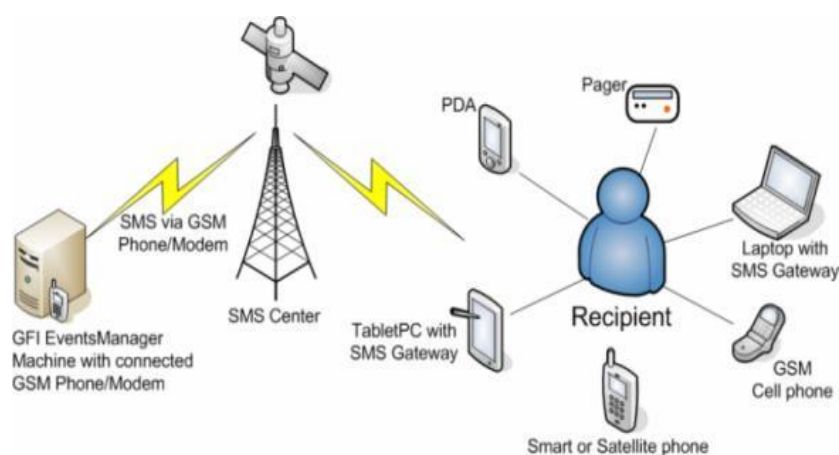


Figura No. 2.3: TIC's SMS - Red estructural de SMS enviados por Módem.

Fuente: <http://bibing.us.es/proyectos/abreproy/11141/fichero/PFC%252F4+Red+GSM.pdf>

En la *figura No. 2.3* se presenta la red estructural para el uso de los módems, teniendo en cuenta que disponen de un conector serial o USB para conectarse al PC. Son muy útiles para permitir que desde cualquier ordenador de una intranet o que tenga acceso a internet pueda enviar mensajes SMS.

2.1.2 Tics para las Redes Sociales

Se considera una red social como el conjunto de personas o entidades que se relacionan entre sí, así como las interacciones que se generan o establecen entre ellas. En Internet, estas interacciones pueden producirse de

maneras muy diversas; por ejemplo, mediante comentarios en blogs, intercambio de mensajes correo electrónico o participación en foros. (<http://www.discapnet.es/>, 2010)

Acotando se puede indicar que la propia Internet es una gran red social en la que todos los usuarios participan de una forma u otra.

En este sentido las redes sociales son un elemento más dentro de esta gran red, que no son ni buenas ni malas en sí mismas.

Estas redes sociales proporcionan las siguientes:

Ventajas

- Favorece la comunicación entre ellos y con su entorno. Además, al ser comunicación virtual hace que desaparezca la timidez.
- Proporciona un espacio de colaboración para compartir apuntes, hacer trabajos en grupo, resolver dudas, etcétera.
- Genera respuestas inmediatas a los usuarios, pero en este caso la información que se ofrece puede resultarle útil también a otras personas en la misma situación.
- Facilitan la creación de redes temáticas de usuarios que, a su vez, generan conocimientos fundamentales y complementarios que permiten que la organización conozca sus intereses y la información generada en estas redes y así poder tenerlos en cuenta.

Desventajas

- Dificultan la utilización del criterio propio de filtrado ya que uno de sus objetivos es promover las relaciones personales no directas, con lo cual se pierde el criterio de referencia y selección.
- Pueden facilitar el acceso a ciertos datos privados, como son el teléfono, correo electrónico, entre otros.

- En muchos casos no se garantiza la privacidad al no poder delimitar el acceso a los contactos.

En este sentido, se puede decir que la mayoría de las aplicaciones web 2.0, tienen un componente de red social, ya que agrupan usuarios con intereses comunes, por ejemplo: Facebook, Twitter, LinkedIn, Google Plus, entre otros

2.1.2.1 Twitter

Twitter es una plataforma de micro mensajería que permite enviar mensajes de texto cortos (hasta 150 caracteres) y conversar con los otros usuarios a través de navegadores web o a través de clientes de escritorio y de telefonía móvil (<http://www.cantabria.ccoo.es/>).

Las conversaciones se publican en Internet (aunque se pueden proteger las conversaciones de una cuenta para que sólo las puedan ver los seguidores de un usuario) y se construyen redes sociales a partir del seguimiento de los usuarios.

Entre las Tipología de cuentas, se tienen las siguientes:

- **Informativos (difusión):** emiten mensajes de manera unidireccional para informar a las personas que se hayan suscrito.
- **Relacionales (comunicación):** emiten, mantienen conversaciones y crean relaciones con personas de la Red.
- **Asistenciales (atención ciudadana):** su peso recae más en la atención de preguntas y dudas que puedan tener los seguidores.

La denominación de las cuentas o nombre de usuario está limitado a 15 caracteres, por ejemplo:

@leomolinan, @cnt @noticiaselcomer, @espe

Estas cuentas de Twitter se crean desde correos electrónicos personales o desde cualquier Instituciones y cada cuenta debe estar asociada a un correo diferente, ya que el sistema que utiliza Twitter no permite que una misma dirección de correo esté asociada a varias cuentas. (Commons, 2013).

Desarrollo Aplicaciones Web Twitter4J

Twitter4J es una biblioteca de Java no oficial de la API de Twitter.

Con Twitter4J se puede integrar fácilmente la aplicación Java con el servicio Twitter.

A través de las API de Twitter cualquiera puede crear aplicaciones que comuniquen con el servicio de la mencionada red social.

Generalidades del API de Twitter4J

El uso de las APIs de Twitter está limitado, por lo que las aplicaciones no pueden conectarse un número indeterminado de veces para solicitar cualquier cosa.

Sin embargo, los límites serían más o menos aceptables para páginas personales y proyectos pequeños.

En el caso que se desee construir sistemas que hagan un uso intensivo del API de Twitter, estaría la posibilidad de registrar la aplicación.

Los límites de acceso al API sin registro son 150 solicitudes por hora, mientras que para aplicaciones registradas en la "whitelist" podrían llegarse a hacer 20.000 solicitudes por hora.

Enviar y Recibir Mensajes con Twitter4J

Permite enviar y recibir mensajes directos vía `Twitter.sendMessage()` / `Twitter.getDirectMessages()`, se puede utilizar el siguiente ejemplo (Twitter Inc, 2014), como lo muestra la *figura No. 2.4*.

```
// The factory instance is re-useable and thread safe.
Twitter sender = TwitterFactory.getSingleton();
DirectMessage message = sender.sendMessage(recipientId, message);
System.out.println("Sent: " + message.getText() + " to @" +
message.getRecipientScreenName()); (twitter4j, 2014)
```

Figura No. 2.4: TIC's Twitter - Envío de mensaje Twitter.

2.1.2.2 Facebook

Esta red social se constituye como una plataforma que permite comunicar y compartir información, fotos, vídeos y enlaces con los diferentes usuarios que se conocen, además, los usuarios pueden participar en las comunidades que les interesen. (Cataluña, 2013)

Facebook ha mejorado y ampliado las posibilidades de la propia sección de mensajería hasta dejarla muy similar a lo que son los webmails. Desde ahí se podrá, con una cuenta de correo de dominio *Facebook.com* (*TuNombre@ facebook.com*) enviar y recibir mensajes sin la necesidad pasarte por tu Hotmail, Gmail, Yahoo o cualquiera sea el servicio que utilizas. Visualización

Correo electrónico en Facebook es otra de las grandes novedades del nuevo servicio de mensajería. Consiste en crear una dirección de correo electrónico *@facebook.com* para poder enviar, desde Facebook, mails a contactos que no están en esta red social.

Para la comunicación entre Facebook y las aplicaciones, la red libera APIs para los desarrolladores, con el fin de acceder a los contenidos y funcionalidades que ofrece la red social Facebook.

2.1.3 TIC's para Correos Electrónicos

El correo electrónico es el medio de comunicación más usado y uno de los primeros en internet, siendo muy útil para el envío y recepción correos electrónicos, de manera rápida, permitiendo adjuntar archivos digitales (imágenes, sonidos, vídeo, texto) para hacerlos llegar a una dirección o a varias a la vez, con muy pocos pasos.

Las cuentas de correo electrónico son recursos asociados a un nombre de dominio, el nombre que hay a continuación de la arroba (@) y muchas empresas proveedoras de servicios relacionados con Internet, entidades públicas, ofrecen la posibilidad de crear cuentas de correo electrónico gratuitas. Los servicios de correo electrónico más utilizados son: Gmail, Hotmail o Yahoo.

E-mail es un servicio de red que permite a los usuarios enviar y recibir mensajes rápidamente (también denominados mensajes electrónicos o cartas electrónicas) mediante sistemas de comunicación electrónicos. Principalmente se use este nombre para denominar al sistema que provee este servicio en Internet, mediante el protocolo SMTP.

2.2. RIA (RICH INTERNET APPLICATION)

2.2.1 Antecedente

La tendencia a desarrollar aplicaciones en Internet viene creciendo a un ritmo muy acelerado y la mayoría de las empresas están empezando a aprovechar los beneficios que ésta brinda. (DUHL, 2003)

Hasta ahora, las aplicaciones de escritorio permitían una mayor riqueza gráfica y mejor respuesta en la interacción con el usuario en comparación con las aplicaciones Web. Se pensaba que las aplicaciones Web nunca alcanzarían la capacidad de interacción de las aplicaciones de escritorio.

En efecto, ante cada acción del usuario que requería al servidor, se debía recargar la página Web utilizada o dirigirse a una nueva, perjudicando el tiempo de respuesta, esto conduce a usuarios insatisfechos lo cual redundaría en pérdida de ingresos para las empresas e instituciones.

El uso de Internet está cambiando junto con las expectativas de los usuarios. La expansión del ancho de banda, la demanda de los consumidores y la competencia de las empresas por llegar a nuevos mercados, crea la necesidad de impulsar nuevas tecnologías de desarrollo de aplicaciones que tengan el alcance de las aplicaciones Web, pero que tengan además el desempeño y la calidad de interacción de las aplicaciones de escritorio.

2.2.2 Aplicaciones de Escritorio y Aplicaciones Web Convencionales

Las aplicaciones de escritorio presentan algunas diferencias con respecto a las aplicaciones web convencionales, de las que se pueden indicar a continuación:

Las **Aplicaciones de Escritorio** ofrecen las siguientes **ventajas**:

- a) Mejora la experiencia del usuario en cuanto a audio, video y comunicaciones.
- b) Son más interactivas que las aplicaciones Web convencionales (ya que mantienen un contacto permanente entre los procesos internos del programa y lo que sucede en la interfaz de usuario).
- c) Permite acciones tales como arrastrar y pegar documentos, textos e imágenes.

Estas **Aplicaciones de Escritorio** tiene su **desventaja** que se basa en el siguiente punto:

- a) Los mayores problemas de las aplicaciones de escritorio son que dependen del sistema operativo del computador en el cual se va a instalar, y que la aplicación (junto con sus actualizaciones) debe ser instalada en cada computador utilizado.

En cambio las **Aplicaciones de Web Convencionales** ofrecen las siguientes **ventajas**:

- a) No es necesario realizar instalación, actualizaciones o parches en cada computador donde se ejecute la aplicación.
- b) Las aplicaciones pueden funcionar en diferentes sistemas operativos (independencia de la plataforma).
- c) Normalmente la interfaz de usuario es simple y estandarizada (baja curva de aprendizaje para los usuarios finales).
- d) Permiten a las empresas llegar a un mayor público a un menor costo.
- e) La comunicación con el servidor es asincrónica.
- f) La instalación se realiza en un único computador (servidor Web).

Estas **Aplicaciones de Web Convencionales** también tienen sus **desventajas** que se basan en los siguientes puntos:

- a) Respuesta lenta y pérdida del contexto durante el envío y recepción de los datos al servidor.
- b) No hay respuesta inmediata a los actos; se tiene que esperar que llegue la siguiente carga de pantalla.
- c) No permite acciones de interfaz gráfica como arrastrar y pegar documentos, textos e imágenes.

2.2.3 Aplicaciones y Tecnologías Enriquecidas para Internet RIA

Las Rich Internet Applications, o RIA (Aplicaciones de Internet Enriquecidas), son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales.

Las aplicaciones RIA son un nuevo tipo de aplicaciones 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, buscando mejorar la experiencia del usuario. (Wikipedia, Rich Internet application, 2014).

Con la definición anteriormente indicada, RIA en general permite una mayor riqueza gráfica y dinamismo en el transporte de la información entre el cliente Web y el servidor.

Normalmente en las aplicaciones web, hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace. De esta forma se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces a recargar la misma página con un cambio mínimo.

En los entornos RIA, en cambio, no se producen recargas de página, ya que desde el principio se carga toda la aplicación, y sólo se produce comunicación con el servidor cuando se necesitan datos externos como datos de una base de datos o de otros ficheros externos.



Figura No. 2.5: RIA - Aplicaciones de Internet Enriquecidas.

Fuente: <http://web-doszero.blogspot.com/2009/08/aplicaciones-de-internet-enriquecidas.html>

En la *figura No. 2.5* se ilustra el modelo que engloba la tecnología RIA, en donde se combina las aplicaciones web con las comunicaciones.

Las aplicaciones RIA trabajan de modo asincrónico, lo que permite que sean más dinámicas que las aplicaciones Web tradicionales. De esta forma, reducen la diferencia con las aplicaciones de escritorio.

Con la gran diferencia de que las RIA no necesitan instalar la aplicación en la máquina local del usuario.

Sin embargo, algunas de estas aplicaciones necesitan instalar pequeños programas o componentes normalmente llamados plug-in la primera vez que se ejecutan en el computador del usuario.

2.2.4 Arquitectura de la Tecnología RIA

Desde los inicios del Internet las aplicaciones Web tradicionales centran su actividad en una arquitectura Cliente - Servidor donde todo el procesamiento es realizado en el servidor y el cliente es utilizado para mostrar contenido estático.

La mayor desventaja con este sistema es que toda Interacción con la aplicación debe pasar por el servidor.

Haciendo uso de tecnologías que puedan ejecutar instrucciones en la máquina del cliente, las aplicaciones “ricas” (RIA) pueden eludir este bucle lento y sincrónico para muchas de las iteraciones del usuario.

Todas las aplicaciones RIA comparten una característica, introducen una capa intermedia, comúnmente llamada “motor cliente” (client engine), entre el cliente y el servidor, responsable de dibujar la interfaz de usuario y de la comunicación con el servidor, siempre de ser posible en forma asincrónica.

La *figura No. 2.6* muestra las capas que integran la arquitectura de una aplicación RIA, donde se puede ver el “contenedor de ejecución de interface rica” comúnmente llamado “motor cliente”. (Campana, 2009).

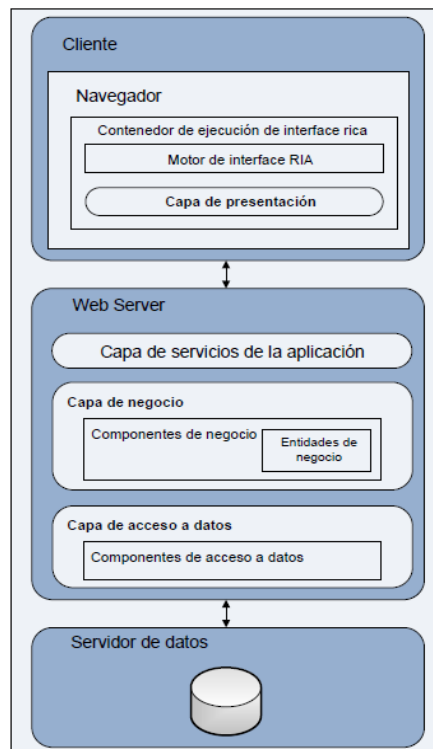


Figura No. 2.6: RIA - Arquitectura (Campana, 2009).

Generalmente se tiene una aplicación cliente y una capa de servicios separada. Las RIA se apoya más sobre un desarrollo “cliente-servidor” en vez de un desarrollo web tradicional, en donde el estado se mantiene en el servidor en sesiones. El cliente sabe acerca de sí mismo y el tipo de datos que está solicitando y únicamente solicita los datos que necesita sin ninguna otra información.

El Cliente, maneja la interacción entre el usuario y la interfaz de usuario, el usuario invoca comandos, actualiza vistas y carga datos. Aquí se mantiene el estado de la aplicación, se manejan todas las peticiones de datos hacia el servidor y se controla como se presentan los datos.

El Servidor, aquí se manejan y se procesan todas las peticiones de la aplicación cliente y delega las acciones en el servidor, estas pueden ser, guardar datos en la base de datos, actualizar los archivos del sistema, retornar datos al servidor, o algún tipo de proceso analítico. Determina y le da formato a los datos que son retornados al cliente.

2.2.5 Características de RIA

- **Accesibilidad:** AJAX es nativo en los navegadores web y es el único *RIA framework* que puede ser encontrado por los diferentes motores de búsqueda. Aunque Adobe Flash ha dado grandes pasos en esta dirección.
- **Comunicaciones avanzadas:** con servidores que soporten nuevas tecnologías se puede mejorar la experiencia del usuario al utilizar protocolos de red optimizados y entradas y salidas asíncronas. Se requiere de una conexión de banda ancha confiable.
- **Complejidad:** soluciones avanzadas puede ser más difíciles de diseñar, desarrollar, implementar y depurar que las aplicaciones web tradicionales.
- **Consistencia:** la interfaz de usuario y las experiencias pueden ser controlada por el sistema operativo, el monitoreo del rendimiento y diagnóstico de errores puede ser difícil.
- **Instalación y mantenimiento:** se requiere de la instalación de un plugin o una máquina virtual, que generalmente es más rápida que la instalación de una aplicación tradicional y esta no se puede automatizar. Las actualizaciones son automáticas.
- **Offline:** puede ser usada sin conexión a Internet, reteniendo el estado en la máquina cliente.
- **Seguridad:** se mejora la seguridad por medio de actualizaciones automáticas. Pero las extensiones por sí mismas son sujetas a vulnerabilidades.
- **Rendimiento:** puede ser mejorado dependiendo de la aplicación y de las características de la red. Aplicaciones que pueden procesar localmente en el cliente evitan viajes hacia el servidor lo que incrementa su rendimiento. Dándole más trabajo al cliente también se puede incrementar el rendimiento del servidor.
- **Riqueza:** añadiendo características que no son nativas en los navegadores web como captura de video. (Fundación Wikimedia, 2014)

2.2.6 Beneficios de RIA

Entre los beneficios principales de aplicaciones RIA, se tiene una mejora importante en la experiencia visual, que hacen del uso de la aplicación algo muy sencillo, ofrece mejoras en la conectividad y despliegue instantáneo de la aplicación, agilizando su acceso, garantizan la desvinculación de la capa de presentación es decir acceso a la aplicación desde cualquier computador en cualquier lugar del mundo.

A pesar de que el desarrollo de aplicaciones multimedia para navegadores web está mucho más limitado y es más difícil que otro tipo de aplicaciones de escritorio, los esfuerzos se justifican por varios motivos:

- No necesitan instalación (solo es necesario mantener actualizado el navegador web).
- Las actualizaciones hacia nuevas versiones son automáticas.
- Se pueden utilizar desde cualquier ordenador con una conexión a Internet sin depender del sistema operativo que este utilice.
- Generalmente es menos probable la infección por virus, que utilizando por ejemplo programas ejecutables.
- Más capacidad de respuesta, ya que el usuario interactúa directamente con el servidor, sin necesidad de recargar la página.
- Ofrecen aplicaciones interactivas que no se pueden obtener utilizando solo HTML, incluyendo arrastrar y pegar, cálculos en el lado del cliente sin la necesidad de enviar la información al servidor.
- Evita la problemática del uso de diferentes navegadores al abstraerse de ello a través de un framework.

2.3. APLICACIONES WEB Y CLIENTE-SERVIDOR

2.3.1 Antecedentes

Si bien desarrollar aplicaciones web, es junto al desarrollo de aplicaciones móviles, la tendencia más fuerte del mercado de desarrollo de software, también es cierto que otros modelos de desarrollo siguen

manteniendo vigencia y son parte fundamental de la evolución de los sistemas informáticos.

Fuente: <http://desarrollandowebapps.blogspot.com/2013/02/aplicaciones-web-y-cliente-servidor.html>

Veinte años atrás las aplicaciones más usadas eran monolíticas, un solo gran archivo ejecutable que hacia todo el procesamiento de la lógica de negocio y atendía al usuario proporcionando una interfaz muy controlada y mucho menos rica que las interfaces actuales. Muchas de esas aplicaciones eran mono-usuario (un único usuario al tiempo), y poco a poco fueron pasando a ser multi-usuario aunque manteniendo el concepto de un solo archivo ejecutable, normalmente ubicado en una carpeta compartida para que los usuarios de la red pudieran accederlo.

2.3.2 Aplicaciones Web

Una aplicación web es aquella herramienta de software que los usuarios utilizan para acceder a un servidor web a través de Internet o una Intranet mediante un navegador Web.

Las aplicaciones web tiene mucho auge gracias a lo fácil que resulta usar un navegador web como cliente ligero, esto además supone otras ventajas como la independencia del sistema operativo que utilice el usuario en su PC. Otra razón de su popularidad es la facilidad para actualizarlas y mantenerlas, ya que no requiere la distribución, instalación y actualización de la aplicación en miles de usuarios; basta actualizar el servidor para contar con nuevas versiones del sistema.

Para que las aplicaciones web funcionen de forma adecuada en cualquier navegador es necesario que se cumplan estándares, tanto por parte de la aplicación desarrollada como del navegador mismo.

Los diversos sistemas operativos existentes soportan navegadores que están contruidos para soportar estándares definidos por la industria (DOM,

HTML, XML, CSS), esto a su vez se conectan a servidores web que retornan información en un formato estándar que debe ser soportado por los navegadores.

La construcción de las aplicaciones deben tener en cuenta estos estándares para garantizar su portabilidad; si en algún momento se toma una decisión sobre una tecnología o herramienta no estandarizada, finalmente se tendrá limitantes y nuestra aplicación solo sería visible por ciertos navegadores, lo que no es recomendable y por tanto se debe evitar (salvo que sea nuestra última opción para cumplir con un requerimiento del cliente).

2.3.3 Aplicaciones Web 2.0

La Web ha ido evolucionando desde las clásicas páginas HTML estáticas de sólo lectura que constituían una simple vitrina de contenidos (Web 1.0) hacia un grupo de nuevas tecnologías y herramientas que la convirtieron en una plataforma abierta basada en la participación de los usuarios, adquiriendo así una nueva dimensión social y participativa denominada Web 2.0. (Tórrez, 2009)

Web 2.0 permite al usuario ser también creador, compartir contenidos e interactuar con otros usuarios, dándole esto un nuevo sentido a la Web, dejando de ser un elemento de consumo y transformándose en un elemento de producción y creación de los usuarios.

Actualmente las páginas web educativas 1.0 son páginas poco interactivas aunque sus contenidos sean muy útiles para los profesores y para los alumnos.

Las Web educativas 2.0, blogs, wikis, RSS, marcadores sociales, redes sociales tienen el potencial de complementar, mejorar y añadir nuevas dimensiones colaborativas al aula.

2.3.4 Diferencias entre Aplicaciones WEB 2.0 y Aplicaciones WEB 1.0

Entre las principales diferencias graficas que se ilustran en la *figura No. 2.7*, se tiene siguiente cuadro comparativo:

Web 1.0 vs. Web 2.0 = la metáfora	
WEB 1.0	WEB 2.0
1. Solo-lectura	1. Lectura-escritura-colaboración
2. Inteligencia individual (solo-para-expertos)	2. Inteligencia colectiva (sabiduría de las multitudes)
3. Control	3. Libertad, autorregulación
4. Actitud pasiva	4. "Conversación", implicación
5. Tecnolog. Propietarias	5. Estándares
6. Impersonal	6. Humana, personal, informal
7. Gestión de versiones	7. Beta Perpetuo
8. Te lo doy hecho	8. Hazlo tú mismo
9. Opacidad	9. Transparencia

Figura No. 2.7: Aplicaciones Web - Diferencia entre Web 1.0 y Web 2.0.

Fuente: <http://www.cea.es/Herramientas/post/Que-es-la-Web-20.aspx>

Ahora se detallan las principales diferencias:

✓ **Por el Modo de Visualización:**

Web 1.0, El modo de visualización era el NAVEGADOR.

Web 2.0, Los modos de visualización son el NAVEGADOR Y EL LECTOR RSS.

Hoy y en día se tiende más a navegar a través de un lector RSS que buscar la información nueva en sitios preferidos.

Esto permite acceder a mucha más información en mucho menos tiempo.

✓ **Por el Protagonismo:**

Web 1.0, los protagonistas eran los USUARIOS AVANZADOS.

Web 2.0, los protagonistas son los USUARIOS AFICIONADOS.

Al contrario de lo que ocurría con la web 1.0, hoy por hoy no es necesario tener conocimientos informáticos para participar activamente en el desarrollo de internet, con solamente conocer lo básico es suficiente para manejar cualquier aplicativo en la Web 2.0.

✓ **Por la Arquitectura:**

Web 1.0, la arquitectura era CLIENTE - SERVIDOR.

Web 2.0, la arquitectura son los SERVICIOS WEB.

En la Web 1.0 los usuarios hacían peticiones a través de un navegador cliente y el servidor les mandaba una respuesta, mientras que hoy por hoy es normal que el cliente interactúe con la información almacenada en el servidor. Se trata de un servicio web.

✓ **Por los Editores:**

Web 1.0, los editores eran los WEBMASTER.

Web 2.0, ya no es necesario ser un Web Máster para editar una WEB, sino que todos los usuarios de esa web pueden participar en la edición del mismo.

✓ **Por el Estado:**

Web 1.0, el estado era ESTÁTICO.

Web 2.0, el estado es DINÁMICO.

Anteriormente, las páginas web permanecían estáticas hasta que el Web Máster las editase, Actualmente las páginas se cambian continuamente gracias a la interacción de los usuarios.

✓ **Por la Mínima Unidad de Contenido:**

Web 1.0, la unidad mínima de contenido era la PÁGINA.

Web 2.0, la unidad mínima de contenido es el ARTÍCULO, POST O MENSAJE.

Actualmente, cada vez que un usuario publica una interacción en un sitio Web, se tiene una nueva Unidad de Contenidos. Anteriormente, al ser el Web Máster el único que podía publicar, la Unidad Mínima de Contenido era la página web.

✓ **Por el Modo:**

Web 1.0, el modo era la ESCRITURA.

Web 2.0, el modo es de ESCRITURA COMPARTIDA.

Ahora no solo el Web Máster escribe o edita una página Web, sino que los usuarios de esa página pueden colaborar con sus aportaciones en el desarrollo de sus contenidos.

✓ **Por el Precio:**

Web 1.0, software con licencias pagas.

Web 2.0, el usuario puede bajar programas en forma gratuita.

✓ **Por su función:**

Web 1.0, la única función es difundir información.

Web 2.0, permite producir, diseñar, construir y compartir información en diferentes soportes.

Actualmente una de las principales diferencias de web2.0 respecto a la web 1.0, es que los usuarios de la World Wide Web pasaron de ser simples lectores con limitados conocimientos de programación a convertirse en usuarios productores de contenido, gracias a la presencia de software abiertos en la web y a la simplicidad para darles uso. La premisa de la web 2.0 es que todo lo que existe en la web puede ser suyo: contenidos y herramienta

2.3.5 Ventajas de la web 2.0

Entre las principales ventajas, se tienen las siguientes:

- **Software legal**, no hay por qué preocuparse por licencia alguna.
- **Disponibles desde cualquier lugar**.
- **Multiplataforma**, funciona independientemente del sistema operativo que se use e incluso se puede acceder desde cualquier dispositivo.
- **Siempre actualizado**, el servicio se encargará de las actualizaciones del software.
- **Menor requerimiento de hardware**, solo se necesita poder utilizar un navegador.
- **Colaboración**, pueden trabajar varias personas a la vez y desde diferentes lugares del mundo.

2.3.6 Arquitectura de la Tecnología Cliente-Servidor

Las diferentes arquitecturas que aparecieron para dar solución a una cantidad cada vez más creciente de necesidades de los usuarios que veían como los computadores aportaban a su negocio. Sin duda uno de los modelos arquitecturales más usados es Cliente-Servidor, en este modelo de aplicaciones distribuidas reparten las tareas en dos roles: un proveedor que proporciona recursos o servicios y al cual se le llama servidor, y consumidor que contacta al servidor con el objetivo de hacer uso de los recursos que este provee, se llama cliente.

Las máquinas **Clientes** son, en general, PC monousuario o puestos de trabajo que ofrecen una interfaz muy amigable para el usuario final. Cada **Servidor** ofrece una serie de servicios de usuario compartidos a los clientes.

El **Servidor** permite a los **Clientes** compartir el acceso a la misma base de datos y permite el uso de un sistema de computación de alto rendimiento para gestionar la base de datos.

En la *figura No. 2.8*, se representada la típica configuración cliente/servidor. Concretamente aparece un servidor de datos, en la parte superior y varios clientes en la inferior.

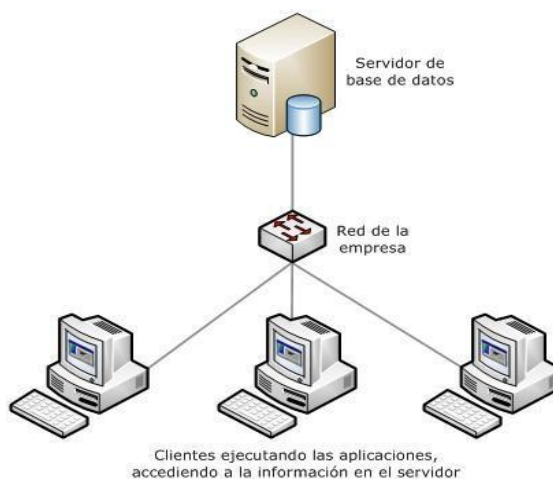


Figura No. 2.8: Aplicaciones Web - Modelo Cliente Servidor.

Fuente: <https://todoaccessvba.wordpress.com/2012/08/29/arquitectura-clienteservidor/>

Con el uso de esta arquitectura para el desarrollo de aplicaciones web, se pasa a tener una sola gran aplicación monolítica a un modelo distribuido en el que el servidor (normalmente de mayor capacidad de cómputo) procesa toda la lógica de negocio, y el cliente provee la interfaz de usuario y realiza validaciones y normalmente procesos menores de la lógica de negocio.

La entrada de ciertos motores de base de datos, en remplazo de las débiles estructuras de almacenamiento de información en archivos planos o tablas sueltas en forma de archivos, contribuyeron a la arquitectura cliente servidor.

Se puede indicar que las bases de datos además de proveer un sistema robusto para almacenar grandes volúmenes de información, proveen también de una estructura sólida de desarrollo basada en SQL como estándar.

2.3.7 Modelo Aplicaciones en 2 Capas

La arquitectura de dos capas en la actualidad es muy utilizada, aunque con muchas fallas, todavía no se ha podido dejar de usar. Estas

arquitecturas fueron las primeras en aprovecharse de la estructura cliente-servidor. El modelo de 2 capas se ilustra en la *figura No. 2.9*.

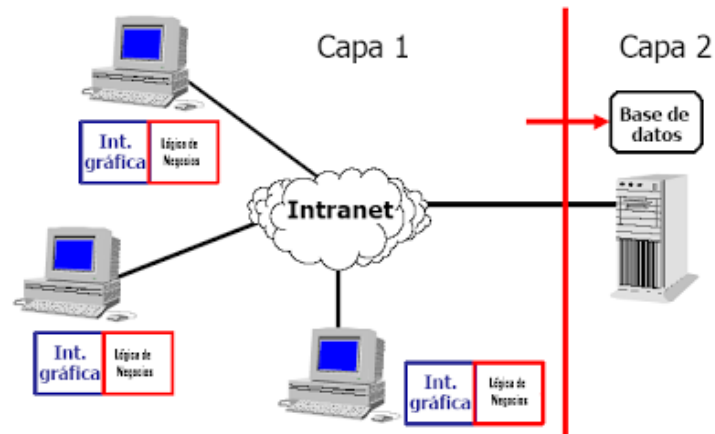


Figura No. 2.9: Aplicaciones Web - Aplicaciones 2 Capa (Tolmo, 2012).

Las capas que esta arquitectura presenta son las siguientes:

Nivel de aplicación: Este nivel es en el que se encuentra toda la interfaz del sistema y es la que el usuario puede disponer para realizar su actividad con el sistema (lógica de negocios).

Nivel de la base de datos. Este nivel de la Base de Datos también llamado el Repositorio de Datos, es la capa en donde se almacena toda la información ingresada en el sistema y que se deposita en forma permanente.

Existen herramientas para el desarrollo en dos capas por ejemplo Visual Basic, Access y SQL.

2.3.8 Modelo Aplicaciones de 3 Capas

La arquitectura de dos capas si bien ayudó en unos años atrás, se vio la necesidad de crear una nueva arquitectura ya que en dos capas se tenían problemas en la capa de aplicación ya que la principal desventaja de esta era el peso que tenía para el cliente.

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración entre sistemas complejos como

pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad e integración.

Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y por último una interfaz gráfica que facilite al usuario el uso del sistema (presentación). (Tolmo, 2012)

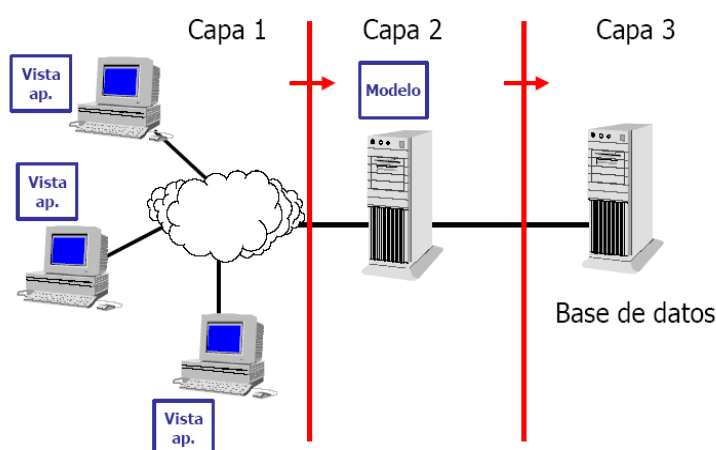


Figura No. 2.10: Aplicaciones Web - Arquitectura 3 Capas (Tolmo, 2012).

Si se establece una separación entre la capa de interfaz gráfica (cliente), replicada en cada uno de los entornos de usuario, y la capa modelo, que quedaría centralizada en un servidor de aplicaciones, según el diagrama que se puede ver en la *figura No. 2.10* se obtiene una potente arquitectura que otorga algunas ventajas:

- Centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo.
- No replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.
- Mayor sencillez de los clientes.

Crear un modelo de cuatro capas, separando cliente, servidor web, modelo y almacén de datos. Esto permite una mayor extensibilidad en caso de que existan también clientes no web en el sistema, que trabajarían directamente contra el servidor del modelo, esto se puede observar en la *figura No. 2.11*.

Sin embargo, la gran mayoría de las aplicaciones web comunes utilizan una arquitectura basada en la de tres capas extendida a las particularidades de la web.

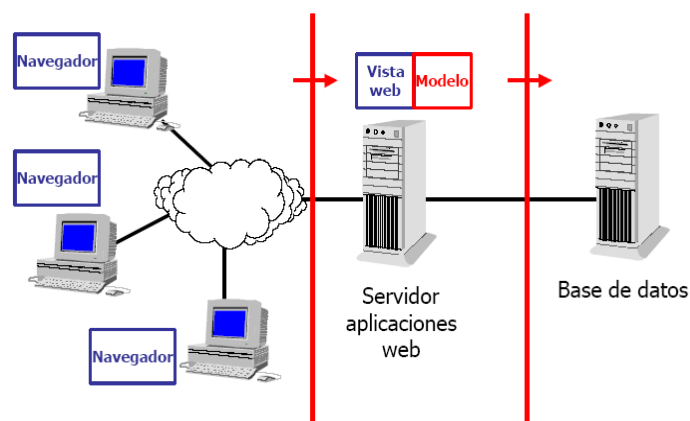


Figura No. 2.11: Aplicaciones Web - Arquitectura 3 Capas Vista Modelo (Tolmo, 2012).

Y es así que se creó la arquitectura de tres capas las cuales son:

- Nivel de Aplicación: La diferencia de este nivel aplicado ahora en una arquitectura de tres capas es que solo tiene que trabajar con la semántica propia de aplicación, sin tener que preocuparse de cómo está implementado este ni de su estructura física.
- Nivel de Dominio de la aplicación: En cambio este nivel se encarga de toda la estructura física y el dominio de aplicación.
- Nivel de Repositorio: Sigue siendo la capa en donde se almacenan los datos y toda la información.

Las herramientas para el desarrollo de tres capas son:

- Visual Basic en lo que se refiere a la capa de Aplicación
- SQL Server en lo que se refiere al repositorio de datos.
- MTS en lo que se refiere al nivel del dominio de Aplicación

2.3.9 Modelo Aplicaciones de N Capas

En una aplicación distribuida en n-capas los diferentes procesos están distribuidos en diferentes capas no sólo lógicas, sino también físicas. Los procesos se ejecutan en diferentes equipos, que pueden incluso residir en plataformas o sistemas operativos completamente distintos. Cada equipo posee una configuración distinta y está optimizado para realizar el papel que le ha sido asignado dentro de la estructura de la aplicación, de modo que tanto los recursos como la eficiencia global del sistema se optimicen. En la *figura No. 2.12* se ilustra un modelo de N capas.

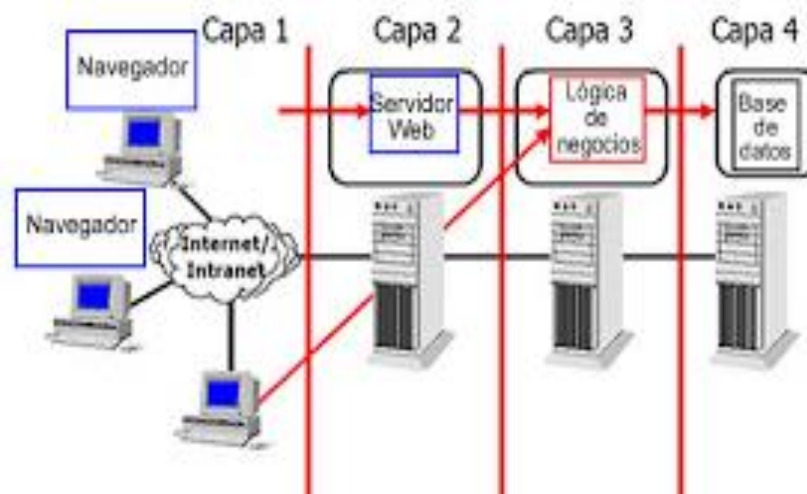


Figura No. 2.12: Aplicaciones Web - Arquitectura N Capas (Tolmo, 2012).

El surgimiento de la tecnología de componentes distribuidos es la clave de las arquitecturas de n-capas. Estos sistemas de computación utilizan un número variable de componentes individuales que se comunican entre ellos utilizando estándares predefinidos y frameworks de comunicación como:

- CORBA: (Common Object Request Broker Architecture) de Object Management Group (OMG).

- DNA : (Distributed Network Applications) de Microsoft (incluye COM/DCOM y COM+ además de MTS, MSMQ, etc.).
- EJB: (Enterprise Java Beans) de Sun Microsystems.
- XML : (extensible Markup Language) del World Wide Web Consortium (W3C)

2.4. PATRONES DE DISEÑO

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser **reutilizable**, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

2.4.1 Objetivos de los patrones

Los patrones de diseño pretenden:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Asimismo, no pretenden:

- Imponer ciertas alternativas de diseño frente a otras.
- Eliminar la creatividad inherente al proceso de diseño.

2.4.2 Categorías de patrones

Según la escala o nivel de abstracción:

- **Patrones de arquitectura**

- Aquellos que expresan un esquema organizativo estructural fundamental para sistemas de software.
- Proporcionan un conjunto de subsistemas predefinidos, con reglas y guías para organizar las relaciones entre ellos

- **Patrones de diseño**

- Aquellos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software.
- Describen el esquema básico para estructurar subsistemas y componentes.

- **Patrones Elementales**

- Patrones de bajo nivel específicos para un lenguaje de programación o entorno concreto.
- Describen cómo implementar componentes particulares de un patrón.

2.4.3 Cómo seleccionar un patrón de diseño

- Que solucionen problemas de diseño.
- Buscar las intenciones de cada patrón.
- Estudiar cómo se interrelacionan los patrones.
- Estudiar patrones de propósito similar.
- Examinar la causa de un rediseño.
- Considerar qué debería ser variable en un diseño.

2.4.4 Cómo usar un patrón de diseño

- Leer el patrón una vez para tener una visión general.
- Volver y estudiar la estructura, los participantes y las colaboraciones.

- Ver un ejemplo concreto codificado del patrón.
- Elegir nombres para los participantes del patrón que sean significativos en el contexto de la aplicación.
- Definir las clases.
- Definir nombres específicos de la aplicación para las operaciones en el patrón.
- Implementar las operaciones que realizarán las responsabilidades y colaboraciones del patrón.

2.4.5 Patrón de la Arquitectura MVC

La arquitectura *Model-View-Controller* surgió como patrón arquitectónico para el desarrollo de interfaces gráficas de usuario. Su concepto se basaba en separar el modelo de datos de la aplicación de su representación de cara al usuario y de la interacción de éste con la aplicación, mediante la división de la aplicación en tres partes fundamentales que se presenta en la *figura No. 2.13*:

- El modelo, que contiene la lógica de negocio de la aplicación.
- La vista, que muestra al usuario la información que éste necesita.
- El controlador, que recibe e interpreta la interacción del usuario, actuando sobre modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización.

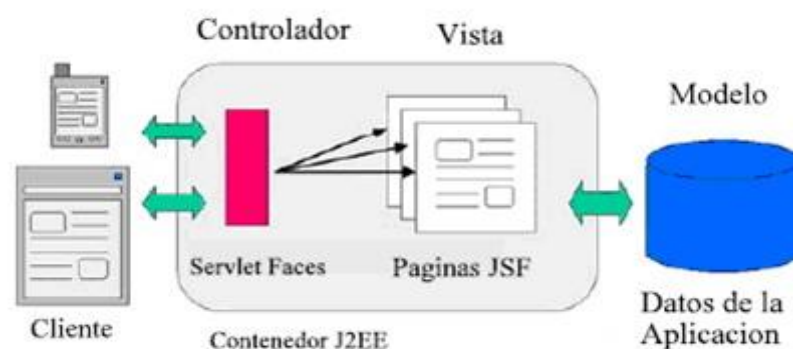


Figura No. 2.13: Patrón Diseño - MVC.

Fuente: https://www.assembla.com/wiki/show/andalucia/Entregable_E2/10

Esta arquitectura demuestra ser muy apropiada para las aplicaciones web y especialmente adaptarse bien a las tecnologías proporcionadas por la plataforma JEE, de manera que:

1. El modelo, que contiene la lógica de negocio, sería modelado por un conjunto de clases Java, existiendo dos claras alternativas de implementación, utilizando objetos java tradicionales llamados POJOs (*Plain Old Java Objects*) o bien utilizando EJB (*Enterprise JavaBeans*) en sistemas con unas mayores necesidades de concurrencia o distribución.
2. La vista proporcionará una serie de páginas web dinámicamente al cliente, siendo para él simples páginas HTML. Existen múltiples marcos de trabajo o frameworks que generan estas páginas web a partir de distintos formatos, siendo el más extendido el de páginas JSP (JavaServer Pages) o JSF (Java Server Faces), que mediante un conjunto de tags XML proporcionan un interfaz sencillo y adecuado a clases Java y objetos proporcionados por el servidor de aplicaciones.

Esto permite que sean sencillas de desarrollar por personas con conocimientos de HTML. Entre estos tags tienen mención especial la librería estándar JSTL (JavaServer Pages Standard Tag Library) que proporciona una gran funcionalidad y versatilidad.

3. El controlador en la plataforma JEE se desarrolla mediante servlets, que hacen de intermediarios entre la vista y el modelo, más versátiles que los JSP para esta función al estar escritos como clases Java normales, evitando mezclar código visual (HTML, XML...) con código Java. Para facilitar la implementación de estos servlets también existe una serie de marcos de trabajo que proporcionan soporte a los desarrolladores.

2.4.6 Patrón Active Record

Active Record se enfoque al problema de poder acceder a los datos de una base de datos relacionales en forma orientada a objetos. Una fila en la tabla de la base de datos (o vista) se envuelve en una clase, de manera que

se asocian filas únicas de la base de datos con objetos del lenguaje de programación usado. (CRUZ, 2008)

Cuando se crea uno de estos objetos, se añade una fila en la tabla de la base de datos. Cuando se modifican los atributos del objeto, se actualiza la fila de la base de datos, también permite consultar registros existentes, actualizarlos y eliminarlos si es necesario.

Una clase Active Record consiste en el conjunto de propiedades que representa las columnas de la tabla más los típicos métodos de acceso como las operaciones CRUD, búsqueda (find), validaciones, y métodos de negocio.

En que situaciones CONVIENE usar este patrón:

- Lógica de negocio simple y poco relacionado con otras entidades.
- Es ideal cuando la estructura de la tabla coincide con la estructura de la clase.

En que situaciones NO CONVIENE de Active Record:

- Es simple, esto es bueno y malo al mismo tiempo. Con lógica de negocio compleja, este patrón pierde coherencia.
- Otra desventaja que al estar tan acoplado a la estructura de la clase, un cambio en el diseño de la tabla, implica cambiar la clase.
- En situaciones de operaciones de alto volumen de datos, el overhead que se paga en el pasaje y carga de datos, es innecesario.

Esta desventaja aplica tanto a Active Record, como a cualquier otro diseño orientado a objetos.

En la *tabla No. 2.1*, se puede observar el ejemplo de los métodos que pertenecen una clase típica de Active Record:

Tabla No. 2.1: Métodos Active Record

Inserta un registro en la tabla	<code>public void Create()</code>
Elimina el registro de la tabla	<code>public void Delete()</code>
Modifica el registro en la tabla	<code>public void Update()</code>
Retorna el total de registros de la tabla	<code>public static int GetCount()</code>
Busca en la tabla el objeto usando como criterio su id.	<code>public static Order FindById(int id)</code>
Carga todos los registros de la tabla.	<code>public static List<Order> LoadAll()</code>

2.5. JPA (JAVA PERSISTENCE API)

Es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE) (http://es.wikipedia.org/wiki/Java_Persistence_API).

JPA viene a solucionar el vacío que hay entre utilizar objetos y persistirlos en una base de datos relacional.

La persistencia en este contexto cubre las siguientes áreas:

- La API en sí misma, definida en el paquete `javax.persistence`.
- El lenguaje de consulta Java Persistence Query Language (JPQL).
- Metadatos objeto/relacional.

El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos (siguiendo el patrón de mapeo objeto-relacional), y permitir usar objetos regulares (conocidos como POJOs).

JPA mapea automáticamente nuestras clases en la base de datos de manera transparente, lo cual entre otras cosas permite poder migrar de motor cuando se quiera, y poder compartir código o trabajar en equipo sin ningún problema.

JPA trabaja fuertemente con anotaciones y para mapear un bean (una clase java) con una tabla de la base de datos, se tendría que escribir lo que un Entity.

Esto es tan sencillo como escribir un Bean, con sus atributos y métodos get y set, añadiendo la anotación “@Entity” y seleccionando a uno de sus atributos como clave primaria con “@Id”. Por ejemplo, el siguiente código podría ser un Entity, que permitirá almacenar, recuperar, o actualizar registros sobre la tabla “usuario”, en la *figura No. 2.14* se ilustra la creación de una entidad:

```
@Entity
public class Usuario
{
    @Id
    private String id;
    private String name;
    private String email;
}
```

Figura No. 2.14: JPA - Creación Entidad.

Con solo esto se tiene creada una “entidad” llamada “Usuario” y donde se podría insertar, actualizar o eliminar entradas en una tabla llamada “Usuario” aunque esta aún no existe, siquiera.

Un fichero muy importante que se tiene que crear a parte de las clases “Entity” es el fichero “**persistence.xml**”, en este fichero se indica precisamente que clases son Entity, sobre qué base de datos y cuál es la política de creación de esta base de datos.

Este fichero tiene los datos de la conexión hacia la base de datos, la cual ya debe estar creada anteriormente.

El fichero XML se estructura de la siguiente forma:

- Tiene un nombre “EjemploTESIS” en el tag, este es importante, pues luego por ese nombre se puede acceder a este recurso o unidad de persistencia.

- Luego se añaden todas las clases Entity que se quiere manipular. Esto también es muy importante, porque si se crea un Entity pero no se añade a esta sección del XML, para JPA no existe.
- Y por último están los properties, que permiten definir los mecanismos de conexión hacia cualquier base de datos.

Con los Entity's ya creados, se presenta un ejemplo de inserción de un registro sobre la base de datos, los pasos a seguir son los siguientes:

- Primero se crea un EntityManager que permita manipular y trabajar con los objeto Entity, esto se ilustra en la *figura No. 2.15*.

```
EntityManagerFactory factory =  
Persistence.createEntityManagerFactory("EjemploTESIS",System.getProperties());  
em = factory.createEntityManager();
```

Figura No. 2.15: JPA - Creación Entity Factory.

El nombre "EjemploTESIS", se lo tiene definido en el fichero "persistence.xml", así sabe cómo recuperar las clases y propiedades que corresponden a esta unidad de persistencia.

Una vez creada la EntityManager, se puede abrir una transacción e incluso almacenar elementos (Entity) en la base de datos. Un ejemplo para insertar un "Usuario" podría ser como se ilustra en la *figura No. 2.16*.

```
em.getTransaction().begin();  
Usuario u = new Usuario();  
u.setId(100);  
u.setName("jose");  
u.setEmail("notengo@gmail.com");  
em.persist(u);  
em.flush();  
em.getTransaction().commit();
```

Figura No. 2.16: JPA - Transacción EntityManager.

Como se puede ver, se ha recuperado una transacción (getTransaction), se ha instanciado un objeto usuario, se han asignado los valores a los

atributos, se ha “persistido” (almacenado) en la base de datos con “persist” y por último un commit de la transacción.

No hay que escribir métodos, insert, update ni delete, ni crear el modelo de la tabla, todo se hace dinámicamente.

2.5.1 Consultas JPA - JPQL

Para hacer las consultas en JPA se emplea un lenguaje denominado JPQL, no es SQL exactamente porque trabaja con “objetos” no con “columnas” pero si muy parecido.

Por ejemplo, si la consulta de todos los campos de una tabla es “SELECT * FROM USUARIOS”, en JPQL la consulta sería “SELECT u FROM Usuario u”, Donde “Usuario” no es el nombre de una tabla, sino de la clase “Entity” y “u” son los identificadores del objeto.

Más ejemplos: “SELECT u.id, u.name FROM Usuario u where u.name LIKE:= PARAMETRO”.

Para ejecutar estas consultas se emplea el objeto “Query”. Query admite dos métodos:

getResultList, cuando el conjunto de valores devuelto es una lista de valores, por ejemplo un SELECT de varios campos como lo presenta la *figura No. 2.17*.

```
Query q = em.createQuery("SELECT c FROM Cliente c");
List<Cliente> clientes= q.getResultList();
int num_clientes= clientes.size();
```

Figura No. 2.17: JPA - Transacción getResultList.

GetSingleResult, cuando solo se devuelve un único objeto (fila).

Si se quisiera ejecutar las consultas con parámetros, las variables a sustituir van precedidas de “:” y se deben pasar con `setParameter`. Ejemplo: una consulta de Clientes cuyo id es “666” sería como lo presenta la *figura No. 2.18*.

```
Query q = em.createQuery("SELECT c FROM Cliente c WHERE p.id : = CLIENTE_ID");
q.setParameter("CLIENTE_ID", "666");
Cliente miClienteBuscado = (Cliente) q.getSingleResult();
```

Figura No. 2.18: JPA - Transacción `getSingleResult`.

2.5.2 Características JPA

- **Persistence:** La clase `javax.persistence.Persistence` contiene métodos estáticos de ayuda para obtener una instancia de `EntityManagerFactory` de una forma independiente al vendedor de la implementación de JPA.
- **EntityManagerFactory:** La clase `javax.persistence.EntityManagerFactory` ayuda a crear objetos de `EntityManager` utilizando el patrón de diseño del Factory (fábrica).
- **EntityManager:** La clase `javax.persistence.EntityManager` es la interfaz principal de JPA utilizada para la persistencia de las aplicaciones. Cada `EntityManager` puede realizar operaciones CRUD (Create, Read, Update, Delete) sobre un conjunto de objetos persistentes.
- **Entity:** La clase `javax.persistence.Entity` es una anotación Java que se coloca a nivel de clases Java serializables y que cada objeto de una de estas clases anotadas representa un registro de una base de datos.
- **EntityTransaction:** Cada instancia de `EntityManager` tiene una relación de uno a uno con una instancia de `javax.persistence.EntityTransaction`, permite operaciones sobre datos persistentes de manera que agrupados

formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos.

- **Query:** La interface *javax.persistence.Query* está implementada por cada vendedor de JPA para encontrar objetos persistentes manejando cierto criterio de búsqueda. JPA estandariza el soporte para consultas utilizando Java Persistence Query Language (JPQL) y Structured Query Language (SQL). Se puede obtener una instancia de Query desde una instancia de un EntityManager.

2.5.3 Implementaciones Hibernate

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Ver la *figura No. 2.19*.

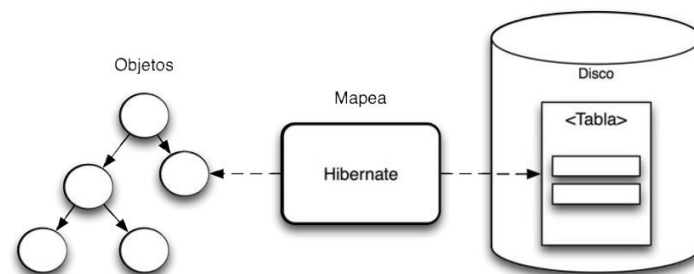


Figura No. 2.19: Hibernate - Arquitectura (Alvarez, 2012).

Todos los frameworks de persistencia trabajan de forma similar, pero cada uno tiene sus propios ficheros de configuración. En el caso del framework Hibernate existe una serie de ficheros y conceptos claves a la hora de comenzar a trabajar con él. Estos se enumeran a continuación basándose en la clase Libro de ejemplo:

Hibernate.cfg.xml: Es el fichero principal de configuración del framework, es donde se configura el driver JDBC de acceso a datos la IP del servidor de base de datos, el usuario y la contraseña ,así como los ficheros de mapeo que van a utilizar las distintas clases.

Libro.xml: Es el fichero de mapeo que almacena la información relevante referida a cómo un objeto que pertenece a una clase determinada, en este caso un Libro, es mapeado a una fila de la tabla Libros.

Libro.java: Clase java que es mapeada.

En la *figura No. 2.20* se muestra un diagrama de alto nivel sobre cuáles son los ficheros y pasos generales que realiza el framework Hibernate a la hora de trabajar con los objetos java que se encarga de persistir.

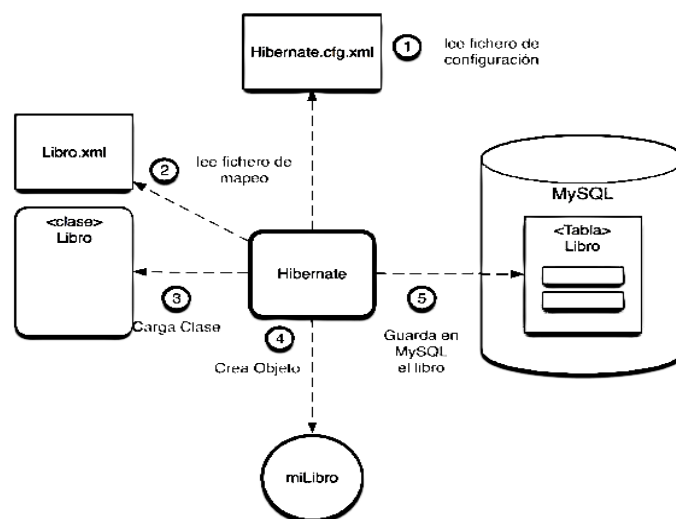


Figura No. 2.20: Hibernate - Diagrama (Alvarez, 2012).

A continuación se explica cada uno de los pasos del diagrama en ejemplo:

- 1) **Leer Hibernate.cfg.xml:** En la primera tarea, el framework busca y lee el fichero de configuración donde se encuentra la información de usuario, clave, driver y url de conexión.

De esta forma se conecta a la base de datos. Junto a esta información se encuentra también La lista de ficheros de mapeo que leerá posteriormente.

- 2) **Leer Fichero de mapeo:** Se encarga de leer todos los ficheros de mapeo de la aplicación para saber cómo mapear un objeto determinado contra alguna de las tablas existentes en la base de datos.
- 3) **Carga la clase:** El programa que se está construyendo carga las clases en memoria.
- 4) **Crear Objeto:** El programa que se está construyendo crea varios objetos de la clase.
- 5) **Salvar:** Hibernate se encarga de guardar los datos en la base de datos (Alvarez, 2012).

2.5.3.1 Características realizadas con Hibernate

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen.

Con esta información Hibernate le permite a la aplicación manipular los datos en la base de datos operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

2.6. JAVA SERVER PAGES (JSF)

La tecnología Java Server Faces es un framework de interfaz de componentes de usuarios del lado del servidor para las aplicaciones web basadas en la tecnología Java.

Esta tecnología simplifica el desarrollo de interfaces de usuario para crear aplicaciones java J2EE basadas en el patrón MVC Modelo Vista Controlador. (Romero)

Su implementación incluye:

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

La especificación de JSF fue desarrollada por la *Java Community Process*.

Este framework hace parte de la vista de la aplicación, donde se tienen las páginas web (.XHTML) que es lo que ve el cliente y por detrás de cada una de estas páginas se tienen a los Beans Administrados (ManagedBeans). Cada uno de estos managedBeans contiene diferentes propiedades de acuerdo a la necesidad del desarrollo.

Por ejemplo, un managedBean llamado FacturaBean.java, donde este tendrá todas las propiedades que tiene una factura.

@ManagedBean:

Esta anotación indica que una clase FacturaBean es de tipo Bean Administrado, por otro lado todos los beans administrados deben de llevar un nombre, en el caso del anterior bean se llama facturaBean, este no tiene que coincidir con el nombre de la clase estrictamente hablando.

@RequestScoped:

El Scope de un bean administrado define cual va a ser el ciclo de vida del elemento, si el scope es de tipo request, este vivirá únicamente con hacer una petición y regresar los datos a la vista, terminadas estas tareas el bean y todas las propiedades que tenía mueren con la instancia del bean.

@SessionScoped:

Esta anotación hace posible que el bean administrado y sus propiedades vivan durante toda la sesión de un usuario.

@ApplicationScoped:

Esta anotación hace posible que el bean administrado y sus propiedades duren hasta que se termine la aplicación web.

En JSF, se destacan los siguientes conceptos:

Managed Bean (o Backing Bean). Es un pojo que se asocia con los componentes de la interfaz de usuario (UIComponent), dentro de sus funciones está:

- ✓ Almacenar referencias a los componentes de la vista.

- ✓ Proveer de las propiedades necesarias para la vista, como el comportamiento de la pantalla, la información que se presenta e inclusive hasta en elementos de diseño.
- ✓ Exponer los métodos que pueden ser usados por los componentes para comunicar al servidor de la iteración con el usuario.

Convertidores, son parte de la vista y es un proceso que transforma una cadena del lado de la vista a un tipo de dato Java en el lado del servidor y así mismo de regreso.

Validaciones, son usadas para validar los datos de los componentes de acuerdo al uso de la vista o las reglas de negocio.

Method expressions, son usados para comunicar eventos de la vista al modelo de presentación como `ActionEvent`, `ValueChangeEvent`, `PhaseEvent`. Soportan invocaciones arbitrarias a métodos públicos del Managed bean.

2.6.1 Ciclo de Vida de JSF

Una página JavaServer Faces está representada por un árbol de componentes de interfaz de usuario, llamada vista (Cédric Simon).

Durante el ciclo de vida, la implementación Java Server Faces debe construir la vista al examinar el estado guardado de una presentación anterior de la página. Cuando el cliente envía una página, la implementación JavaServer Faces realiza varias tareas, tales como la validación de la entrada de datos y la conversión de los datos de entrada a los tipos especificados en el lado del servidor.

Para el desarrollo en JSF (de aplicaciones o de componentes) es fundamental comprender y dominar las seis fases del ciclo de vida de la petición dentro del controlador JSF, como lo demuestra la *figura No. 2.21*.

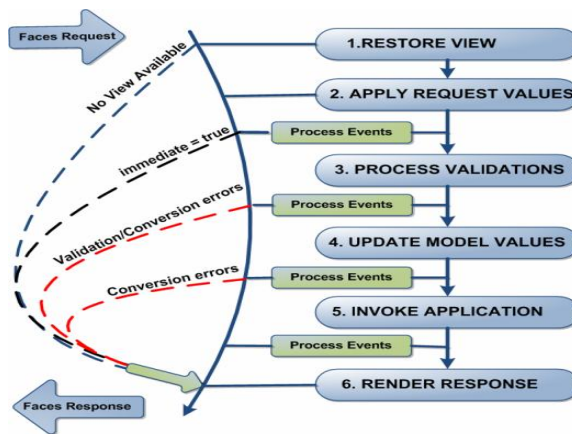


Figura No. 2.21: JSF - Ciclo vida.

Fuente: <http://osl2.uca.es/wikiCE/index.php/Archivo:JSF-Lifecycle.png>

En el ciclo de vida de procesamiento de peticiones JSF se realizan la mayoría de las operaciones del lado del servidor de una manera automática basada en eventos. Enlaza directamente campos con propiedades de una clase java, de modo que procesa los parámetros de entrada y gestiona un conjunto de componentes del lado del servidor y los sincroniza para que lo vea el usuario en el navegador. En la *figura No. 2.22* se presentan los procesos de cada ciclo de vida del JSF.

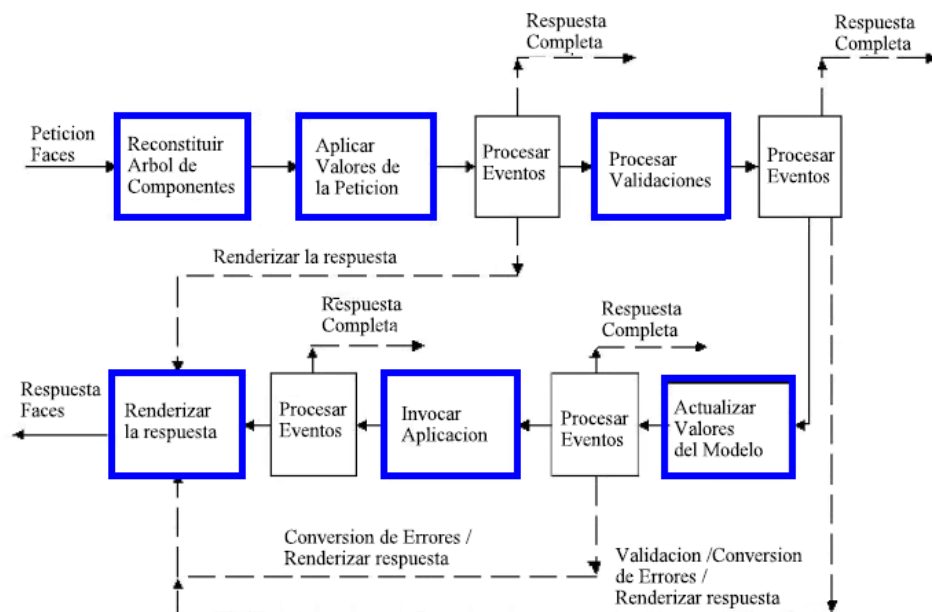


Figura No. 2.22: JSF - Proceso del Ciclo vida.

Fuente: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso>

JSF mantiene una vista en el lado del servidor que representa partes importantes del estado actual del cliente. Las fases del ciclo de vida son:

1. Crea o Restaura la Vista - Restore View

Cuando se hace una petición de una página JavaServer Faces, o cuando se produce un evento como pulsar sobre un enlace o un botón, el sistema JavaServer Faces entra en el estado reconstituir el árbol de componentes.

Durante esta fase, la implementación JavaServer Faces construye el árbol de componentes (vistas) que representan los componentes de interfaz de usuario del cliente. Las vistas se crean y almacenan en un contenedor de vistas llamado "*FacesContext*". Si no encuentra datos enviados por método POST, se salta directamente a la etapa de producir respuesta.

2. Aplicar valores a petición - Apply Request Values

En esta etapa se recuperan los valores de la request y se asignan a los beans de la página. Se invocan los convertidores definidos, si falla alguno, se marca el componente `UIComponent` como erróneo con `setValid(true)` y se agrega un mensaje al contexto de faces con `addMessage()`, luego se va directamente a producir respuesta.

Finalmente se guardan los valores enviados por la forma como cadenas dentro del componente (setters del `UIComponent`).

Se puede usar el atributo **immediate** para saltar el proceso de conversión y validación de datos que generalmente es usado para botones como Cancelar.

3. Procesar validaciones - Process Validation

En esta fase se verifican los parámetros de entrada según el conjunto de reglas definidas del fichero de configuración, se usan los valores

enviados por la fase de aplicación de valores. Si un valor incumple una regla, la implementación JavaServer Faces añade un mensaje de error al FacesContext, se marca el componente como erróneo con `setValid(true)` y se agrega un mensaje con `addMessage()`, después salta directamente a la fase producir respuesta.

El ciclo de vida avanza directamente hasta la fase renderizar las respuestas para que la página sea dibujada de nuevo incluyendo los mensajes de error.

4. Actualizar modelo - Update Model

Una vez que la implementación JavaServer Faces determina que el dato es leído y válido, procede a actualizar los beans (mediante los setters del ManagedBean) de los modelos asociados al componente. Esta asociación es generalmente usada con el atributo `value` del componente JSF.

Una vez que se pasan las validaciones y las conversiones, se realizan las actualizaciones de los objetos que modelan los datos en el lado del servidor (Managed Beans). El mecanismo es similar a los anteriores, se invoca al interfaz `processUpdates()` de manera recursiva en el árbol de componentes `UIViewRoot`.

5. Invocar la aplicación - Invoke Application

Durante esta fase, la implementación Java Server Faces maneja cualquier evento a nivel de la aplicación, como enviar un formulario o enlazar a otra página, si la aplicación necesita redirigirse a un recurso de aplicación web diferente o generar una respuesta que no contenga componentes JavaServer Faces, puede llamar a `FacesContext.responseComplete`.

Posteriormente, la implementación JavaServer Faces configura el árbol de componentes de la respuesta a esa nueva página y, por último, transfiere el control a la fase Renderizar la Respuesta.

6. Renderizar respuesta (final) - Render Response

En esta fase se genera la página que será enviada al usuario (respuesta HTML) con todos sus elementos y valores actualizados, invocando los atributos de codificación de los componentes para dibujar los componentes del árbol de componentes grabado en el FacesContext.

Si se encontraron errores durante las fases aplicar los valores a la petición, procesar validaciones o actualizar los valores del modelo, se dibujará la página original. Si las páginas contienen etiquetas `output_errors`, cualquier mensaje de error que haya en la cola se mostrará en la página.

Al llamar por primera vez a la página, sólo se ejecutan las dos primeras fases, debido a que no hay información de la petición para rellenar el modelo. Ahora el usuario rellena los datos y pueden ocurrir dos cosas, que haya algún error de **validación**, en cuyo caso no pasará del paso de **validación** y no se rellenará el modelo (nunca se ejecuta el setter) ni se ejecutará ninguna acción (no se ejecuta el método `save ()`).

O bien que todo esté correcto, entonces ejecutará el siguiente paso de **actualizar los valores del modelo** y si no hay ningún error en este paso, se pasarían a ejecutar las **invocaciones de los eventos**, en este caso, se ejecutaría el método `save ()`.

2.6.2 Beans y páginas JSF

Las aplicaciones web correctamente planificadas tienen dos partes: la parte de presentación y la lógica de negocio.

La parte de presentación afecta a la apariencia de la aplicación, y en el contexto de una aplicación basada en navegadores, la apariencia está determinada por las etiquetas HTML, esto comprende marcos, tipos de caracteres, imágenes, etc. La lógica de negocio se implementa en Java y determina el comportamiento de la aplicación.

En el contexto de JSF, la lógica de negocio está contenida en los beans, y el diseño está contenido en las páginas web. Un bean es una clase Java que contiene atributos. Un atributo es un valor identificado por un nombre, pertenece a un tipo determinado y puede ser leído y/o escrito sólo a través de métodos a tal efecto llamados métodos getter y setter.

En una aplicación JSF, se deben usar beans para todos los datos accedidos por una página. Los beans son los conductos entre la interfaz de usuario y la trastienda de la aplicación. Una vez que un bean ha sido definido, puede ser accedido a través de etiquetas. JSF.

Por ejemplo, la siguiente etiqueta lee y actualiza el atributo password del bean usuario `<h:inputSecret value="#{usuario.password}"/>`

Las características más importantes de un bean son los atributos que posee, también llamados propiedades.

2.6.3 Beans

Un bean es una clase Java que contiene atributos. Un atributo es un valor identificado por un nombre, pertenece a un tipo determinado y puede ser leído y/o escrito sólo a través de métodos a tal efecto llamados métodos getter y setter:

- Para conocer el valor de un campo llamado xxx se utiliza la función `getXxx` (o `isXxx` si es de tipo boolean).
- Para asignarle valor a un campo llamado xxx se usa la función `setXxx`.

Por ejemplo, `UsuarioBean.java` (*figura No. 2.23*) tiene dos atributos, nombre y password, ambos de la clase `String`:

```
public class UsuarioBean {
    private String nombre;
    private String password;
    // ATRIBUTO: nombre
    public String getNombre() { return nombre; }
    public void setNombre(String nuevoValor) { nombre = nuevoValor; }
    // ATRIBUTO: password
    public String getPassword() { return password; }
    public void setPassword(String nuevoValor) { password = nuevoValor; }
}
```

Figura No. 2.23: JSF - Creación de Bean.

En una aplicación JSF, se deben usar beans para todos los datos accedidos por una página. Los beans son los conductos entre la interfaz de usuario y la trastienda de la aplicación.

2.6.4 Eventos

Las aplicaciones de Web a menudo necesitan responder a los eventos del usuario, como los elementos seleccionados de un menú o dando un clic sobre un botón.

Típicamente, se registran manejadores de eventos con componentes; por ejemplo, podría registrar un oyente de cambio de valor (`valueChangeListener`) con una etiqueta menú en una página JSF de la siguiente manera (*figura No. 2.24*):

```
< h:selectOneMenu valueChangeListener="#{formulario.ciudad}"...>
...
< /h:selectOneMenu>
```

Figura No. 2.24: JSF - Ejemplo de Eventos.

JSF soportar tres clases de eventos que son:

- Eventos de cambio de valor.
- Eventos de acción.
- Eventos de fase.

Los eventos de cambio de valor son disparados con los componentes de entrada, como `h:inputText`, `h:selectOneRadio`, y `h:selectManyMenu`, cuando el valor del componente cambia y el formulario es tramitado.

Los eventos de acción son disparados por componentes de comando, como `h:commandButton` y `h:commandLink`, cuando el botón o el enlace es activado.

Los eventos de fase son disparados por el ciclo de vida de JSF.

2.7. AJAX - TECNOLOGÍA RIA

Es el acrónimo de **Asynchronous JavaScript And XML (JavaScript asíncrono y XML)**. AJAX es una de las técnicas de desarrollo de sistemas web que une varias tecnologías para la creación de aplicaciones interactivas.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor. (Eguiluz, 2013).

La importancia de esta técnica ya que al utilizarla, está se ejecuta en el lado del cliente (en el navegador web, pero por debajo mantiene una comunicación asíncrona con el servidor), permitiendo realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

AJAX se basa en la combinación de 4 tecnologías existentes, entre las que están:

Javascript: es un lenguaje interpretado por los navegadores modernos, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

XML: lenguaje de marcas utilizado para almacenar datos en forma comprensible. Se utiliza como un estándar para el intercambio de información estructurada entre diferentes plataformas.

HTML: es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

CSS: son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

Con las tecnologías resumidas, se indica que AJAX permite la comunicación asíncrona entre un servidor y un navegador en formato XML mediante programas escritos en Javascript, en donde su principal y primordial objetivo, es intercambiar información entre el servidor y el cliente sin la necesidad de recargar la página. De esta forma, se gana en usabilidad, experiencia y productividad del usuario final. (LibrosWeb, 2013).

A continuación en la *figura No. 2.25* que ilustra el modelo de aplicación Web básico comparado con el modelo AJAX, donde la imagen de la izquierda muestra el modelo tradicional de las aplicaciones web y la imagen de la derecha muestra el nuevo modelo propuesto por AJAX:

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

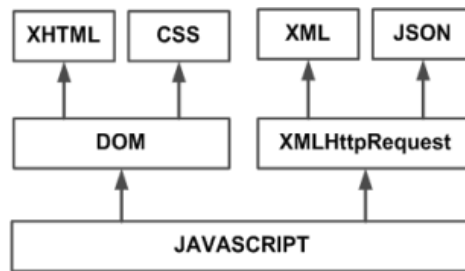


Figura No. 2.25: AJAX - Tecnologías bajo AJAX (LibrosWeb, 2013).

En las aplicaciones web tradicionales, las acciones del usuario en la página (pinchar en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al usuario.

En la *figura No. 2.26*, la imagen de la izquierda muestra el modelo tradicional de las aplicaciones web. La imagen de la derecha muestra el nuevo modelo propuesto por AJAX

Para comunicar la aplicación Web con el servidor, AJAX se vale del objeto JavaScript XMLHttpRequest, objeto con el cual una gran parte de los navegadores pueden recuperar y enviar datos en XML directamente, en background.

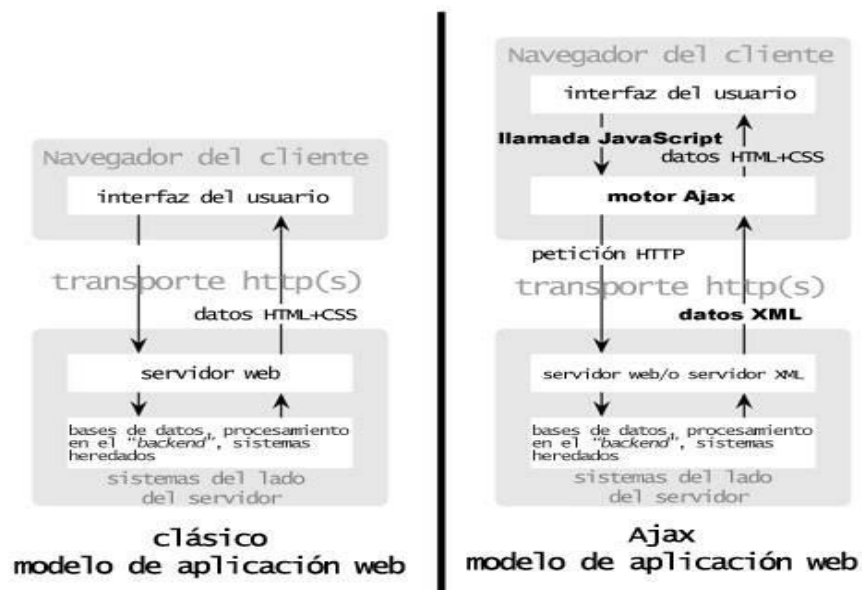


Figura No. 2.26: AJAX - Modelo tradicional y modelo AJAX (LibrosWeb, 2013).

Esta técnica tradicional para crear aplicaciones web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, su uso se convierte en algo molesto.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, porque en una misma página se puede hacer varias cosas, sin tener que saltar a otra “página Web”.

En la siguiente *figura No. 2.27* muestra la diferencia más importante entre una aplicación web tradicional y una aplicación web creada con AJAX. La imagen superior muestra la interacción síncrona propia de las aplicaciones web tradicionales. La imagen inferior muestra la comunicación asíncrona de las aplicaciones creadas con AJAX.

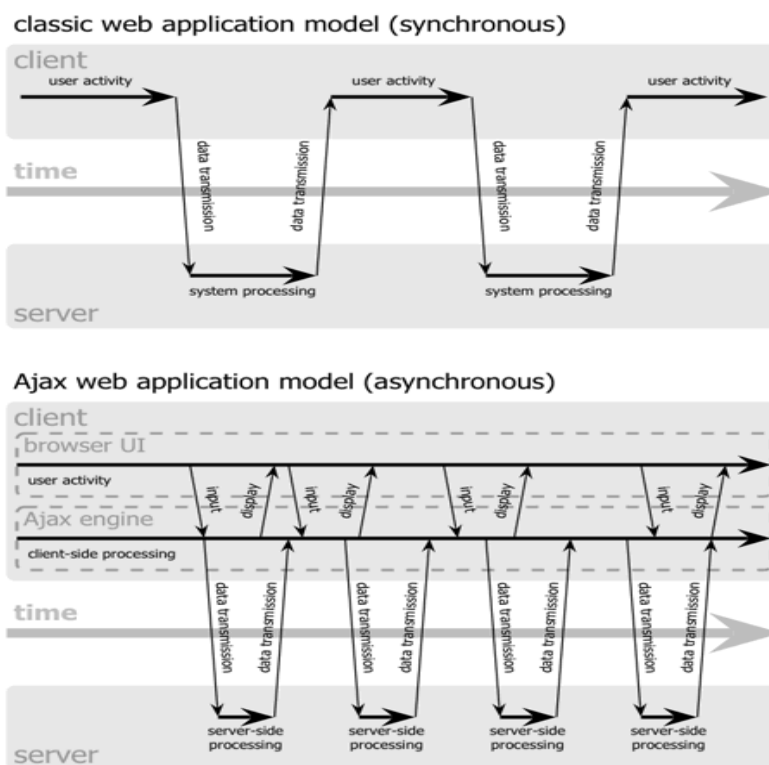


Figura No. 2.27: AJAX - Comparación síncronas y asíncronas AJAX.

(LibrosWeb, 2013)

Las peticiones HTTP al servidor se sustituyen por peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción requiere una respuesta del servidor, la petición se realiza de forma asíncrona mediante AJAX. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

Desde su aparición, se han creado cientos de aplicaciones web basadas en AJAX. En la mayoría de casos, AJAX puede sustituir completamente a otras técnicas como Flash. Además, en el caso de las aplicaciones web más avanzadas, pueden llegar a sustituir a las aplicaciones de escritorio.

A continuación se muestra una lista de algunas de las aplicaciones más conocidas basadas en AJAX:

- Gestores de correo electrónico: [Gmail](#), [Yahoo Mail](#), [Windows Live Mail](#).
- Cartografía: [Google Maps](#), [Yahoo Maps](#), [Windows Live Local](#).
- Aplicaciones web y productividad: [Google Docs](#), [Zimbra](#), [Zoho](#).

Ventajas

- Rapidez en las operaciones.
- Menos carga del servidor (menos transferencia de datos cliente/servidor).
- Menos ancho de banda.
- Soportada por la mayoría de navegadores.
- Interactividad (El usuario no tiene que esperar hasta que lleguen los datos del servidor).
- Portabilidad.
- Usabilidad.
- Velocidad (Debido a que no hay que recargar la página nuevamente).

Desventajas

- Se pierde el concepto de “volver a la página anterior”.
- Problemas con navegadores antiguos.
- No funciona si el usuario tiene desactivado el Javascript en su navegador.
- Se requieren conocimiento sobre las tecnologías que forman AJAX.

Ejemplos de uso de AJAX son las aplicaciones de mapas, búsquedas en tiempo real, chat, validación de formularios en tiempo real, aplicaciones de carro de compras, aplicaciones de correo electrónico, aplicaciones móviles. Las aplicaciones web y la interacción mediante el uso de AJAX han dado origen al término "Web 2.0" - término que sin duda indica una gran revolución en el modo en que se hace uso de la web.

A partir de la versión de 2.X de Primefaces, se puede encontrar con un amplio soporte AJAX, invocando JavaScript's desde algunos componentes JSF. Con JSF 2.X se puede utilizar la etiqueta <f:ajax> dentro de nuestros componentes.

Previamente, en nuestro facelet se debe registrar el componente <h:head>, en lugar de declarar directamente la etiqueta <head>. De otro modo, los scripts JSF no podrían registrarse y no estarían disponibles para la aplicación.

Algunos atributos de la etiqueta <f:ajax> a destacar:

- **Event:** Permite identificar el nombre del evento AJAX que se espera para el componente. Previamente en la definición del componente, se comprobará que lo soporte.

```
<p:ajax event="keyup" update="out1" />
```

```
<p:ajax event="blur" update="out2" />
```

- **Execute:** Es una colección de identificadores de elementos (separados

por espacios) que se tendrán en cuenta en la petición (ajax request). Estará compuesto por los id de los componentes, o bien, por una referencia @this, @form, @all, @none. Si no se especifica este atributo, por defecto tomará la referencia @this.

- **Render:** Nuestra colección de id's de componentes que se renderizarán. También deben estar separados por espacios, utilizando referencias de componentes (@this, @all, etc). Si no aparece esta etiqueta asume como valor @none.
- **Listener:** Mepresenta el método que se ejecutará en el lado servidor.

Los componentes Primefaces y la etiqueta <p:ajax>, añaden más atributos:

- **update:** Id's de componentes a renderizar.
- **Immediate:** Permite indicar cuándo se procesarán las peticiones. Cuando es true, las acciones son procesadas inmediatamente cuando se asigna un valor, es decir, en la fase "apply_request_values". Cuando es "false" el procesamiento se realiza en la fase "invoke_application_phase". Es útil, cuando se desea procesar de forma parcial nuestros formularios.
- **Process:** Colección de identificadores de componentes que se procesarán en las peticiones parciales. También se puede utilizar referencias para identificar nuestros elementos: @this, @form, @all, @none, y @parent.
- **widgetvar.** Permite añadir un nombre a nuestro componente en el lado cliente, que permite referenciar nuestro componente una vez renderizado desde nuestros scripts javascript en el lado cliente.

Añadiendo Ajax a nuestros formularios: En la *figura No. 2.28*, se ilustra un ejemplo sencillo de Ajax utilizando nuestra etiqueta <f:ajax>:

```

<h:form>
  <h:inputText id="name" value="#{myBean.field}" />
  <h:commandButton value="Submit">
    <f:ajax event="click" execute="@this" render="display"/>
  </h:commandButton>
  <h:outputText value="#{myBean.field}" id="display" />
</h:form>

```

Figura No. 2.28: AJAX - Etiqueta.

Procesamiento parcial: Se modifica el ejemplo para realizar un procesamiento parcial de nuestro formulario. Se recuerda que por defecto la ejecución apunta al propio componente (@this), siendo necesario especificar los componentes a procesar en la petición. Si no se hiciera de esta manera, no se remitirá ningún valor en la petición Ajax, ver *figura No. 2.29*.

```

<h:form>
<h:inputText id="name" value="#{myBean.field}" />
  <h:commandButton value="Submit">
    <f:ajax execute="@form" render="display">
  </h:commandButton>
<h:outputText value="#{myBean.field}" id="display" />
</h:form>

<h:form>
<p:inputText id="name" value="#{myBean.field}" />
  <p:commandButton value="Submit" process="name" update="display"/>
<p:outputText value="#{myBean.field}" id="display" />
</h:form>

```

Figura No. 2.29: AJAX - Procesamiento Parcial.

Renderizando componentes fuera de nuestro form: Hasta ahora, se ha referenciado los identificadores (id) de nuestros componentes directamente por su nombre. Además, se debe considerar que están enmarcados dentro del primer componente padre que sea contenedor (**namingContainer**).

Por ejemplo, un componente UIForm (implementa la clase namingContainer) y, es responsable de englobar y dar un espacio de nombres a los componentes que contiene.

En JSF se puede utilizar el carácter “dos puntos” para desmarcarse del namingContainer dónde se encuentra nuestro componente, es decir, permite acceder directamente a otro espacio de nombres de otro namingContainer.

Para acceder al viewRoot directamente se empezará directamente con el

carácter "dos puntos", al que se añadirá las referencias de los contenedores:

```
<h:form id="form">
```

```
<h:outputText id="text" value="#{myBean.myField}"/>
```

```
</h:form>
```

Para referencia nuestro componente "text" fuera del form, se utiliza la siguiente referencia:

```
<f:ajax render="form:text"/>
```

O, si estuviera en la raíz de componentes:

```
<f:ajax render=":form:text"/>
```

En conclusión cabe anotar que **AJAX no es una tecnología**, sino una **combinación de varias de tecnologías y técnicas**, y tampoco se le puede confundir con un lenguaje de programación.

2.7.1 Primefaces

Primefaces es una librería de componentes de código abierto para JSF 2.0 con más de 100 ricos componentes incluyendo Ajax, *figura No. 2.30*.



Figura No. 2.30: Primefaces - Página oficial del Showcase de Primeface.

Fuente: <http://www.primefaces.org/showcase/>

Primefaces es mucho mejor que muchas otras bibliotecas de componentes JSF, debido a las siguientes características:

- ✓ Es compatible con otras librerías de componentes, como JBoss RichFaces.
- ✓ Es un proyecto open source, activo y bastante estable entre versiones.
- ✓ Un amplio conjunto de componentes de Interfaz de usuario (DataTable, AutoComplete, HtmlEditor, Charts, editor HTML, autocompletado etc). No se requiere extra configuración, ni son necesarias dependencias.
- ✓ Excelente documentación con ejemplos de código.
- ✓ Soporte para Ajax, basándose en el estándar JSF 2.0 Ajax API.
- ✓ Sin dependencias, ni configuraciones, además de ser muy ligero.
- ✓ Soporte para interfaces de usuario sobre dispositivos móviles, provee de un kit para este menester.
- ✓ Múltiples temas de apariencia, listos para usar.

Algunos inconvenientes podrían ser:

- ✓ Para utilizar el soporte de Ajax, se lo tiene que indicar explícitamente por medio de atributos específicos de cada componente.
- ✓ No se puede utilizar el soporte de Ajax de JSF 2 (mediante `<f:ajax>`) con los componentes de Primefaces.

Se puede decir que Primefaces cumple sobradamente con todas esas características, por lo que puede ser una buena elección. Además hay que tener en cuenta que no se limita a utilizar una librería, aunque en ese caso habría investigar un poco más para estar seguros de que no se produzcan incompatibilidades entre ellas, que podrían dar muchos quebraderos de cabeza.

2.8. SMSLIB

SMSLib es una biblioteca de mensajería SMS que proporciona una API

de mensajes de texto universal, que se puede utilizar para enviar y recibir mensajes a través de módems GSM y / o los operadores de SMS (<http://smslib.org/>, 2015).

2.8.1 Características

SMSLib se encuentra actualmente en su tercera versión y conserva la antigua funcionalidad:

- Soporta los teléfonos GSM y módems GSM conectados a través de interfaces de puerto serie o interfaces IP.
- Funciona con los protocolos UDP / TEXT.
- Soporta mensajes de texto simples entrantes y salientes.
- Funciona con codificaciones de mensajes 7bit, 8bit y UCS2 (Unicode).
- Soporta grandes mensajes entrantes y salientes.
- Mensajería Flash.
- Los mensajes salientes con información de puerto / direccionamiento.
- Estado de los mensajes (entrega) de informe.
- Información básica de GSM disponible: Módem, Fabricante, S / W de revisión, el nivel de señal, etc.
- Soporta unos operadores, utilizando protocolos HTTP / HTTPS.
- Apoya el protocolo SMPP.

2.8.2 Varias puertas de enlace

SMSLib v3 ha introducido el concepto de la puerta de enlace, que es un interfaz para un dispositivo o un servicio que puede enviar y/o recibir mensajes SMS. Una puerta de enlace puede ser un módem GSM o un proveedor de SMS masivo. SMSLib puede manejar múltiples pasarelas al mismo tiempo.

2.8.3 Recibir síncrono / asíncrono

SMSLib presenta dos maneras para recibir mensajes:

- **Sincrónica:** usted toma la responsabilidad y periódicamente llama al método de lectura ().
- **De forma asincrónica:** esperar a SMSLib llame a la recepción de mensajes.

2.8.4 Síncrono / asíncrono Envío

SMSLib le da dos opciones para el envío de mensajes:

- **Sincrónica:** cuando llame SMSLib enviar un mensaje, el subproceso se bloquea hasta que el mensaje es enviado o falla.
- **De forma asincrónica:** cuando llame SMSLib enviar un mensaje, el hilo no se bloquea y SMSLib vuelve inmediatamente. SMSLib pone en cola el mensaje (s) y los envía en el fondo. Usted puede elegir para ser notificado por el destino de los mensajes.

Los mensajes entrantes: Cada vez que se recibe un mensaje, SMSLib puede configurarse para llamar a su método especificado.

Los mensajes salientes: Cuando se envía de forma asincrónica mensajes, SMSLib puede configurarse para llamar a su método especificado con el mensaje enviado (actualizado con el estado, etc.)

Estado de puerta de enlace cambia: SMSLib puede notificarle cada vez que una puerta de entrada cambia su estado.

2.9. Maven

Maven es una herramienta principalmente utilizada en el desarrollo de software Java. Aparece ante la necesidad de modelar el concepto de "proyecto" y artefacto en forma estándar independientemente del IDE de desarrollo.

El objetivo principal del proyecto es, que el desarrollo se complete en el menor tiempo posible. Para que éste objetivo se cumpla hay algunas áreas que es necesario abordar.

- Fácil proceso de construcción (build) de un proyecto o módulo.
- Proveer un sistema de construcción uniforme.- Maven permite el desarrollo usando el Project Object Model (POM) y un conjunto de plugins que son compartidos por todos los proyectos que usen Maven. Una vez familiarizado con este sistema se podrá incluir o utilizar paquetes de software que se necesite, únicamente añadiendo un nuevo repositorio o modificando su versión (Archivo POM).
- Proporcionar información de calidad del proyecto.
- Una migración transparente a nuevas características. Como ya se mencionó en el punto dos, únicamente será necesario cambiar el archivo POM.xml.

2.9.1 Ciclo de vida Maven

Para ello, en Maven se definen tres ciclos de build del software con una serie de etapas diferenciadas. Por ejemplo el ciclo por defecto tiene las etapas de:

- **Validación** (validate): Validar que el proyecto es correcto.
- **Compilación** (compile). Genera los ficheros .class compilando los fuentes .java
- **Test** (test): Probar el código fuente usando un framework de pruebas unitarias.
- **Empaquetar** (package): Empaquetar el código compilado y transformarlo en algún formato tipo .jar o .war.
- **Pruebas de integración** (integration-test): Procesar y desplegar el código en algún entorno donde se puedan ejecutar las pruebas de integración.
- **Verificar** que el código empaquetado es válido y cumple los criterios de calidad (verify).

- **Instalar** el código empaquetado en el repositorio local de Maven, para usarlo como dependencia de otros proyectos (install). Copia el fichero .jar a un directorio de nuestro ordenador donde maven deja todos los .jar. De esta forma esos .jar pueden utilizarse en otros proyectos maven en el mismo ordenador.
- **Desplegar** el código a un entorno (deploy). Copia el fichero .jar a un servidor remoto, poniéndolo disponible para cualquier proyecto maven con acceso a ese servidor remoto.

Pero estas metas pueden ser añadidas al ciclo de vida a través del Project Object Model (POM). Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

2.9.2 Artefactos y POM

Para definir un Artefacto se necesita crear un fichero POM.xml (Project Object Model) que es el encargado de almacenar toda la información que se ha indicado anteriormente, ver *figura No. 2.31*.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>com.genbetadev.proyecto1</groupId>
  <artifactId>proyecto1</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
  <dependencies>

  <dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
  </dependency>
  </dependencies>
</project>
```

Figura No. 2.31: Maven - Configuración POM.

La estructura del fichero puede llegar a ser muy compleja y puede llegar a depender de otros POM. En este ejemplo se o viendo el fichero más sencillo posible. En él se define el nombre del Artefacto (artifactID) el tipo de empaquetado (jar) y también las dependencias que tiene (log4j). De esta manera nuestra librería queda definida de una forma mucho más clara.

Si a Maven se le indica el jar externo que se necesita, es capaz de ir a buscarlos a internet y descargarlos por su uso.

2.9.3 Integración con IDE

Maven sustituye el entorno integrado de desarrollo (IDE por sus siglas en inglés), por tanto la integración con diferentes IDEs es muy importante. Existen plugins de Maven para crear archivos de configuración del IDE a partir de los POMs. Actualmente se soportan: Eclipse, Netbeans, IntelliJ IDEA, JDeveloper 11G.

2.9.4 Maven y Artefactos

Un Artefacto puede verse como una librería con esteroides (aunque agrupa más conceptos). Contiene las clases propias de la librería pero además incluye toda la información necesaria para su correcta gestión (grupo, versión, dependencias etc). Esto se lo puede ilustrar en la *figura No. 2.32*.

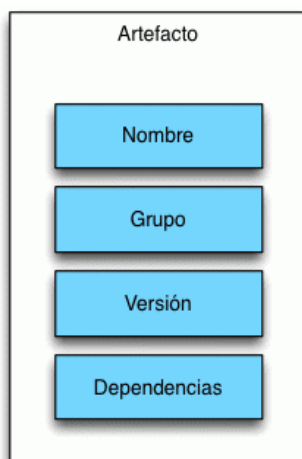


Figura No. 2.32: Maven - Artefacto.

groupId: representa la organización autora/dueña del artefacto. Por ejemplo: com.ibm, org.apache, org.jboss, etc. Este id por sí mismo no identifica un artefacto por las razones obvias de que una organización puede tener más de un artefacto.

artifactId: nombre del proyecto/artefacto actual. Por ejemplo: http-client, hibernate, tomcat, commons-collections, etc.

Versión: como su nombre dice, identifica al número de versión del artefacto.

Maven establece un estándar a nivel mundial. Tanto nuevos proyectos como todos los de la comunidad java+maven van a tener la misma estructura e identificación, esto se lo puede observar en la *figura No. 2.33*.

```
<groupId>org.springframework</groupId>
<artifactId>spring</artifactId>
<version>2.5.6.SEC02</version>
```

Figura No. 2.33: Maven - Variable Artefacto.

2.9.5 Instrucciones básicas Maven

Maven tiene una arquitectura de plugins, para poder ampliar su funcionalidad, aparte de los que ya trae por defecto.

Ejemplos de goals serían:

- **mvn clean:clean (o mvn clean):** limpia todas las clases compiladas del proyecto.
- **mvn compile:** compila el proyecto
- **mvn package:** empaqueta el proyecto (si es un proyecto java simple, genera un jar, si es un proyecto web, un war, etc...)
- **mvn install:** instala el artefacto en el repositorio local (/Users/home/.m2)

2.10. METODOLOGÍA

En la actualidad, la demanda que existen sobre sistemas de software eficientes, con alto desempeño funcional y con una alta calidad, hace la necesidad de contar con metodologías que gestionen proyectos, bajo un enfoque disciplinado y sistemático.

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente, cada uno tiene prioridades, requerimientos, y tecnologías muy diferentes.

Sin embargo, en todos los proyectos, se debe minimizar el riesgo, garantizar buenos resultados y entregar software de calidad superior a tiempo. **Rational Unified Process, o RUP**, es una plataforma flexible de procesos de desarrollo de software que ayuda brindando guías consistentes y personalizadas de procesos para todo el equipo de proyecto.

Es de suma importancia elegir la metodología adecuada, así como las herramientas de implementación adecuadas, es por ello que la metodología RUP basada en UML proporciona todas las bases para llevar al éxito la elaboración del software.

2.10.1 RUP (Proceso Unificado de Rational)

Las siglas RUP en ingles significa Rational Unified Process (Proceso Unificado de Rational) es un producto del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo. Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos, ver *figura No. 2.34*.

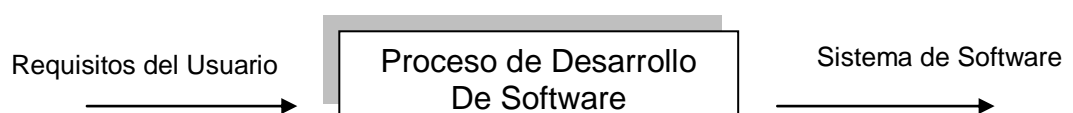


Figura No. 2.34: RUP (Wikipedia, Proceso Unificado de Rational, 2014).

2.10.2 Las mejores prácticas

RUP pretende implementar las mejores prácticas actuales en ingeniería de software:

- Desarrollo iterativo del software.
- Administración de requerimientos.
- Uso de arquitecturas basadas en componentes.
- Modelamiento visual del software.
- Verificación de la calidad del software.
- Control de cambios.

2.10.3 Ciclos y fases

RUP divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo, cada ciclo se divide en cuatro Fases: Inicio, Elaboración, Construcción y Transición.

Cada fase concluye con un hito bien definido donde deben tomarse ciertas decisiones.

2.10.4 Dimensiones del RUP

El RUP tiene dos dimensiones:

El **eje horizontal** representa tiempo y demuestra los aspectos del ciclo de vida del proceso.

El **eje vertical** representa las disciplinas, que agrupan actividades definidas lógicamente por la naturaleza.

La primera dimensión representa el aspecto dinámico del proceso y se expresa en términos de fases, de iteraciones, y la finalización de las fases. La segunda dimensión representa el aspecto estático del proceso: cómo se describe en términos de componentes de proceso, las disciplinas, las actividades, los flujos de trabajo, los artefactos, y los roles.

En la *figura No. 2.35* se puede observar como varía el énfasis de cada disciplina en un cierto plazo en el tiempo, y durante cada una de las fases. Por ejemplo, en las iteraciones tempranas, se pasa más tiempo en requerimientos, y en las últimas iteraciones se pasa más tiempo en poner en práctica la realización del proyecto en sí.

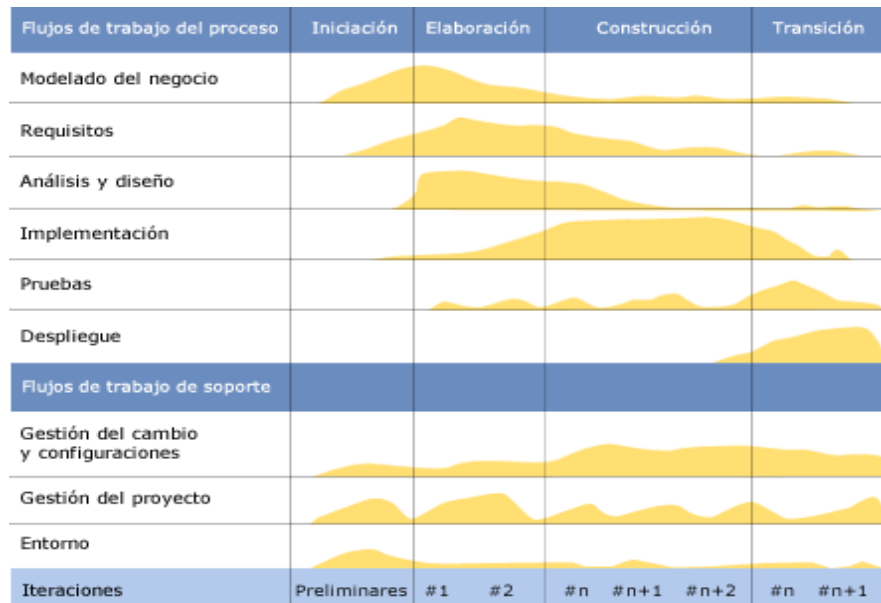


Figura No. 2.35: RUP - Fases (Wikipedia, Proceso Unificado de Rational, 2014).

Se puede hacer mención de las tres características esenciales que definen al RUP:

- **Proceso Dirigido por los Casos de Uso:** Con esto se refiere a la utilización de los Casos de Uso para el desenvolvimiento y desarrollo de las disciplinas con las actividades necesarias. Un Caso de Uso es una secuencia de pasos a seguir para la realización de un fin o propósito, y se relaciona directamente con los requerimientos, ya que un Caso de Uso es la secuencia de pasos que conlleva la realización e implementación de un Requerimiento planteado por el Cliente. Esta es la base para la implementación de las fases y disciplinas del RUP.
- **Proceso Iterativo e Incremental:** Es el modelo utilizado por RUP para el desarrollo de un proyecto de *software*. Este modelo plantea la

implementación del proyecto a realizar en Iteraciones, con lo cual se pueden definir objetivos por cumplir en cada iteración y así poder ir completando todo el proyecto iteración por iteración, con lo cual se tienen varias ventajas, entre ellas se puede mencionar la de tener pequeños avances del proyectos que son entregables al cliente el cual puede probar mientras se está desarrollando otra iteración del proyecto, con lo cual el proyecto va creciendo hasta completarlo en su totalidad.

- **Proceso Centrado en la Arquitectura:** Define la Arquitectura de un sistema, y una arquitectura ejecutable construida como un prototipo evolutivo. Una arquitectura ejecutable es una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades.

2.10.5 Fases del RUP

El ciclo de vida del software del RUP se descompone en cuatro fases secuenciales (figura). En cada extremo de una fase se realiza una evaluación (actividad: Revisión del ciclo de vida de la finalización de fase) para determinar si los objetivos de la fase se han cumplido. Una evaluación satisfactoria permite que el proyecto se mueva a la próxima fase.

El ciclo de vida consiste en una serie de ciclos, cada uno de los cuales produce una nueva versión del producto, cada ciclo está compuesto por fases y cada una de estas fases está compuesta por un número de iteraciones, estas fases son:

a) Fases Inicio

En esta fase se centra en la etapa de inicio del proyecto, con la finalidad de generar las actividades de modelamiento y requerimientos basado en la recopilación de datos, procesos, documentos y demás información relevante para la construcción del plan de trabajo que garantice la ejecución del proyecto de forma exitosa, logrando los objetivos propuestos.

La fase inicio, identifican los siguientes puntos para su planteamiento

- Análisis de los diferentes requerimientos relevantes para el funcionamiento del servicio de comunicación.
- Se define el alcance del proyecto.
- Se definen los costos y actividades a realizarse.
- Se definen los Riesgos de Servicio, mitigando posibles inconvenientes con el módem y las consultas de la información académica del Sistema Banner.
- Se define el Modelo de Negocio del Servicio de Comunicación.
- Se desarrollan los diferentes casos de uso en referencia a los requerimientos definidos.
- Se Identifica y se propone la Arquitectura del desarrollo.

b) Fases Elaboración

El objetivo de la fase de elaboración es refinar los requisitos obtenidos en la fase de iniciación y con ellos definir la arquitectura base del Servicio de Comunicación, el monitoreo de los posibles los riesgos, los planes de contingencia y producir un plan de administración del proyecto.

Entre lo que se destaca en esta fase son los siguientes puntos de vista:

- Especificación de requerimientos a detalle.
- Diseño del Sistema.
- Generación de los Diagramas de Casos de Uso.
- Generación de Diagramas de Actividades.
- Generación del Diagrama de Clases.
- Generación del Diagrama de Entidad - Relación.
- Analizar el dominio del problema a profundidad.
- A partir de aquí la arquitectura, los requerimientos y los planes de desarrollo son estables.
- Desarrollar un plan de proyecto.

c) Fases de RUP: Construcción

El producto se desarrolla a través de iteraciones en donde se involucra las tareas de análisis, diseño e implementación.

- Se realiza el desarrollo del Sistema o Servicio de Comunicación.
- Gran parte del trabajo es programación y pruebas internas y funcionales del Servicio de Comunicación.
- Además las pruebas son usadas para validar el producto con las expectativas del cliente.
- Se realiza la documentación del servicio construido como los documentos técnicos y funcionales.
- Implementación, el producto de software integrado y corriendo en la plataforma adecuada.
- Para la construcción del servicio de comunicación se debe proveer al desarrollador de los siguientes modelos y diagramas definidos del análisis de requerimientos.
 - Vista Lógica:
 - Diagrama de Clases.
 - Modelo Entidad Relación.
 - Vista de Implementación:
 - Diagrama de secuencias.
 - Diagrama de estados.
 - Diagrama de colaboración.
- Para su implementación, el producto de software debe estar integrado con las demás plataformas y además debe estar corriendo en la plataforma adecuada.

d) Fases de RUP: Transición

El objetivo de esta fase se centra en la liberación del producto y se entrega al usuario para un uso real, una vez instalado surgirán nuevos elementos que implicarán nuevos desarrollos (ciclos).

Entre la documentación entregable, se indican las siguientes:

- Manuales de usuario y técnico.
- Entrenamiento de usuarios.

2.10.6 Disciplinas

En RUP, las disciplinas son las fuentes principales y estas deben alternarse en cada interacción.

Se identifican las siguientes disciplinas:

Modelado del negocio, en esta disciplina se pretende entender el negocio de la organización referente al Sistema. Para esto se analiza los cambios que se produzcan con el Sistema, se verifica el procedimiento actual y de ser necesario se busca una posible reingeniería en los procedimientos actuales.

Requerimientos, se genera un documento de acuerdo al alcance del proyecto y se define las partes del negocio que se construyen con el proyecto y que partes no.

Análisis y Diseño, tomando en cuenta los requerimientos y restricciones se diseña la solución a ser implementada. En esta disciplina se establece y valida la arquitectura, se comprenden los requerimientos, se diseñan los módulos, la base de datos, interfaz de usuario.

Implementación, se pasa el diseño a código ejecutable con un nivel primario de pruebas para entender y evolucionar el diseño. Así mismo se escribe el código fuente, se implementa los módulos y se integra el código en subsistemas.

Verificación, en la verificación se detectan fallos en el sistema o servicio, se valida el diseño, se comprueba que el sistema satisfaga los requerimientos, además se ejecutan pruebas y se reportan defectos para una futura iteración.

Puesta en marcha, se da la liberación del sistema o servicio y se entrena a los usuarios.

Configuración y gestión del cambio, se gestiona las peticiones de cambio, se planea el control y reportes de la configuración, y se administra la versión base del sistema o servicio.

Administración del proyecto, el objetivo es dirigir las actividades que toman lugar en el proyecto como gestión del riesgo, dirección del equipo de trabajo, coordinación externa, planeación y culminación del proyecto.

Entorno, el objetivo es asegurar que se pueda ejecutar el proceso por medio de la identificación, evaluación, instalación y configuración de las herramientas para el equipo del proyecto.

2.10.7 Lenguaje de Modelamiento Unificado (UML)

El UML (Lenguaje Unificado de Modelado), es una de las herramientas más emocionantes en el mundo del desarrollo de sistemas. Esto se debe a que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarles a otras personas.

La clave está en organizar el proceso de diseño de tal forma que los analistas, usuarios, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan y convengan con él. El UML proporciona tal organización.

2.10.8 Diagrama de Casos de Uso

Los diagramas de Casos de Uso documentan el comportamiento de un sistema desde el punto de vista del usuario, por tanto los casos de uso determinan los requisitos funcionales del sistema o representan las funciones que un sistema puede ejecutar (Pisco, 2013).

Estos se pueden usar durante las siguientes fases del desarrollo:

- Captura de Requisitos.
- Planificación de Iteraciones de desarrollo.
- Validación del Sistema.

Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente, como lo demuestra la *figura No. 2.36*.

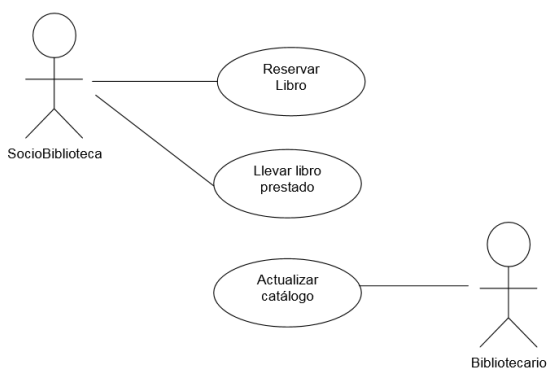


Figura No. 2.36: UML - Caso de Uso.

Fuente: <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>

Los elementos básicos de los casos de uso son los siguientes:

Actores: Representan un tipo de usuario del Sistema, en los diagramas de caso de uso los actores se dibujan como una silueta humana, como lo demuestra la *figura No. 2.37*.

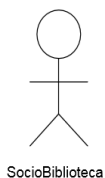


Figura No. 2.37: UML - Actor.

Fuente: <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>

Se interpreta como usuario a cualquier cosa externa que interactúa con el Sistema.

Otros actores pueden ser: Sistemas Informáticos o Unidades Organizativas.

Al identificar el interés de los actores por los casos de uso permite:

- Priorizar los desarrollos de unos casos con respecto a otros.
- Planificar mejor las iteraciones.

Caso de Uso: es una tarea que debe poder llevarse a cabo con el apoyo con apoyo del sistema que se está desarrollando, este se representa mediante un óvalo.

Cada uno de los casos de uso debe detallarse habitualmente por una descripción textual.

Asociaciones: Hay una asociación entre el actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo el caso de uso, como lo demuestra la *figura No. 2.38*.

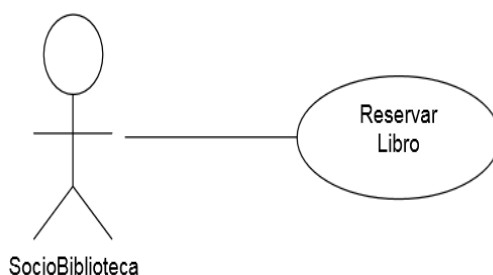


Figura No. 2.38: UML - Asociación en Casos de Uso.

Fuente: <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>

Si en un diagrama de casos de uso aparece una asociación entre un actor y un caso, indica que puede que ese actor interactúe con el sistema en ese caso de uso.

Existen 3 tipos de asociaciones: Incluye, Extiende y Generaliza.

Incluye.- Se puede incluir una relación entre dos cosas de uso tipo “include” si se desea especificar comportamiento común entre dos o más casos de uso.

En el diagrama, se indica mediante una flecha a trazos y abierta, como lo demuestra la *figura No. 2.39*.

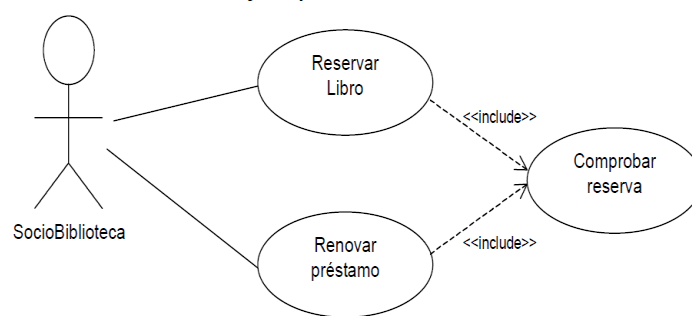


Figura No. 2.39: UML - Inclusión en Casos de Uso.

Fuente: <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>

En el ejemplo se indica que tanto “reservar libro” como “renovar préstamo” hacen algo en común: “comprobar reservar”.

Ventajas:

- Las descripciones de los casos de uso son más cortas y se entienden mejor.
- La identificación de funcionalidad común puede ayudar a descubrir el posible uso de componentes ya existente en la implementación.

Desventajas:

- La inclusión de estas relaciones hace que los diagramas sean más difíciles de leer, sobre todo para los clientes.

Extiende.- Se puede incluir una relación entre dos cosas de uso tipo “extend” si se desea especificar diferentes variantes del mismo caso de uso,

este implica que el comportamiento de un caso de uso es diferente dependiendo de ciertas circunstancias.

En principio, esas variaciones pueden también mostrarse como diferentes descripciones de escenarios asociados al mismo caso de uso, esto se demuestra en la *figura No. 2.40*.

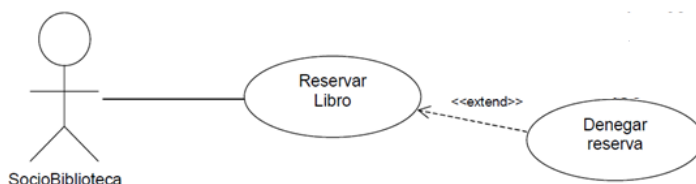


Figura No. 2.40: UML - Extensión en Casos de Uso.

Fuente: <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>

La flecha en el caso de las relaciones “extend” va hacia el caso de uso original.

Es útil dibujar los límites del sistema cuando se pretende hacer un diagrama de casos de uso para parte del sistema, esto se demuestra en la *figura No. 2.41*.

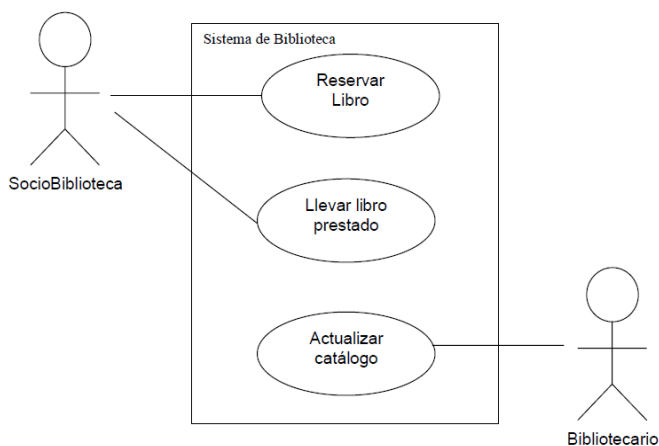


Figura No. 2.41: UML - Límites en Casos de Uso.

Fuente: <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>

Diagrama de Clases

Una clase es una descripción de conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica.

Las clases son gráficamente representadas por cajas con compartimentos para:

- Nombre de la clase, atributos y operaciones / métodos (ver *figura No. 2.42*).
- Responsabilidades, Reglas, Historia de Modificaciones, etc.

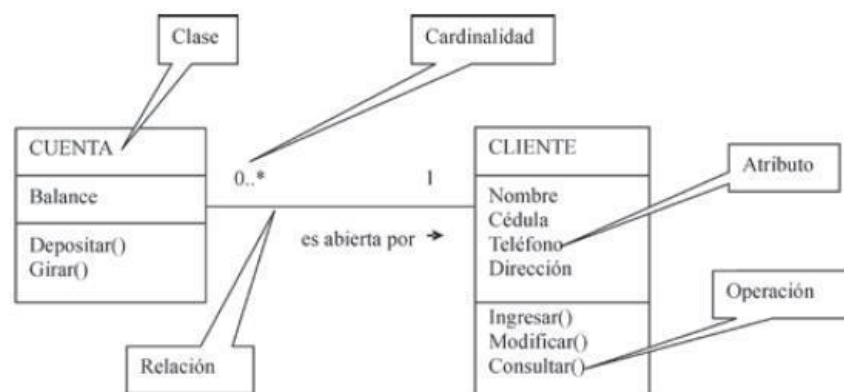


Figura No. 2.42: UML - Diagrama de Clases.

Fuente: http://www.scielo.org.co/scielo.php?pid=S1692-33242006000200010&script=sci_arttext

Diagrama de Secuencias

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista business del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

En la *figura No. 2.43* se identifica el diagrama de secuencia y sus interacciones.

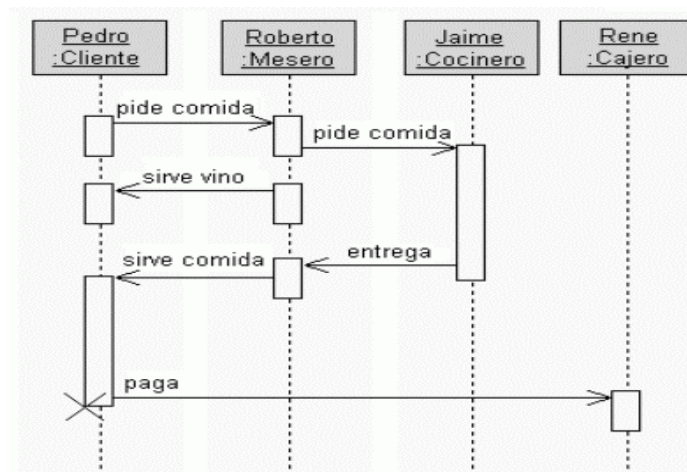


Figura No. 2.43: UML - Diagrama de Secuencia.

Fuente: <http://utnsimocametodologia.blogspot.com/2011/10/aprendiendo-uml-hora-9-diagrama-de.html>

Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos. Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

Diagrama de Colaboración o Comunicación

Se llamaban Diagramas de Colaboración en UML y sirven para enfatizar los vínculos de datos entre los participantes de una interacción. Usan numeración para mostrar la secuencia de un mensaje y usualmente. *En la figura No. 2.44* se ilustra el diagrama de colaboración.

Estos diagramas de Comunicación modelan las interacciones entre objetos o partes en términos de mensajes en secuencia, además representan una combinación de información tomada desde el diagrama de clases, secuencia, y de los casos de uso describiendo tanto la estructura estática como el comportamiento dinámico de un sistema.

Para mantener el orden de los mensajes en un diagrama de comunicación, los mensajes son etiquetados con un número cronológico y colocado cerca del enlace por el cual se desplaza el mensaje.

Leer un diagrama de comunicación conlleva comenzar en el mensaje 1.0, y seguir los mensajes desde un objeto hasta el siguiente, sucesivamente.

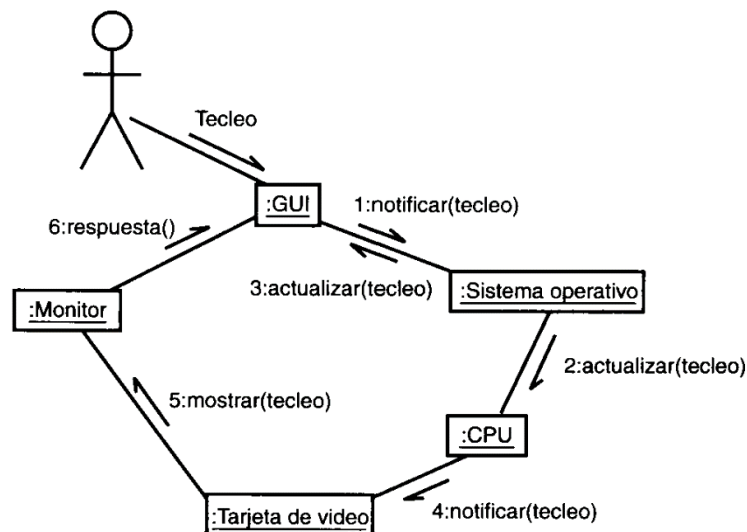


Figura No. 2.44: UML - Diagrama de Colaboración.

Fuente: <http://aprendiendoumlfacil.blogspot.com/2012/05/los-diagramas-de-colaboraciones.html>

Diagrama de Estados

Los diagramas de estado son una técnica conocida para describir el comportamiento de un sistema. Describen todos los estados posibles en los que puede entrar un objeto particular y la manera en que cambia el estado del objeto, como resultado de los eventos que llegan a él.

En la mayor parte de las técnicas Orientadas a Objetos, los diagramas de estado se dibujan para una sola clase, mostrando el comportamiento de un solo objeto durante todo su ciclo de vida. En la *figura No. 2.45* se ilustra un ejemplo del diagrama de estado.

El estado en el que se encuentra un objeto determina su comportamiento. Cada objeto sigue el comportamiento descrito en el Diagrama de Estados asociado a su clase. Los Diagramas de Estados y escenarios son complementarios, los Diagramas de Estados son autómatas jerárquicos que permiten expresar concurrencia, sincronización y jerarquías de objetos, son grafos dirigidos y deterministas.

La transición entre estados es instantánea y se debe a la ocurrencia de un evento.



Figura No. 2.45: UML - Diagrama de Estados.

Fuente: http://www.milestone.com.mx/articulos/la_vida_de_un_objeto_diagrama_de_estados.htm

CAPÍTULO 3

3. DESARROLLO DE LA APLICACIÓN

3.1. METODOLOGÍA DE DESARROLLO

La Metodología utilizada para el análisis y desarrollo de los servicios de comunicación es RUP (Proceso Unificado de Desarrollo). RUP describe de manera efectiva el uso de las reglas de negocio y procedimientos probados para el desarrollo de software, conocidos como “mejores prácticas”.

RUP captura varias de las mejores prácticas en el desarrollo moderno de software para un amplio rango de proyectos y organizaciones. Además es una guía para el uso efectivo de UML, proporcionando a cada miembro del equipo de una base de conocimiento como guías, plantillas y herramientas para todas las actividades del desarrollo. (Brito, 2009).

Esta metodología permitió definir el análisis, diseño, implementación y documentación del sistema orientado a objetos, permitiendo llevar un orden correcto de ejecución de cada una de las etapas o fases de vida del proyecto.

En conjunto a esta metodología se implementó el uso del lenguaje Unificado de Modelado (UML) bajo la herramienta de desarrollo Enterprise Architect, la misma que tiene como propósito la de representar de manera gráfica los elementos necesario para el modelado del proyecto.

"Los modelos sirven para captar y enumerar exhaustivamente los requisitos y el dominio del conocimiento, de forma que todos los implicados puedan entenderlos y puedan estar de acuerdo con ellos." (James Rumbaugh, 2000).

Varios de estos modelos aportaron diferentes vistas para el desarrollo del proyecto, permitiendo comprenderlo desde varios puntos de referencia, para esto UML recomienda la utilización de diagramas que permitan crear un

lenguaje comprensible para los desarrolladores como son: los diagramas de casos de uso con sus correspondientes escenarios, excepciones y caminos alternos, los diagramas de actividad, los diagramas de estado, los diagramas de actividades.

"Los modelos sirven para captar y enumerar exhaustivamente los requisitos y el dominio del conocimiento, de forma que todos los implicados puedan entenderlos y puedan estar de acuerdo con ellos." (James Rumbaugh, 2000).

3.2. FASE DE INICIACIÓN

En esta fase se centra en el inicio del proyecto, con la finalidad de generar las actividades de modelamiento y requerimientos basado en la recopilación de datos, procesos, documentos y demás información relevante para la construcción del plan de trabajo que garantice la ejecución del proyecto de forma exitosa, logrando los objetivos propuestos.

3.2.1 Alcance Del Proyecto

El presente proyecto tiene como objetivo la elaboración del servicio de comunicación instantánea en modo de prototipo para la Universidad de las Fuerzas Armadas ESPE, resaltando la creación de los servicios de la entrega y recepción de las diferentes gestiones académicas por medio de mensajes SMS, correos y redes sociales, así como el registro de anuncios académicos y de la suscripción al sistema.

Este alcance tiene la concepción de mejorar los procesos de consulta y entrega de la información académica, basado en los siguientes procesos:

3.2.1.1 Autenticación y Autorización al Servicio de Comunicación.

Elaborar la autenticación al Servicio de Comunicación TREEONE mediante una pantalla de autenticación, el cual deberá manejar 2 tipos de validaciones de acceso:

- a) Para el **Usuario Administrador propio** del Servicio de Comunicación, quien tendrá el acceso al rol administrativo y de configuración, además almacenará una contraseña propia del servicio.
- b) Para los **Usuarios Suscriptores**, quienes serán los estudiantes, docentes y personal administrativo de la ESPE. Estos usuarios del sistema Banner, estarán almacenados en una base de datos de pruebas y serán obtenidos o consumidos mediante Servicios Web.

3.2.1.2 Gestión de Suscripción Académica

Para este proceso, se debe obtener la información del usuario registrado en la base de datos de prueba Banner, esto una vez validado su existencia mediante un Servicio Web.

El Servicio de Comunicación deberá manejar las siguientes funcionalidades de creación de suscriptores:

- ✓ Pantalla para la presentación de la Suscripción al Servicio de Comunicación.
- ✓ Dialogo para el registro de los Contactos del Suscriptor.
 - ✓ Para el registro de la cuenta de correo ESPE, esta se cargará con la información que se obtenga de la base de datos de prueba Banner, mediante un Servicio Web.
 - ✓ Validar que los datos de contactos registrados, no deben estar ya registrados o asociado a un suscriptor activo (número móvil, cuenta correo ESPE, cuenta correo Personal, cuenta Twitter, cuenta Facebook).
 - ✓ Validar el estándar para el registro de las cuentas de correo.
- ✓ Permitir la creación automática de los permisos al Servicio de Comunicación según el rol obtenido por el Servicio Web.
- ✓ Permitir enviar un correo electrónico de tipo informativo por la suscripción al Servicio de Comunicación TREEONE. Se deberá enviar al correo ESPE del personal académico.

3.2.1.3 Gestión Administración del Servicio de Comunicación

El Servicio de Comunicación TREEONE manejará las siguientes funcionalidades administrativas, como:

- Permitir configurar los diferentes **Parámetros** del Servicio de Comunicación.
- Permitir configurar, activar y desactivar el dispositivo **Módem**.
- Permitir la creación y eliminación del permiso **Administrativo** para los suscriptores.
- Permitir la administración de **Periodos Académicos**.
- Permitir la consulta de todos los **Mensajes SMS** que reciben y envían los estudiantes suscriptos en el Servicio de Comunicación.
- Permitir la consulta de todos los **Anuncios Académicos** que reciben los estudiantes, docentes y personal administrativo suscriptos en el Servicio de Comunicación.

3.2.1.4 Gestión Notificaciones Salientes

En este módulo se deberá notificar al estudiante sobre las gestiones académicas registradas o actualizadas en la base de datos de pruebas Banner, entre los procesos a seguir serían los siguientes:

- Permitir el envío de mensajes móviles SMS, correo electrónico y mensaje red social sobre el registro de Notas Académicas.
- Permitir el envío de mensajes móviles SMS, correo electrónico y mensaje red social sobre el registro de Impedimentos Académicos.
- Permitir el envío de mensajes móviles SMS, correo electrónico y mensaje red social sobre el registro de Cuentas Pendientes.

3.2.1.5 Gestión Notificaciones Entrantes

En este módulo se deberá notificar al estudiante sobre las gestiones académicas consultadas por medio del envío de un mensaje móvil SMS, entre los procesos a seguir serían los siguientes:

- a) Permitir la consulta vía mensaje móvil SMS de las gestiones académicas (Notas, Impedimentos y Pagos Pendientes).
 - Se deberá enviar un mensaje de texto o SMS al número configurado.
 - El contenido o código de consulta deberá estar configurado como: para recibir la Notas Académicas de todas sus materias, se deberá mandar la palabra “notas”, para recibir los Impedimentos Académicos, se deberá mandar la palabra “impedimentos” y para recibir los valores pendientes de pago, se deberá mandar la palabra “cuentas”.
- b) Permitir la recepción del mensaje mediante un código de consulta.
 - El Servicio de Comunicación, deberá validar que el número del originador del mensaje de texto, este asociado a un suscriptor activo en el Servicio de Comunicación.
- c) Permitir leer e identificar la consulta realizada.
 - El Servicio de Comunicación por medio de Servicios Web, deberá obtener la información en la base de datos de pruebas Banner.
- d) Permitir enviar el resultado de consulta realizada.
 - Permitir el envío de la información consultada por medio de un mensaje de texto o SMS, correo electrónico y mensaje red social.

3.2.1.6 Anuncios Académicos

En este módulo se deberá proporcionar el servicio de anuncios académicos, que permitirá enviar notificaciones a todos los suscriptores

registrados en el Servicio de Comunicación. Los suscriptores que dispondrán de este módulo son:

- ✓ Personal.
- ✓ Docente.

Para el registro de anuncios por parte del Docente, estos permitirán enviar comunicados a todos sus estudiantes suscriptos en el Servicio de Comunicación.

Para el registro de anuncios por parte del Personal, estos permitirán enviar comunicados a todo el personal suscripto en el Servicio de Comunicación.

Para la notificación, se utilizarán los siguientes medios de comunicación:

- Correos electrónico que son: Correo Personal, Correo ESPE.
- Redes Sociales (correo Facebook y cuenta Twitter).

La información suministrada, será proporcionada por la base de datos de pruebas Banner, con el apoyo y auspicio de la Unidad de Tecnología de la Información y Comunicación (TIC's).

3.2.2 Definición de Los Riesgos

Se puede identificar el siguiente riesgo que puede afectar o demorar el desarrollo de proyecto de tesis, entre ellos se tiene lo siguiente:

a) Inaccessibilidad a las fuentes información del Sistema Banner.

No se puede tener acceso a la información del sistema Banner debido a la restricción que tiene el ambiente de producción por seguridad de la integración de sus datos. Esto puede provocar demoras en la evolución del proyecto, ya que los requerimientos para la realización del proyecto no podrán ser bien especificados.

Magnitud:

La magnitud que tendría sería alta ya que complicaría el desarrollo del producto de software.

Impacto:

El riesgo afecta, en mayor grado al desarrollo y a satisfacer las necesidades del cliente ya que no se podrá gestionar con los valores actuales del Sistema Banner. Además no se podrá gestionar los consumos de los servicios web solicitados.

Tratamiento:

Este riesgo será transferido para lo cual se realizan las siguientes actividades:

- a) Solicitar el certificado o justificativo de las razones por la cual no se puede tener acceso a la información del Sistema Banner Mi ESPE.
- b) Solicitar el acta de entrega y recepción del prototipo, justificando el uso de una base de pruebas para la construcción del mismo.
- c) Explicar el impacto que se presentaría por la presencia de este riesgo.
- d) Informar que la responsabilidad del riesgo queda a cargo de la organización, y que los desarrolladores del proyecto se liberan de cualquier responsabilidad.
- e) De la implementación del proyecto se deberá cambiar a un proyecto de tipo prototipo, utilizando una base de datos de pruebas Banner y Servicios Web que permita simular el acceso a la información del Sistema Banner Mi ESPE. Esto con el fin de cumplir con la ejecución del proyecto de tesis.

3.2.3 Especificación de requerimientos de Software estándar IEEE 830.

La especificación de requerimientos de software (SRS) surge como un conjunto de información unificada y detallada sobre los diferentes requisitos

de software que debe cumplir el Servicio de Comunicación TREEONE, como Prototipo.

Este documento precisa con exactitud las funciones, capacidad y restricciones del software, ya que comprende la totalidad de los requerimientos sin hacer uso de presunción alguna de parte del cliente y de los desarrolladores, si cualquier requerimiento funcional o no funcional no es identificado en el documento, (Méndez, 2008).

El objetivo principal del Servicio de Comunicación TREEONE, es apoyar en los diferentes procesos de comunicación que brinda el Sistema Banner Mi ESPE hacia su comunidad universitaria, entre ellos está la entrega y consulta de las gestiones académicas de los estudiantes, sean estas: Notas Académicas, impedimentos y valores pendiente de cobro del estudiante, estos mediante los siguientes medios de comunicación: mensajería celular SMS, correos ESPE, correo personal y cuentas sociales, así mismo permitirá al docente y al personal administrativo, el envío de anuncios académicos por los siguientes medios de comunicación como: mensajería de redes sociales y correos electrónicos.

El Servicio a desarrollar tomará la información de las gestiones académicas (Notas, Impedimentos y estados de Cuentas) registradas en la base de datos pruebas Banner, por medio de Servicios Web.

Esta estrategia de la especificación de requerimiento cumple con las directrices establecidas por el estándar “IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830”.

3.2.3.1 Propósito

El objetivo de esta especificación consiste en definir de manera clara y precisa todas las funcionalidades y restricciones que se necesitan para la implementación de una aplicación WEB que permitirá administrar, consultar

y comunicar la información de las gestiones académicas del Sistema Banner Mi ESPE.

3.2.3.2 **Ámbito del Sistema**

Diseño de un prototipo para el servicio interactivo de comunicación instantánea para la Universidad de las Fuerzas Armadas ESPE. El servicio funcionará en un entorno WEB y permitirá dar apoyo a los siguientes procesos:

- **Gestión de Autenticación y Autorización**, permite el acceso al Servicio de Comunicación.
- **Gestión de Administración**, permite la configuración y administración del Servicio de Comunicación.
- **Gestión de Suscripción**, permite la suscripción al Servicio de Comunicación.
- **Gestión de Contactos**, permite el registro de la información de contacto del personal suscripto en el Servicio de Comunicación.
- **Gestión Automática de Notificaciones Salientes**, permite el envío de las gestiones académicas registradas en el ambiente de pruebas Banner, por medio de mensajes de texto o SMS, correos y cuentas sociales.

Este proceso obtendrá la información de las gestiones académicas registradas en el ambiente de pruebas banner mediante servicios web, entre las gestiones académicas se indican las siguientes:

- Notas Académicas.
 - Impedimentos Académicos.
 - Valores o cuentas pendientes de cobro.
-
- **Gestión de Notificaciones Entrantes**, permite consultar las gestiones académicas registradas en el ambiente de pruebas Banner, mediante el envío de un mensaje de texto o SMS de consulta.

Este proceso obtendrá la información de la gestión académica consultada por el suscriptor, entre ellas tenemos:

- Notas Académicas.
 - Impedimentos, del periodo presente.
 - Valores pendientes de pago, del periodo presente.
- **Gestión de Anuncios Académicos**, permite crear anuncios académicos y notificarlos por:
 - ✓ Correo Electrónico.
 - ✓ Mensaje Twitter.
 - ✓ Correo Facebook.
 - **Gestión de Reportes**, permite generar y exportarlos a Excel y PDF, entre ellos se tienen los reportes de:
 - ✓ Anuncios Académicos.
 - ✓ Mensajes Académicos.
 - ✓ Contactos.
 - **Características Principales**, para el correcto funcionamiento del servicio de comunicación, se debe contar con las siguientes características:
 - ✓ El Servicio permitirá el registro y mantenimiento de la información de contactos de los suscriptores estudiantes, docentes y personal administrativo que opten por este servicio comunicativo.
 - ✓ Para el envío de los mensajes de texto o SMS, se utiliza un dispositivo de comunicación como lo es el MÓDEM GSM o 3G con chip o sim card de cualquier operadora de telefonía móvil.
 - ✓ El chip deberá tener la característica de poder enviar y recibir mensajes de texto o SMS.

- ✓ Los mensajes de texto o SMS tienen un costo envío, para lo cual la ESPE deberá realizar un convenio con las operadoras de telefonía móvil para la adquisición de paquetes mensajería "SMS".
- ✓ Los mensajes de texto recibos no tendrán costo para el suscriptor.
- ✓ Los mensajes de texto enviados por el suscriptor tendrán costo.
- ✓ Los mensajes por correo electrónico, por Facebook y por Twitter, no tendrán costos.
- ✓ En caso de utilizar cantidades mayores de 5000 mensajes por Twitter, este tendrá un costo por su uso.
- ✓ Para el servicio de mensajería móvil SMS, se trabajará con las operadoras de telefonía celular: CNT, Claro y Movistar.
- ✓ Para los servicios de mensajería por redes sociales, se trabajarán con las redes Facebook (opcional) y Twitter (opcional).
- ✓ Para los servicios de correos electrónicos, se trabajarán con el correo ESPE (obligatorio) y con el correo personal del suscriptor (obligatorio).
- ✓ Los mensaje de texto o SMS, manejarán un máximo de 150 caracteres.
- ✓ Los mensajes por correo electrónico y por redes sociales, manejarán un máximo de 140 caracteres.
- ✓ La cuenta de Twitter deberá ser registrada sin el signo @.
- ✓ El suscriptor deberá seguir al usuario Twitter creado para la ESPE, con esta configuración el suscriptor puede recibir los anuncios académicos.
- ✓ La cuenta de Facebook deberá ser registrada con el estándar nayo.molina@facebook.com
- ✓ Los mensajes por Facebook, serán por medio del correo registrado en la cuenta, debido a que Facebook cerró el servicio de mensajes directos por asuntos de seguridad.

3.2.3.3 Definiciones, acrónimos y abreviaturas

a) *Definiciones del Sistema*

Administrador del Sistema: Persona encargada de ofrecer el soporte técnico y operativo al Sistema.

Comunidad social.- Permite una mejor socialización y participación de los estudiantes en las redes sociales de la ESPE.

Correo electrónico, herramienta institucional que permite integrar una comunicación con otros miembros de la comunidad universitaria.

Estudiante: Persona que está cursando estudios para un carrera de grado.

Docente: Persona que enseña una determinada ciencia o arte.

Personal Universitario: Personal que labora en la Universidad de las Fuerzas Armadas ESPE.

Suscriptor: Personal Universitario que se ha suscripto al servicio de comunicación.

Suscripción: Proceso por la cual el Personal Universitarios se suscribe en el Sistema de comunicación TREEONE para tener el beneficio de las gestiones académicas.

Gestión Académica: Servicios de Información Académica que permite controlar desde la matrícula de estudiantes, registro de profesores, entrega de notas académicas (calificaciones), entrega de impedimentos, entrega de valores pendientes de pago, ingreso de tareas, reportes y estadísticas, asistencia, horarios y una larga lista de características.

Mensajería Celular (SMS): es un servicio que permite el envío y recepción de mensajes de texto en teléfonos celulares.

Notificación Entrante: Proceso en el que cual se entrega o se notifica las gestiones académicas al personal Académico por medio de un mensaje por SMS, Red Social o Correo Electrónico.

Notificación Saliente: Proceso que permite la consulta de las gestiones académicas por medio de un mensaje móvil SMS.

Anuncios Académicos: Proceso que permite el registro de anuncios académicos por medio de correos electrónicos o por medio de las redes sociales.

Permiso: Parámetro que especifica si su poseedor dispone de acceso a una determinada función del Servicio de Comunicación,

Usuario: Nombre o alias que se le da a una persona para permitirle el acceso al Sistema, en este caso son los usuarios de la base de pruebas Banner.

TREEONE: Servicio de Comunicación Instantánea para la Universidad de las Fuerzas Armadas ESPE.

b) Definiciones de tecnología

AJAX (Asynchronous JavaScript + XML): Java Script asíncrono + XML, Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes. Es una técnica de desarrollo web para crear aplicaciones interactivas.

Aplicación: Es un programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo.

Base de Datos: Cualquier conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar. La información se organiza en campos y registros. Un campo

se refiere a un tipo o atributo de información, y un registro, a toda la información sobre un individuo.

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.

Encriptar: Ocultar el contenido de un mensaje mediante un proceso de codificación de manera tal que el acceso al contenido del mensaje sea restringido.

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.

HTTP: El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW).

Internet: Es un método de interconexión descentralizada de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP.

Java Script: Java Script es un lenguaje interpretado en el cliente por el navegador al momento de cargarse la página, es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento HTML.

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

Postgresql: Lenguaje de consulta estructurado, en informática, un lenguaje utilizado en bases de datos para consultar, actualizar y manejar bases de datos relacionales.

RIA (Rich Internet Applications): Aplicaciones ricas en Internet, 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.

Spring Framework: Es un Framework de código abierto de desarrollo de aplicaciones para la plataforma Java.

URL (Uniform Resource Locator): 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.

c) Acrónimos

ESPE: Universidad de las Fuerzas Armadas ESPE.

URL: Uniform Resource Locator, esquema utilizado para la localización de un recurso en Internet.

SMS: Short Message Service, Servicio de Mensajes cortos.

DBA: Es aquella persona que tiene el control central del Sistema de base de datos.

ERS: Acrónimo de Software Requirements Specifications (Especificación de Requerimientos de Software).

URL: Un URL es una cadena de caracteres que identifica el tipo de documento, la computadora, el directorio y los subdirectorios en donde se encuentra el documento y su nombre.

d) Abreviaturas

HW: Hardware.

SW: Software.

3.2.3.4 Referencias

IEEE Recommended Practice for Software Requirements Specification.
ANSI/IEEE std. 830, 1998.

3.2.3.5 Visión general

El documento respeta la estructura recomendada por el estándar “IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830-1998, este consta de tres secciones.

La Primera Sección se encuentra la Introducción, es aquella que proporciona y se detalla los objetivos con la visión general de la ERS.

En la Segunda Sección 2 se da una descripción general del Servicio de Comunicación, en donde se describe la perspectiva general del producto a desarrollarse, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo,

En la Tercera Sección se definen detalladamente los requisitos que debe satisfacer el Servicio de Comunicación.

3.2.3.6 Descripción General

En esta sección se realiza una descripción a alto nivel, focalizándose en las funciones que realizará, así como en aquellas restricciones generales que afectan al mismo.

El Servicio de Comunicación constará de dos aplicaciones, una FRONT que permitirá la suscripción del personal universitario, también existirá una

aplicación BACK para los procesos de entrega de información hacia el personal universitario.

3.2.3.7 Perspectiva del producto

El Servicio de Comunicación será un producto diseñado para trabajar en entornos WEB, lo que permitirá su utilización de forma descentralizada, además trabajará de manera independiente, por lo que únicamente obtendrá la información de las gestiones académicas del personal registrado en la base de pruebas Banner.

El Servicio de Comunicación interaccionará con cuatro productos software, dichos productos serán los siguientes:

- Un servidor de Aplicaciones WEB. No existe ninguna restricción acerca de qué servidor se debe utilizar, por tal motivo se utiliza el servidor JBOSS.
- Software de acceso a redes TCP/IP. El protocolo de comunicaciones utilizado por el Servicio será TCP/IP, SMTP.
- Base de datos. La única exigencia será la utilización de una Base de Datos Relacional, por lo demás, el tipo de base de datos y el mecanismo de comunicación con la misma (ODBC, JDBC, interfaz propietario, etc.) se utilizará al Postgresql como la base de datos relacional.
- El protocolo utilizado para la transmisión de datos entre redes de computadoras es el protocolo TCP/IP. Además, el protocolo http será el que permita el acceso a la página Web.

3.2.3.8 Funciones del Servicio de Comunicación

El Sistema permitirá realizar las siguientes funciones:

A. Funciones Automáticas del Servicio

Se presentan las siguientes funcionalidades automáticas:

i. Servicio Web

Se creará un Servicio Web **BannerServicio** para consultar los usuarios registrados en una base de pruebas en MYSQL.

Este servicio web, tiene como fin el simular la consulta del usuario del sistema académico Banner.

Este Servicio Web permitirá realizar las siguientes consultas:

- **BannerServicio.buscarPorClaveCodigo**, permite consultar si el usuario se encuentra en la base de pruebas mediante los parámetros CLAVE y CODIGO.

BannerServicio.buscarPorClaveCodigo, retorna un objeto con los siguientes campos: nombres, apellidos, identificación, codigo-_banner, rol, clave, periodo, vigencia y correo.

- **BannerServicio.buscarPorCodigo**, permite consultar si el usuario se encuentra en la base de pruebas mediante el parámetro CODIGO.

BannerServicio.buscarPorCodigo, retorna un objeto con los siguientes campos: nombres, apellidos, identificación, codigo-_banner, rol, clave, periodo, vigencia y correo.

- **BannerServicio.obtenerDocentesPorPeriodo**, permite obtener los docentes por Periodo.

Recibe: codigoPeriodo.

Retorna: listaDocentesPorPeriodo.

- **BannerServicio.obtenerMateriasPorDocentesPeriodo**, permite obtener las materias Docente y por Periodo.

Recibe: codigoDocente, codigoPeriodo.

Retorna: listaMateriasPorDocentePeriodo.

- **BannerServicio.obtenerParalelosPorDocenteMateriaPeriodo**, permite obtener los paralelos o NRC's por Docente, por Materia y por Periodo.
Recibe: codigoDocente, codigoMateria, codigoPeriodo.
Retorna: listaParalelosPorDocenteMateriaPeriodo.
- **BannerServicio.obtenerAlumnoPorDocenteMateriaParaleloPeriodo**, permite obtener los alumnos por Docente, por Materia, por NRC y por Periodo.
Recibe: codigoDocente, codigoMateria, codigoParalelo, codigoPeriodo.
Retorna: listaAlumnosPorDocenteMateriaParaleloPeriodo.
- **BannerServicio. actualizarNotasPorAlumnoMateria**, permite actualizar las notas registradas.
Recibe: codigoAlumno, codigoMateria, codigoPeriodo, nota1, nota2, nota3.
Retorna: N/A.
- **BannerServicio.consultarNotasPorAlumno**, permite consultar las Notas Académicas registradas en la base de pruebas Banner.
Recibe: codigoAlumno, codigoPeriodo.
Retorna: alumnoNotas.
- **BannerServicio.consultarImpedimentosPorAlumno**, permite obtener a los impedimentos académicos registradas en la base de pruebas Banner.
Recibe: codigoAlumno, codigoPeriodo.
Retorna: alumnoImpedimentos.
- **BannerServicio.consultarCuentasPorAlumno**, permite obtener a las cuentas o pagos pendientes registrados en la base de pruebas Banner.

Recibe: *codigoAlumno, codigoPeriodo.*

Retorna: *alumnoCuentas.*

- **BannerServicio. obtenerNotificacionesRegistradas,** permite obtener a las notificaciones entrantes con el estado Registrado "R".

Recibe: *N/A.*

Retorna: *getListaNotificaciones.*

- **BannerServicio.actualizarNotificacionesRegistradas,** permite actualizar las notificaciones entrantes con el estado Entregado "E".

Recibe: *listaNotificacionesEspe.*

Retorna: *getListaNotificaciones.*

ii. Timer

Permite la ejecución automática de los envíos de mensajes de texto o SMS's, correo electrónico y mensaje red social de las Notificaciones Salientes. Este proceso se ejecuta cada cierto periodo de tiempo y consultará la existencia de notificaciones pendientes de envío de la base de datos de pruebas Banner, mediante un Servicio Web.

iii. Gestión Notificaciones Salientes

Este módulo permite gestionar el proceso de notificación vía mensaje de texto o SMS, correo electrónico y mensaje red social, de las diferentes gestiones académicas registradas en la base de datos de pruebas Banner. Estas gestiones se basan en:

- Registro y Modificación de Notas Académicas.
- Registro y Modificación de Impedimentos Académicos.
- Registro y Modificación de Valores Pendientes de Pago.

Para notificar, el Servicio de Comunicación consultará vía Servicio Web a todas las notificaciones que se encuentran en la tabla **notificaciones_espe** de la base de datos de pruebas Banner.

iv. Gestión Notificaciones Entrantes

Este módulo permite gestionar el proceso de notificación de las gestiones académicas consultada por vía mensaje de texto o SMS. Estas gestiones se basan en:

- Consulta de Notas Académicas.
- Consulta de Impedimentos Académicos.
- Consulta de Valores Pendientes de Pago.

Para notificar, el Servicio de Comunicación consultará vía Servicio Web la gestión consultada sobre las tablas **notas_espe**, **cuentas_espe** o **impedimentos_espe**, todas de la base de datos de pruebas Banner

B. Funciones de Administración de Catálogos

- Gestión de Menús. Permite gestionar la edición de menús o ejecutables del Servicio de Comunicación. Esta gestión tendrá las siguientes opciones: modificar y listar.
- Gestión de Roles. Permite gestionar la creación de Roles, esta gestión tendrá las siguientes opciones: agregar, eliminar y listar.
- Gestión de Parámetros. Permite gestionar la creación de parámetros y esta gestión tendrá las siguientes opciones: modificar, y listar.
- Gestión de Periodos. Permite gestionar la creación de periodos y esta gestión tendrá las siguientes opciones: agregar, eliminar y listar.
- Conexión Módems. Permite la configuración de los valores de conexión, además permite activar y desactivar el Módem.

- Creación Suscriptor Administrativo. Permite otorgar permisos de Administrador a un suscriptor.

C. Funciones de Consultas de Información

- Consulta de Materias. Proceso que permitirá al suscriptor estudiante o docente, verificar las materias asignadas en del presente periodo.
- Consulta de Mensajes. Proceso que permitirá al suscriptor estudiante verificar los mensajes recibidos.
- Consulta de Anuncios. Proceso que permitirá al suscriptor estudiante, docente y personal verificar los mensajes y anuncios enviados y recibidos.
- Sistema de ayudas, Instructivo del uso del servicio para los diferentes suscriptores del Servicio de Comunicación.

D. Funciones de Gestión de Procesos

i. Autenticación y Autorización al Servicio de comunicación

El acceso al servicio de comunicación TREEONE, será mediante la pantalla de autenticación, el cual dispondrá de 2 tipos de validaciones:

- Para el usuario administrador del Servicio de Comunicación, quien tendrán acceso al rol administrativo, y además cuenta con una contraseña encriptada para su acceso al Servicio. Este usuario no será validado por el servicio web y se manejará con una clave que será entregada en la documentación.
- Para los usuarios suscriptores, quienes serán los estudiantes, docentes y personal administrativo de la ESPE, estos usuarios suscriptores serán validados por el servicio web BannerServicio. El Servicio Web BannerServicio, verificará si el usuario que se está autenticando en el Servicio de Comunicación, se encuentre registrado en la base de datos de pruebas de Mysql, esto con el fin

de poder simular al proceso de verificación de usuarios del sistema de gestión académica "Banner".

- El Servicio de Comunicación TREEONE no proveerá de claves para el ingreso, solamente accederán los usuarios que se encuentren en la base de datos de prueba en MySql

ii. Anuncios Académicos

Este módulo proporcionará un servicio de anuncios académicos, que permitirá enviar notificaciones a todos los usuarios registrados en el Servicio de Comunicación. Los usuarios que dispondrán de este módulo serán:

- Personal
- Docente

Para el registro de **anuncios por parte del Docente**, estos permitirán enviar comunicados a todos sus estudiantes suscriptos en el Servicio de Comunicación TREEONE por periodo Universitario.

Para el registro de **anuncios por parte del Personal**, estos permitirán enviar comunicados a todo el personal suscriptos en el Servicio de Comunicación TREEONE por periodo Universitario.

Para la notificación se utilizarán los siguientes medios de comunicación:

- Correos electrónico que son: Correo Personal, Correo ESPE.
- Redes Sociales (correo Facebook y cuenta Twitter).

iii. Gestión de Suscripción Académica

Proceso que permitirá al personal académico suscribirse en el Servicios Interactivo de Comunicación Académica.

El Servicio de Comunicación manejará las siguientes funcionalidades de creación de suscriptores:

- Registro de los contactos personales del suscriptor, sean estos contactos el número celular, las cuentas de redes sociales y las cuentas de correo electrónico.
- Creación automática de los suscriptores en el Servicio de Comunicación TREEONE con los contactos registrados.
- Creación automática de los accesos en el Sistema TREEONE según el rol obtenido del **Servicio Web BannerServicio**.
- Envío de un correo electrónico informativo por la suscripción al Servicio de Comunicación TREEONE. Se enviará al correo personal ESPE.

iv. Gestión de Contactos.

Permite gestionar la creación y edición de contactos.

v. Gestión de cancelación de la suscripción.

Permite cancelar la suscripción.

E. Características de Roles y Usuarios

i. Características de los Roles

El Servicio de Comunicación maneja 4 tipos de roles, entre los cuales se tienen:

1. Rol Administrador General del Servicio de Comunicación ("ADM_GEN").
2. Rol Estudiante Suscripto ("STUDENT").
3. Rol Docente Suscripto ("FACULTY").
4. Rol Personal Administrativo Suscripto ("EMPLOYED").

El Servicio de Comunicación puede manejar las siguientes combinaciones de roles:

- a. Rol Administrador del Servicio de Comunicación ("ADM_GEN") + Rol Estudiante Suscripto ("STUDENT").
- b. Rol Administrador del Servicio de Comunicación ("ADM_GEN") + Rol Docente Suscripto ("FACULTY").
- c. Rol Administrador Configurador del Servicio de Comunicación ("ADM_GEN") + Usuario Rol Personal Administrativo Suscripto ("EMPLOYED").
- d. Rol Docente Suscripto ("FACULTY") + Rol Personal Administrativo Suscripto ("EMPLOYED").
- e. Rol Administrador Configurador del Servicio de Comunicación ("ADM_CONF") + Rol Docente Suscripto ("FACULTY") + Rol Personal Administrativo Suscripto ("EMPLOYED").

ii. Rol Administrador General del Servicio de Comunicación ("ADM_GEN").

El rol indicado tendrá las siguientes opciones:

- Configuración de los diferentes parámetros.
- Configuración, Activación y Desactivación del Módem.
- Creación y Eliminación del Permiso Administrativo para los suscriptores del Servicio de Comunicación.
- Administración de Periodos Académicos.
- Cambio de Contraseña.
- Consulta de todos los **Mensajes de texto o SMS** que reciben y envían los estudiantes suscriptos en el Servicio de Comunicación.
- Consulta de todos los **Anuncios Académicos** que reciben los estudiantes, los docentes y el personal administrativo suscriptos en el Servicio de Comunicación.

iii. Rol Estudiante Suscriptor ("STUDENT")

El rol indicado tendrá las siguientes opciones:

- Suscripción al Servicio de Comunicación
- Cancelación al Servicio de Comunicación.
- Administración de Contactos.
- Instructivo del Servicio de Comunicación.
- Consulta de Materias Asignadas.
- Consulta de Mensajes Recibidos.
- Consulta de los Mensajes de texto o SMS Recibidos.

iv. Rol Docente Suscriptor ("FACULTY")

El rol indicado tendrá las siguientes opciones:

- Suscripción al Servicio de Comunicación
- Cancelación al Servicio de Comunicación.
- Administración de Contactos.
- Registro de Anuncios Académicos
- Instructivo del Servicio de Comunicación.
- Consulta de los Anuncios Académicos Enviados y Recibidos.
- Consulta de Materias Asignadas.

v. Rol Personal Administrativo Suscriptor ("EMPLOYED")

El rol indicado tendrá las siguientes opciones:

- Suscripción al Servicio de Comunicación
- Cancelación al Servicio de Comunicación.
- Administración de Contactos.
- Registro de Anuncios Académicos
- Consulta de los Anuncios Académicos Enviados y Recibidos.
- Instructivo del Servicio de Comunicación.

vi. Características del Usuario

La interface de usuario es de manejo intuitivo, fácil de aprender y sencillo de manejar, con el fin de familiarizar a los suscriptores de una manera rápida y siendo de uso inmediato para las funciones dentro del Servicio de Comunicación.

Cada suscriptor tendrá un perfil específico para que su interacción con el Servicio de Comunicación sea correcta y no conlleve a fallos.

- ***Usuario Administrador General del Servicio de Comunicación***

El usuario "**ADMGENSISCOM**" administrador general del Servicio de Comunicación, será el encargado de manejar el Servicio de Comunicación con gran responsabilidad sobre los criterios de permisos que tiene asignado.

Este usuario tendrá acceso al ROL: *Rol Administrador General del Servicio de Comunicación ("ADM GEN")*

- ***Usuario Suscriptor Estudiante***

Persona que interactuará continuamente con el Servicio de Comunicación para lo cual debe tener un conocimiento básico sobre navegación en páginas Web.

Este usuario tiene acceso al ROL: *Rol Estudiante Suscripto ("STUDENT")*

- ***Usuario Suscriptor Docente***

Persona que interactúa continuamente con el Servicio de Comunicación para lo cual debe tener un conocimiento básico sobre navegación en páginas Web.

Este usuario tiene acceso al ROL: Rol Docente Suscripto ("FACULTY").

- **Usuario Suscriptor Personal Administrativos**

Persona que interactúa continuamente con el Servicio de Comunicación, para lo cual debe tener un conocimiento básico sobre navegación en páginas Web.

Este usuario tiene acceso al ROL: Rol Personal Administrativo Suscripto ("EMPLOYED").

F. Funciones de Auditoria

Gestión que permite guardar los eventos de creación y actualización de registros, como:

- Usuario creador del registro: Se registrará id banner del suscriptor creador del registro.
- Usuario actualizador del registro: Se registrará id banner del suscriptor actualizador del registro.
- Fecha de creación del registro: Se registrará la fecha de creación del registro.
- Fecha de actualización del registro: Se registrará la fecha de actualización del registro.

Los registros eliminados serán de modo lógico y no de modo físico, actualizando el campo vigente a FALSE.

3.2.3.9 Políticas de la empresa (reguladoras)

La aplicación se desarrollará mediante software de licencia abierta por lo tanto no se deberá pagar por el uso de: servidor WEB (JBoss), Sistema de Gestión de base de datos (POSTGRESQL) y el lenguaje de programación (JSF, PRIMEFACE, JAVA).

3.2.3.10 Limitaciones de hardware

Las características del hardware en los equipos donde se instalará el Servicio de Comunicación, serán las mismas en donde se encuentre el Sistema Banner.

Para esta aplicación será necesario instalará el servidor JBOSS, la base de datos POSTGRESQL y la Aplicación TREEONE.

3.2.3.11 Interfaces con otras aplicaciones

Debido a que el Servicio de Comunicación interactuará con una base de datos de prueba Banner, se deberá desarrollar conexiones necesarias para la utilización de esta base, con el fin de obtener la información académica de los estudiantes.

Estas conexiones serán por medio de Servicios Web.

3.2.3.12 Funciones de control

El Servicio de Comunicación controlará los permisos que tiene cada usuario para que puedan acceder a la información que le corresponda de acuerdo al rol configurado. Tiene los controles adecuados para la validación de datos, de igual manera la parametrización de los procesos del Servicio de Comunicación.

3.2.3.13 Lenguajes de Programación.

Todo el material que realice el usuario y la aplicación debe de estar en lenguaje español. Aunque también se puede configurar el proceso de internacionalizar el lenguaje del Servicio.

3.2.3.14 Protocolos de Comunicación

Se usará protocolos de comunicación TCP/IP, HTTP y SMTP.

3.2.3.15 Requisitos de fiabilidad

Puesto que para el proceso de suscripción se pretende que tenga un acceso simultáneo de estudiantes, el servidor tiene que ser capaz de proporcionar un acceso concurrente a un número considerable de estudiantes.

3.2.3.16 Credibilidad de la aplicación

Para garantizar una buena credibilidad el Servicio de Comunicación deberá ser sometido a una serie de pruebas para establecer que se encuentra acorde a los requerimientos que se plasman en el documento en tanto a la consistencia de datos como al rendimiento de la aplicación, tales como tiempos de respuesta.

3.2.3.17 Consideraciones de seguridad

El Servicio de Comunicación no proveerá de claves de acceso, ya que estará inmerso en la información de usuarios de la base de datos de pruebas Banner, esto significa que los usuarios para ingresar, primero deberían estar almacenados en la base pruebas.

Si un usuario desearía ingresar al Servicio de Comunicación colocando directamente la dirección de acceso en la barra de direcciones, se le denegará el acceso y se cerrará la sesión.

La clave para el administrador propio del Servicio de Comunicación, se registra de modo encriptada.

3.2.3.18 Suposiciones y Dependencias

A. Suposiciones

Los requisitos descritos son estables ya que fueron aprobados por el equipo de diseño y personal del departamento de la tecnología de la comunicación.

Cualquier petición de cambios en la especificación debe ser aprobada por el equipo de trabajo.

También se debería efectuar la capacitación adecuada y acorde a lo que cada usuario va a realizar.

Su capacitación se hará en el momento que sea necesario, con esto los usuarios se familiarizarán con el funcionamiento del Servicio de Comunicación.

B. Dependencias

El Servicio de Comunicación interactuará con la información de la base de datos de pruebas, especialmente con los usuarios banner y la información de las gestiones académicas.

La red interna deberá estar configurada para el manejo de protocolos TCP/IP, HTTP y SMTP, principalmente todo lo relacionado en cuanto a desempeño y seguridad.

3.2.3.19 Requerimientos Futuros.

La lógica del negocio y las Utilidades del Servicio de Comunicación, pueden ser utilizadas para la creación de otros servicios de comunicación.

3.2.3.20 Requisitos Específicos

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el Servicio de Comunicación.

Todos los requisitos aquí expuestos son esenciales, es decir, no sería aceptable el Servicio de Comunicación que no satisfaga alguno de los requisitos aquí presentados.

3.2.3.21 Interfaces Externas

A. Interfaz con el usuario

Las interfaces de usuario están relacionadas con las páginas que ofrece el Servicio de Comunicación y que debe manipular el suscriptor para realizar una operación determinada.

Esta interfaz tiene un alto grado de usabilidad y sencillez, cumpliendo todos los requisitos especificados, por lo que se dichas interfaces incluirán:

- Botones.
- Menús despegables.
- Mensajes informativos.
- Mensajes de error.
- Cuadros de diálogo.
- Páginas para el ingreso, modificación, actualización, eliminación, búsquedas y listados de datos.

Existen tres interfaces gráficas de usuario web:

Interfaz para el usuario final, está orientada al estudiante, docente y personal administrativo, que se registran o se suscriban en el Servicio de Comunicación, Podrá recibir y consultar las gestiones académicas mediante los medios de comunicación como: mensajería celular, mensajes por redes sociales o mensajes por correo electrónico.

Interfaz de parametrización y configuración, esta interfaz está orientado para el usuario administrativo, el será el encargado de configurar, parametrizar las diferentes funcionalidades del Servicio de Comunicación, así como dar mantenimiento a las diferentes tablas referenciales.

Interfaz de seguridad, interfaz orientada para el suscriptor administrativo encargado de gestionar la modificación de Menús y de dar permisos de administración para el suscriptor.

B. Interfaz con el software

Cualquier usuario que desee conectarse al Servicio de Comunicación le bastará con un navegador moderno para que pueda visualizarla.

Por otra parte, el servidor que hospede la aplicación estará basado en Jboss y la base de datos Postgresql, disponibles para diversos Sistemas operativos: Linux y Windows. En este caso la interfaz funcionará bajo un entorno Windows.

C. Interfaces de hardware

Para acceder al Servicio de Comunicación, los usuarios únicamente necesitarán un ordenador desde el cual, mediante una interfaz tengan acceso a Internet.

Este computador puede estar basado en cualquier arquitectura pero deberá disponer al menos de un procesador con capacidad suficiente para ejecutar un navegador Web moderno.

La pantalla del monitor.- el software deberá mostrar información al usuario a través de la pantalla del monitor.

Ratón.- el software debe interactuar con el movimiento del ratón y sus botones.

Teclado.- el software deberá interactuar con las pulsaciones del teclado. El teclado de entrada de datos en el área activa de la base de datos.

D. Interfaces de Comunicación

Los usuarios suscriptores (Estudiantes, Docentes, Personal Administrativo y Administradores del Servicio de Comunicación) se conectarán mediante Internet. Esta conexión la realizarán desde la Universidad, desde la casa, desde su celular o desde cualquier otro lugar en donde tenga disponibilidad de acceso.

La interfaz de comunicación entre los usuarios y gestiones académicas de pruebas y el Servicio de Comunicación será mediante conexiones vía Servicios Web.

3.2.4 Funciones del Servicio de Comunicación

Entre las principales funcionalidades que ofrece el Servicio de Comunicación TREEONE, se diferencian entre los requisitos funcionales y los requisitos no funcionales.

3.2.4.1 Requisitos Funcionales

Entre los principales requisitos funcionales que manejará el servicio de comunicación TREEONE, se destacan los siguientes:

Acceso al Servicio de Comunicación

Este proceso indica los requerimientos a seguir para ingresar al Servicio de Comunicación TREEONE, mediante los usuarios registrados en la base de datos de pruebas Banner.

El Servicio de Comunicación no proveerá de claves para el ingreso, solamente accederán los usuarios registrados en la base de datos de pruebas Banner. En la *tabla No. 3.1*, se indican los requerimientos para el acceso al Servicio de Comunicación:

Tabla No. 3.1: Requerimientos - Acceso al Servicio de Comunicación.

Requerimientos	Descripción
RQ1. Autenticación Usuario Banner	<i>Proceso que permitirá la autenticación de los usuarios, el servicio de comunicación validará el tipo de usuario registrado y obtendrá la información registrada en la base de datos de pruebas Banner.</i>
RQ2. Validar Usuario Suscrito en TREEONE	<i>Proceso que permitirá validar si el usuario se encuentra suscrito en el servicio de comunicación TREEONE.</i>
RQ3. Obtener Información Usuario Banner	<i>Permite obtener la Información del personal académico suscrito en la base de datos de pruebas Banner, la información a obtener es: cédula, codigo_banner, nombres, apellidos, rol, periodo, vigencia y correo ESPE.</i>

Suscripción

Requerimientos para la suscripción al Servicio de Comunicación por parte del personal universitario de la ESPE. La suscripción se realiza mediante el registro de los contactos personales sean estos: el número celular, las cuentas de redes sociales y las cuentas de correo electrónico. En la *tabla No. 3.2*, se presentan los requerimientos para la suscripción:

Tabla No. 3.2: Requerimientos - Suscripción.

Requerimientos	Descripción
RQ4. Creación Automática del Suscriptor	Proceso que permitirá crear al suscriptor automáticamente con los datos obtenidos de la base de datos de pruebas Banner como: Nombre, Apellido, Rol, Identificación y Código Banner.
RQ5. Creación Automática de Permisos	Proceso que permitirá crear automáticamente los permisos al servicio de comunicación TREEONE.
RQ6. Creación de Contactos	Proceso que permite el registro de la Información de contacto del personal académico suscripto.

Cancelación de la Suscripción

Requerimientos para el proceso de cancelación de la suscripción. En la *tabla No. 3.3* se presentan los requerimientos para la cancelación de la suscripción.

Tabla No. 3.3: Requerimientos - Cancelación de la Suscripción.

Requerimientos	Descripción
RQ7. Actualización Automática del Suscriptor	Proceso que permitirá actualizar al suscriptor con vigencia a FALSE.
RQ8. Actualización Automática de Permisos	Proceso que permitirá actualizar los permisos del suscriptor con vigencia a FALSE
RQ9. Actualización Automática de Permisos	Proceso que permitirá actualizar los contactos del suscriptor con vigencia a FALSE.

Notificación Entrante Gestión Académica

Requerimiento para el proceso de notificaciones de SMS entrantes por la consulta de las Gestiones Académicas.

En la *tabla No. 3.4* se presentan los requerimientos para las notificaciones entrantes.

Tabla No. 3.4: Requerimientos - Notificación Entrante Gestión Académica.

Requerimientos	Descripción
RQ10. Leer SMS	Proceso que permitirá leer los SMS entrantes al servicio de comunicación TREEONE.
RQ11. Validar Número Celular	Proceso que permitirá validar si el número del mensaje SMS entrante, se encuentra registrado a un suscriptor activo en el Servicio de Comunicación TREEONE.
RQ12. Valida Usuario Suscriptor	Proceso que permitirá validar si el suscriptor se encuentra activo en el servicio de comunicación TREEONE.
RQ13. Consultar Petición	Proceso que realizará la consulta por la petición solicitada a través de SMS entrante, retornará las Notas o Impedimentos o Valores Impagos del estudiante.
RQ14. Enviar Mensaje SMS	Proceso que permitirá enviar el mensaje SMS, correos electrónicos, correo Facebook y mensajes Twitter.
RQ15. Guardar Mensajes	Proceso que permitirá guardar el mensaje notificado, sea exitoso o fallido.

Notificación Saliente Gestión Académica

Requerimientos para el proceso de notificaciones Salientes por medios de SMS. En la *tabla No. 3.5* se presentan los requerimientos para la notificación saliente.

Tabla No. 3.5: Requerimientos - Notificación Saliente Gestión Académica.

Requerimientos	Descripción
RQ16. Obtención Notificación Saliente	Proceso o TIMER que estará consultando cada cierto tiempo si existen notificaciones salientes registradas o pendientes de ser enviadas.
RQ17. Enviar Mensaje por Medios Comunicación	Proceso que permitirá enviar la notificación de las gestiones académicas (NOTAS, IMPEDIMENTOS, VALORES PENDIENTES) por SMS, correo electrónico y mensaje red social.
RQ18. Validar Destinatario	Proceso que permitirá validar si el Código Banner de la Notificación Saliente, se encuentra registrado a un suscriptor activo en el Servicio de Comunicación TREEONE.
RQ19. Verificar Configuración de Mensajes Notificación Entrante	Proceso que permitirá verificar las configuraciones y parámetros de envío de los mensajes. Verifica si el servicio SMS se encuentra configurado y activo para ser usado.

Gestión de Anuncios Académicos

Este proceso indica los requerimientos para la generación de anuncios académicos y permitirá enviar comunicados a todos los suscriptores registrados en el Servicio de Comunicación TREEONE. En la *tabla No. 3.6* se presentan los requerimientos para el registro de anuncios académicos.

Tabla No. 3.6: Requerimientos - Anuncios.

Requerimientos	Descripción
RQ20. Obtener Destinatarios Estudiantes	Proceso que permitirá obtener los Estudiantes Suscritos (destinatarios) en el Servicio de Comunicación TREEONE, por DOCENTE y por el PERIODO presente. Se utiliza un Servicio Web.
RQ21. Cargar Materias Docente, Estudiante	Proceso que permitirá obtener las MATERIAS que cursa el ESTUDIANTE, por PERIODO presente y, por Docente. Se utiliza un Servicio Web.
RQ22. Validar Destinatario	Proceso que permitirá validar si Código Banner del Destinatario se encuentra registrado a un suscriptor activo en el Servicio de Comunicación TREEONE.
RQ23. Obtener Suscriptores Personal	Proceso que permitirá obtener los usuarios suscritos (destinatarios) en el Servicio de Comunicación, de tipo PERSONAL.
RQ24. Crear Anuncio Personal	Proceso que permitirá generar el ANUNCIO ACADÉMICO para los suscriptores tipo PERSONAL.
RQ25. Crear Anuncio Docente	Proceso que permitirá generar el ANUNCIO ACADÉMICO para los suscriptores tipo DOCENTE.
RQ26. Enviar Anuncio	Proceso que permitirá enviar el ANUNCIO ACADÉMICO.

Gestión Parametrización

Requerimiento para el proceso de parametrización de los diferentes estados del Servicio de Comunicación TREEONE. En la *tabla No. 3.7* se presentan los requerimientos para la configuración de parámetros.

Tabla No. 3.7: Requerimientos - Parametrización.

Requerimientos	Descripción
RQ27. Configurar Parámetros	Proceso que permitirá configurar (cambiar los valores de los parámetros) los diferentes parámetros del Servicio de Comunicación TREEONE.
RQ28. Guardar Configuración	Proceso que permitirá guardar los cambios realizados sobre los parámetros del Servicio de Comunicación TREEONE.

Gestión de Suscriptos Administradores

Requerimientos para el proceso de otorgar el rol administrativo al usuario suscriptor. En la *tabla No. 3.8* se presentan los requerimientos para la gestión de suscriptores administradores.

Tabla No. 3.8: Requerimientos - Administración de Usuarios Suscriptos.

Requerimientos	Descripción
RQ29. Configurar Suscriptor Administrador	Proceso que permitirá configurar (ADICIONAR O ELIMINAR) SUSCRIPTORES con rol ADMINISTRATIVO en el Servicio de Comunicación TREEONE.

Administración de Periodos

Requerimiento para el proceso de Administración de Periodos. En la *tabla No. 3.9* se presenta el requerimiento para la administración de periodos académicos.

Tabla No. 3.9: Requerimientos - Administración de Periodos.

Requerimientos	Descripción
RQ30. Administrar Periodos	Proceso que permitirá CREAR, ELIMINAR y ACTUALIZAR los periodos.

Administración de Contactos

Requerimiento para el proceso de Administración de los contactos del Suscriptor.

En la *tabla No. 3.10* se presenta el requerimiento para la administración para la gestión de contactos.

Tabla No. 3.10: Requerimientos - Administración de Contactos.

Requerimientos	Descripción
RQ31. Administración de Contactos	Proceso que permitirá configurar (ADICIONAR y EDITAR) CONTACTOS del Suscriptor.

Gestión administración MÓDEMS

Requerimientos para el proceso de Configuración y Activación del MÓDEM, en donde se especificarán las gestiones para la configuración de los diferentes parámetros que se necesitan para la conexión del módem móvil usb.

En la *tabla No. 3.11* se presentan los requerimientos para la administración del módem.

Tabla No. 3.11: Requerimientos - Configuración y Activación del Módem.

Requerimientos	Descripción
RQ32. Configurar Parámetros Módem	Proceso que permitirá configurar los parámetros del MÓDEM para su funcionamiento
RQ33. Activar/Desactivar Módem	Proceso que permitirá gestionar (ACTIVAR O DESACTIVAR) el módems.

Otras Funcionalidades

Varios Requerimientos que se ejecutan en el Servicio de Comunicación TREEONE. En la *tabla No. 3.12* se presentan los requerimientos para las otras funcionalidades.

Tabla No. 3.12: Requerimientos - Otras Funcionalidades.

Requerimientos	Descripción
RQ34. Generar Consultas de Anuncios	Proceso que permitirá al Suscriptor, ver los anuncios recibidos por periodos.
RQ35. Generar Consultas de Mensajes	Proceso que permitirá a los ESTUDIANTES, poder ver los mensajes enviados y recibidos por periodos.
RQ36. Generar Consultas de Materias por Docente	Proceso que permitirá a los DOCENTES, poder ver las materias asignadas por periodo. Se utiliza un Servicio Web.
RQ37. Generar Consultas de Materias por Estudiante	Proceso que permitirá a los ESTUDIANTES, poder ver las materias asignadas por periodo.
RQ38. Generar Ayuda	Proceso que permitirá desplegar un instructivo para el manejo del Servicio de Comunicación TREEONE.

3.2.4.2 Requisitos No Funcionales

Requisitos de Interfaz del usuario

La interfaz con el usuario debe ser realizada en un entorno web para ser ejecutada por los navegadores más comunes del mercado. Las pantallas deben ser sencillas e intuitivas y ser mostradas en el idioma español.

Además se debe mantener la misma distribución física en las pantallas, es decir si en más de una pantalla existe el mismo icono, en todas debe ubicarse en el mismo lugar y orden.

Requisitos de rendimiento

Los recursos de consumo y carga del Servicio de Comunicación deben ser mínimos, debido a que no existe un software extra y su complejidad es baja.

El Servicio de Comunicación debe estar soportado por un equipo Dual Core 2 con 6Gb de memoria principal.

Tiempo de repuesta

Las consultas deben consumir la menor cantidad posible de recursos del servidor web que soporta el Servicio de Comunicación. Estas consultas no se deben tardar más de 10 segundos, dependiendo de la cantidad de registros existentes, debido a que las consultan manejan filtros, esta no serán complejas.

Cabe aclarar que esta velocidad de respuesta se refiere a conexiones realizadas a través de la red de la Universidad.

La mayoría del proceso se debe realizar en el equipo cliente y sólo realizar las consultas hacia la base de datos del Servicio de Comunicación, mediante los comandos PLSQL estándares.

Ciclo de vida

Para el desarrollo del Servicio de Comunicación se debe utilizar la metodología seleccionada, respetando el ciclo de vida orientado a objetos, lo que permite realizar cambios de acuerdo a las necesidades del usuario a medida que éste use las distintas etapas de desarrollo del prototipo.

Recursos Tecnológicos

El sistema en el entorno del usuario debe ser soportado por cualquier equipo que pueda ejecutar un navegador de Internet, y el sistema operativo Windows 2007 o superior.

Consumo de recursos

El Servicio de Comunicación deberá dar un servicio simultaneo a toda la ESPE, con un tiempo de respuesta inmediata en caso del registro y

suscripción, para el caso del mensaje dependerá de la velocidad con la cual el módem envíe el o los mensajes.

El tiempo de servicio exigido debe cumplirse, salvo en el caso de congestión de la red.

Restricciones de Diseño o Desarrollo

- El Servicio de Comunicación debe tener una arquitectura de 3 capas:
 1. Capa de consumo (web).
 2. Capa de servicio (lógica de negocio).
 3. Capa de persistencia (acceso a datos).
- El Servicio de Comunicación debe tener una arquitectura orientada a objetos.
- Para el desarrollo de la interfaz se utilizó PRIMEFACES.

Atributos del Sistema

Fiabilidad: la fiabilidad es uno de los factores que dará confianza a los suscriptores, para lo cual el Servicio de Comunicación estará controlando todo tipo de transacción y estará apto para responder todo tipo de incidente.

Mantenibilidad, El Servicio de Comunicación es susceptible para ser ampliado y diseñado para que su mantenimiento sea fácil, y de esta manera pueda ser corregido en caso de ser necesario.

Las labores de mantenimiento las realizará el administrador del Servicio de Comunicación, siendo él un personal académico suscripto en el sistema.

En cuanto a la configuración del módem, si se llegase a tener que hacer un mantenimiento, estos pueden ser configurados con los nuevos valores.

Software Libre: El proyecto será de software libre y, por tanto, cualquier componente software que reutilice también deberá ser libre.

Seguridad: El Servicio de Comunicación no proveerá de una creación manual de usuarios, este proceso se realizará automáticamente cuando el usuario acceda al módulo de suscripción.

Los usuarios únicamente se autenticarán utilizando a los usuarios de pruebas de la base de datos Banner, luego se validará si el usuario se encuentra suscripto en el Servicio de Comunicación, con esta validación (sea por la clave o por un mecanismo de control), el Servicio de Comunicación dará los accesos pertinentes.

Auditoría: Todas las suscripciones, mantenimientos de las tablas referenciales (catálogos), movimientos usuarios y las comunicaciones o avisas por mensajes, quedarán registrados para su estadística con: fecha de creación, fecha de actualización, usuario de creación y usuario de actualización.

3.2.4.3 Otros Requisitos

a) *Requerimientos De Hardware*

Computador Que Alojara La Aplicación

Para que la aplicación se ejecute con normalidad no es necesario grandes requerimientos de hardware ya que se desarrollará usando la tecnología de Java que es muy liviana en cuanto a consumo de memoria RAM para el funcionamiento de la JVM (Máquina Virtual de JAVA), ya que en los casos más críticos los mensajes se encolarán, se enviarán o recibirán uno a la vez, es decir que las variables y consumo de memoria que se genere en un mensaje luego de ser procesado se liberará, dando espacio al nuevo proceso que generará el siguiente mensaje.

En Microsoft Windows XP la JVM (Máquina Virtual de JAVA) necesita físicamente de:

Memoria RAM: mínimo 4 GB y máximo 6GB

El sistema operativo instalado donde alojará el software de comunicación es un Dual Core 3, teniendo como requisitos mínimos las siguientes características:

- Velocidad del Procesador: 3.00GHz.
- Memoria RAM: 6GB.
- Puertos USB: 1 puerto USB por módem GSM.
- Puerto LAN: 1 puerto LAN para conexión con la red.

b) Módem GSM o 3G

El módem GSM o 3G que se use debe soportar la librería SMSLIB para poder establecer la comunicación con el computador que alojará la aplicación y al cual estará conectado éste, también debe tener una memoria interna para almacenar los SMS ya que es allí donde llegan en primera instancia los mensajes de petición de información.

El modelo del módem adquirido para este proyecto es HUAWEY Mobile Broadband E173. Se elige este modelo y marca porque es un Módem que no tiene bloqueo alguno en el registro de la operadora que la vende, es decir que puede trabajar con SIM de CNT, MOVISTAR o CLARO.

c) Justificación Herramienta A Utilizar

Para la implementación del servicio de comunicación se planteó utilizar de manera preferente y en la medida viable para los requerimientos de la institución, soluciones basada en software libre y código abierto. El uso del software libre viene dado por razones técnicas de la Universidad de las Fuerzas Armadas ESPE, en las cuales se indican a continuación:

- Costos cero eliminando los costes derivados de las licencias.
- Gran flexibilidad que presta el poder modificar y adaptar el código a los requerimientos particulares del cliente.
- Fácil integración al permitir la modificación del código y ser un

software usado y creado de las necesidades de los usuarios.

- El mantenimiento es flexible.

En vista de lo detallado sobre uso de software libre, se especifican las siguientes herramientas, framework, servidores y base de datos usados por la construcción del servicio de comunicación.

1. Para el proyecto se utilizó el **LENGUAJE DE PROGRAMACIÓN JAVA**, ya que permite el desarrollo de aplicaciones web orientado a objetos, además es un lenguaje de programación multiplataforma ya que permite desarrollar muchos programas y llevarlos a ejecutar en distintos sistemas operativos sin la necesidad de modificarlos. Para el proyecto se utilizó la **versión 1.6 DE JAVA**.
2. Para abarcar el desarrollo Java, se utilizó el **SERVIDOR DE APLICACIONES JBOSS versión 7.1.1**, que es un servidor de aplicaciones Java EE de código abierto implementado en Java puro.
3. Para el soporte de datos se utilizó el manejador de Base de Datos **POSTGRESQL versión 9.1**, diseñada para entornos de alta transaccionalidad, es el que más se acopla a los lineamientos java opensource.
4. Se utilizó el framework **JAVASERVER FACES (JSF) 2.0**, que es una tecnología para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE, permitiendo manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad, así como la de administrar los diferentes Beans desarrollados.
5. Para el desarrollo de las interfaces se utilizó **PRIMEFACES versión 4.0 FINAL** (Framework para el desarrollo de páginas web con soporte AJAX). Primefaces es una librería de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de

componentes enriquecidos que facilitan la creación de las aplicaciones web.

6. Se utilizó **AJAX**, las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor.

La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor. (Eguiluz, 2013).

7. Para el envío y recepción de los mensajes se utilizó la biblioteca de mensajes **SMSLIB versión 3.5.2**, ya que proporciona una API de mensajes de texto universal, que se puede utilizar para enviar y recibir mensajes a través de módems GSM y está disponible para entornos Java y Microsoft NET Framework.
8. El IDE de desarrollo utilizado es **SPRING TOOL SUITE (STS) versión: 3.3.0.RELEASE**, q es un IDE basado en la versión Java EE de Eclipse, pero altamente customizado para trabajar con Spring Framework.
9. Se usó **Maven Versión 3.0.4** (Herramienta de software para la gestión y construcción de proyectos JAVA).
10. Se maneja Servicio Web para simular la consulta del usuario banner, registro y consultas de las gestiones académicas, aquí se utilizó una base **MySQL**, en donde se encuentran registrados varios usuarios del sistema banner.

En fin todas las herramientas estas soportadas por Java y son opensource.

3.2.4.4 Especificaciones suplementarias

En este apartado las especificaciones suplementarias capturan los requerimientos del Servicio de Comunicación que no se ven de alguna manera, tales requerimientos incluyen los puntos que se describen a continuación:

Usabilidad: El Servicio de Comunicación será usado por miembros del personal académico (estudiantes, docentes y personal administrativo). Además debe soportar un número de usuarios conectados simultáneamente.

Entre sus características se destacan los siguientes:

- El Servicio de Comunicación debe proporcionar una interfaz gráfica orientada al usuario final, que le permita utilizar la herramienta de manera fácil.
- El Servicio de Comunicación no debe emitir mensajes de errores fatales poco entendibles para el usuario.

Disponibilidad: El Servicio de Comunicación deberá estar disponible 24x7, debido a los siguientes eventos que se registren:

1. Que un Estudiante Suscrito solicite y reciba la información de sus Gestiones Académicas, enviando un mensaje SMS.
2. Que un Personal Suscrito consulte y registre Anuncios Académicas.

En caso de que se requiera mantenimiento, este será planificado.

Seguridad: El Servicio de Comunicación debe ser confiable, robusto y los datos por ninguna circunstancia deben perderse.

Desempeño: En cuanto a los requerimientos de desempeño se puede indicar.

Tiempo de acceso y Respuesta, se debe permitir de un acceso y tiempo de respuesta rápida.

Capacidad, debe tener capacidad suficiente para registrar al 6000 estudiantes y 200 entre docentes y personal administrativo.

3.2.4.5 Requerimientos de Ayuda.

El Servicio de Comunicación dispondrá de un espacio de ayuda, el cual permitirá desplegar la información sobre el uso de los servicios de comunicación.

3.3. FASE DE ELABORACIÓN

El objetivo de la fase de elaboración es refinar los requisitos obtenidos en la fase de iniciación y con ellos definir la arquitectura base del Servicio de Comunicación, el monitoreo de los posibles los riesgos, los planes de contingencia y la producción de un plan para la administración del proyecto.

3.3.1 Desarrollo Del Modelado Del Negocio

En esta sección se presenta los modelos de negocio definido por el RUP, para el proyecto de comunicación e información.

La ESPE cuenta con el departamento de las Tecnologías de Información y Comunicación (TIC), el cual maneja el sistema de gestión académica BANNER, permitiendo tener control de la información de todo el personal académico de la Universidad.

El modelo de procesos de negocio que se presenta en la siguiente *figura No. 3.1*, indica de manera global la distribución de los procesos del Servicio de Comunicación, en donde la BASE DE DATOS DE PRUEBAS es el encargado de suministrar la información académica de pruebas para la ejecución de los procesos de notificación, teniendo con interfaz mediadora al sistema de pruebas BANNER.

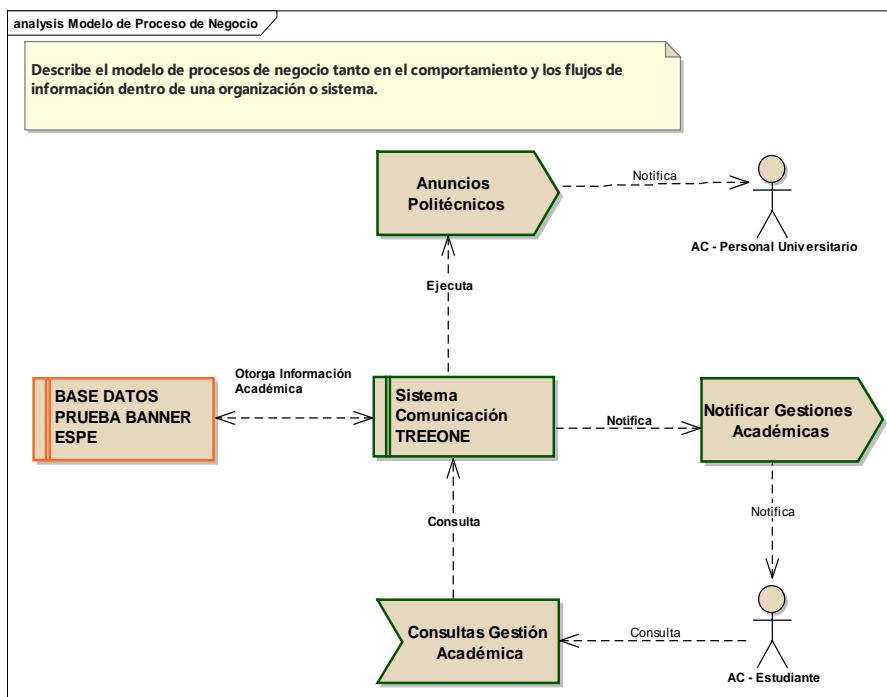


Figura No. 3.1: Modelo - Modelado Del Negocio.

3.3.2 Identificación del Modelo General

En la *Figura No. 3.2*, se indica el modelo general del Servicio de Comunicación, teniendo como acceso a la base de datos de pruebas Banner, la comunicación entre los 2 ambientes son a nivel de servicios web.

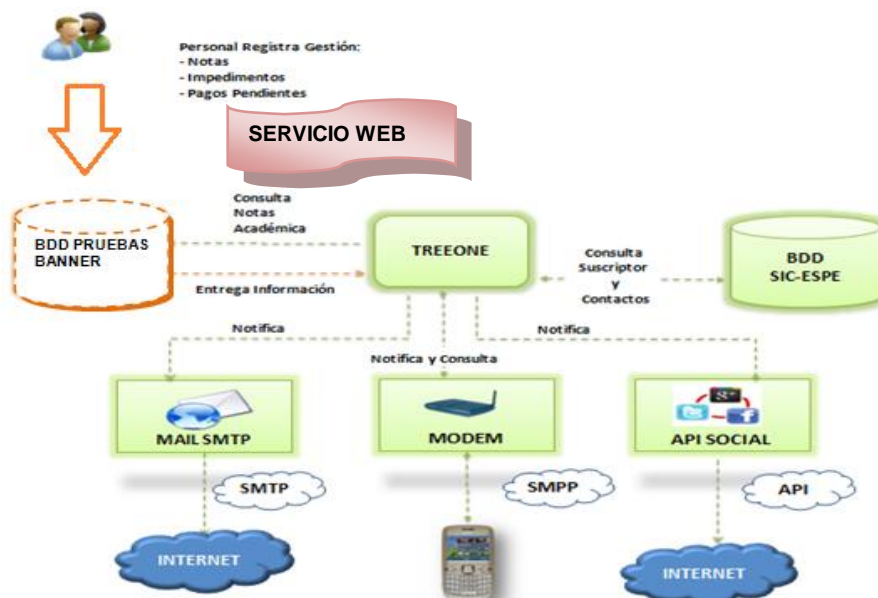


Figura No. 3.2: Modelo - General.

3.3.3 Especificación de los Requerimientos a Detalle

Actores

La definición de este requerimiento consiste en listar los diferentes usuarios que interactuarán con el Servicio de Comunicación de una forma distinta e indistintamente.

En el desarrollo se han identificado los siguientes actores que interactuarán con cada uno de los subsistemas definidos en el proyecto.

En la *tabla No. 3.13* se detallan los actores del servicio de comunicación.

Tabla No. 3.13: Especificaciones - Actores del Servicio de Comunicación.

Actores	Descripción
AC - Personal Académico	<i>Es el Personal Académico de la ESPE que pueden suscribirse, sean estos: Estudiantes, Docentes o Personal Administrativo ESPE</i>
AC - Administrador	<i>Es el personal suscrito, que tendrá el rol de Administrador.</i>
AC - Estudiante	<p><i>Este actor se encargará de:</i></p> <ul style="list-style-type: none"> • <i>Consultar y Recibir las Notas académicas mediante el envío de SMS's.</i> • <i>Recibir las gestiones académicas (notas académicas, impedimentos y estados de cuentas) por medio de: Correos electrónicos y Redes Sociales.</i> • <i>Recibir anuncios académicos por medio de: Correos electrónicos y Redes Sociales.</i>
AC - Docente	<i>Actor referente al personal Docente de la ESPE, que se encarga de registrar Anuncios Académicos.</i>
AC - Personal	<i>Actor referente al personal de la ESPE, que se encarga de Registrar los Anuncios Académicos.</i>
AC - BannerPruebas	<i>Actor referente a los usuarios registrados en la base de pruebas del Sistema Banner.</i>
AC - Suscriptor	<i>Actor referente al personal universitario suscrito en el servicio de comunicación.</i>
AC - Treeone	<i>Actor referente al Actores del Servicio de Comunicación TREEONE.</i>

Características del Software

Este requerimiento se debe a las necesidades que el usuario ha propuesto, para que el Actores del Servicio de Comunicación cumpla o satisfaga dichas necesidades.

En la *tabla No. 3.14*, se presentan las características del software

Tabla No. 3.14: Especificaciones - Características del Software


Características	Descripción
CS1 - Gestión Suscripción	<i>Gestión para la suscripción y cancelación del servicio de comunicación de todo el personal universitario.</i>
CS2 - Gestión Contactos	<i>Gestión para el registro de las diferentes cuentas de correo, cuentas sociales y número móvil del personal universitario.</i>
CS3 - Gestión Notificaciones Automáticas	<i>Proceso para la notificación de las diferentes gestiones académicas vía mensajes de telefonía móvil SMS, como:</i> <ul style="list-style-type: none"> • <i>Notas Académicas</i> • <i>Impedimentos Académicos</i> • <i>Valores Pendientes de Pago</i>
CS4 - Consulta Gestiones Académicas	<i>Proceso para las consultas de las diferentes gestiones académicas, mediante el envío de un mensaje de telefonía móvil SMS.</i>
CS5 - Anuncios Académicos	<i>Gestión para la creación de anuncios académicos mediante el envío de un correo ESPE, correo personal, mensaje Twitter y mensaje Facebook.</i>
CS6 - Gestión Administración y Configuración	<i>Gestión para la configuración de los diferentes parámetros.</i>
CS7 - Gestión Periodos	<i>Gestión que permite la creación de periodos universitarios.</i>
CS8 - Gestión Activación Módem	<i>Gestión que permite la activación y la desactivación del módem.</i>
CS9 - Gestión Suscriptores Administradores	<i>Gestión que permite configurar suscriptores administradores del servicio de comunicación.</i>

Casos de Uso

Los casos de uso son derivados de los requerimientos, de las especificaciones de las características software y del análisis de las necesidades de los usuarios, ver *tabla No. 3.15*.

Tabla No. 3.15: Especificaciones - Casos de Uso del Servicio TREEONE

Paquete	Casos de Uso	Descripción
Autenticación	CU - Acceder TREEONE	<i>Gestión para la autenticación y autorización al Servicio de Comunicación TREEONE.</i>
Autenticación	CU - Verificar Usuario	<i>Gestión que permite verificar si el usuario autenticado se encuentra registrado en la base de datos de pruebas de usuarios Banner.</i>
Suscripción Académica	CU_Suscripción Académica	<i>Gestión que permitirá al personal académico, suscribirse al Servicio</i>

CONTINÚA 

Cancelación Suscripción Académica	CU_Cancelación Suscripción Académica	Gestión que permitirá la cancelación de la suscripción al Servicio de Comunicación.
Notificación Entrante Gestión Académica	CU - Obtener Petición	<i>Gestión que permite obtener la información de las gestiones académicas registradas en las tablas de la base de pruebas Banner (NotasEspe, ImpedimentosEspe y CuentasEspe).</i>
Notificación Entrante Gestión Académica	CU - Enviar Mensaje SMS	<i>Gestión que permitirá enviar un mensaje de texto o SMS al estudiante suscriptor.</i>
Notificación Saliente Gestión Académica	CU_Registrar Gestiones Académicas	<i>Gestión que permitirá obtener el registro de las Notas Académicas, de los Impedimentos y de los valores pendientes de pago.</i>
Notificación Saliente Gestión Académica	CU - Obtener Información de Notificación	<i>Gestión que permitirá obtener los registros pendientes a enviar sobre las gestiones académicas, estas notificaciones serán enviadas por medio de un mensaje de texto o SMS. Las gestiones académicas que se enviará son: Notas Académicas, Impedimentos Académicos y Cuentas pendientes de pago.</i>
Anuncios Académicos	CU - Registrar Anuncios Docentes	<i>Gestión que permitirá el registro de Anuncios Académicos por parte del Suscriptor Docente hacia sus estudiantes suscriptos. El Docente podrá enviar anuncios a todos sus alumnos que se encuentran suscriptos en el servicio de comunicación, esto mediante la selección de, la o las materias que está a cargo en el periodo académico.</i>
Anuncios Académicos	CU - Registrar Anuncios Personal	<i>Gestión que permitirá el registro de Anuncios Académicos por parte del Suscriptor Personal hacia todos los suscriptores personales activos en el servicio de comunicación.</i>
Parametrización	CU - Configurar Parámetros	<i>Gestión para la configuración de los diferentes parámetros del Servicio de Comunicación.</i>
Administración Suscriptores	CU - Configurar Suscriptor Administrador	<i>Gestión para la creación de suscriptores administrativos del Servicio de Comunicación.</i>
Administración Periodos	CU - Generar Periodo	<i>Gestión para la creación de los diferentes periodos académicos del Servicio de Comunicación.</i>
Configuración Módems	CU - Configurar Parámetros Módem	<i>Gestión que permite configurar los valores de conexión del módem.</i>
Configuración Módems	CU - Activar/Desactivar Módem	<i>Gestión que permite activar y desactivar el módem.</i>
Administración Contactos	CU - Editar Contactos	<i>Gestión para la edición de los Contacto registrados en el Servicio de Comunicación.</i>

Clases

Son derivados de los requerimientos y de los casos de uso para la representación del modelo de datos.

No se especifican como clases a las entidades de la base de datos de pruebas Banner, ya que al pertenecer a otro esquema fuera del servicio de comunicación, no es parte del proyecto.

En la *tabla No. 3.16* se presentan las clases necesarias para el desarrollo del proyecto.

Tabla No. 3.16: Especificaciones - Clases del Servicio de Comunicación

Clases	Descripción
CL_Acceso	<i>Accesos que se dan a los Roles creados</i>
CL_Anuncios	<i>Clase que contienen y manejan los Anuncios registrados</i>
CL_Contacto	<i>Clase que contiene y maneja los Contactos del suscriptor</i>
CL_Mensajes	<i>Clase que contiene y maneja los Mensajes enviados y recibidos.</i>
CL_Menu	<i>Clase que contiene y maneja los Menús o páginas del servicio de Comunicación</i>
CL_NotificacionEntrante	<i>Clase que contiene y maneja las Notificaciones recibidas.</i>
CL_NotificacionSaliente	<i>Clase que contiene y maneja las Notificaciones de salida.</i>
CL_Periodo	<i>Clase que contiene y maneja los Periodos del Servicio de Comunicación</i>
CL_Permission_Rol	<i>Clase que contiene y maneja los Permisos y Roles del Servicio de Comunicación</i>
CL_Suscriptor	<i>Clase que contiene y maneja los Suscriptores del Actores del Servicio de Comunicación</i>

Las clases que manejarán los procesos automáticos del servicio de comunicación serán las de notificaciones y mensajes, en cambio las clases que manejarán los procesos manuales serán los de administración y de contactos.

3.4. DISEÑO DEL SISTEMA

En esta sección tiene como propósito, el definir de manera clara y precisa todas las funcionalidades y restricciones que se necesitan para la implementación del prototipo WEB, permitiendo administrar, consultar y comunicar las gestiones académicas del Sistema Banner Mi ESPE.

Estas funcionalidades se encuentran detalladas en la sección de especificación de requerimientos (**3.24 Funciones del Actores del Servicio de Comunicación**).

3.4.1 DIAGRAMAS DE CASOS DE USO

A continuación se presentan los diagramas de actores y de casos de uso derivados de los requerimientos y del análisis de las necesidades de los usuarios.

3.4.1.1 Actores/Stakeholders.

En la *figura No. 3.3*, se describen todos los actores/Stakeholders:

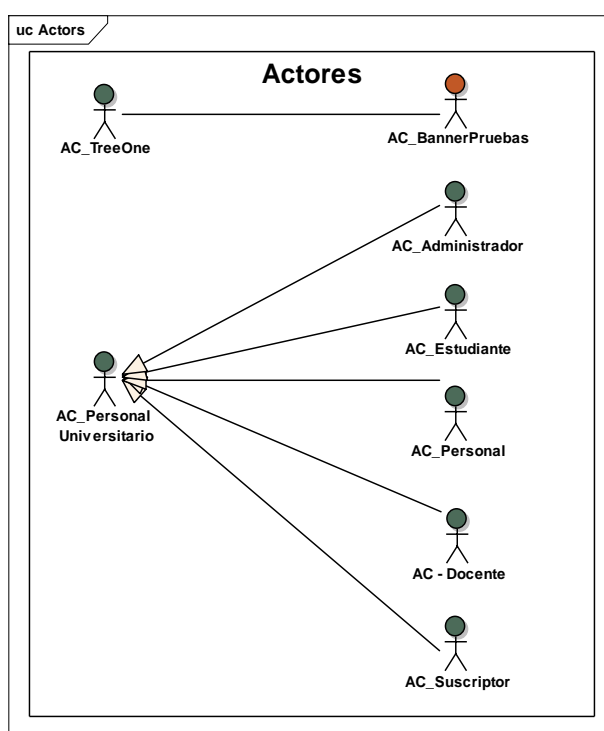


Figura No. 3.3: Diagrama - Actores del Servicio de comunicación.

Para los diagramas de casos de uso del servicio de comunicación, se presentan los diferentes diagramas:

3.4.1.2 Proceso de AUTENTIFICACIÓN.

Este proceso de autenticación consiste primeramente en verificar por medio de un servicio web, la existencia del usuario sobre la base de datos de pruebas banner. Con el usuario registrado se puede acceder al Servicio de Comunicación. En la *figura No. 3.4* se representa el diagrama los Casos de Uso del proceso de Autenticación y Autorización.

Actores/Stakerholder: Personal Universitario, Treeone y Banner Prueba.

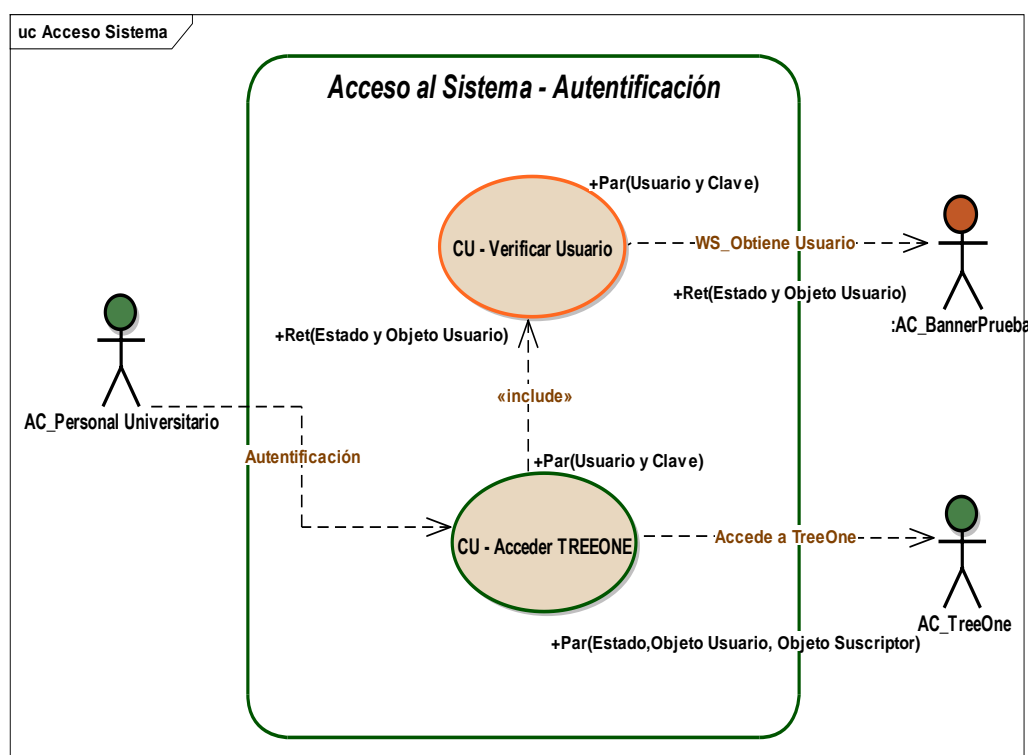



Figura No. 3.4: Caso de Uso - Autenticación.

Caso de Uso Acceder TREEONE

En la *tabla No. 3.17*, se presentan los escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.17: Escenarios - Acceder TREEONE.

Nombre del Caso de Uso:	CU - Acceder TREEONE
Notas:	Gestión para la autenticación y autorización al Servicio de Comunicación TREEONE.
RESTRICCIONES:	
Pre-condition	Recibir Estado de la Consulta del Web Service
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC Personal Universitario</u> accede al servicio de comunicación por medio de la pantalla <u>Pantalla Login</u> 2. el <u>AC Personal Universitario</u> ingresa el USUARIO y CLAVE sobre la <u>Pantalla Login</u> 3. La <u>Pantalla Login</u> abre la comunicación con <u>AC TreeOne</u> 4. <u>AC TreeOne</u> realiza la verificación del usuario mediante la inclusión del caso de uso <u>CU - Verificar Usuario</u> <p>Usos: <i>Web Service:</i> servicio.getBannerPort().buscarPorClaveCodigo(usuario, clave)</p> <ol style="list-style-type: none"> 5. <u>CU - Verificar Usuario</u> retorna el objeto Usuario a <u>AC TreeOne</u> 6. <u>AC TreeOne</u> valida si el usuario se encuentra suscrito en el servicio de comunicación, objeto <u>Suscriptores</u> <p>Usos: <i>Suscriptores.buscarSuscriptorVigentePorCodigo(usuario())</i></p> <ol style="list-style-type: none"> 7. Se retorna el objeto <u>Suscriptores</u> a <u>AC TreeOne</u> 8. <u>AC TreeOne</u> presenta <u>Pantalla Principal</u> del servicio de comunicación <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>
<p>Nombre: Usuario no Existe</p> <p>Tipo: Excepción</p>	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC Personal Universitario</u> accede al servicio de comunicación por medio de la pantalla <u>Pantalla Login</u> 2. el <u>AC Personal Universitario</u> ingresa el USUARIO y CLAVE sobre la <u>Pantalla Login</u> 3. La <u>Pantalla Login</u> abre la comunicación con <u>AC TreeOne</u> 4. <u>AC TreeOne</u> realiza la verificación del usuario mediante la inclusión del caso de uso <u>CU - Verificar Usuario</u> <p>Usos: <i>Web Service:</i> servicio.getBannerPort().buscarPorClaveCodigo(usuario, clave)</p> <p>Excepción: Cuando [Usuario no Existe] 4a_1. Se presenta mensaje "CREDENCIALES INCORRECTAS "</p> <p>Resultado: Usuario no Existe completo. Volver al punto 1.</p> <p>Fin</p>

CONTINÚA 

<p>Nombre: <i>Usuario no Suscrito</i></p> <p>Tipo: <i>Excepción</i></p>	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC_Personal Universitario</u> accede al servicio de comunicación por medio de la pantalla <u>Pantalla_Login</u> 2. el <u>AC_Personal Universitario</u> ingresa el USUARIO y CLAVE sobre la <u>Pantalla_Login</u> 3. La <u>Pantalla_Login</u> abre la comunicación con <u>AC_TreeOne</u> 4. <u>AC_TreeOne</u> realiza la verificación del usuario mediante la inclusión del caso de uso <u>CU - Verificar Usuario</u> <p>Usos: <i>Web Service:</i> <i>servicio.getBannerPort().buscarPorClaveCodigo(usuario, clave)</i></p> <ol style="list-style-type: none"> 5. <u>CU - Verificar Usuario</u> retorna el objeto Usuario a <u>AC_TreeOne</u> 6. <u>AC_TreeOne</u> valida si el usuario se encuentra suscrito en el servicio de comunicación, objeto <u>Suscriptores</u> <p>Usos: <i>Suscriptores.buscarSuscriptorVigentePorCodigo(usuario())</i></p> <p>Excepción: Cuando [Usuario no Suscrito]</p> <p>6a_1. <u>AC_TreeOne</u> presenta pantalla para la suscripción al servicio de comunicación por medio de la <u>Pantalla_Suscripción</u></p> <p>Resultado: Usuario no Suscrito completo.</p> <p>Fin</p>
<p>Nombre: <i>Usuario Administrador Treeone</i></p> <p>Tipo: <i>Camino Alterno</i></p>	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC_Personal Universitario</u> accede al servicio de comunicación por medio de la pantalla <u>Pantalla_Login</u> <p>Camino Alterno: Cuando [Usuario Administrador Treeone]</p> <p>2a_1. <u>AC_TreeOne</u> verifica las credenciales de acceso sobre el objeto <u>Suscriptores</u></p> <p>Usos: <i>Suscriptores.buscarPorUsuarioClave(usuario, claveEncriptada)</i></p> <p>Continúa en el punto</p> <p>8. <u>AC_TreeOne</u> presenta <u>Pantalla_Principal</u> del servicio de comunicación</p> <p>Fin</p>

Caso de Uso - Verificar Usuario

En la *tabla No. 3.18*, se presentan los diferentes escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.18: Escenarios - Verificar Usuario.

Nombre del Caso de Uso:	CU - Verificar Usuario
Notas:	Permite verificar si el usuario autenticado se encuentra registrado o activo en la base de pruebas de usuarios Banner. Recibe: Usuario y Clave. Retorna: Estado y objeto Usuario (Si el estado fuera activo).
RESTRICCIONES:	
Pre-condition	Recibe Usuario y Contraseña
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <p>1. <u>AC_TreeOne</u> solicita verificar las credenciales en <u>AC_BannerPruebas</u> por medio de un Web Service. Usos: Web Service: <i>buscarPorClaveCodigo(usuario, clave)</i></p> <p>2. <u>AC_BannerPruebas</u> retorna a <u>AC_TreeOne</u> la respuesta de la verificación de credenciales por medio del Web Service Usos: Web Service: <i>buscarPorClaveCodigo(usuario, clave)</i></p> <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>

3.4.1.3 Proceso de SUSCRIPCIÓN ACADÉMICA

Proceso que indica el caso de uso para la gestión de la Suscripción al Servicio de Comunicación TREEONE.

Una vez validado el usuario, se obtiene los datos del personal académico mediante la consulta vía servicio web (nombres, apellidos, número de identificación, id banner, correo electrónico, periodo, rol), luego valida si el usuario que se está suscribiendo se encuentra suscripto o no en el servicio de comunicación TREEONE. El resultado es un suscriptor nuevo creado según su rol obtenido a través del servicio web.

En la *figura No. 3.5* se presenta el diagrama de caso de uso del proceso de Suscripción Académica.

Actores: Personal Universitario, Banner Pruebas y Suscriptor.

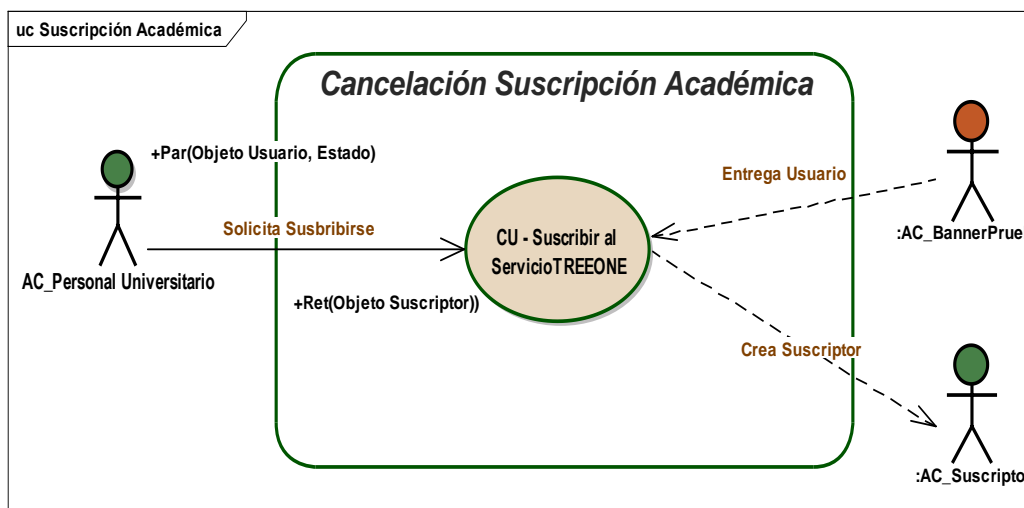


Figura No. 3.5: Caso de Uso - Suscripción Académica.

Caso de Uso - Suscribir al Servicio de Comunicación TREEONE

En la *tabla No. 3.19*, se presentan los escenarios, excepciones y caminos alternos del caso de uso

Tabla No. 3.19: Escenarios - Suscribir al Servicio TREEONE.

Nombre del Caso de Uso:	CU - Suscribir al Servicio TREEONE
Notas:	Proceso que permitirá suscribirse al servicio de mensajería mediante el registro de la información de contacto del personal académico
RESTRICCIONES:	
Pre-condition	Usuario Pruebas Banner Activo
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC TreeOne</u> obtiene el usuario y presenta la <u>Pantalla Suscripción</u> Usos: Web Service: buscarPorCodigo 2. <u>AC TreeOne</u> internamente trae el usuario de <u>AC BannerPruebas</u> Usos: Web Service: servicio.getBannerPort().buscarPorCodigo(codigoBanner) 3. <u>AC TreeOne</u> carga la cuenta ESPE sobre la <u>Pantalla Suscripción</u> Usos: Se carga automáticamente el correo ESPE del usuario de pruebas 4. <u>AC Personal Universitario</u> registra las cuentas sobre la <u>Pantalla Suscripción</u> 5. <u>AC Personal Universitario</u> guarda y cierra la <u>Pantalla Suscripción</u> 6. <u>Pantalla Suscripción</u> guardar los datos en <u>AC TreeOne</u> 7. <u>AC TreeOne</u> internamente crea el <u>Suscriptores</u>, crea los <u>Permisos</u> y crea los <u>Contactos</u>

CONTINÚA

<p>Tipo: Camino Básico</p>	<p>8. <u>AC TreeOne</u> presenta <u>Pantalla Principal</u> al nuevo <u>AC Suscriptor</u></p> <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>
<p>Nombre: Campos Vacíos</p> <p>Tipo: Excepción</p>	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC TreeOne</u> obtiene el usuario y presenta la <u>Pantalla Suscripción</u> Usos: Web Service: <i>buscarPorCodigo</i> 2. <u>AC TreeOne</u> internamente trae el usuario de <u>AC BannerPruebas</u> Usos: Web Service: <i>servicio.getBannerPort().buscarPorCodigo(codigoBanner)</i> 3. <u>AC TreeOne</u> carga la cuenta ESPE sobre la <u>Pantalla Suscripción</u> Usos: <i>Se carga automáticamente el correo ESPE del usuario de pruebas.</i> 4. <u>AC Personal Universitario</u> registra las cuentas sobre la <u>Pantalla Suscripción</u> 5. <u>AC Personal Universitario</u> guarda y cierra la <u>Pantalla Suscripción</u> <p>Excepción: Cuando [Campos Vacíos] 5c_1. Se presenta un mensaje "Campos obligatorios"</p> <p>Resultado: Campos Vacíos complete. Volver al punto 4.</p> <p>Fin</p>
<p>Nombre: Formato Incorrecto</p> <p>Tipo: Excepción</p>	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC TreeOne</u> obtiene el usuario y presenta la <u>Pantalla Suscripción</u> Usos: Web Service: <i>buscarPorCodigo</i> 2. <u>AC TreeOne</u> internamente trae el usuario de <u>AC BannerPruebas</u> Usos: Web Service: <i>servicio.getBannerPort().buscarPorCodigo(codigoBanner)</i> 3. <u>AC TreeOne</u> carga la cuenta ESPE sobre la <u>Pantalla Suscripción</u> Usos: <i>Se carga automáticamente el correo ESPE del usuario de pruebas.</i> 4. <u>AC Personal Universitario</u> registra las cuentas sobre la <u>Pantalla Suscripción</u> 5. <u>AC Personal Universitario</u> guarda y cierra la <u>Pantalla Suscripción</u> <p>Excepción: Cuando [Formato Incorrecto] 5b_1. Se presenta un mensaje "Formato Incorrecto"</p> <p>Resultado: Formato Incorrecto complete. Volver al punto 4.</p> <p>Fin</p>
<p>Nombre: Cuentas Repetidas</p> <p>Tipo: Excepción</p>	<p>Inicio:</p> <ol style="list-style-type: none"> 1. <u>AC TreeOne</u> obtiene el usuario y presenta la <u>Pantalla Suscripción</u> Usos: Web Service:<i>buscarPorCodigo</i> 2. <u>AC TreeOne</u> internamente trae el usuario de <u>AC BannerPruebas</u> Usos: Web Service: <i>servicio.getBannerPort().buscarPorCodigo(codigoBanner)</i>

Nombre: <i>Cuentas Repetidas</i>	4. <u>AC Personal Universitario</u> registra las cuentas sobre la <u>Pantalla Suscripción</u>
Tipo: <i>Excepción</i>	5. <u>AC Personal Universitario</u> guarda y cierra la <u>Pantalla Suscripción</u>
	Excepción: Cuando [Cuentas Repetidas] 5a_1. Se presenta un mensaje "Cuenta ya existe"
	Resultado: Cuentas Repetidas complete. Volver al punto 4.
	Fin

3.4.1.4 Proceso de CANCELACIÓN SUSCRIPCIÓN ACADÉMICA

Proceso que indica el caso de uso para la gestión de Cancelación de la Suscripción al Servicio de Comunicación TREEONE. En la *figura No. 3.6*, se presenta el caso de uso de Cancelación Suscripción Académica

Actores: Suscriptor y TreeOne.

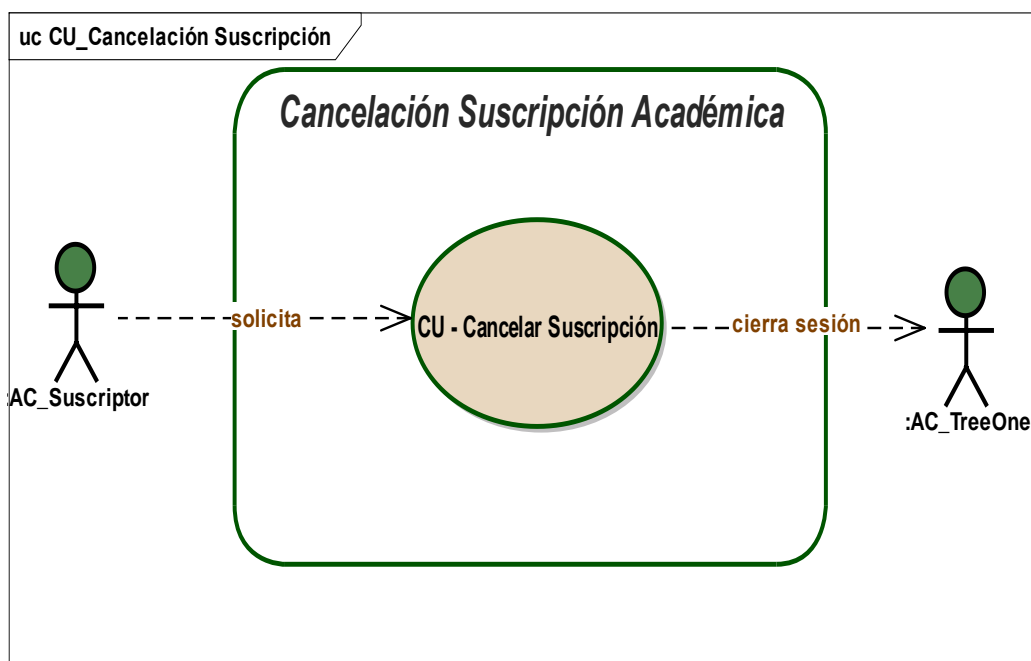


Figura No. 3.6: Caso de Uso - Cancelación Suscripción Académica.

Caso de Uso - Cancelación Suscripción

En la *tabla No. 3.20* se presentan los escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.20: Escenarios - Cancelación Suscripción.

Nombre del Caso de Uso:	CU - Cancelar Suscripción
Notas:	Gestión que permite al personal académico, cancelar la suscripción al Servicio de Comunicación
RESTRICCIONES:	
Pre-condition	Suscriptor Activo (El suscriptor debe encontrarse activo)
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC_Suscriptor</u> accede a la <u>Pantalla_CancelarSuscripción</u> 2. la <u>Pantalla_CancelarSuscripción</u> presenta botón Cancelar al <u>AC_Suscriptor</u> 3. <u>AC_Suscriptor</u> cancela suscripción sobre la <u>Pantalla_CancelarSuscripción</u> 4. <u>Pantalla_CancelarSuscripción</u> se cierra sesión e invoca <u>AC_TreeOne</u> 5. <u>AC_TreeOne</u> internamente elimina los <u>Contactos</u> Usos: vigencia es false 6. <u>AC_TreeOne</u> internamente elimina los <u>Permisos</u> Usos: vigencia es false 7. <u>AC_TreeOne</u> internamente elimina el suscriptor en <u>Suscriptores</u> Usos: vigencia es false 8. <u>AC_TreeOne</u> cierra sesión del y presenta <u>Pantalla_Login</u> <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>

3.4.1.5 Proceso de NOTIFICACIÓN ENTRANTE GESTIÓN ACADÉMICA

Gestión que permite consultar las gestiones académicas mediante el envío de un mensaje de texto - SMS.

Este servicio estará presente solamente para los suscriptores estudiantes, ya que por medio de un mensaje de texto, podrán consultar sus notas académicas, sus impedimentos o sus valores pendientes de pago.

El código y el número a enviar en el mensaje de texto, se encuentra descrito en la página instructivo de uso.

Con esto es estudiante podrá consultar gestiones académicas del presente periodo configurado en el sistema TREEONE, y podrá recibir la notificación de la consulta ejecutada, mediante un mensaje de texto - SMS, por los correos electrónicos, por el correo Facebook y por un mensaje Twitter.

En la *figura No. 3.7* se presenta el caso de uso *Notificación Entrante de Gestión Académica*.

Actores: Estudiante, Treeone y BannerPruebas.

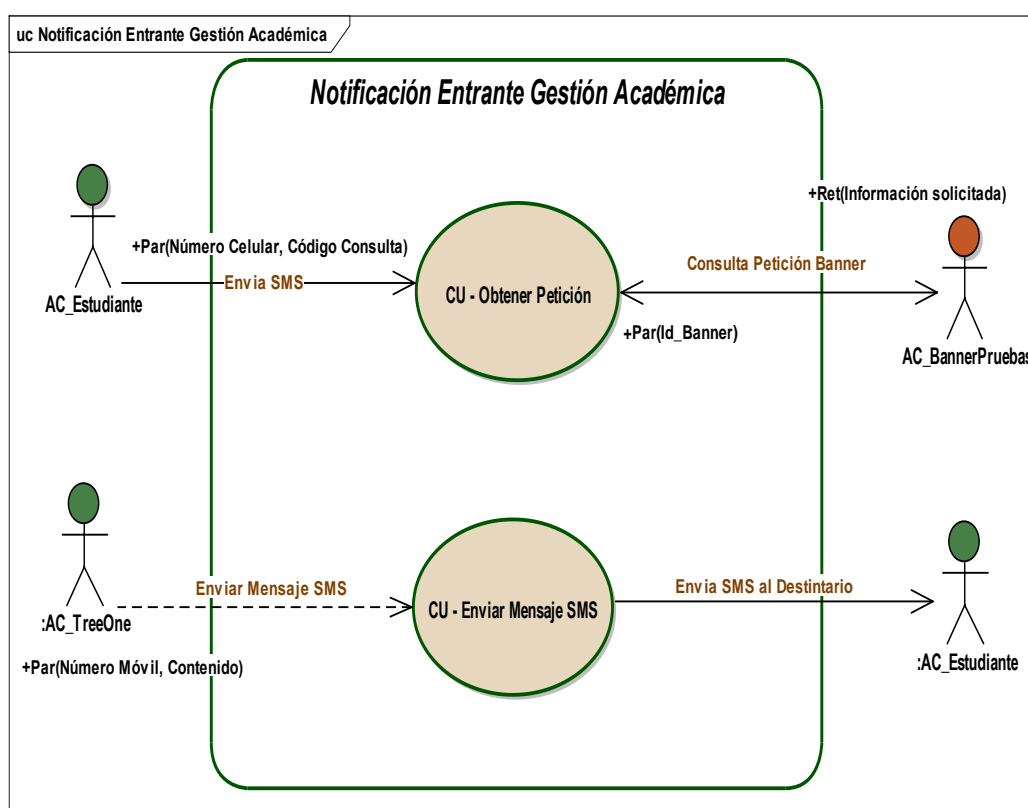



Figura No. 3.7: Caso de Uso - Notificación Entrante Gestión Académica.

Caso de Uso - Consultar Petición


En la *tabla No. 3.21*, se presentan los escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.21: Escenarios - Consultar Petición.

Nombre del Caso de Uso:	CU - Obtener Petición
Notas:	Gestión que permite obtener la información de las gestiones académicas registradas en las tablas de pruebas (NotasEspe, ImpedimentosEspe y CuentasEspe). Según el código de consulta (impedimentos, notas o cuentas), este proceso realizará la obtención de la información consultada.
RESTRICCIONES:	
Pre-condition	Recibir el mensaje de texto
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC_Estudiante</u> envía un SMS al número móvil configurado en <u>AC_TreeOne</u>. Usos: <i>Número móvil ESPE y código gestión</i> 2. <u>AC_TreeOne</u> recibe el mensaje SMS Usos: <i>SMSLIB InboundNotification</i> 3. <u>AC_TreeOne</u> valida el código del mensaje SMS Usos: <i>Case</i> 4. <u>AC_TreeOne</u> valida que el número del origen del mensaje SMS este asociado a un <u>Contactos</u> registrado. Usos: <i>validaNumero(número)</i> 5. <u>AC_TreeOne</u> obtiene el ID del suscriptor de <u>Contactos</u> 6. <u>AC_TreeOne</u> valida si el suscriptor se encuentra activo en <u>Periodos</u> Usos: <i>verificaSiUsuarioEstaSuscripto(Contactos.suscriptor)</i> 7. <u>AC_TreeOne</u> obtiene el suscriptor del <u>AC_Estudiante</u> 8. <u>AC_TreeOne</u> consulta el tipo de gestión en <u>AC_BannerPruebas</u> Usos: <i>Web Service</i> 9. <u>AC_BannerPruebas</u> indica a <u>AC_TreeOne</u> Código es "notas" 10. <u>AC_BannerPruebas</u> entrega información de NOTAS a <u>AC_TreeOne</u> Usos: <i>Web Service:</i> <i>servicio.getBannerPort().consultarNotasPorAlumno</i> <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>
Nombre: Código No Existe Tipo: Excepción	<p>Inicio:</p> <ol style="list-style-type: none"> 1. <u>AC_Estudiante</u> envía un SMS al número móvil configurado en <u>AC_TreeOne</u>. Usos: <i>Número movil ESPE y código gestión</i> 2. <u>AC_TreeOne</u> recibe el mensaje SMS Usos: <i>SMSLIB InboundNotification</i> 3. <u>AC_TreeOne</u> valida el código del mensaje SMS Usos: <i>Case</i>

CONTINÚA 

<p>Nombre: <i>Código No Existe</i></p> <p>Tipo: <i>Excepción</i></p>	<p>3a_1. Se guarda en la tabla <u>NotificacionesEntrantes</u> con el estado "Código no Existe"</p> <p>Resultado: Código No Existe complete.</p> <p>Fin:</p>
<p>Nombre: <i>Suscriptor No Existe</i></p> <p>Tipo: <i>Excepción</i></p>	<p>Inicio:</p> <ol style="list-style-type: none"> 1. <u>AC_Estudiante</u> envía un SMS al número móvil configurado en <u>AC_TreeOne</u>. Usos: <i>Número movil ESPE y código gestión</i> 2. <u>AC_TreeOne</u> recibe el mensaje SMS Usos: <i>SMSLIB InboundNotification</i> 3. <u>AC_TreeOne</u> valida el código del mensaje SMS Usos: <i>Case</i> 4. <u>AC_TreeOne</u> valida que el número del origen del mensaje SMS este asociado a un <u>Contactos</u> registrado. Usos: <i>validaNumero(número)</i> 5. <u>AC_TreeOne</u> obtiene el ID del suscriptor de <u>Contactos</u> 6. <u>AC_TreeOne</u> valida si el suscriptor se encuentra activo en <u>Periodos</u> Usos: <i>verficaSiUsuarioEstaSuscripto(Contactos.suscriptor)</i> <p>Excepción: Cuando [Suscriptor No Existe]</p> <p>6a_1. Se guarda en la tabla <u>NotificacionesEntrantes</u> con el estado "Suscriptor no Existe"</p> <p>Resultado: Suscriptor No Existe complete.</p> <p>Fin</p>
<p>Nombre: <i>Código es "cuentas"</i></p> <p>Tipo: <i>Camino Alternante</i></p>	<p>Inicio:</p> <ol style="list-style-type: none"> 1. <u>AC_Estudiante</u> envía un SMS al número móvil configurado en <u>AC_TreeOne</u>. Usos: <i>Número movil ESPE y código gestión</i> 2. <u>AC_TreeOne</u> recibe el mensaje SMS Usos: <i>SMSLIB InboundNotification</i> 3. <u>AC_TreeOne</u> valida el código del mensaje SMS Usos: <i>Case</i> 4. <u>AC_TreeOne</u> valida que el número del origen del mensaje SMS este asociado a un <u>Contactos</u> registrado. Usos: <i>validaNumero(número)</i> 5. <u>AC_TreeOne</u> obtiene el ID del suscriptor de <u>Contactos</u> 6. <u>AC_TreeOne</u> valida si el suscriptor se encuentra activo en <u>Periodos</u> Usos: <i>verficaSiUsuarioEstaSuscripto(Contactos.suscriptor)</i>

CONTINÚA 


<p>Nombre: Código es "cuentas"</p> <p>Tipo: Camino Alternante</p>	<p>7. <u>AC_TreeOne</u> obtiene el suscriptor del <u>AC_Estudiante</u></p> <p>8. <u>AC_TreeOne</u> consulta el tipo de gestión en <u>AC_BannerPruebas</u></p> <p>Usos: Web Service</p> <p>Camino Alternante: Cuando [Código es "cuentas"] 9b_1. <u>AC_BannerPruebas</u> entrega información de CUENTAS a <u>AC_TreeOne</u></p> <p>Resultado: Código es "cuentas" complete.</p> <p>Fin:</p>
<p>Nombre: Código es "impedimentos"</p> <p>Tipo: Camino Alternante</p>	<p>Inicio:</p> <p>1. <u>AC_Estudiante</u> envía un SMS al número móvil configurado en <u>AC_TreeOne</u>.</p> <p>Usos: Número movil ESPE y código gestión</p> <p>2. <u>AC_TreeOne</u> recibe el mensaje SMS</p> <p>Usos: SMSLIB InboundNotification</p> <p>3. <u>AC_TreeOne</u> valida el código del mensaje SMS</p> <p>Usos: Case</p> <p>4. <u>AC_TreeOne</u> valida que el número del origen del mensaje SMS este asociado a un <u>Contactos</u> registrado.</p> <p>Usos: validaNumero(número)</p> <p>5. <u>AC_TreeOne</u> obtiene el ID del suscriptor de <u>Contactos</u></p> <p>6. <u>AC_TreeOne</u> valida si el suscriptor se encuentra activo en <u>Periodos</u></p> <p>Usos: verificaSiUsuarioEstaSuscriptor(Contactos.suscriptor)</p> <p>7. <u>AC_TreeOne</u> obtiene el suscriptor del <u>AC_Estudiante</u></p> <p>8. <u>AC_TreeOne</u> consulta el tipo de gestión en <u>AC_BannerPruebas</u></p> <p>Usos: Web Service</p> <p>Camino Alternante: Cuando [Código es "impedimentos"] 9a_1. <u>AC_BannerPruebas</u> entrega información de IMPEDIMENTOS a <u>AC_TreeOne</u></p> <p>Resultado: Código es "impedimentos" complete.</p> <p>Fin:</p>

Caso de Uso - Enviar Mensaje SMS

En la *tabla No. 3.22*, se presentan los escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.22: Escenarios - Enviar Mensaje SMS.

Nombre del Caso de Uso:	CU - Enviar Mensaje SMS
Notas:	Gestión que permite enviar un mensaje de texto o SMS al estudiante suscriptor.
RESTRICCIONES:	
Pre-condition	Número Destinatario (Número móvil de destinatario del mensaje)
Pre-condition	Contenido (El texto o contenido del Mensaje)
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC_TreeOne</u> recibe número y contenido del mensaje a enviar. 2. <u>AC_TreeOne</u> valida la configuración de los <u>Parámetros</u> de notificación <u>Usos:</u> <i>buscarVigentesNotiEntrante(contenido)</i> 3. <u>Parámetros</u> entrega la validación a <u>AC_TreeOne</u> <u>Usos:</u> <i>Booleano (T,F)</i> 4. <u>AC_TreeOne</u> ejecuta el envío del mensaje SMS hacia el <u>AC Estudiante</u> <u>Usos:</u> <i>SendThread(celular, mensaje, gateway)</i> 5. <u>AC_TreeOne</u> guarda el estado del mensaje enviado en <u>Mensajes</u> <u>Usos:</u> <i>guardarMensaje()</i> 6. <u>AC_TreeOne</u> guarda el estado del mensaje enviado en <u>NotificacionesEntrantes</u> <u>Usos:</u> <i>guardarNotificacionEntrante</i> <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>
<p>Nombre: <i>Parámetro envío SMS Inactivo</i></p> <p>Tipo: <i>Excepción</i></p>	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC_TreeOne</u> recibe número y contenido del mensaje a enviar 2. <u>AC_TreeOne</u> valida la configuración de los <u>Parámetros</u> de notificación <u>Usos:</u> <i>buscarVigentesNotiEntrante(contenido)</i> 3. <u>Parámetros</u> entrega la validación a <u>AC_TreeOne</u> <u>Usos:</u> <i>Booleano (T,F)</i> <p>Excepción: Cuando [Parámetro envío SMS Inactivo]</p> <p>3a_1. <u>AC_TreeOne</u> guarda el mensaje SMS no enviado en la tabla <u>NotificacionesEntrantes</u></p> <p>Resultado: Parámetro envío SMS Inactivo complete.</p> <p>Fin</p>

CONTINÚA 

<p>Nombre: <i>Es Mensaje de Notificación Saliente</i></p> <p>Tipo: <i>Camino Alterno</i></p>	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC_TreeOne</u> recibe número y contenido del mensaje a enviar 2. <u>AC_TreeOne</u> valida la configuración de los <u>Parámetros</u> de notificación <u>Usos:</u> <i>buscarVigentesNotiEntrante(contenido)</i> 3. <u>Parámetros</u> entrega la validación a <u>AC_TreeOne</u> <u>Usos:</u> <i>Booleano (T,F)</i> 4. <u>AC_TreeOne</u> ejecuta el envío del mensaje SMS hacia el <u>AC_Estudiante</u> <u>Usos:</u> <i>SendThread(celular, mensaje, gateway)</i> 5. <u>AC_TreeOne</u> guarda el estado del mensaje enviado en <u>Mensajes</u> <u>Usos:</u> <i>guardarMensaje()</i> <p>Camino Alterno: Cuando [Es Mensaje de Notificación Saliente] 6a_1. <u>AC_TreeOne</u> guarda el estado del mensaje enviado en <u>Mensajes</u> y en <u>NotificacionesEntrantes</u></p> <p>Resultado: <u>eliminar secciones en word</u> Es Mensaje de Notificación Saliente complete.</p> <p>Fin</p>
<p>Nombre: <i>Mensaje SMS Falló</i></p> <p>Tipo: <i>Camino Alterno</i></p>	<p>Inicio:</p> <ol style="list-style-type: none"> 1. <u>AC_TreeOne</u> recibe número y contenido del mensaje a enviar 2. <u>AC_TreeOne</u> valida la configuración de los <u>Parámetros</u> de notificación <u>Usos:</u> <i>buscarVigentesNotiEntrante(contenido)</i> 3. <u>Parámetros</u> entrega la validación a <u>AC_TreeOne</u> <u>Usos:</u> <i>Booleano (T,F)</i> 4. <u>AC_TreeOne</u> ejecuta el envío del mensaje SMS hacia el <u>AC_Estudiante</u> <u>Usos:</u> <i>SendThread(celular, mensaje, gateway)</i> <p>Excepción: Cuando [Mensaje SMS Falló] 4a_1. <u>AC_TreeOne</u> guarda en la tabla en <u>NotificacionesEntrantes</u>, el estado y el mensaje de error.</p> <p>Resultado: Mensaje SMS Falló complete.</p> <p>Fin:</p>

3.4.1.6 Proceso de NOTIFICACIÓN SALIENTE GESTIÓN ACADÉMICA.

Proceso automático que permite notificar o enviar la información de la gestiones académica registradas en la base de datos de pruebas Banner, por medio de un mensaje de texto o SMS, de correos electrónicos, de correo facebook y por medio de un mensaje social Twitter.

En la *figura No. 3.8*, se presenta el caso de uso Notificación Saliente Gestión Académica.

Actores: Personal Universitario, Treeone, BannerPruebas y Estudiante.

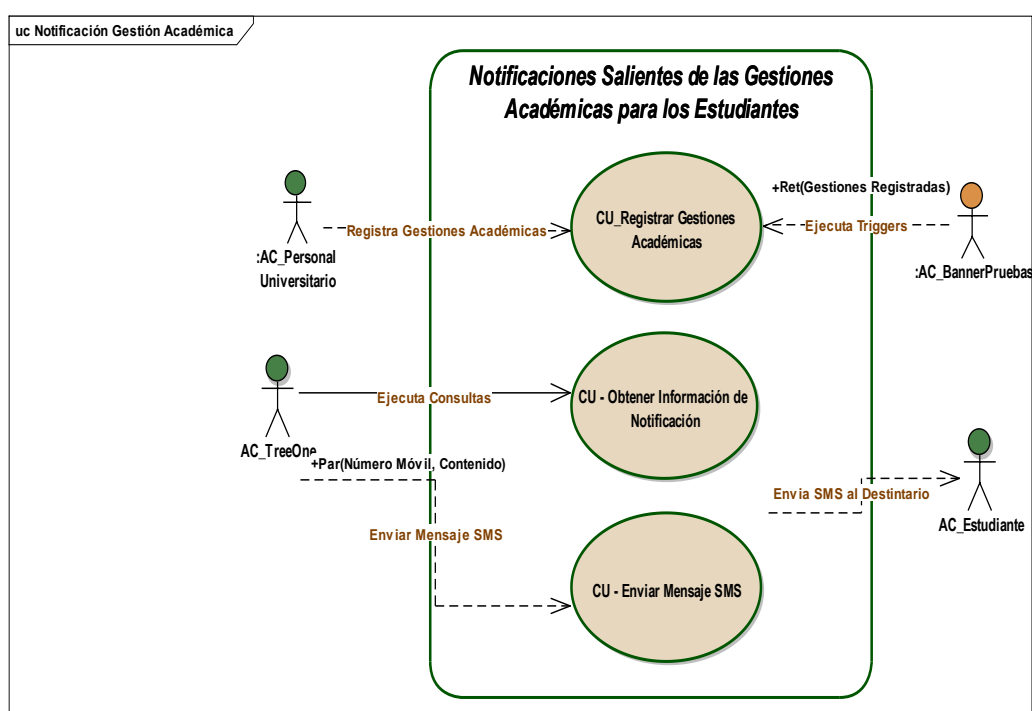


Figura No. 3.8: Caso de Uso - Notificación Saliente Gestión Académica.

Caso de Uso - Registrar Gestiones Académicas.

En la *tabla No. 3.23*, se presentan los distintos escenarios, excepciones y caminos alternos que cumple el caso de uso para satisfacer las necesidades de los requerimientos:

Tabla No. 3.23: Escenarios - Registrar Gestiones Académicas.


Nombre del Caso de Uso:	UC_Registrar Gestiones Académicas
Notas:	Gestión que permite obtener el registro de las Notas Académicas, de los Impedimentos y de los valores pendientes de pago.
RESTRICCIONES:	
Pre-condition	N/A
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio:</p> <p>1. <u>AC_Personal Universitario</u> accede a las pantallas de registro de gestiones académicas que se encuentra en la base de pruebas <u>AC_BannerPruebas</u>.</p> <p style="text-align: center;">Usa: Interfaces de registro</p> <p>2. <u>AC_Personal Universitario</u> registra las gestiones académicas en la base de pruebas <u>AC_BannerPruebas</u>.</p> <p style="text-align: center;">Usa: Interfaces de registro</p> <p>3. <u>AC_BannerPruebas</u> guarda sobre una tabla temporal con el estado "Registrado"</p> <p style="text-align: center;">Usa: Web Service: RegistrarGestion</p> <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>

Caso de Uso - Obtener Información de Notificación.

En la *tabla No 3.24*, se presentan los escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.24: Escenarios - Obtener Información de Notificación.

Nombre del Caso de Uso:	CU - Obtener Información de Notificación
Notas:	Proceso que permite obtener los registros pendientes a enviar sobre las gestiones académicas. Las gestiones académicas que se enviará son: - Notas Académicas. - Impedimentos. - Estados de cuentas o valores pendientes de cobro.
RESTRICCIONES:	
Pre-condition	N/A
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <p>1. <u>AC_TreeOne</u> consulta por medio del TIMER los registros en estado "R" de la tabla Notificaciones Salientes <u>AC_BannerPruebas</u></p>

CONTINÚA 

<p>Tipo: Camino Básico</p>	<p>Usos: <i>TIMER runtask</i></p> <p>2. <u>AC_TreeOne</u> toma los registros en estado "R" y los coloca con el estado "E" en la <u>AC_BannerPruebas</u></p> <p>Usos: <i>Web Service</i></p> <p>3. <u>AC_TreeOne</u> valida si el IdBanner del registro de encuentra asociado a un suscriptor activo en <u>Suscriptores</u></p> <p>4. <u>AC_TreeOne</u> obtiene la información del suscriptor en <u>Suscriptores</u></p> <p>5. <u>AC_TreeOne</u> valida el Periodo Activo en <u>Periodos</u></p> <p>6. <u>AC_TreeOne</u> obtiene la información para notificar al <u>AC Estudiante</u></p> <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>
<p>Nombre: <i>Suscriptor no se encuentra Activo</i></p> <p>Tipo: <i>Excepción</i></p>	<p>Inicio</p> <p>1. <u>AC_TreeOne</u> consulta por medio del TIMER los registros en estado "R" de la tabla Notificaciones Salientes de <u>AC_BannerPruebas</u></p> <p>Usos: <i>TIMER runtask</i></p> <p>2. <u>AC_TreeOne</u> toma los registros en estado "R" y los coloca con el estado "E" en la <u>AC_BannerPruebas</u></p> <p>Usos: <i>Web Service</i></p> <p>3. <u>AC_TreeOne</u> valida si el IdBanner del registro de encuentra asociado a un suscriptor activo en <u>Suscriptores</u></p> <p>Excepción: Cuando [Suscriptor no se encuentra Activo] 3a_1. Actualiza el registro en la tabla Notificaciones Salientes con el estado "No Suscripto".</p> <p>Resultado: Suscriptor no se encuentra Activo complete.</p> <p>Fin</p>
<p>Nombre: <i>Periodo No Activo</i></p> <p>Tipo: <i>Excepción</i></p>	<p>Inicio</p> <p>1. <u>AC_TreeOne</u> consulta por medio del TIMER los registros en estado "R" de la tabla Notificaciones Salientes de <u>AC_BannerPruebas</u></p> <p>Usos: <i>TIMER runtask</i></p> <p>2. <u>AC_TreeOne</u> toma los registros en estado "R" y los coloca con el estado "E" en la <u>AC_BannerPruebas</u></p> <p>Usos: <i>Web Service</i></p> <p>3. <u>AC_TreeOne</u> valida si el IdBanner del registro de encuentra asociado a un suscriptor activo en <u>Suscriptores</u></p> <p>4. <u>AC_TreeOne</u> obtiene la información del suscriptor en <u>Suscriptores</u></p>

Nombre: <i>Periodo No Activo</i>	5. <u>AC_TreeOne</u> valida el Periodo Activo en <u>Periodos</u>
Tipo: <i>Excepción</i>	<p>Excepción: Cuando [Periodo No Activo] 5a_1. Actualiza el registro en la tabla Notificaciones Salientes con el estado "Periodo No Activo".</p> <p>Resultado: Periodo No Activo complete.</p> <p>Fin</p>

3.4.1.7 Proceso de Anuncios Académicos

Proceso por el cual los suscriptores Docentes y Personal, podrán registrar anuncios académicos. En la *figura No. 3.9*, se presenta el caso de uso de Anuncios Académicos.

Actores: Docente, Personal, Estudiante y Personal Universitario.

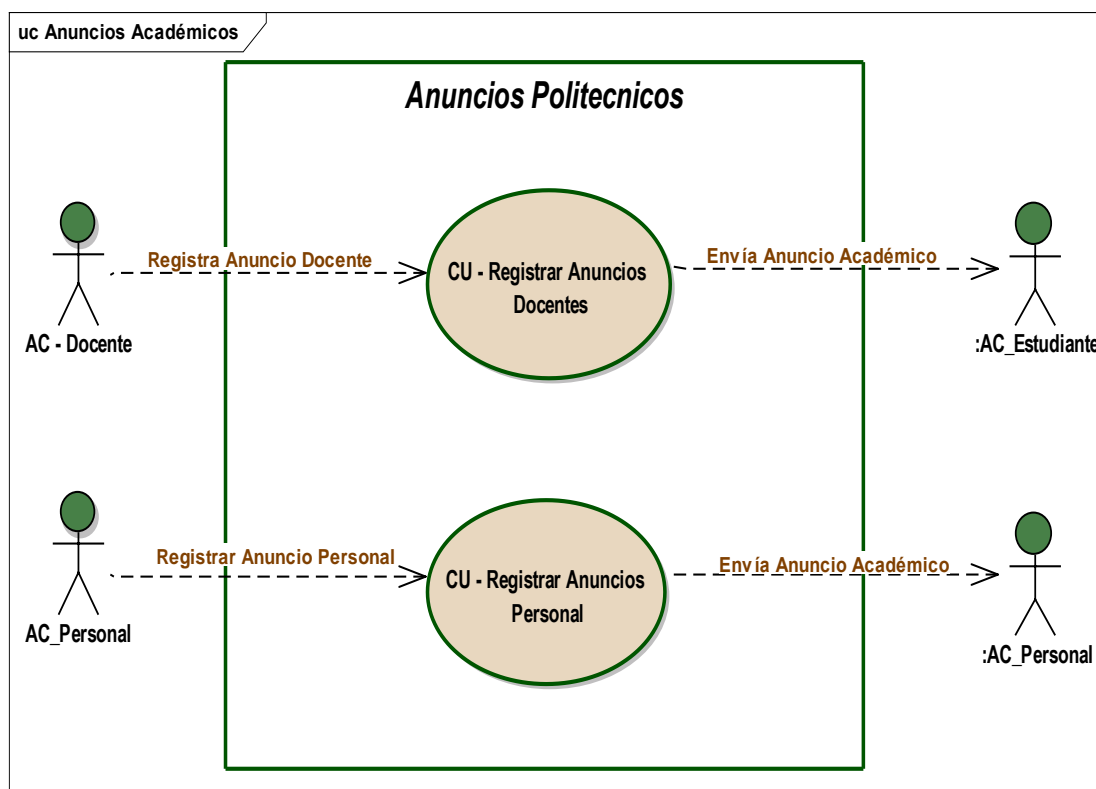



Figura No. 3.9: Caso de Uso - Anuncios Académicos.

Caso de Uso - Registrar Anuncios Docentes

En la *tabla No 3.25*, se presentan los escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.25: Escenarios - Registrar Anuncios Docentes.

Nombre del Caso de Uso:	CU - Registrar Anuncios Docentes
Notas:	<p>Proceso que permite el registro de Anuncios Académicos por parte del Suscriptor Docente hacia sus estudiantes suscriptores activos.</p> <p>El Docente podrá enviar anuncios a todos sus alumnos que se encuentran suscriptos en el servicio de comunicación, esto mediante la selección de, la o las materias que está a cargo en el periodo académico.</p>
RESTRICCIONES:	
Pre-condition	N/A
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. El <u>AC - Docente</u> accede a la <u>Pantalla RegistrarAnuncio</u> 2. <u>AC - Docente</u> pulsa el botón "Anuncio por Docente" de la <u>Pantalla RegistrarAnuncio</u> 3. La <u>Pantalla RegistrarAnuncio</u> presenta lista de Materias del <u>AC - Docente</u> <p>Usos: <i>Web Service:(obtenerMateriasPorDocente)</i></p> 4. <u>AC - Docente</u> selecciona una materia sobre la lista en la <u>Pantalla RegistrarAnuncio</u> 5. la <u>Pantalla RegistrarAnuncio</u> presenta lista de NRC's para el <u>AC - Docente</u> <p>Usos: <i>Web Service:(obtenerParalelosPorDocente)</i></p> 6. <u>AC - Docente</u> selecciona el o los NRC de la Materia seleccionada sobre la <u>Pantalla RegistrarAnuncio</u> 7. <u>Pantalla RegistrarAnuncio</u> solicita cargar alumnos suscriptos en <u>AC TreeOne</u> 8. <u>AC TreeOne</u> cargaSuscriptores alumnos en la <u>Pantalla RegistrarAnuncio</u> <p>Usos: <i>CargarSuscriptores</i></p> 9. El <u>AC - Docente</u> registra el asunto y contenido del anuncio sobre la <u>Pantalla RegistrarAnuncio</u> 10. El <u>AC - Docente</u> genera el anuncio sobre la <u>Pantalla RegistrarAnuncio</u> 11. <u>AC TreeOne</u> valida campos obligatorios sobre <u>Pantalla RegistrarAnuncio</u> <p>Usos: <i>ValidaCamposObligatorios</i></p> 12. <u>AC TreeOne</u> valida disponibilidad de los medios sobre <u>Pantalla RegistrarAnuncio</u> <p>Usos: <i>buscarVigentesMediosAnuncios: Twitter, Facebook, Correo personal y Correo ESPE</i></p>

CONTINÚA 

<p>Tipo: Camino Básico</p>	<p>13. <u>AC_TreeOne</u> envía el anuncio al <u>AC_Estudiente</u> suscriptor obtenido</p> <p>Usos: <i>Hilos por cada notificación enviada.</i></p> <p>14. <u>AC_TreeOne</u> guarda el anuncio sobre la tabla <u>Anuncios</u></p> <p>Usos: <i>Estado "Notificado"</i></p> <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>
<p>Nombre: Campos Obligatorios vacíos</p> <p>Tipo: Excepción</p>	<p>Inicio</p> <ol style="list-style-type: none"> 1. El <u>AC - Docente</u> accede a la <u>Pantalla_RegistrarAnuncio</u> 2. <u>AC - Docente</u> pulsa el botón "Anuncio por Docente" de la <u>Pantalla_RegistrarAnuncio</u> 3. La <u>Pantalla_RegistrarAnuncio</u> presenta lista de Materias del <u>AC - Docente</u> <p>Usos: <i>Web Service:(obtenerMateriasPorDocente)</i></p> <ol style="list-style-type: none"> 4. <u>AC - Docente</u> selecciona una materia sobre la lista en la <u>Pantalla_RegistrarAnuncio</u> 5. la <u>Pantalla_RegistrarAnuncio</u> presenta lista de NRC's para el <u>AC - Docente</u> <p>Usos: <i>Web Service:(obtenerParalelosPorDocente)</i></p> <ol style="list-style-type: none"> 6. <u>AC - Docente</u> selecciona el o los NRC de la Materia seleccionada sobre la <u>Pantalla_RegistrarAnuncio</u> 7. <u>Pantalla_RegistrarAnuncio</u> solicita cargar alumnos suscriptos en <u>AC_TreeOne</u> 8. <u>AC_TreeOne</u> cargaSuscriptores alumnos en la <u>Pantalla_RegistrarAnuncio</u> <p>Usos: <i>CargarSuscriptores</i></p> <ol style="list-style-type: none"> 9. El <u>AC - Docente</u> registra el asunto y contenido del anuncio sobre la <u>Pantalla_RegistrarAnuncio</u> 10. El <u>AC - Docente</u> genera el anuncio sobre la <u>Pantalla_RegistrarAnuncio</u> 11. <u>AC_TreeOne</u> valida campos obligatorios sobre <u>Pantalla_RegistrarAnuncio</u> <p>Usos: <i>ValidaCamposObligatorios</i></p> <p>Excepción: Cuando [Campos Obligatorios vacíos]</p> <p>11a_1. <u>AC_TreeOne</u> presenta un mensaje "Campos Obligatorios"</p> <p>Usos: <i>mensaje con componente Growl</i></p> <p>Resultado: Campos Obligatorios vacíos complete. Volver al punto 3.</p> <p>Fin</p>
<p>Nombre: Falló el envío de una Notificación</p> <p>Tipo: Excepción</p>	<p>Inicio</p> <ol style="list-style-type: none"> 1. El <u>AC - Docente</u> accede a la <u>Pantalla_RegistrarAnuncio</u>

<p>2. <u>AC - Docente</u> pulsa el botón "Anuncio por Docente" de la <u>Pantalla_RegistrarAnuncio</u></p> <p>3. La <u>Pantalla_RegistrarAnuncio</u> presenta lista de Materias del <u>AC - Docente</u></p> <p><u>Usos:</u> <i>Web Service:(obtenerMateriasPorDocente)</i></p> <p>4. <u>AC - Docente</u> selecciona una materia sobre la lista en la <u>Pantalla_RegistrarAnuncio</u></p> <p>5. la <u>Pantalla_RegistrarAnuncio</u> presenta lista de NRC's para el <u>AC - Docente</u></p> <p><u>Usos:</u> <i>Web Service:(obtenerParalelosPorDocente)</i></p> <p>6. <u>AC - Docente</u> selecciona el o los NRC de la Materia seleccionada sobre la <u>Pantalla_RegistrarAnuncio</u></p> <p>7. <u>Pantalla_RegistrarAnuncio</u> solicita cargar alumnos suscriptos en <u>AC_TreeOne</u></p> <p>8. <u>AC_TreeOne</u> cargaSuscriptores alumnos en la <u>Pantalla_RegistrarAnuncio</u></p> <p><u>Usos:</u> <i>CargarSuscriptores</i></p> <p>9. El <u>AC - Docente</u> registra el asunto y contenido del anuncio sobre la <u>Pantalla_RegistrarAnuncio</u></p> <p>10. El <u>AC - Docente</u> genera el anuncio sobre la <u>Pantalla_RegistrarAnuncio</u></p> <p>11. <u>AC_TreeOne</u> valida campos obligatorios sobre <u>Pantalla_RegistrarAnuncio</u></p> <p><u>Usos:</u> <i>ValidaCamposObligatorios</i></p> <p>12. <u>AC_TreeOne</u> valida disponibilidad de los medios sobre <u>Pantalla_RegistrarAnuncio</u></p> <p><u>Usos:</u> <i>buscarVigentesMediosAnuncios: Twitter, Facebook, Correo personal y Correo ESPE.</i></p> <p>13. <u>AC_TreeOne</u> envía el anuncio al <u>AC_Estudiante</u> suscriptor obtenido</p> <p><u>Usos:</u> <i>Hilos por cada notificación enviada.</i></p> <p><u>Excepción:</u> Cuando [Falló el envío de una Notificación] 13a_1. <u>AC_TreeOne</u> registra el error sobre la tabla <u>Anuncios</u></p> <p><u>Resultado:</u> Falló el envío de una Notificación complete.</p> <p><u>Fin</u></p>
--

Caso de Uso - Registrar Anuncios Personal

En la *tabla No. 3.26*, se presentan los escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.26: Escenarios - Registrar Anuncio Personal.

Nombre del Caso de Uso:	CU - Registrar Anuncios Personal
Notas:	Proceso que permite el registro de Anuncios Académicos por parte del Suscriptor Personal hacia todos los suscriptores personales activos en el servicio de comunicación.
RESTRICCIONES:	
Pre-condition	N/A
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC_Personal</u> accede a la <u>Pantalla_RegistrarAnuncio</u> 2. <u>AC_Personal</u> pulsa sobre el botón "Anuncio por Personal" de la <u>Pantalla_RegistrarAnuncio</u> 3. La <u>Pantalla_RegistrarAnuncio</u> presenta al <u>AC_Personal</u> la lista del Personal Suscripto <p>Usos: <i>cargarFiltroPersonal()</i></p> <ol style="list-style-type: none"> 4. <u>AC_Personal</u> selecciona el personal suscripto de la <u>Pantalla_RegistrarAnuncio</u> 5. El <u>AC_Personal</u> registra el asunto y contenido del anuncio sobre la <u>Pantalla_RegistrarAnuncio</u> 6. El <u>AC_Personal</u> genera el anuncio sobre la <u>Pantalla_RegistrarAnuncio</u> 7. <u>Pantalla_RegistrarAnuncio</u> solicita realizar validaciones en <u>AC_TreeOne</u> 8. <u>AC_TreeOne</u> valida que todos los campos obligatorios estén registrados sobre la <u>Pantalla_RegistrarAnuncio</u> 9. <u>AC_TreeOne</u> valida disponibilidad de medios. <p>Usos: <i>buscarVigentesMediosAnuncios: Twitter, Facebook, Correo personal y Correo ESPE.</i></p> <ol style="list-style-type: none"> 10. <u>AC_TreeOne</u> envía el anuncio al <u>AC_Personal</u> <p>Usos: <i>Hilos: SendThreadMail y SendThreadTwitter</i></p> <ol style="list-style-type: none"> 11. <u>AC_TreeOne</u> guarda el anuncio sobre la tabla <u>Anuncios</u> <p>Usos: <i>guardarAnuncio</i></p> <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>
Nombre: Campos Obligatorios vacíos	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC_Personal</u> accede a la <u>Pantalla_RegistrarAnuncio</u> 2. <u>AC_Personal</u> pulsa sobre el botón "Anuncio por Personal" de la <u>Pantalla_RegistrarAnuncio</u>
Tipo: Excepción	

<p>Nombre: Campos Obligatorios vacíos</p> <p>Tipo: Excepción</p>	<p>3. La <u>Pantalla_RegistrarAnuncio</u> presenta al <u>AC_Personal</u> la lista del Personal Suscripto</p> <p>Usos: <i>cargarFiltroPersonal()</i></p> <p>4. <u>AC_Personal</u> selecciona el personal suscripto de la <u>Pantalla_RegistrarAnuncio</u></p> <p>5. El <u>AC_Personal</u> registra el asunto y contenido del anuncio sobre la <u>Pantalla_RegistrarAnuncio</u></p> <p>6. El <u>AC_Personal</u> genera el anuncio sobre la <u>Pantalla_RegistrarAnuncio</u></p> <p>7. <u>Pantalla_RegistrarAnuncio</u> solicita realizar validaciones en <u>AC_TreeOne</u></p> <p>8. <u>AC_TreeOne</u> valida que todos los campos obligatorios estén registrados sobre la <u>Pantalla_RegistrarAnuncio</u></p> <p>Excepción: Cuando [Campos Obligatorios vacíos]</p> <p>8a_1. <u>AC_TreeOne</u> presenta un mensaje "Campos Obligatorios"</p> <p>Usos: <i>componente Growl</i></p> <p>Resultado: Campos Obligatorios vacíos complete. Volver al punto 3.</p> <p>Fin</p>
<p>Nombre: Falló el envío de una Notificación</p> <p>Tipo: Excepción</p>	<p>Inicio</p> <p>1. <u>AC_Personal</u> accede a la <u>Pantalla_RegistrarAnuncio</u></p> <p>2. <u>AC_Personal</u> pulsa sobre el botón "Anuncio por Personal" del la <u>Pantalla_RegistrarAnuncio</u></p> <p>3. La <u>Pantalla_RegistrarAnuncio</u> presenta al <u>AC_Personal</u> la lista del Personal Suscripto</p> <p>Usos: <i>cargarFiltroPersonal()</i></p> <p>4. <u>AC_Personal</u> selecciona el personal suscripto de la <u>Pantalla_RegistrarAnuncio</u></p> <p>5. El <u>AC_Personal</u> registra el asunto y contenido del anuncio sobre la <u>Pantalla_RegistrarAnuncio</u></p> <p>6. El <u>AC_Personal</u> genera el anuncio sobre la <u>Pantalla_RegistrarAnuncio</u></p> <p>7. <u>Pantalla_RegistrarAnuncio</u> solicita realizar validaciones en <u>AC_TreeOne</u></p> <p>8. <u>AC_TreeOne</u> valida que todos los campos obligatorios estén registrados sobre la <u>Pantalla_RegistrarAnuncio</u></p> <p>9. <u>AC_TreeOne</u> valida disponibilidad de medios.</p> <p>Usos: <i>buscarVigentesMediosAnuncios: Twitter, Facebook, Correo personal y Correo ESPE.</i></p>

	<p>10. <u>AC_TreeOne</u> envía el anuncio al <u>AC_Personal</u></p> <p>Usos: Hilos: <i>SendThreadMail</i> y <i>SendThreadTwitter</i></p> <p>Excepción: Cuando [Falló el envío de una Notificación] 10a_1. <u>AC_TreeOne</u> registra el error sobre la tabla <u>Anuncios</u></p> <p>Resultado: Falló el envío de una Notificación complete.</p> <p>Fin</p>
--	---

3.4.1.8 Proceso de PARAMETRIZACIÓN.

Proceso por el cual se configuran los parámetros del Actores del Servicio de Comunicación. En la *figura No. 3.10* se presenta el caso de uso Parametrización.

Actores: Administrador y Treeone.

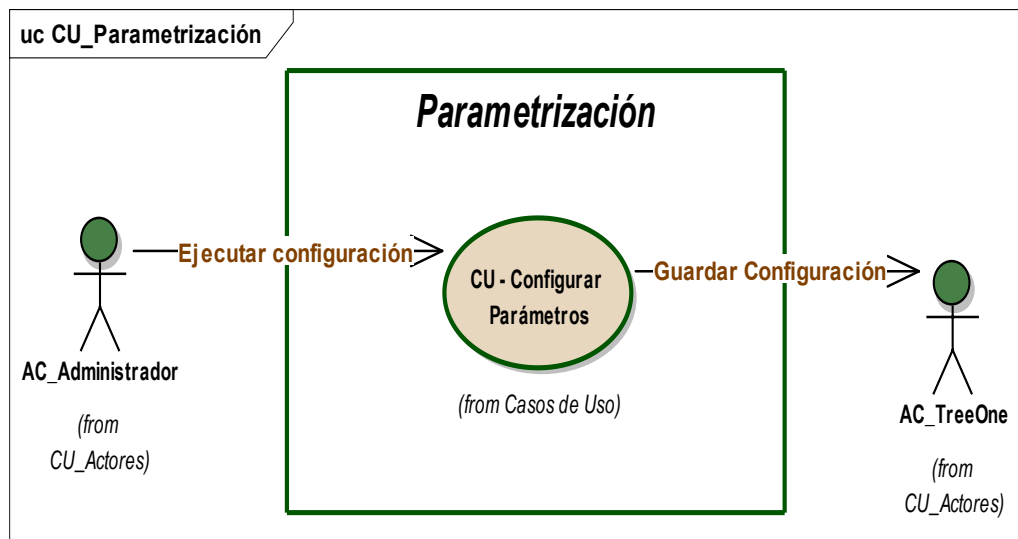


Figura No. 3.10: Caso de Uso - Parametrización.

Caso de Uso - Configurar Parámetros

En la *tabla No. 3.27*, se presentan los distintos escenarios, excepciones y caminos alternos que cumple el caso de uso para satisfacer las necesidades de los requerimientos:

Tabla No. 3.27: Escenarios - Configurar Parámetros.

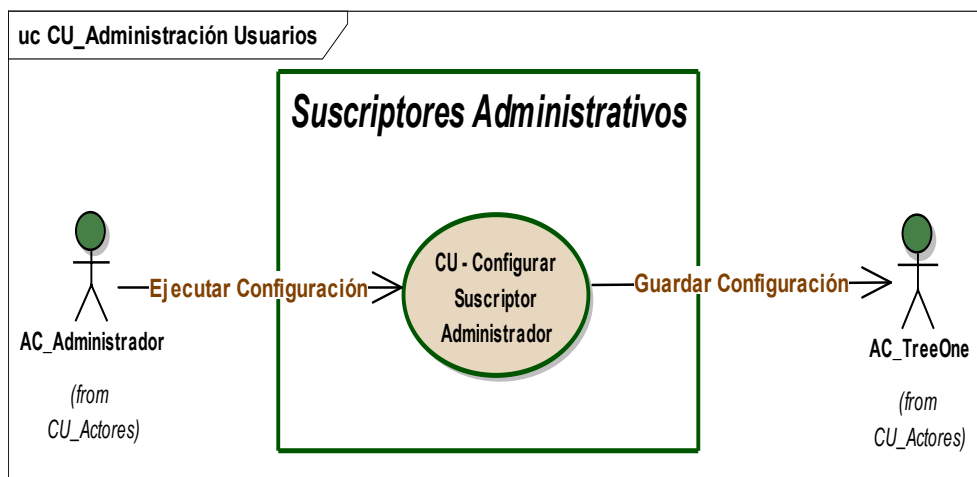
Nombre del Caso de Uso:	CU - Configurar Parámetros
Notas:	Proceso que permite configurar los parámetros del servicio TREEONE.
RESTRICCIONES:	
Pre-condition	N/A
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. El <u>AC_Suscriptor</u> accede a pantalla de configuración de parámetros del Servicio de Comunicación. 2. El <u>AC_Suscriptor</u> edita la información de los parámetros configurados. 3. <u>AC_TreeOne</u> guarda y presenta un mensaje de "Guardado Exitosamente". <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>

Proceso de ADMINISTRACIÓN USUARIOS

Proceso por el cual se configuran los suscriptores como administradores del Servicio de Comunicación

En la *figura No. 3.11*, se presenta el caso de uso Administración Usuarios.

Actores: Administrador y Treeone.


**Figura No. 3.11:** Caso de Uso - Administración Usuarios.

Caso de Uso - Configurar Suscriptor Administrador

En la *tabla No. 3.28*, se presentan los escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.28: Escenarios - Configurar Suscriptor Administrador.

Nombre del Caso de Uso:	CU - Configurar Suscriptor Administrador
Notas:	Permite la administración para la creación y eliminación de usuarios suscriptores del sistema
RESTRICCIONES:	
Pre-condition	N/A
ESCENARIOS:	
Tipo: <i>Camino Básico</i>	<p>Inicio</p> <ol style="list-style-type: none"> 1. El <u>AC Administrador</u> accede a la pantalla de administración de suscriptores Administradores. 2. El <u>AC Administrador</u> selecciona un suscriptor de una lista de suscriptores activos. 3. El <u>AC Administrador</u> selecciona el botón de agregar suscriptor como administrador. 4. <u>AC TreeOne</u> guarda y presenta mensaje "Suscriptor creado como Administrador". <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>
<p>Nombre: <i>Eliminación de Suscriptor Administrador</i></p> <p>Tipo: <i>Camino Alterno</i></p>	<p>Inicio</p> <ol style="list-style-type: none"> 1. El <u>AC Administrador</u> accede a la pantalla de administración de suscriptores Administradores. <p>Alternos: Cuando [Eliminación de Suscriptor Administrador]</p> <p>2a_1. El <u>AC Administrador</u> selecciona el botón de eliminación del suscriptor como administrador.</p> <p>2a_2. <u>AC TreeOne</u> guarda y presenta mensaje "Suscriptor eliminado como Administrador".</p> <p>Resultado: Eliminación de Suscriptor Administrador Completo.</p> <p>Fin</p>
<p>Nombre: <i>Cuentas Ya Registradas</i></p> <p>Tipo: <i>Excepción</i></p>	<p>Inicio</p> <ol style="list-style-type: none"> 1. <u>AC TreeOne</u> presenta la pantalla para el registro de los datos de contactos del Suscriptor 2. <u>AC TreeOne</u> presenta la información del correo ESPE. 3. <u>AC Personal Universitario</u> registra número móvil, cuenta de correo personal, cuenta TWITTER y cuenta FACEBOOK.

CONTINÚA 

<p>Nombre: Cuentas Ya Registradas</p> <p>Tipo: Excepción</p>	<p>Alternativo:</p> <p>Cuando [Cuentas Ya Registradas]</p> <p>4a_1. AC_TreeOne presenta mensaje "Cuenta ya se encuentra registrada".</p> <p>Resultado:</p> <p>Cuentas Ya Registradas Completo.</p> <p>Retorna al punto 3.</p> <p>Fin</p>
--	---

3.4.1.9 Proceso de ADMINISTRACIÓN PERIODOS

Proceso por el cual se crean periodos académicos. Los periodos académicos se crean con el estado ABIERTO y si lo eliminan se actualizará al estado CERRADO. Debe existir un solo periodo activo. En la *figura No. 3.12*, se presenta el caso de uso Administración Periodos.

Actores: Administrador y Treeone.

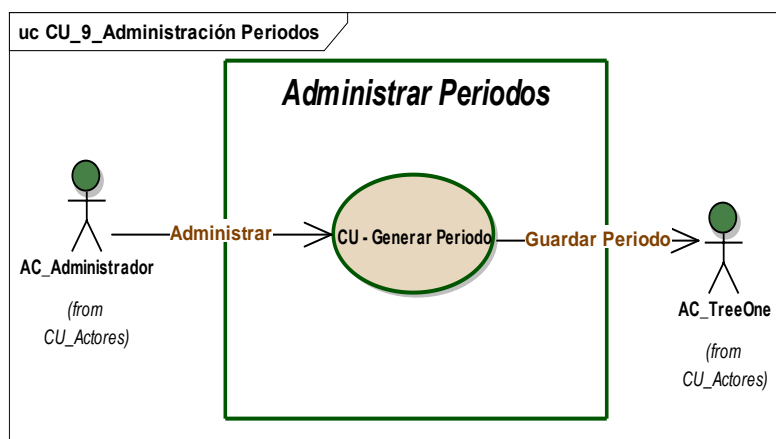


Figura No. 3.12: Caso de Uso - Administración Periodos.

Caso de Uso - Generar Periodo

En la *tabla No. 3.29*, se presentan los distintos escenarios, excepciones y caminos alternos que cumple el caso de uso para satisfacer las necesidades de los requerimientos:

Tabla No. 3.29: Escenarios - Generar Periodo.

Nombre del Caso de Uso:	CU - Generar Periodo
Notas:	Permite la Creación y Eliminación de periodos académicos.
RESTRICCIONES:	
Pre-condition	N/A
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. El <u>AC Administrador</u> accede a la pantalla de administración de Periodos. 2. El <u>AC Administrador</u> crea, edita y elimina un periodo académico. 3. <u>AC TreeOne</u> guarda y presenta un mensaje de "Guardado Exitosamente". <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>

3.4.1.10 Proceso de EDICIÓN DE CONTACTOS

Proceso por el cual se modifica los contactos creados. En la *figura No. 3.13*, se presenta el caso de uso Administración Contactos.

Actores: Personal Universitario y Treeone.

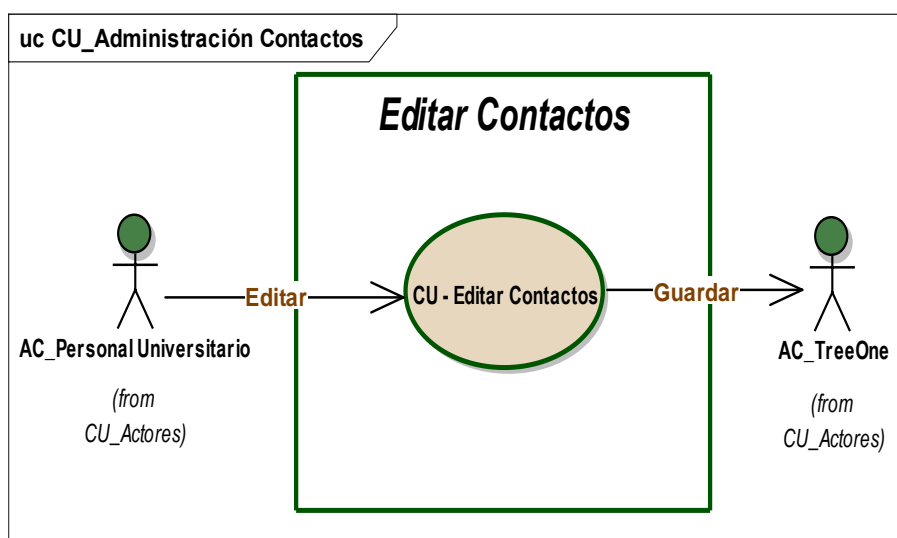


Figura No. 3.13: Caso de Uso - Administración Contactos.

Caso Uso - Editar Contactos

En la *tabla No. 3.30* se presentan los escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.30: Escenarios - Editar Contactos.

Nombre del Caso de Uso:	CU - Editar Contactos
Notas:	Permite la Edición de la información de contacto del suscriptor.
RESTRICCIONES:	
Pre-condition	Suscriptor Activo.
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. El <u>AC_Suscriptor</u> accede a pantalla para editar la información de los contactos registrados. 2. El <u>AC_Suscriptor</u> edita la información de los contactos. 3. <u>AC_TreeOne</u> guarda y presenta un mensaje de "Guardado Exitosamente". <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>

3.4.1.11 Proceso de CONFIGURACIÓN MÓDEM

Proceso por el cual se configuran los parámetros del Módem, activación y desactivación del mismo. Ver *figura No. 3.14*.

Actores: Administrador.

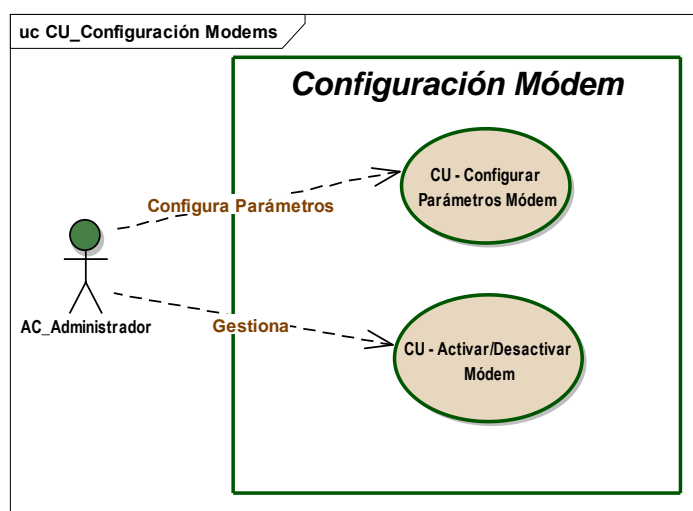


Figura No. 3.14: Caso de Uso - Configuración Módem.

Caso de Uso - Configurar Parámetros Módem

En la *tabla No. 3.31*, se presentan los escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.31: Escenarios - Configurar Parámetros Módem.


Nombre del Caso de Uso:	CU - Configurar Parámetros Módem
Notas:	Gestión que permite configurar los valores de conexión del módem.
RESTRICCIONES:	
Pre-condition	N/A
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. El <u>AC Administrador</u> accede a la pantalla de configuración del Módem. 2. El <u>AC Administrador</u> verifica y edita los datos de conexión del módem. 3. <u>AC TreeOne</u> guarda y presenta un mensaje de "Guardado Exitosamente". <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>

Proceso para Activar/Desactivar Módem

En la *tabla No. 3.32*, se presentan los escenarios, excepciones y caminos alternos del caso de uso:

Tabla No. 3.32: Escenarios - Activar/Desactivar Módem.

Nombre del Caso de Uso:	CU - Activar/Desactivar Módem
Notas:	Gestión que permite activar y desactivar el módem.
RESTRICCIONES:	
Pre-condition	N/A
ESCENARIOS:	
Tipo: Camino Básico	<p>Inicio</p> <ol style="list-style-type: none"> 1. El <u>AC Administrador</u> accede a la pantalla de configuración del módem. 2. El <u>AC Administrador</u> selecciona el botón activar el módem. 3. <u>AC TreeOne</u> presenta mensaje "Módem Conectado". <p>Resultado: Camino Básico Completo.</p> <p>Fin</p>

CONTINÚA 

3.4.2 DIAGRAMAS DE ACTIVIDADES

Aquí se presentan los procesos de negocio como un flujo de trabajo a través de una serie de acciones descritas de los casos de uso.

Diagrama de Actividades - Acceder TREEONE

La representación del flujo de acciones se presenta en la figura No. 3.15.

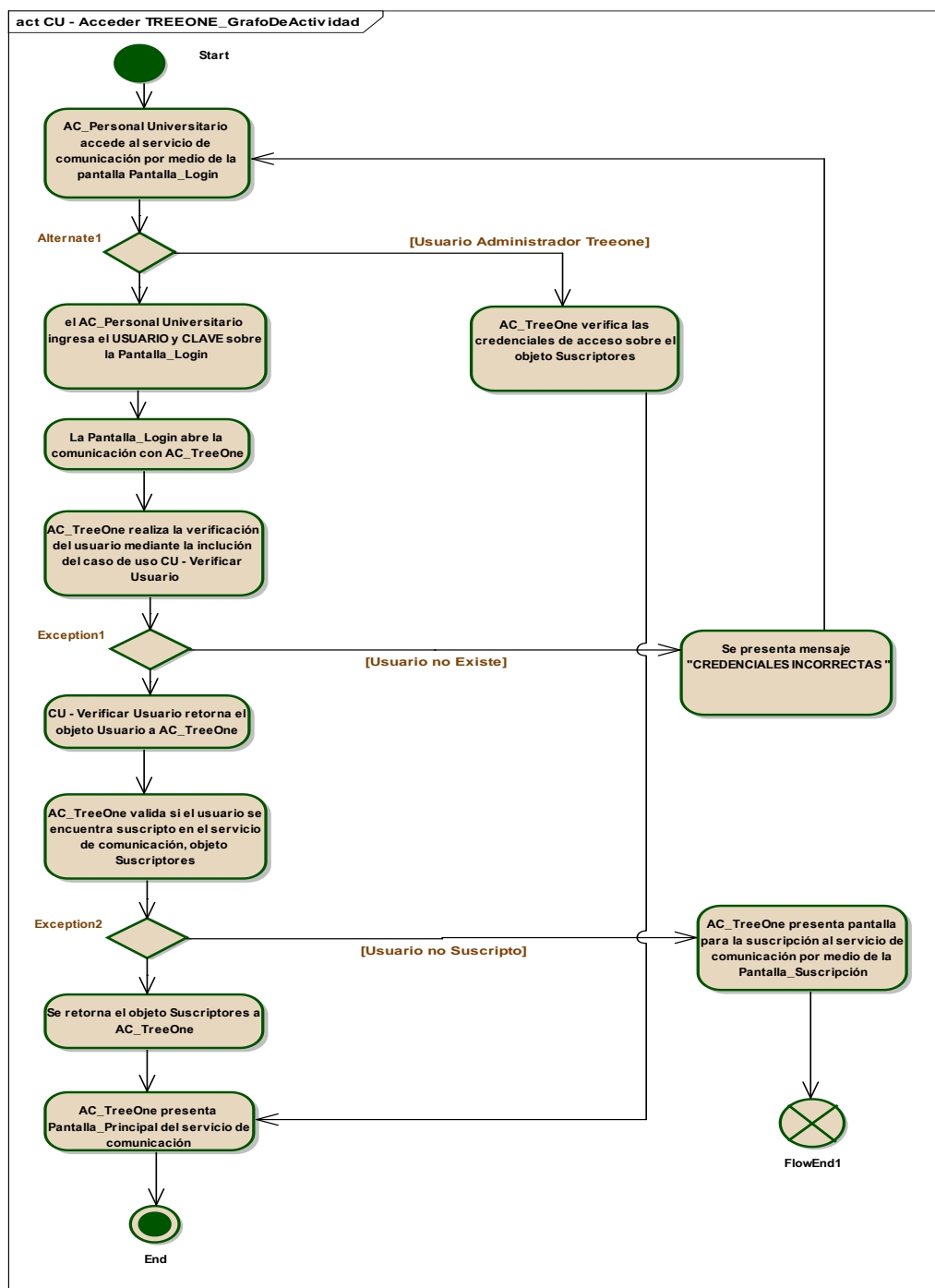


Figura No. 3.15: Diagrama Actividades - Acceder TREEONE.

Diagrama de Actividades - Verificar Usuario

La representación del flujo de acciones se presenta en la *figura No. 3.16*.

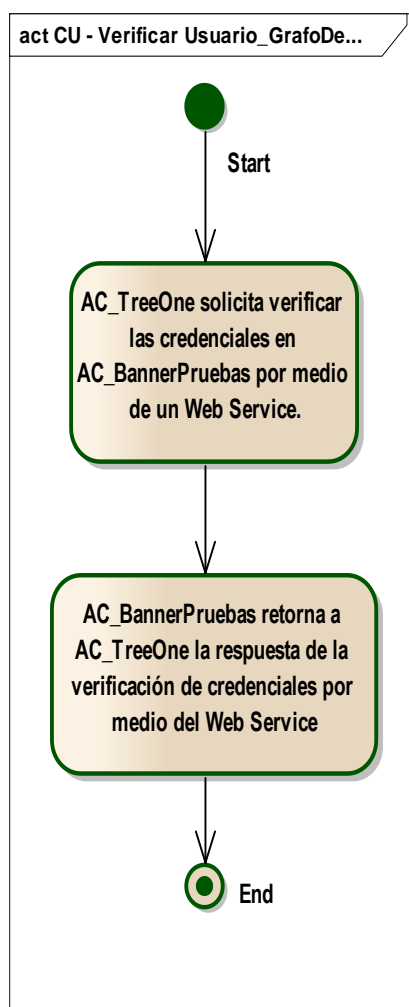


Figura No. 3.16: Diagrama Actividades - Verificar Usuario.

En esta representación se indica la forma en que se verifica las credenciales de autenticación al Servicio de Comunicación, en donde por medio de un Servicio Web, se obtiene la información del usuario que se encuentra registrado en la tabla de usuarios de la base de datos de pruebas Banner.

El usuario obtenido se lo utiliza para el proceso de suscripción al Servicio de Comunicación.

Diagrama de Actividades - Suscribir al servicio TREEONE

La representación del flujo de acciones se presenta en la *figura No. 3.17*.

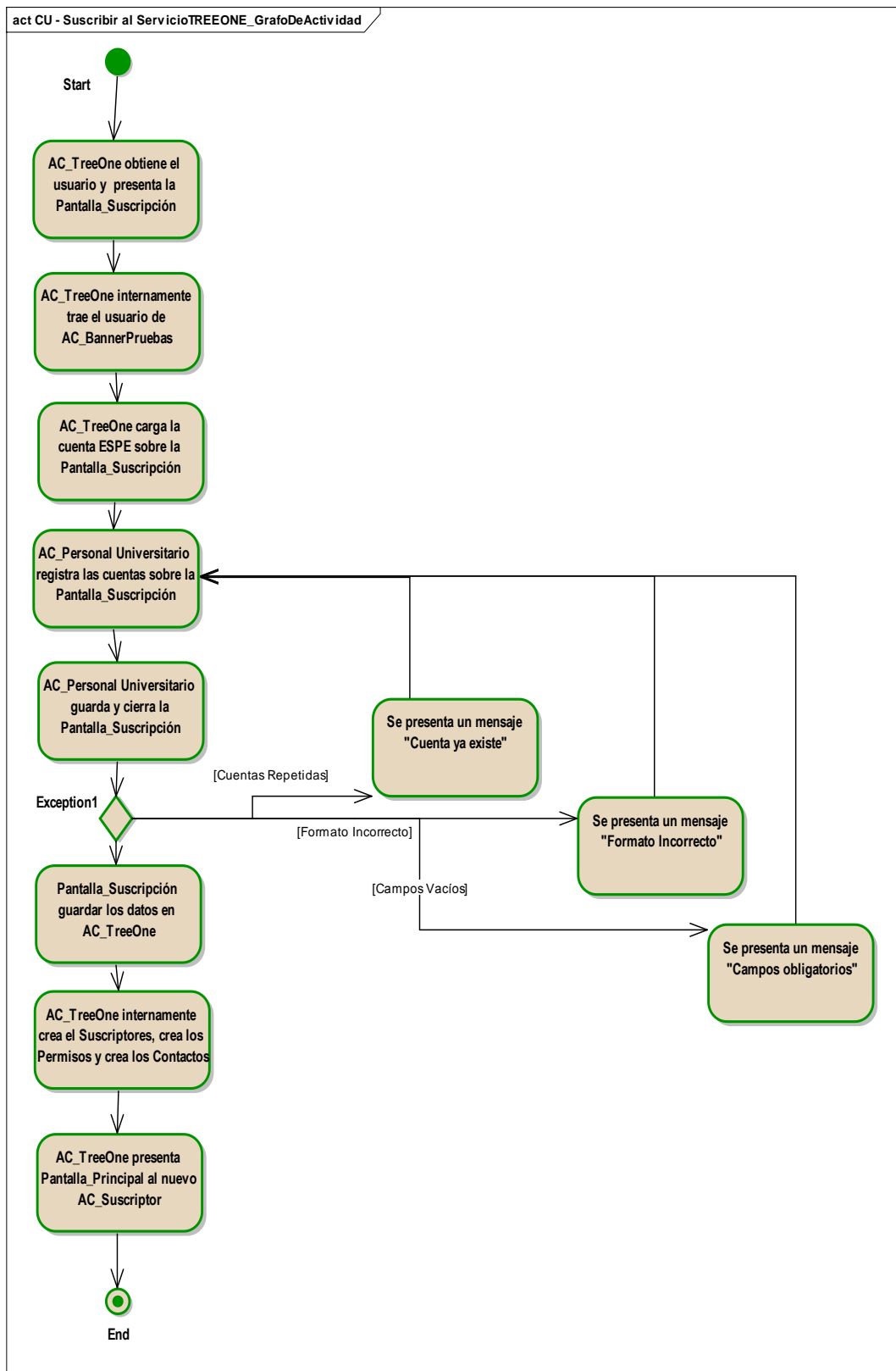


Figura No. 3.17: Diagrama Actividades - Suscribir a TREEONE.

Diagrama de Actividades - Suscribir al servicio TREEONE

La representación del flujo de acciones se presenta en la *figura No. 3.18*.

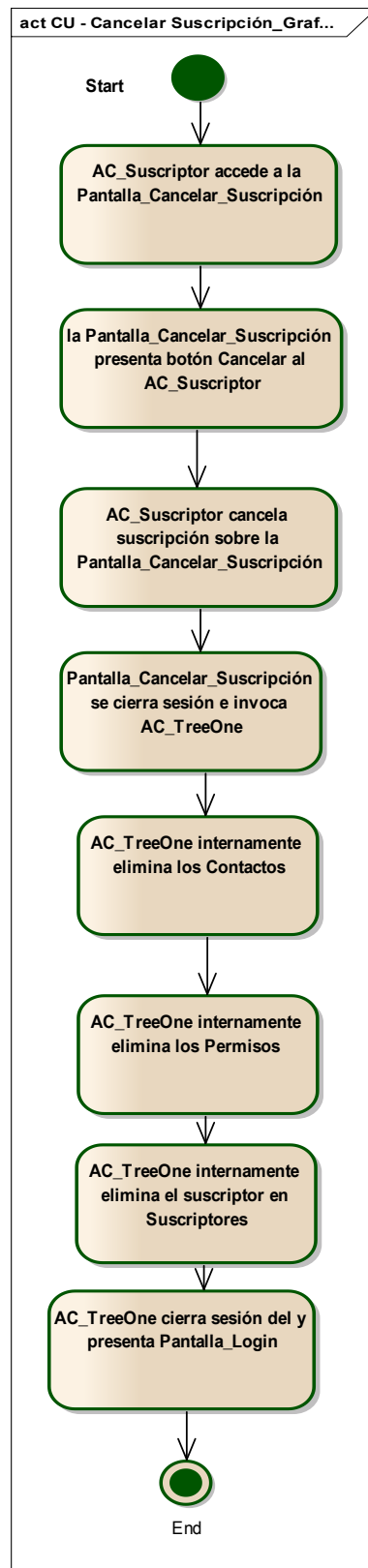


Figura No. 3.18: Diagrama Actividades - Cancelar Suscripción.

Diagrama de Actividades - Obtener Petición

La representación del flujo de acciones se presenta en la *figura No. 3.19*.

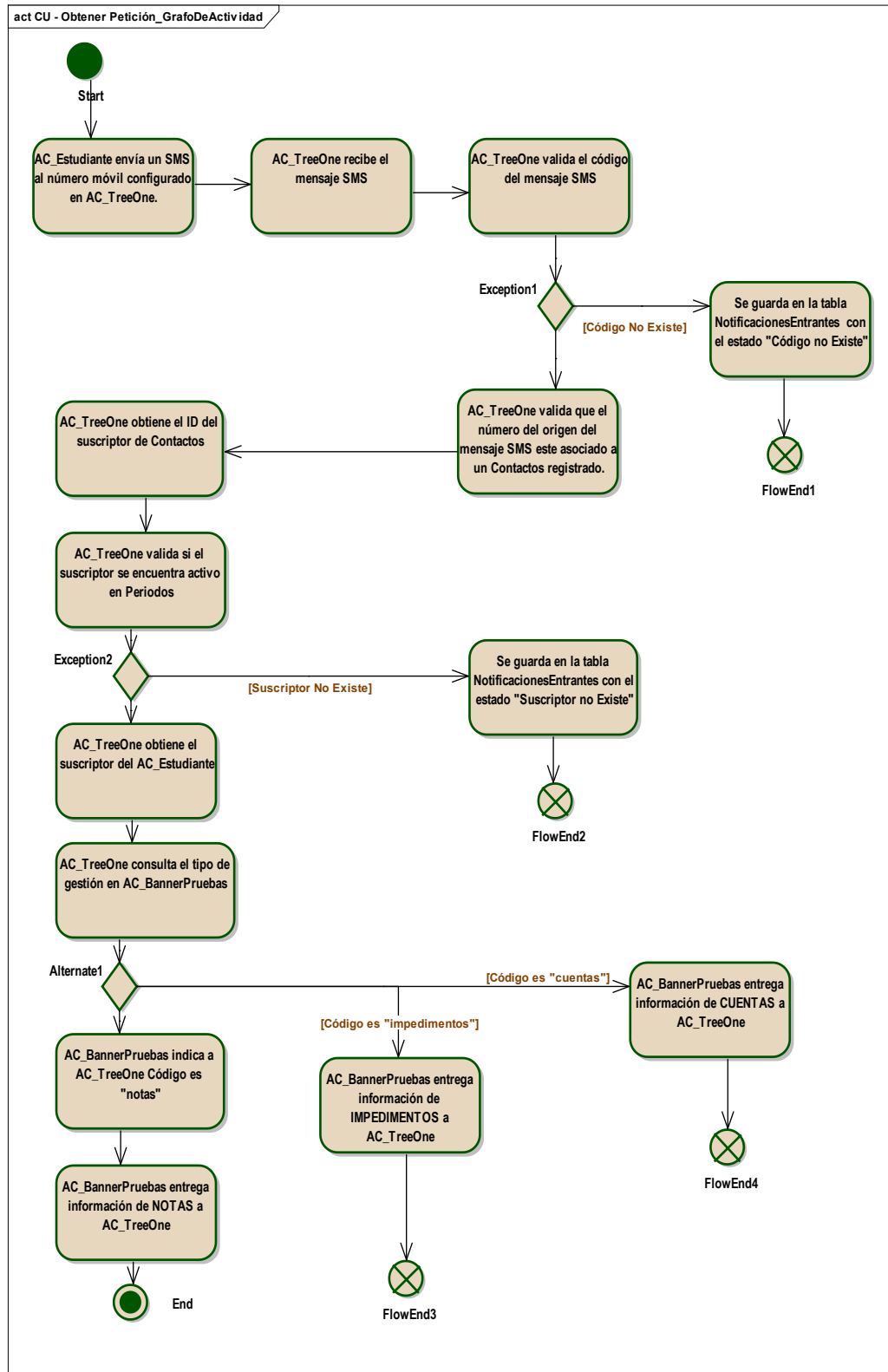


Figura No. 3.19: Diagrama Actividades - Obtener Petición.

Diagrama de Actividades - Enviar Mensaje SMS

La representación del flujo de acciones se presenta en la *figura No. 3.20*.

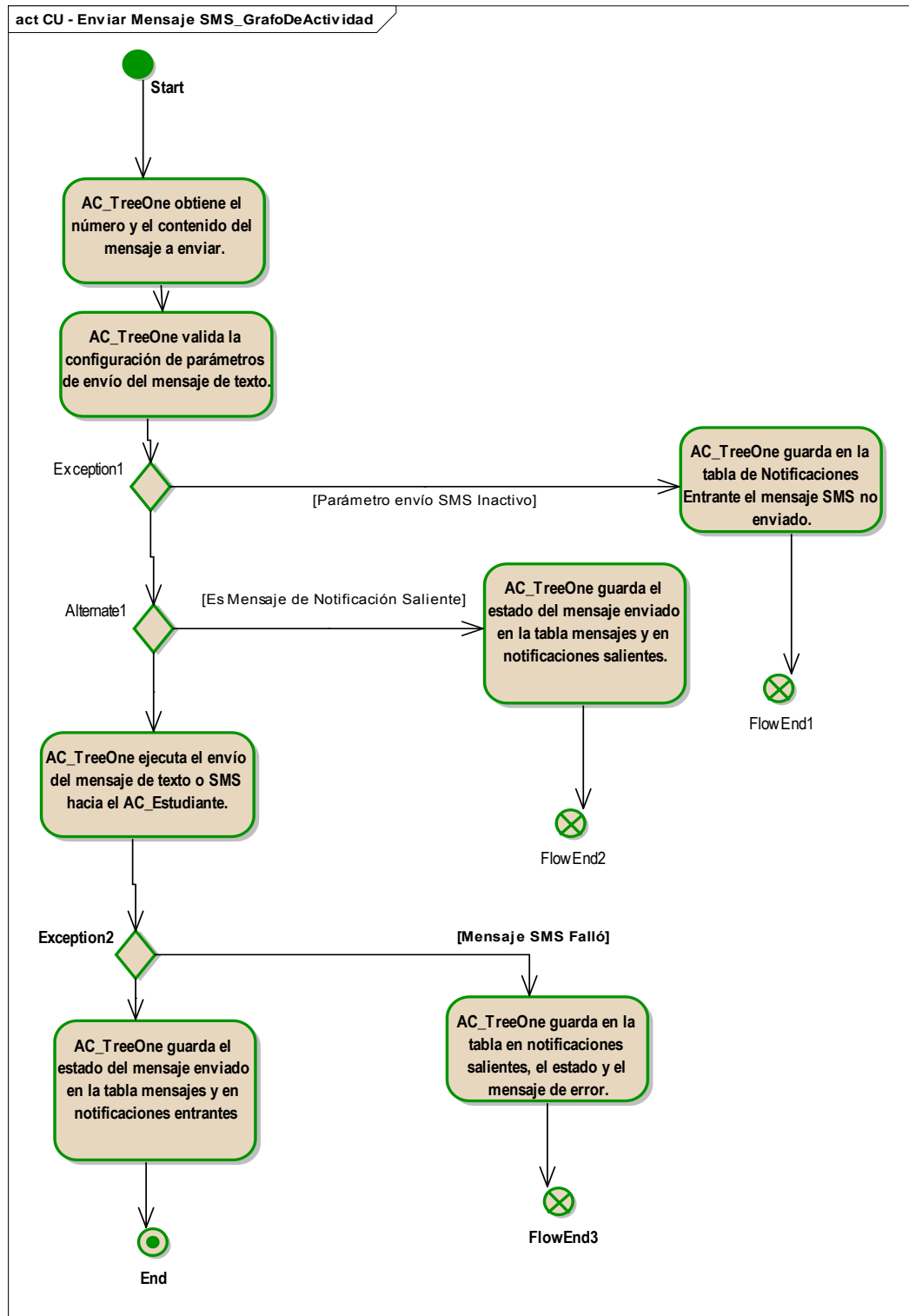


Figura No. 3.20: Diagrama Actividades - Enviar Mensaje SMS.

Diagrama de Actividades - Obtener Información de Notificación

La representación del flujo de acciones se presenta en la *figura No. 3.21*.

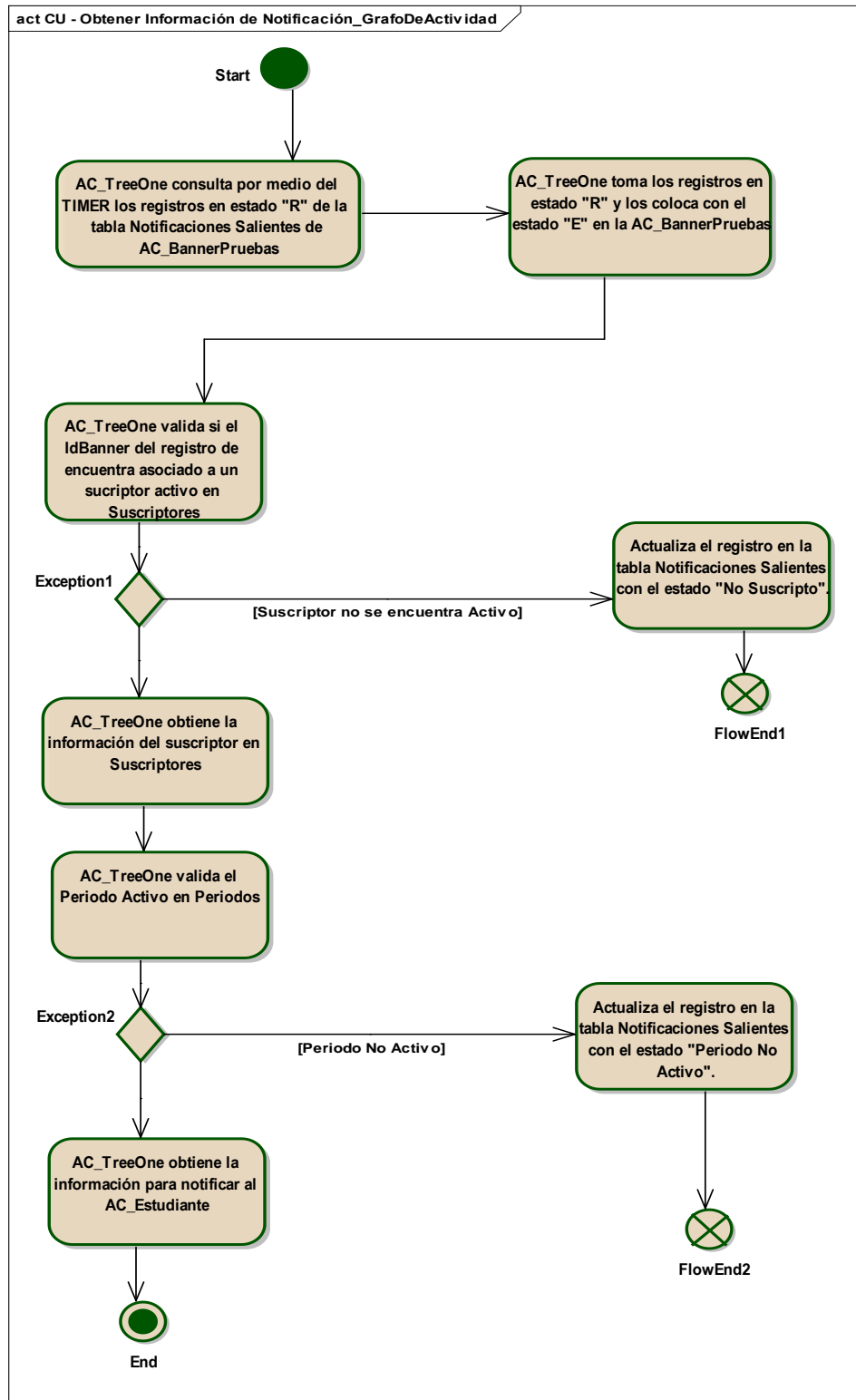


Figura No. 3.21: Diagrama Actividades - Obtener Información de Notificación.

Diagrama de Actividades - Registrar Anuncio Docentes

La representación del flujo de acciones se presenta en la *figura No. 3.22*.

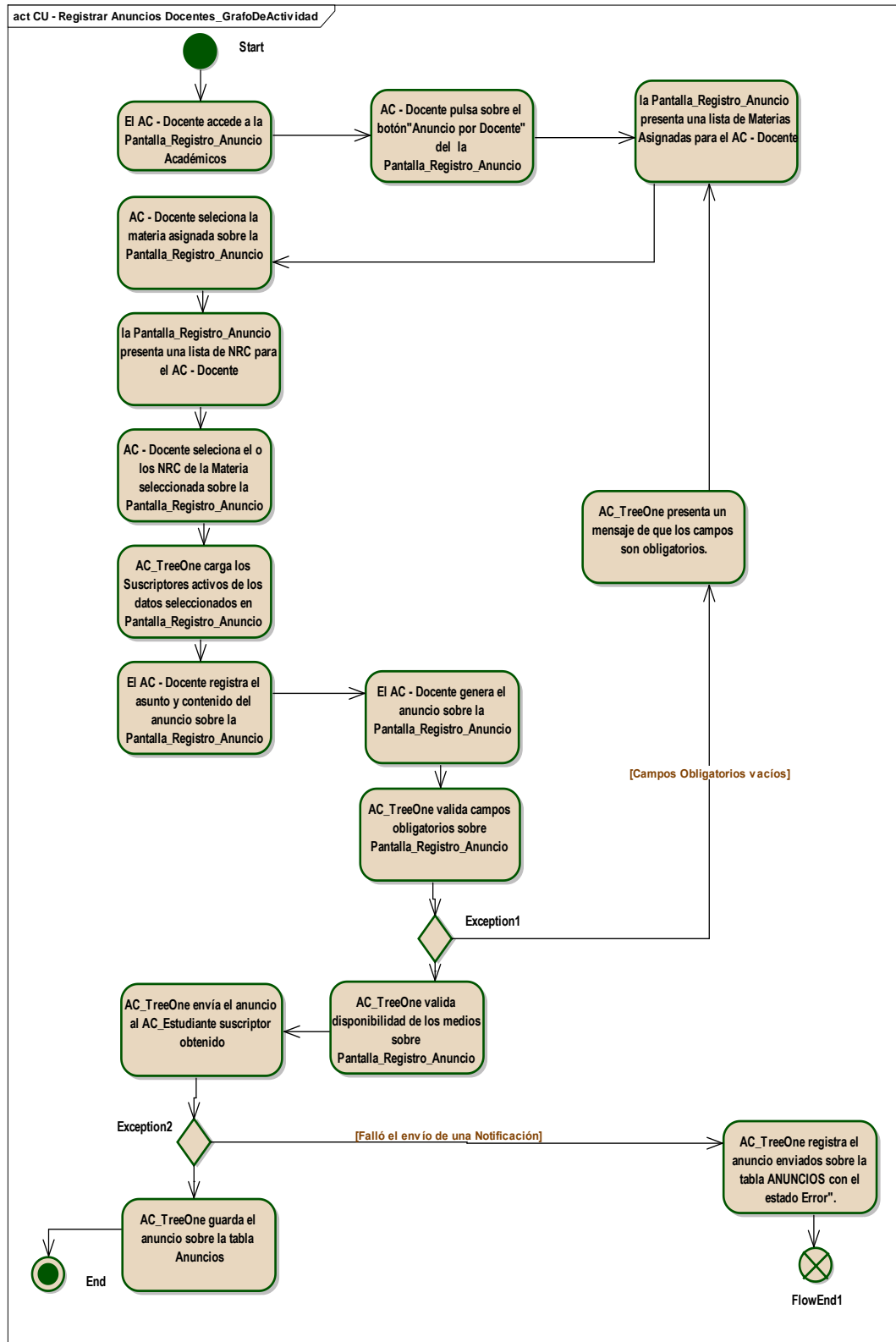


Figura No. 3.22: Diagrama Actividades - Registro Anuncio Docentes.

Diagrama de Actividades - Registrar Anuncio Personal

La representación del flujo de acciones se presenta en la *figura No. 3.23*.

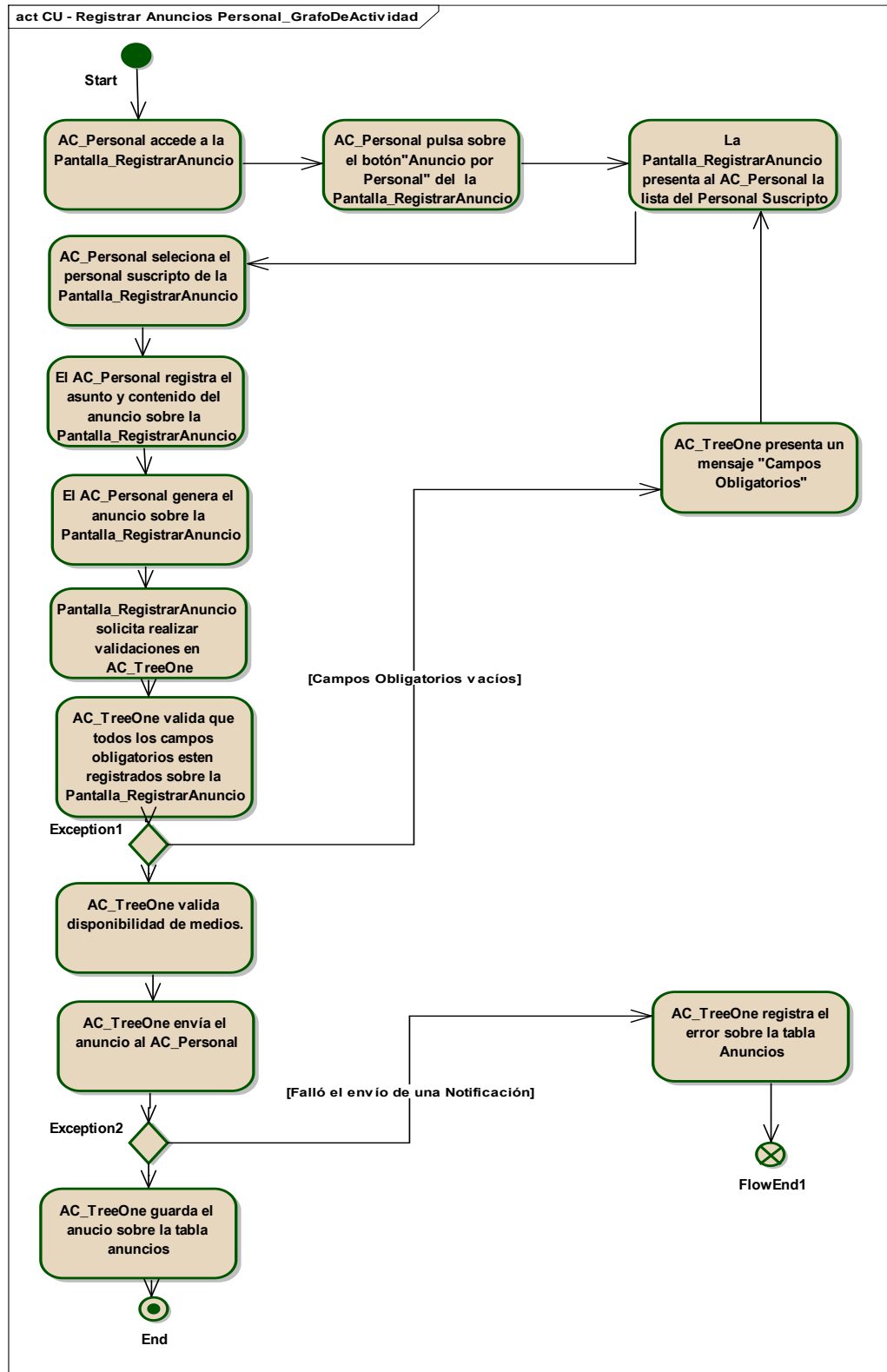


Figura No. 3.23: Diagrama Actividades - Registro Anuncio Personal.

3.4.3 Diagrama de Modelo Lógico

En la *figura No. 3.24*, se presenta el diagrama del modelo lógico del servicio de comunicación.

En este modelo se describen todos los objetos que propios del ambiente del Servicio de Comunicación y los objetos externos del ambiente de pruebas Banner que se son: notas_espe, pagos_espe e impedimentos_espe.

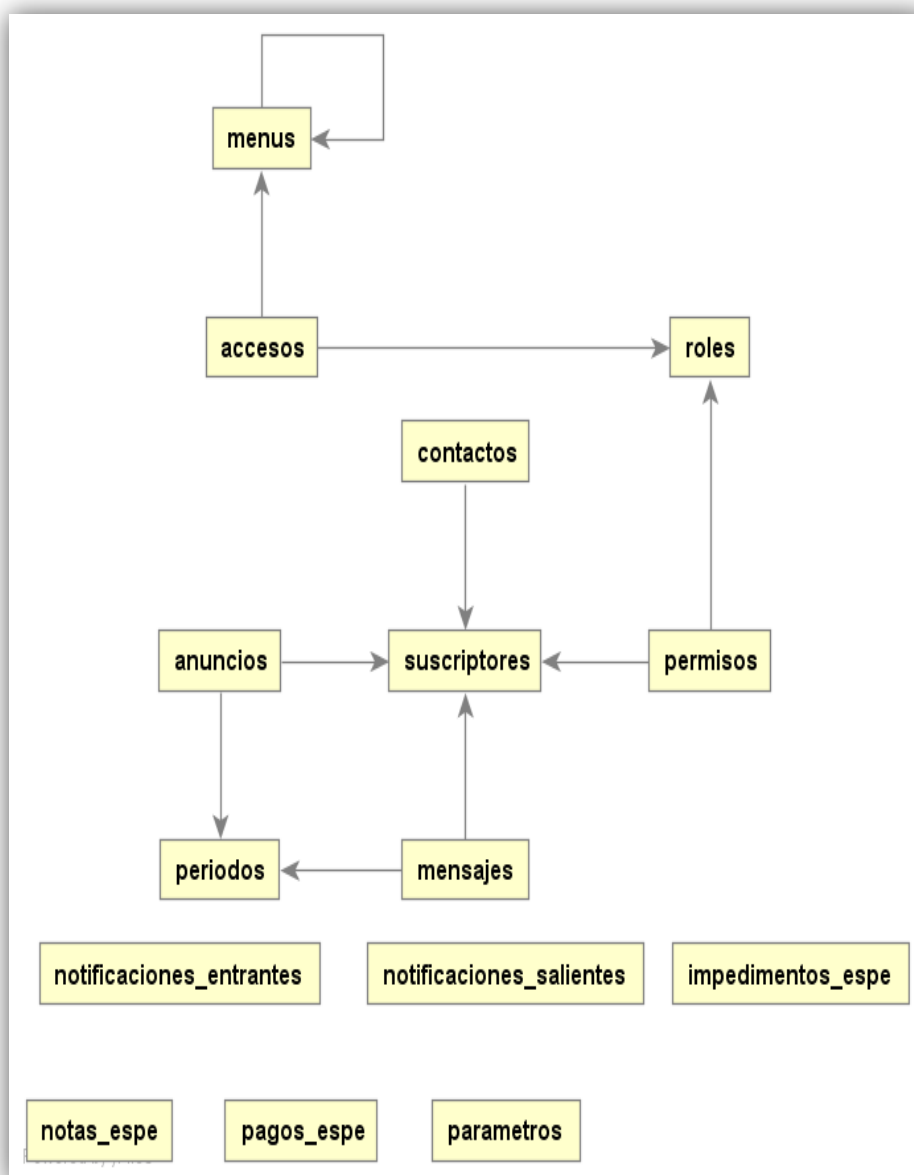


Figura No. 3.24: Diagrama - Modelo Lógico.

3.4.4 Diagrama de Modelo Físico.

En la *figura No. 3.25* se presenta el diagrama del modelo Físico del Servicio de Comunicación (Parte 1).

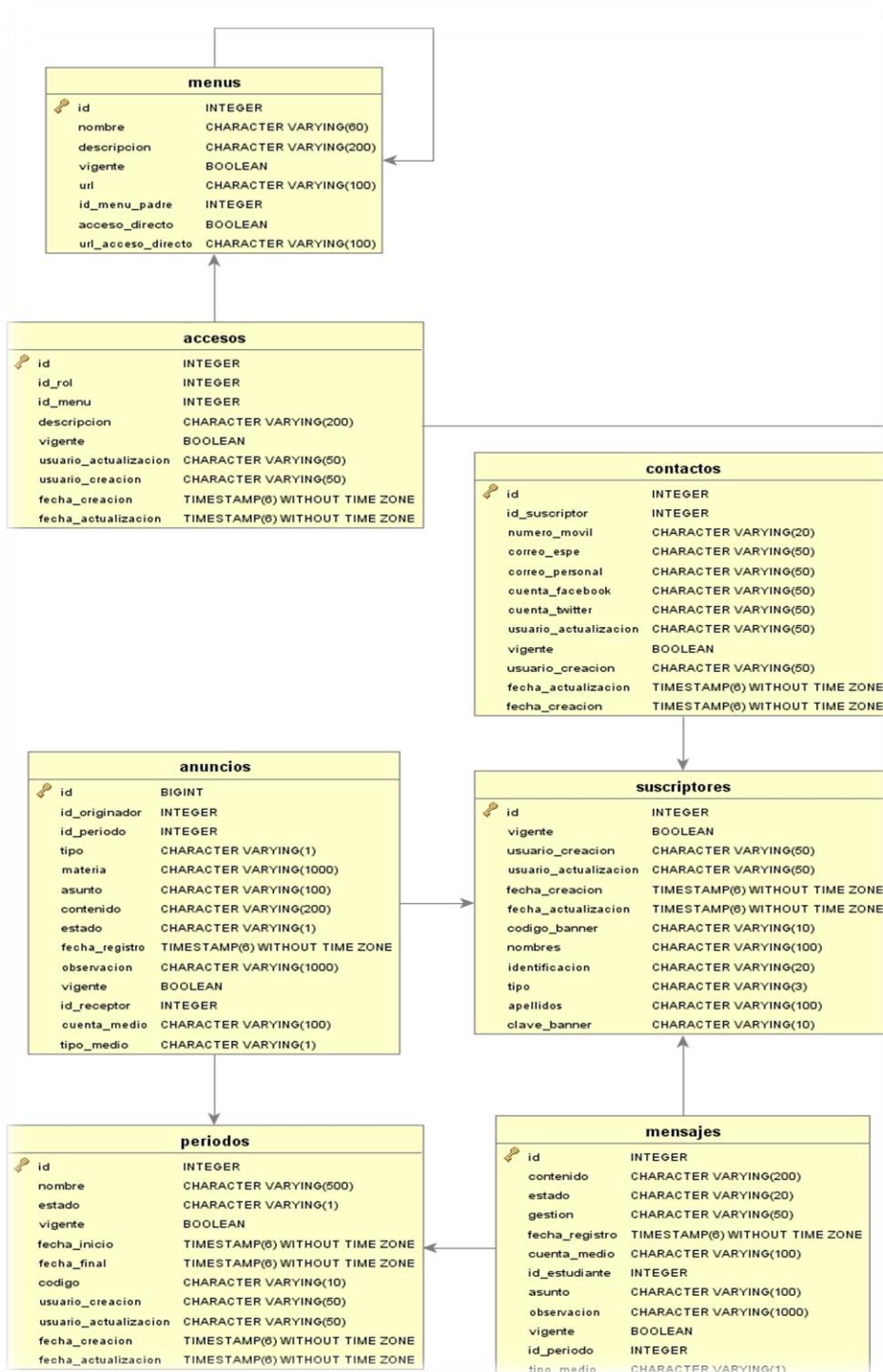


Figura No. 3.25: Diagrama - Modelo Físico Parte 1.

En la *figura No. 3.26* se presenta el diagrama del modelo Físico del Servicio de Comunicación (Parte 2).

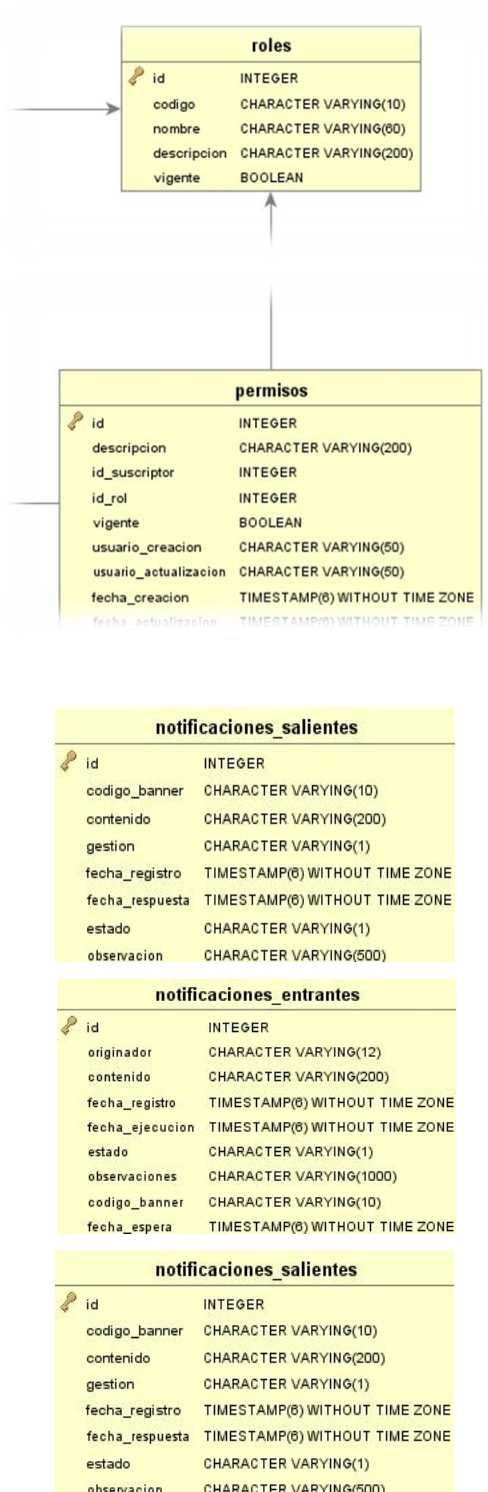


Figura No. 3.26: Diagrama - Modelo Físico Parte 2.

En la *figura No. 3.27* se presenta el diagrama del modelo Físico de la base de datos de pruebas Banner (Parte 3).



Figura No. 3.27: Diagrama - Modelo Físico Parte 3.

3.4.5 Arquitectura del Servicio de Comunicación

En la *figura No. 3.28* se presenta la arquitectura del Servicio de Comunicación, representada por una aplicación distribuida de tres niveles definida por un modelo MVC (Modelo-Vista-Controlador).

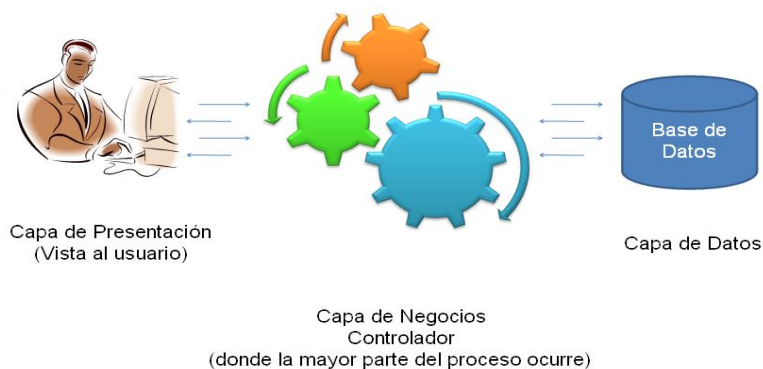


Figura No. 3.28: Arquitectura - MVC del Sistema.

Este patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar las tareas de desarrollo de aplicaciones.

En la *figura No. 3.29* se presenta la arquitectura del proyecto dividida en las tres capas del modelo.

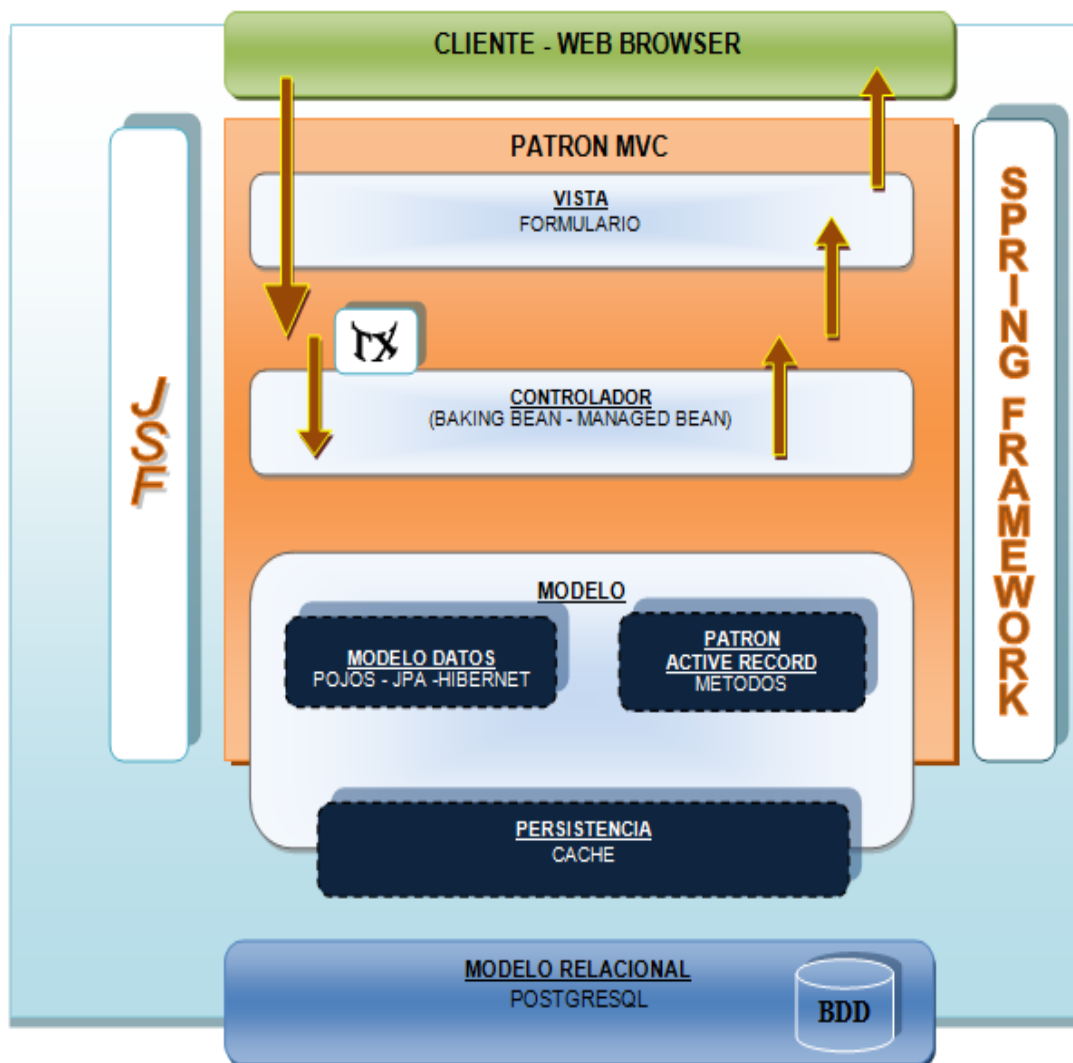


Figura No. 3.29: Arquitectura - Estructura General del Sistema.

- I. **Varias VISTAS:** son las páginas web que se muestran al usuario.
- Capa encargada de presentar la información hacia el usuario, permitiendo el manejo de peticiones y eventos.
 - Esta capa de presentación está desarrollado con los Frameworks JSF y PRIMEFACES (páginas de administración, páginas de gestión y páginas de consultas).

II. Varios CONTROLADORES: encargados de manejar las peticiones del usuario.

- Código que obtiene los datos dinámicamente y genera el contenido para el HTML.
- Esta capa maneja todos los procesos de negocio, el modelo y controladores del servicio de comunicación basados en Backing Bean y los Managed Bean.
- Además se tienen definidos las diferentes extensiones (BasicBean.java), interfaces (AdministradorRegistros.java) e implementaciones de las interfaces del Servicio de Comunicación (guardar(), iniciarVariablesNuevoRegistro(), iniciarVariablesEditarRegistro(), verListaRegistros(), borrar(), editar() y resetear()).
- En el controlador se maneja de manera transaccional o TX, se ejecuta todo o nada, esto se encuentra controlado por las excepciones que se ejecutan al presentarse un error sobre el bucle de desarrollo definido, si falla se ejecuta un rollback, caso contrario realiza el commit.
- Todo esto se encuentra soportado por SpringFramework.

III. Un MODELO: se encarga del almacenamiento y persistencia de datos.

- Esta capa es la encargada de la persistencia de datos, permitiendo las comunicaciones entre las relacionales y a las tablas orientadas a objetos, así como crear los procesos CRUD.
- Aquí se definen los POJOs de cada entidad (contacto.java), la persistencia y los métodos transaccionales como: persist (), merge (), remove (), que realizan operaciones automáticas de create, delete, update y select de consultas con o sin filtros.

- Se maneja cache para los casos en donde se generen las mismas consultas hacia la base datos, permitiendo mejorar lo tiempos de respuesta sobre la información solicitada.

El flujo de control se demuestra en los siguientes pasos:

1. El usuario realiza una acción en la interfaz (vista o web browser).
2. El controlador trata el evento de entrada.
3. El controlador notifica al modelo la acción del usuario, lo que puede implicar un cambio del estado del modelo.
4. Se realiza las diferentes acciones de persistencia.
5. Se genera una nueva vista, la vista toma los datos del modelo (el modelo no tiene conocimiento directo de la vista).
6. La interfaz de usuario espera otra interacción del usuario, para comenzar un nuevo ciclo.

3.4.6 Modelo de Clases.

En la *figura No. 3.30* se representa el modelo de Clases del Servicio de Comunicación, en donde se indican además los enumeradores utilizados en el proyecto.

Es este diagrama de clases se puede visualizar las relaciones entre las clases que se involucran, las cuales están compuestas por:

- atributos.
- métodos.
- enumeradores.
- relaciones.

También se indican los enumeradores, ya que se referencian a vistas pasos a seguir, estados, etc., ya que resultan ser fáciles de reconocer.

3.5. FASE DE CONSTRUCCIÓN

3.5.1 ELABORACIÓN DEL SISTEMA.

Para la construcción del Servicio de Comunicación, se debe contar con la instalación y configuración del servidor de aplicaciones (JBOSS), la Base de Datos (POSTGRESQL) con información propia del Servicio y la Creación de la Base de Datos de pruebas Banner (MYSQL), con sus respectivas tablas y datos de prueba.

La presentación de las diferentes interfaces que se maneja dentro del Servicio de comunicación, se debe referir al Anexo 2 "Manual de Usuarios" y Anexo 3 "Manual Administrativo".

3.5.2 DIAGRAMAS DE ESTADOS

Para poder identificar cada una de las rutas o caminos que puede tomar un flujo de información luego de ejecutarse cada proceso.

Se ha modelado los siguientes diagramas de estado, según los procesos de ejecución y cambio de estado del Servicio de Comunicación

3.5.2.1 Diagrama de Estados - Suscripción Académica

Diagrama que permite describir los cambios de estados entre los diferentes escenarios del caso de uso, indicando:

- Los cambios de estados al realizar la creación del usuario suscriptor.
- La creación de los datos de contactos y de los permisos respectivos para el acceso al Servicio de Comunicación.

En la *figura No. 3.31* se puede observar el cambio de estado de este caso de uso:

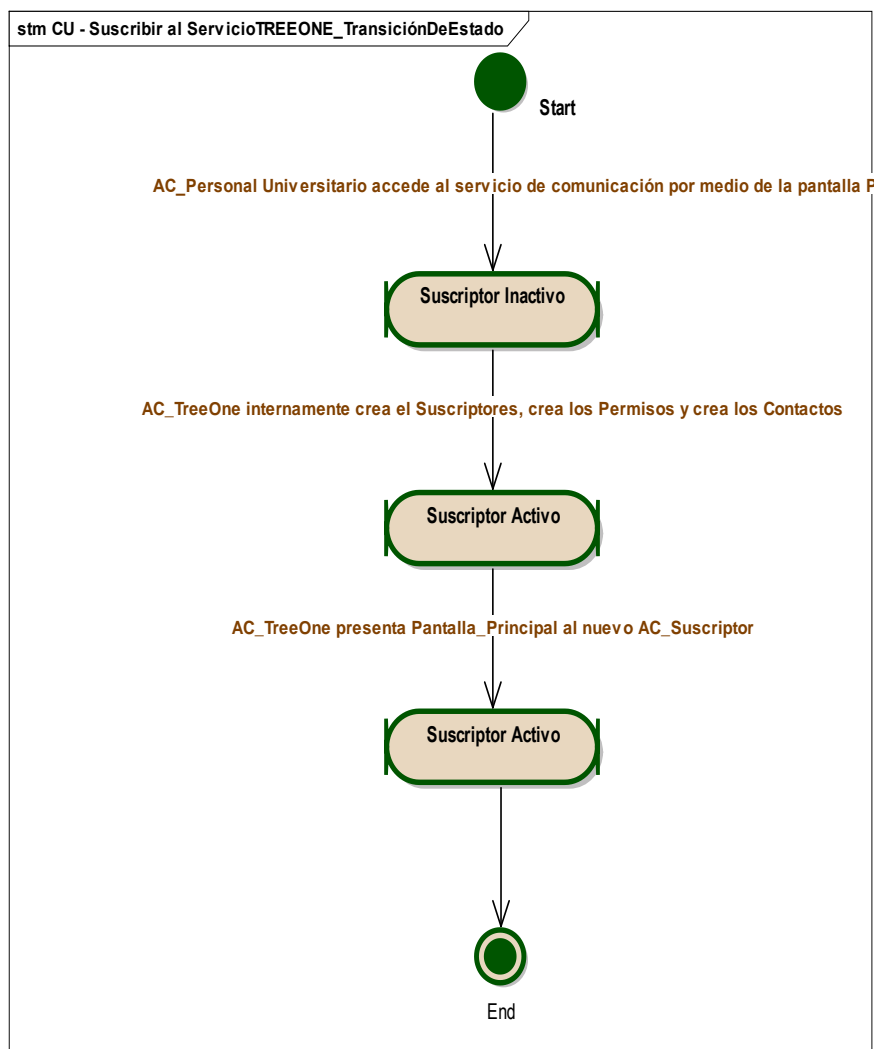


Figura No. 3.31: Diagrama de Estados - Suscripción Académica.

3.5.2.2 Diagrama de Estados - Cancelación Suscripción Académica

Diagrama que permite describir los cambios de estados entre los diferentes escenarios del caso de uso, indicando la eliminación del suscriptor, la eliminación de los datos de contactos y de los permisos respectivos.

El borrado es lógico y no físico.

En la *figura No. 3.32* se puede observar el cambio de estado de este caso de uso:

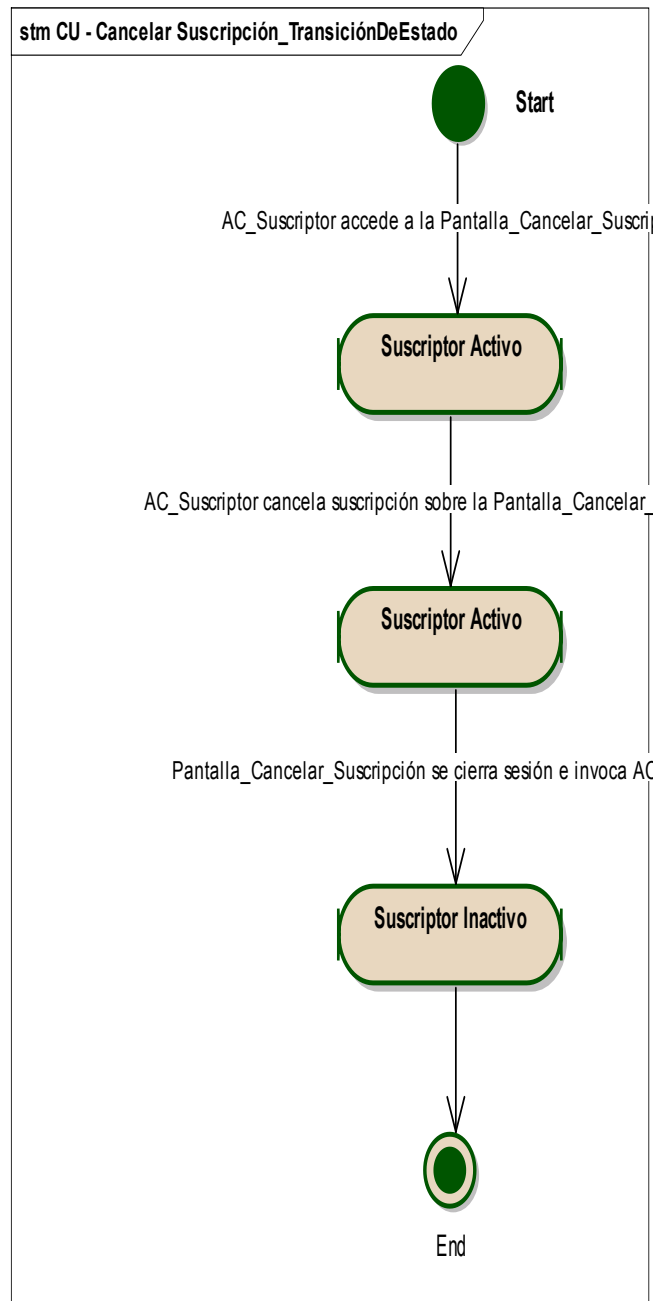


Figura No. 3.32: Diagrama de Estados - Cancelación Suscripción Académica.

3.5.3 DIAGRAMA DE SECUENCIAS

Los diagramas de secuencia de la aplicación se describen a continuación.

3.5.3.1 Diagrama de Secuencia - Verificar Usuario

Diagrama que permite definir la secuencia para el acceso y autenticación al servicio de comunicación. Ver *figura No. 3.33*.

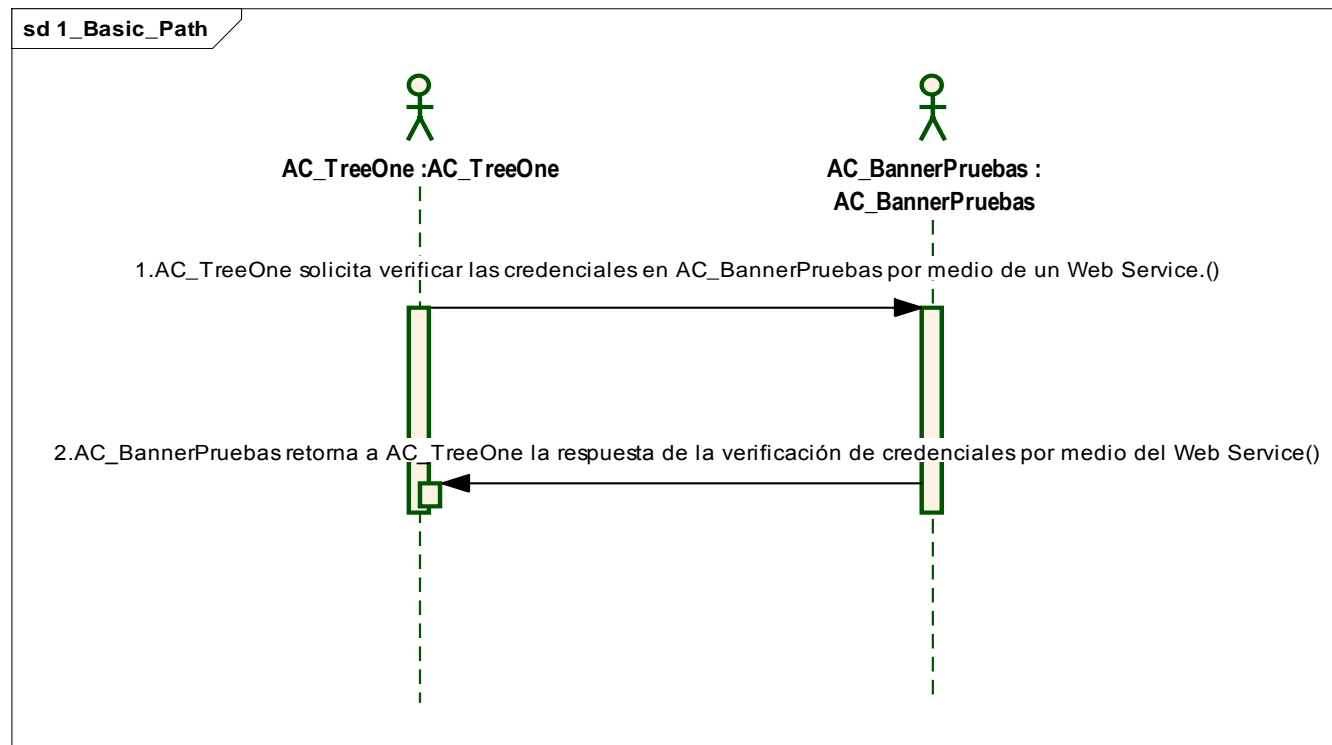


Figura No. 3.33: Diagrama de Secuencia - Autenticación.

Diagrama de Secuencia - Acceder TREEONE

Diagrama que permite definir la secuencia para el acceso y autenticación al servicio de comunicación. Ver *figura No. 3.34*.

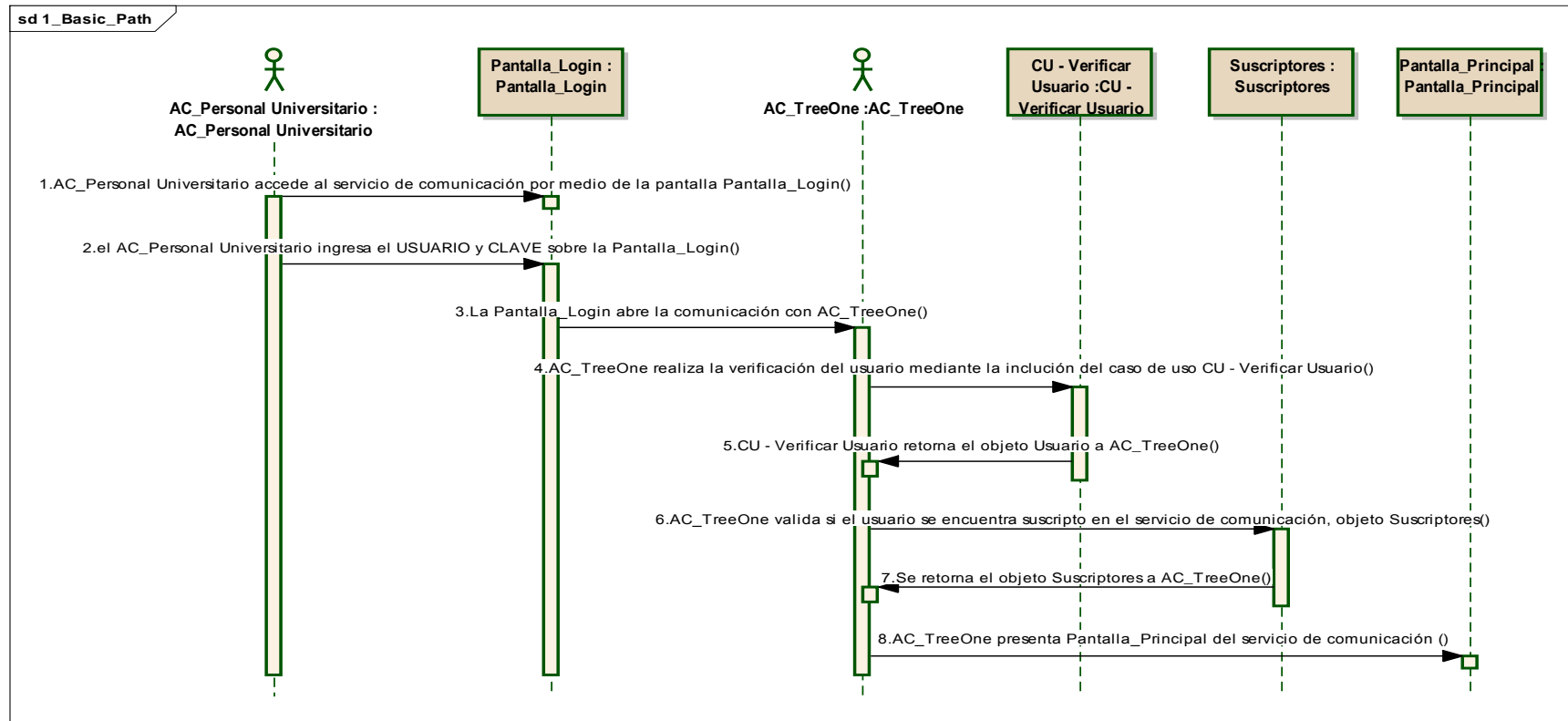


Figura No. 3.34: Diagrama de Secuencia - Autenticación.

3.5.3.2 Diagrama de Secuencia - Suscribir Servicio TREEONE

Diagrama que permite definir la secuencia para la suscripción al servicio de comunicación. Ver *figura No. 3.35*.

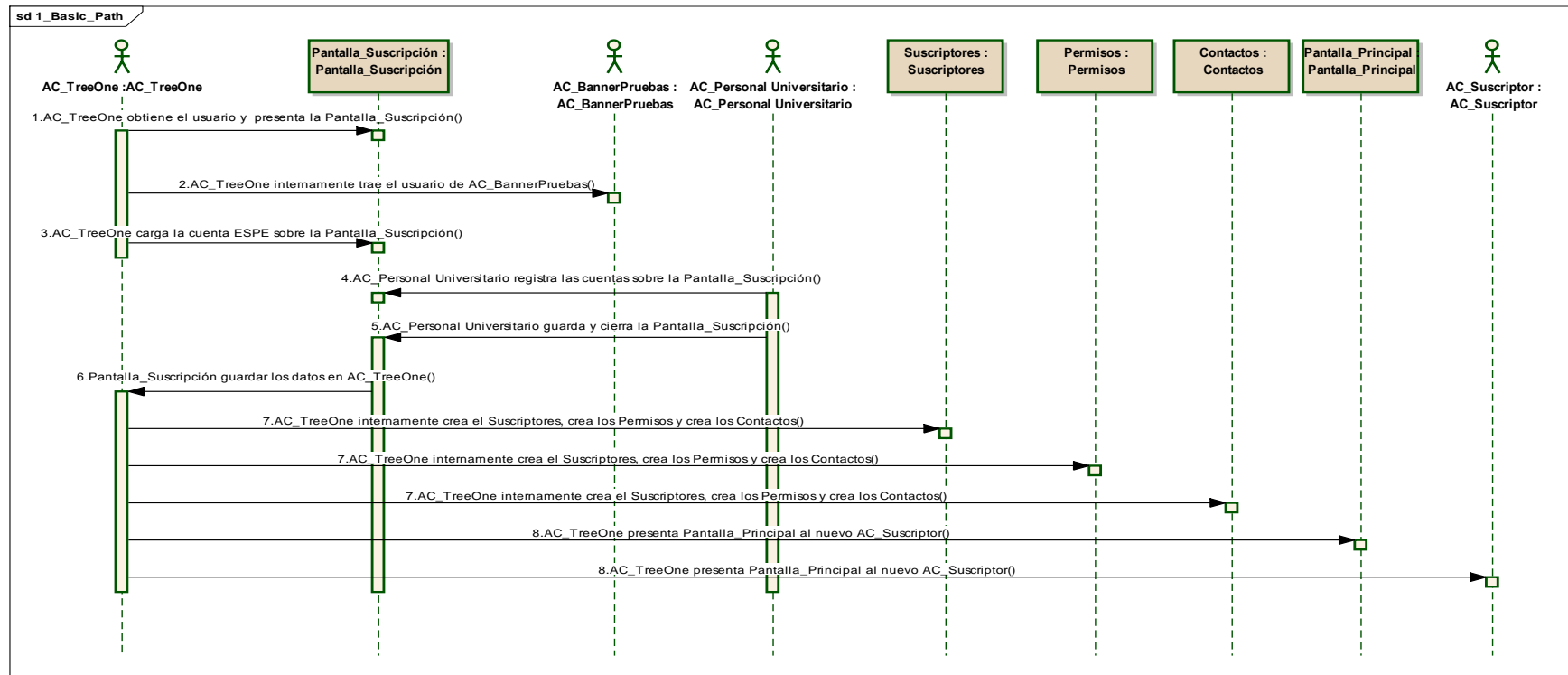


Figura No. 3.35: Diagrama de Secuencia - Suscripción Académica.

3.5.3.3 Diagrama de Secuencia - Cancelación Suscripción Académica

Diagrama que permite definir la secuencia para la cancelación Suscripción. Ver figura No. 3.36.

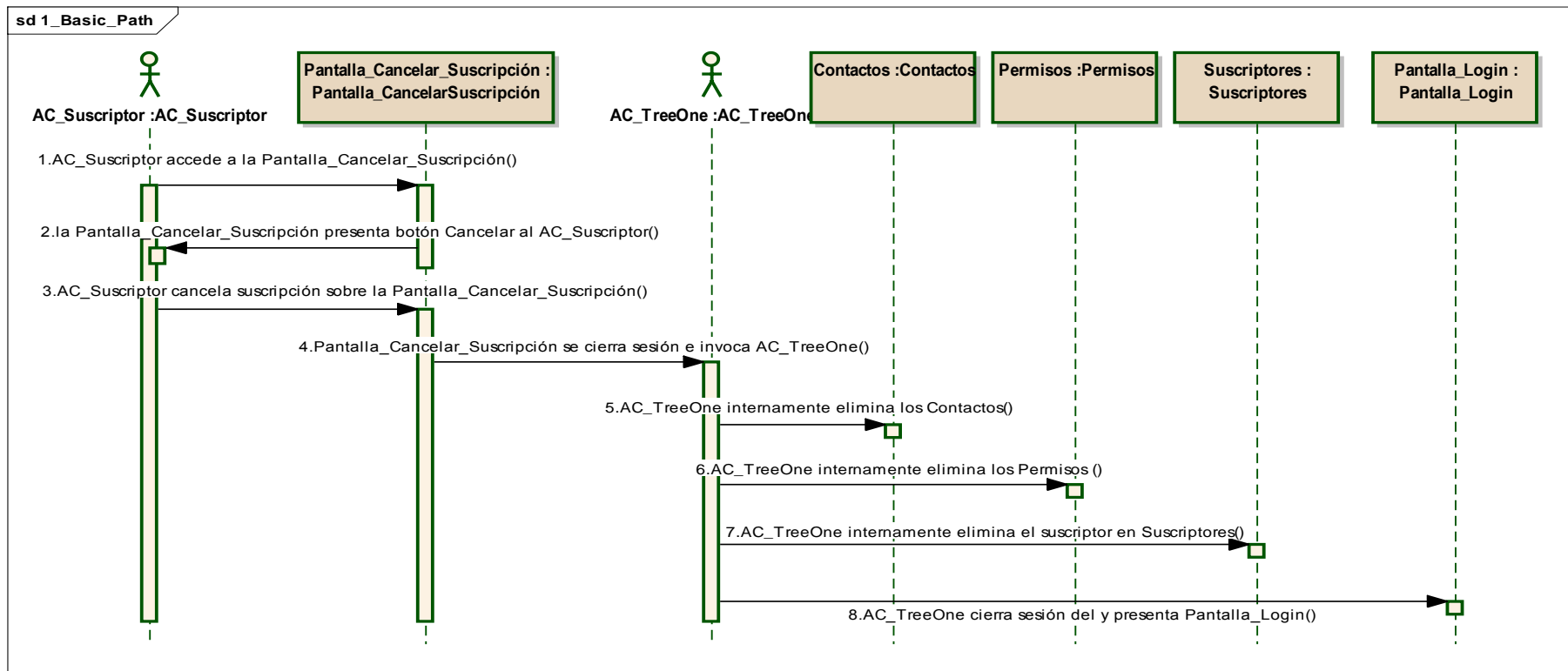


Figura No. 3.36: Diagrama de Secuencia - Cancelación Suscripción Académica.

3.5.3.4 Diagrama de Secuencia - Obtener Petición

Diagrama que permite definir la secuencia para el envío de las consultas académicas. Ver *figura No. 3.37*.

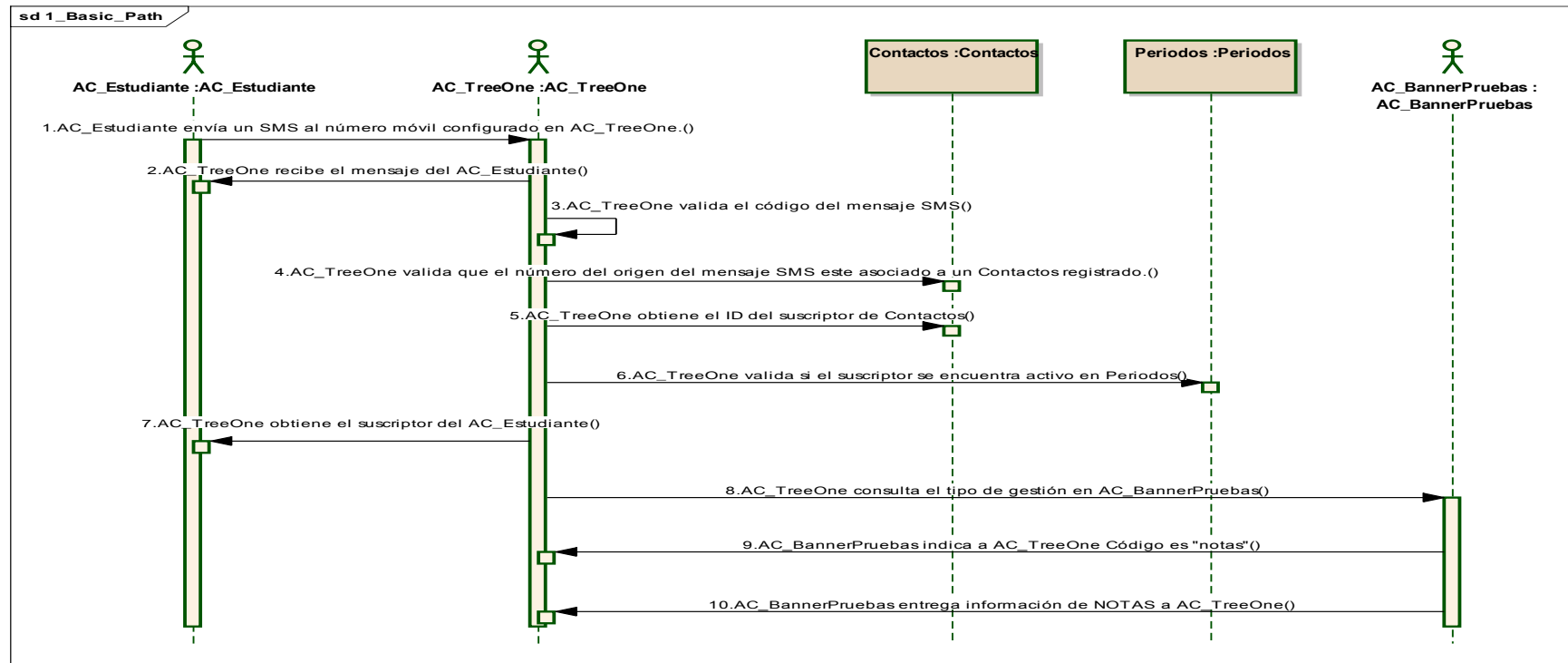


Figura No. 3.37: Diagrama de Secuencia - Obtener Petición.

3.5.3.5 Diagrama de Secuencia - Registrar Gestiones Académicas

Diagrama que permite definir la secuencia para el registro de las gestiones académicas en Pruebas. Ver *figura No. 3.38*.

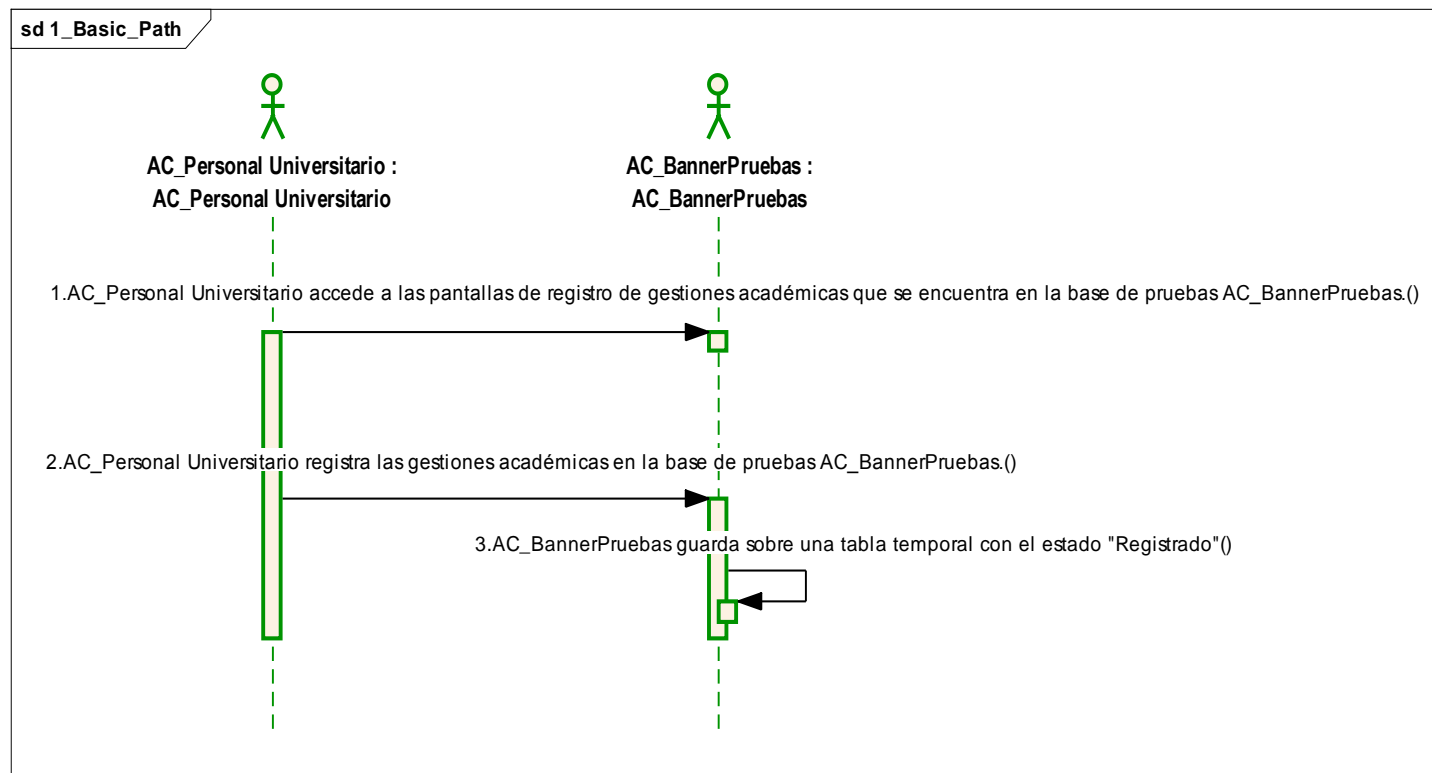


Figura No. 3.38: Diagrama de Secuencia - Registrar Gestiones Académica.

3.5.3.6 Diagrama de Secuencia - Registrar Anuncios Docentes

Permite definir la secuencia para el Registro de Anuncios. Ver *figura No. 3.39*.

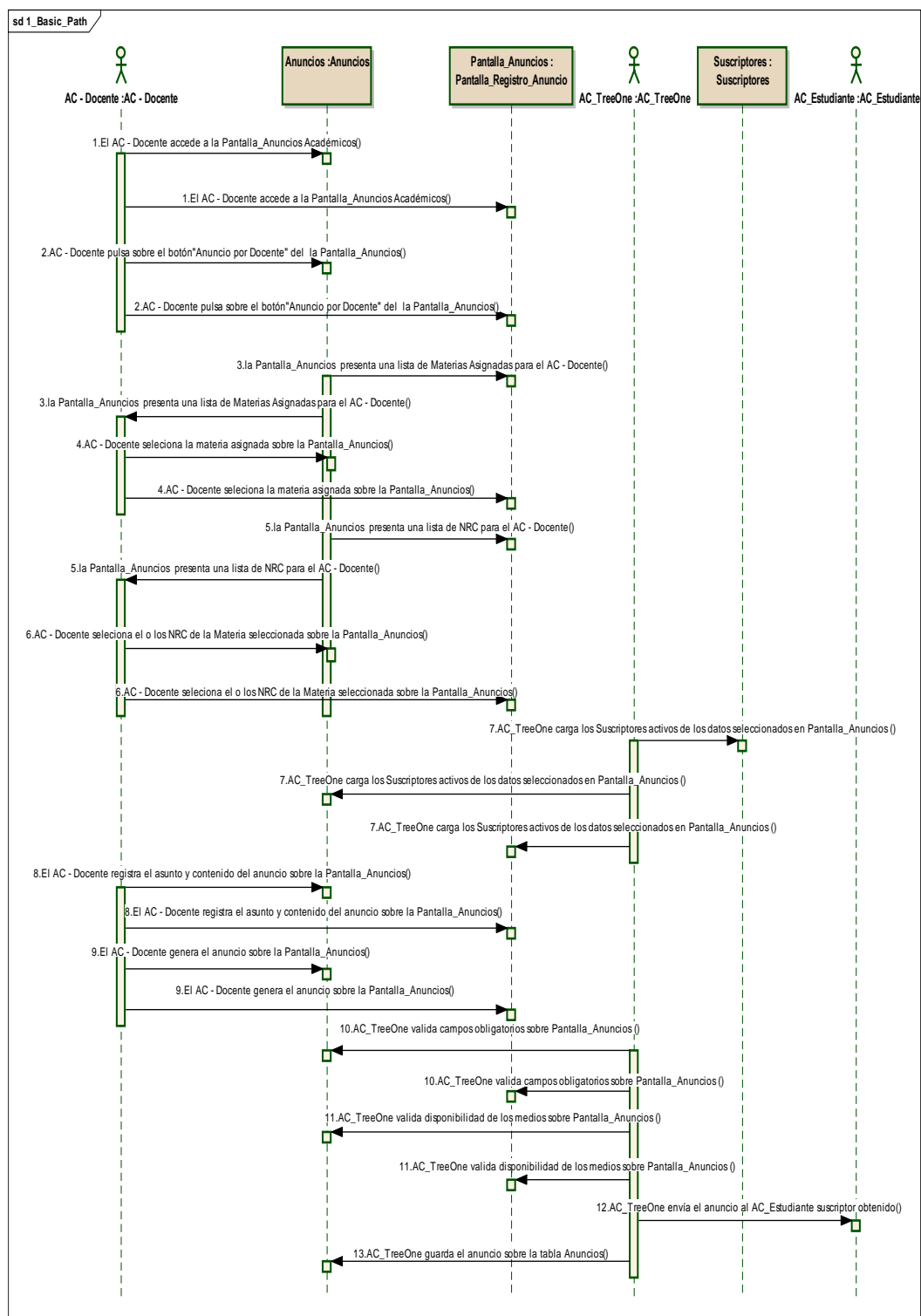


Figura No. 3.39: Diagrama de Secuencia - Registrar Anuncios Docentes.

3.5.3.7 Diagrama de Secuencia - Registrar Anuncios Personal

Diagrama que permite definir la secuencia para el registro de anuncios académicos por parte del Personal. Ver figura No. 3.40.

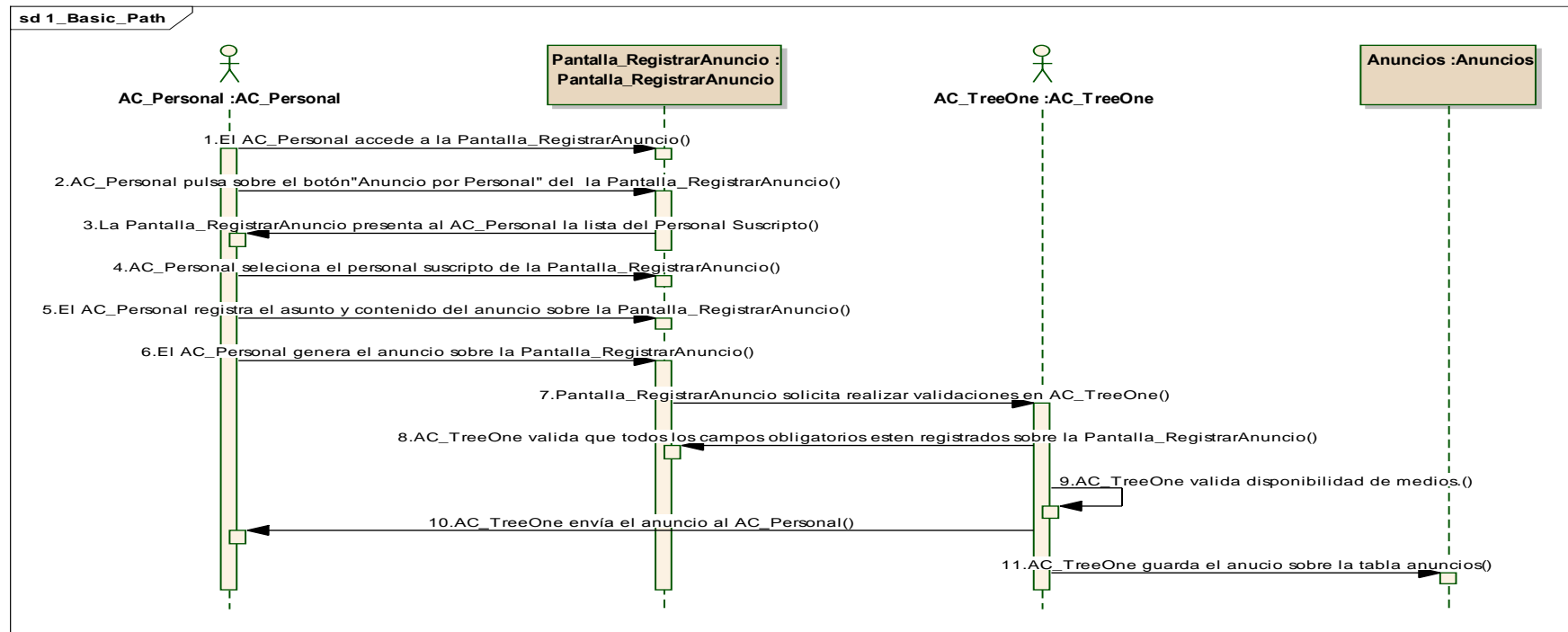


Figura No. 3.40: Diagrama de Secuencia - Registrar Anuncio Personal.

3.5.3.8 Diagrama de Secuencia - Enviar Mensaje SMS

Diagrama que permite definir la secuencia para el envío de los mensajes de texto o SMS. Ver *figura No. 3.41*.

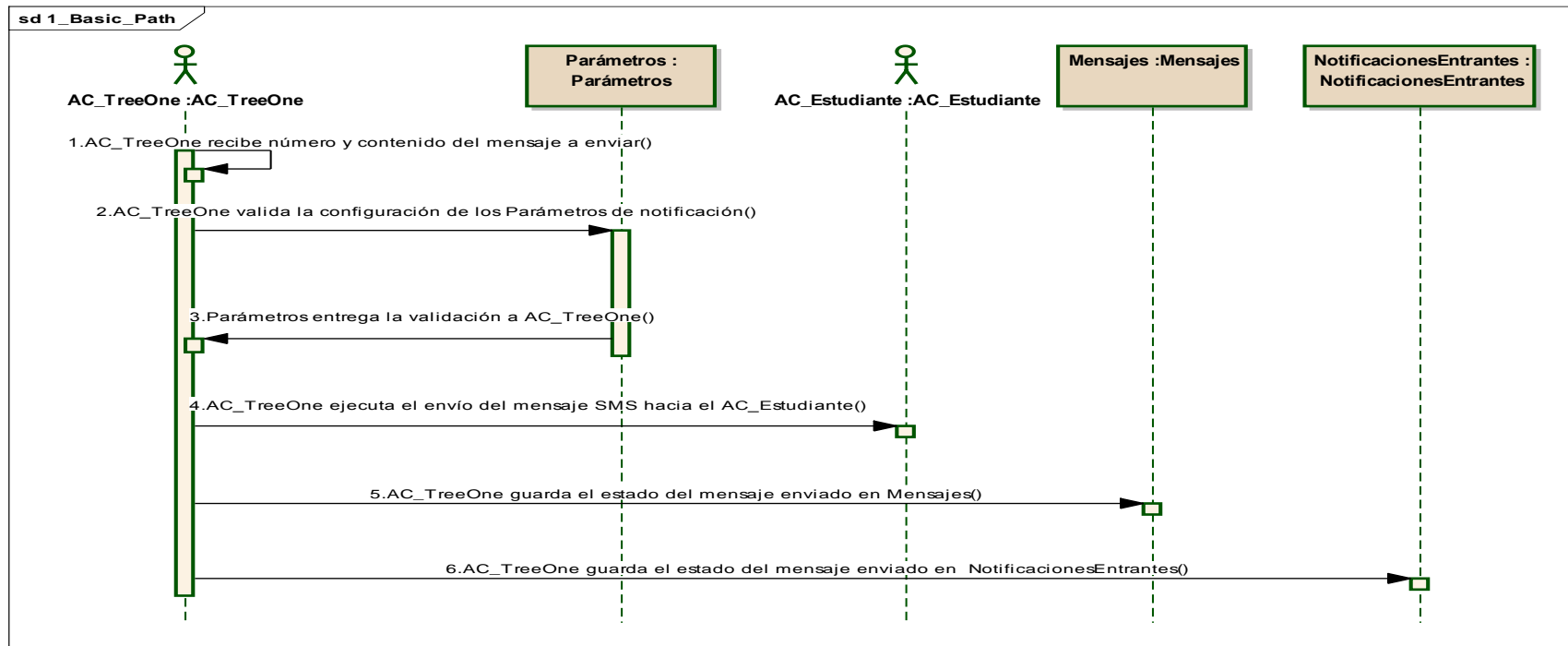


Figura No. 3.41: Diagrama de Secuencia - Enviar mensajes SMS.

3.5.3.9 Diagrama Secuencial - Parametrización

Diagrama que permite definir la secuencia para la configuración de los parámetros del servicio de comunicación. Ver *figura No. 3.42*.

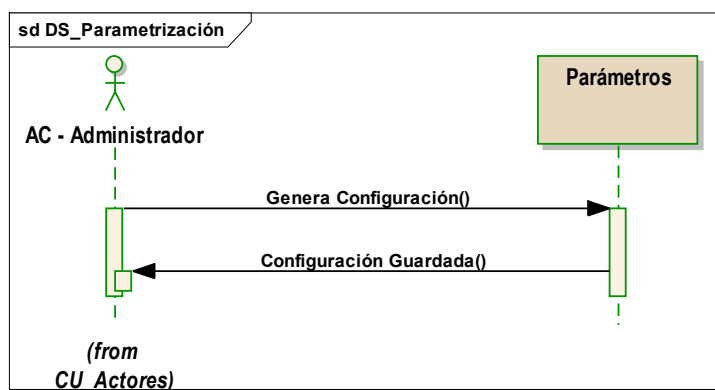


Figura No. 3.42: Diagrama de Secuencia - Parametrización.

3.5.3.10 Diagrama de Secuencia - Administración Usuarios

Diagrama que permite definir la secuencia para la creación de nuevos suscriptores administradores del servicio de comunicación. Ver *figura No. 3.43*.

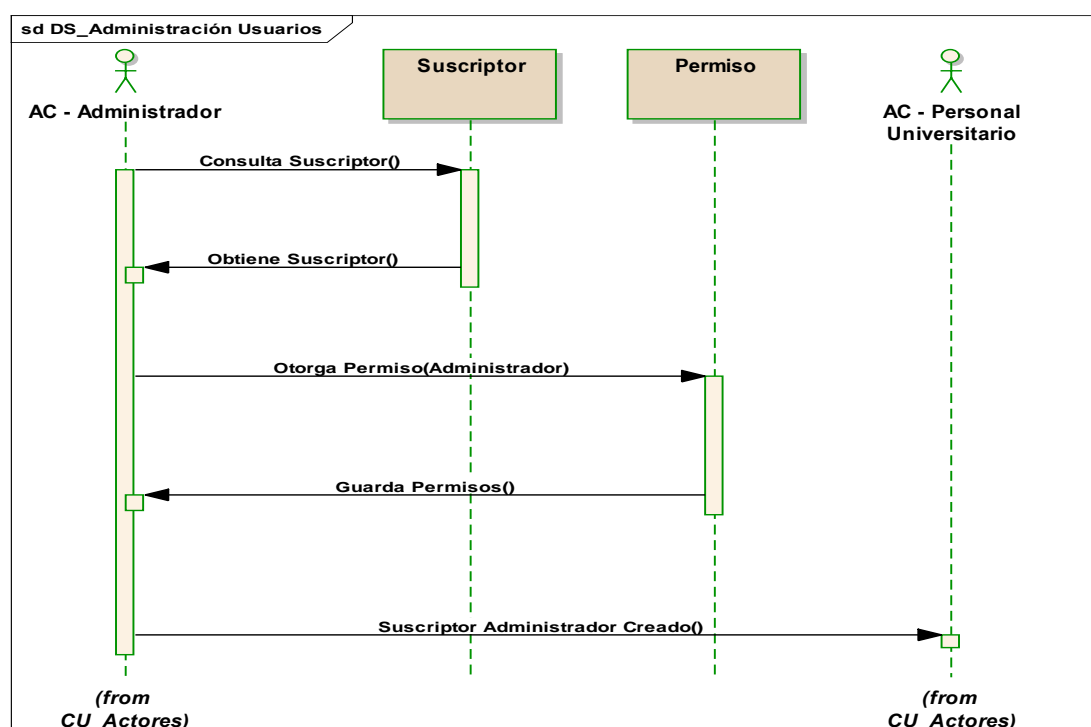


Figura No. 3.43: Diagrama de Secuencia - Administración Usuarios.

3.5.3.11 Diagrama de Secuencia - Administración Periodos

Diagrama que permite definir la secuencia para la creación de los periodos académicos del servicio de comunicación. Ver *figura No. 3.44*.

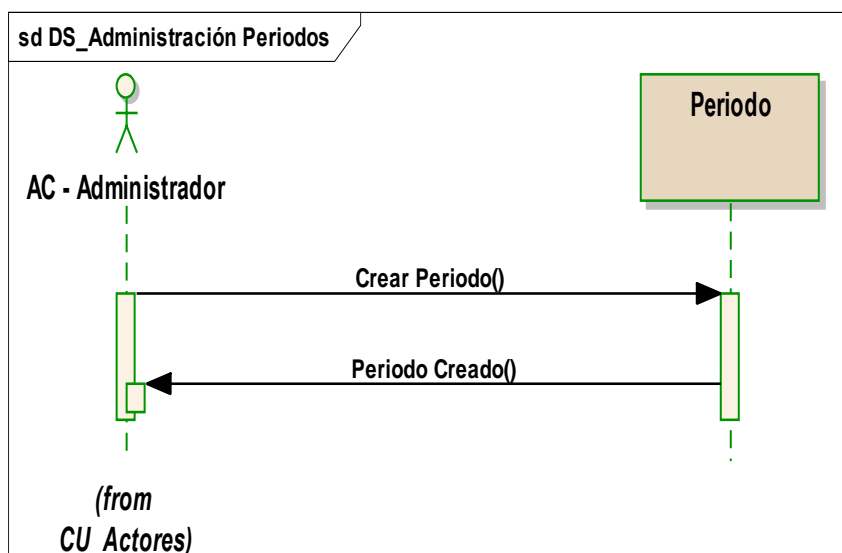


Figura No. 3.44: Diagrama de Secuencia - Administración Periodos.

3.5.3.12 Diagrama de Secuencia - Administración Contactos

Diagrama que permite definir la secuencia para la creación de los contactos del suscriptor. Ver *figura No. 3.45*.

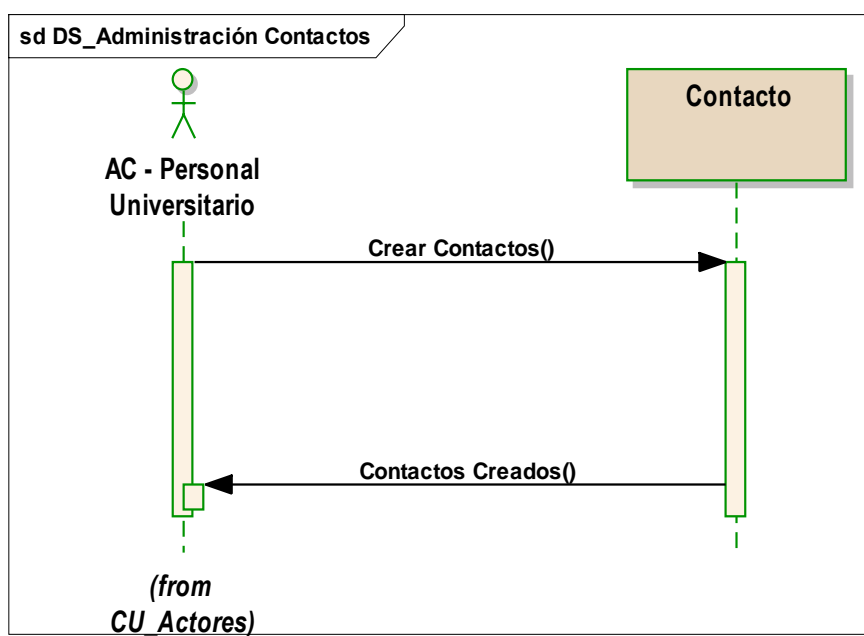


Figura No. 3.45: Diagrama de Secuencia - Administración Contactos.

3.5.4 ESTRUCTURA Y DESARROLLO DEL SISTEMA.

El estándar de programación utilizado en el proyecto es básicamente **JAVA**, para la presentación se utilizó los frameworks **JSF y PRIMEFACE**. A continuación se detallan las estructuras en la *figura No. 3.46*.

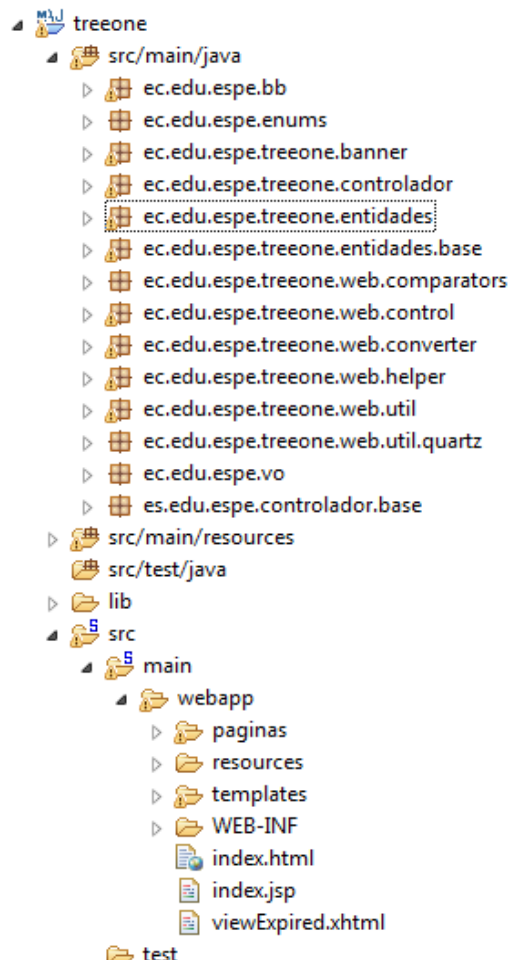


Figura No. 3.46: Estructura de directorios del proyecto.

- Los **Pojos**, se encuentra en la estructura **ec.edu.espe.entidad**.
- Los **Baking Beans**, se encuentra en la estructura **ec.edu.espe.bb**.
- Los **Enumeradores** se encuentran en la estructura **ec.edu.espe.enums**.
- Los Controladores **Managed Bean** se encuentran en la estructura **ec.edu.espe.controlador**.
- Los Utilitarios se encuentran en la estructura **ec.edu.espe.util**.
- El Vo se encuentran en la estructura **ec.edu.espe.vo**.

Además se menciona la interfaz **AdministradorRegistros** y su respectiva implementación en los controladores, se manejan las extensiones **EntidadBasica** y **BasicBean**.

Cada directorio se describe a continuación:

- **ec.edu.espe.treeone.entidades**

Aquí se encuentran todos los POJOS del Servicio de Comunicación, en donde se detalla lo siguiente:

- Se mapean los atributos a objetos (JPA).
- Se generan validaciones de cada atributo.
- Se generan las relaciones.
- Se genera el Entity Manager para la persistencia.
- Se crean las transacciones de persistencia CRUD.
 - ✓ Persist equivale al Create.
 - ✓ Find equivale al Read.
 - ✓ Merge equivale al Update.
 - ✓ Remove equivale al Delete.
- Se generan los tipos de consultas con el **createQuery**.

En la *figura No. 3.47* se presenta el pojo del Objeto Contactos, indicando sus atributos, sus relaciones o mapeos y sus transacciones.

Contactos.java

```
// Pojo para la creación de objeto Contactos, sus atributos y sus metodos CRUD (active records)
@Configurable
@Entity
@Table(schema = "public", name = "contactos")
public class Contactos extends EntidadBasica {
```

```
    private static final long serialVersionUID = 1L;
```

```
    // declaración de atributo
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    @Column(name = "id")
```

```
    private Integer id;
```



```

// relación muchos a uno con la tabla Suscriptores
@ManyToOne
@JoinColumn(name = "id_suscriptor", referencedColumnName = "id", nullable = false,
insertable = false, updatable = false)
private Suscriptores suscriptor;

// declaración de atributo
@Column(name = "numero_movil", length = 20)
@Size (min = 10, max = 10, message = "El número móvil debe tener 10 dígitos")
private String numeroMovil;
// inicialización del entity Manager, persistencia para el CRUD
public static final EntityManager entityManager() {
    EntityManager em = new Contactos().entityManager();
    if (em == null)
        throw new IllegalStateException(
            "Entity manager has not been injected (is the Spring
            Aspects JAR configured as an AJC/AJDT aspects library?");
    return em;
}

// Método que permite persistir o guardar el registro sobre la base de datos
@Transactional
public void persist() {
    if (this.entityManager == null)
        this.entityManager = entityManager();
    this.entityManager.persist(this);
}

// Método que permite eliminar el registro sobre la base de datos
@Transactional
public void remove() {
    if (this.entityManager == null)
        this.entityManager = entityManager();
    if (this.entityManager.contains(this)) {
        this.entityManager.remove(this);
    } else {
        Contactos attached = Contactos.findContactos(this.id);
        this.entityManager.remove(attached);
    }
}

// Método que permite modificar el registro sobre la base de datos
@Transactional
public Contactos merge() {
    if (this.entityManager == null)
        this.entityManager = entityManager();
    Contactos merged = this.entityManager.merge(this);
    this.entityManager.flush();
    return merged;
}

// Método que permite Consultar los registros con parámetros.
public static TypedQuery<Contactos> findContactosesByIdSuscriptor(Suscriptores
                                                                    idSuscriptor) {
    if (idSuscriptor == null)
        throw new IllegalArgumentException("Es requerido");
}

```

```

    EntityManager em = Contactos.entityManager();
    TypedQuery<Contactos> q = em.createQuery("SELECT o FROM Contactos AS o
WHERE          o.vigente is true and o.idSuscriptor = :idSuscriptor",
Contactos.class);
    q.setParameter("idSuscriptor", idSuscriptor);
    return q;
}
}

```

Figura No. 3.47: Ejemplo Entidad Pojo.

➤ **ec.edu.espe.bb**

Aquí se encuentran todos los Backing Beans o variables de sesión, en donde se detalla las siguientes propiedades:

- Las variables almacenarán los datos.
- Se inicializan las variables.
- Se crean el setter's y el getter's.

En la *figura No. 3.48* se presenta el Bean ContactosBB.

ContactosBB.java

```

// Permite declarar las variable a inyectarse con el controlador
// Aquí se detallan todas las variables con sus respectivos getters y setters.
// Es de tipo SessionScope para mantener los valores en sesión

@ManagedBean(name = "contactosBB")
@SessionScoped

public class ContactosBB implements Serializable {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private Contactos;
    private List<Contactos> listaContactos;

    // Constructor
    public ContactosBB() {
        super();
        iniciador(); }

    // Inicializador de las variables
    public void iniciador() {
        contactos = new Contactos();
        listaContactos = new ArrayList<Contactos>(); }

    // Getters y Setters
    public Contactos getContactos() {
        return contactos; }
}

```

```

public void setContactos(Contactos contactos) {
    this.contactos = contactos; }

public List<Contactos> getListaContactos() {
    return listaContactos;    }

public void setListaContactos(List<Contactos> listaContactos) {
    this.listaContactos = listaContactos; }
}

```

Figura No. 3.48: Ejemplo Backing Bean.

➤ **ec.edu.espe.treeone.controlador**

Aquí se encuentran todos los Controladores Beans, en donde se detalla lo siguiente:

- Se genera las llamadas al modelo.
- Se ejecuta la inyección de dependencias.
- Permite extenderse de otras clases.
- Permite implementar una interfaz.

En la *figura No. 3.49* se presenta el Bean ContactosBean:

```

ContactosBean.java
// Controlador encargado de la ejecución de las acciones generadas por la interfaz de usuario
// Es de tipo ViewScope, para que los valores duren y se muestren en la pantalla
// Se genera una inyección de dependencia con el Baking Bean

@ManagedBean(name = "contactosBean")
@SessionScoped
@RoosjsfManagedBean(entity = Contactos.class, beanName = "contactosBean")

// Se extiende del método BasicBean (Herencia)
// Se implementa de una interfaz AdministradorRegistro
public class ContactosBean extends BasicBean implements AdministradorRegistros{
public static final Logger LOG = Logger.getLogger(ContactosBean.class.getCanonicalName());

// Se inyecta con el Baking Bean
@ManagedProperty("#{contactosBB}")
private ContactosBB;

    public ContactosBean() {
        super();
    }

    @PostConstruct
    public void init() {

```

```

        iniciarDatos();
        verListaRegistros();
    }

    public void iniciarDatos() {
        // TODO Se añade la lógica de inicialización
    }

    @Override
    public String guardar() {
        // TODO Se añade la implementación de la interface
        contactosBB.getContactos().persist();
        return null;
    }

    @Override
    public String iniciarVariablesNuevoRegistro() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public String iniciarVariablesEditarRegistro() {
        // TODO Se añade la implementación de la interface
        return null;
    }

    @Override
    public String verListaRegistros() {
        // TODO Se añade la implementación de la interface
        contactosBB.getContactos().findContactosesByldSuscriptor;
        return null;
    }

    @Override
    public String editar() {
        // TODO Se añade la implementación de la interface
        contactosBB.getContactos().merge();
        return null;
    }

    @Override
    public String borrar() {
        // TODO Se añade la implementación de la interface
        contactosBB.getContactos().remove();
        return null;
    }

    @Override
    public String obtenerMensajeConfirmacionEliminacion(Object entidadEliminar) {
        // TODO Se añade la implementación de la interface
        return null;
    }
}

```

Figura No. 3.49: Ejemplo Controlador.

➤ **c.edu.espe.enums**

Aquí se encuentran todas las clases enumeradores del Servicio de Comunicación.

Un ENUM, también llamado enumeración o tipo enumerado es un tipo de dato definido por el usuario que solo puede tomar como valores los definidos en una lista. En la *figura No. 3.50* se presenta el Enumerador MedioEnum.java:

```
MedioEnum.java
// Declaración de variables valores asociados a variables.

public enum MedioEnum {

    // declaración para los diferentes medios
    SMS("S", "Notificación SMS", "Número Celular"),
    FACEBOOK("F", "Notificación FACEBOOK", "Cuenta Facebook"),
    TWITTER("T", "Notificación TWITTER", "Cuenta Twitter"),
    CORREO("C", "Notificación CORREO", "Correo Electrónico");

    // declaración de variables.
    private String codigo;
    private String nombre;
    private String tipo;

    // declaración de los getters y setters.
    public String getCodigo() {
        return codigo;
    }

    public String getNombre() {
        return nombre;
    }

    public String getTipo() {
        return tipo;
    }

    // constructor.
    private MedioEnum(final String codigo, final String nombre, final String tipo) {
        this.codigo = codigo;
        this.nombre = nombre;
        this.tipo = tipo;
    }
}
```

```

// método para obtener el valor o código del enumerador.
public static MedioEnum buscarPorCodigo(final String codigo) {
    MedioEnum medE =null;
    for(MedioEnum me: values()){
        if(codigo.equals(me.getCodigo())){
            medE = me;
            break;
        }
    }
    return medE;
}
}

```

Figura No. 3.50: Ejemplo Enumerador.

➤ **ec.edu.espe.treeone.web.util**

Aquí se encuentran detalladas todas las diferentes utilidades del sistema como las siguientes:

- Utilidad para envíos y recepción de SMS.
- Utilidad para envíos de CORREOS.
- Utilidad para envíos de mensajes TWITTER.

En la *figura No. 3.51* se presenta la utilidad UtilMail.java:

```

UtilMail.java
// Utilidad para enviar Mails

public class UtilMail {
    @Autowired
    private transient MailSender;

    @Autowired
    private transient SimpleMailMessage templateMessage;

    // Método que permite enviar el correo electrónico.
    public String sendMail(String to, String subject, String msg) {
        SimpleMailMessage message = new SimpleMailMessage(this.templateMessage);
        try {

            message.setTo(to);
            message.setSubject(subject);
            message.setText(msg);
            mailSender.send(message);
            return ("succes");
        }catch(MailException e){
            return ("error");
        }
    }
}

```

```

public void setTemplateMessage(SimpleMailMessage templateMessage) {
    this.templateMessage = templateMessage;
}

public MailSender getMailSender() {
    return mailSender;
}

public void setMailSender(MailSender mailSender) {
    this.mailSender = mailSender;
}

public SimpleMailMessage getTemplateMessage() {
    return templateMessage;
}
}

```

Figura No. 3.51: Ejemplo Utilidad Mail.

También se presenta la utilidad UtilSMS.java, se puede ver en la *figura No. 3.52:*

UtilSMS.java

```

// Utilidad para enviar mensajes de texto o SMS's

public class UtilSms {

    private NotificacionesEntrantes notiEnt;
    public static final Logger LOG = Logger.getLogger(UtilSms.class.getCanonicalName());

    public UtilSms() {
        super();
    }

    // Declaración para la declaración del módem
    public String cargarConfiguracion(String modemName, String port, int baudRate, String
manufacturer, String model){
        SerialModemGateway gateway = new SerialModemGateway(modemName, port,
baudRat, manufacturer, model);

        gateway.setProtocol(Protocols.PDU);
        gateway.setInbound(true);
        gateway.setOutbound(true);
        gateway.setSimPin("0000");
        try {
            Service.getInstance().addGateway(gateway);
            Service.getInstance().setOutboundMessageNotification(new
OutboundNotification());
            Service.getInstance().setInboundMessageNotification(new
InboundNotification());
            Service.getInstance().setGatewayStatusNotification(new
GatewayStatusNotification());
        } catch (GatewayException e) {
            e.printStackTrace();
        }
    }
}

```

```

        try {
            comenzarServicio();
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        return Service.getInstance().getServiceStatus().toString();
    }

// Declaración para activar el módem
public void comenzarServicio() throws Exception{
    Service.getInstance().getServiceStatus().toString();
    if (Service.getInstance().getServiceStatus().toString().equals("STOPPED")){
        try{
            Service.getInstance().startService();
        }catch(Exception e){
            LOG.info("Error al iniciar servicio: " + e);
        }
    }
    else{
        Service.getInstance().stopService();
        while (!Service.getInstance().getServiceStatus().toString().equals("STOPPED"))
        {
            System.out.println("Esperando Para Modem-->" +
Service.getInstance().getServiceStatus().toString());
        }
        Service.getInstance().startService();
    }

    while (!Service.getInstance().getServiceStatus().toString().equals("STARTED")) {
        Thread.sleep(3 * 1000);
    }
}

// Método para detener el servicio del módem.
public void pararServicio() throws SMSLibException, TimeoutException, GatewayException,
IOException, InterruptedException {
    Service.getInstance().stopService();
}

// Método para enviar el mensaje.
public String SendMessage(final String celular, final String mensaje) throws
TimeoutException, SMSLibException,
IOException, InterruptedException {
    System.out.println(".....Enviando Mensaje.....");
    OutboundMessage msg = new OutboundMessage(celular, mensaje);
    Service.getInstance().sendMessage(msg);
    return null;
}

// Método automático para detectar la llegada de mensajes SMS's
public class InboundNotification implements InboundMessageNotification {
    public void process(AGateway gateway, MessageType messageType, InboundMessage
msg) {
        if (msgType == MessageType.INBOUND)
            System.out.println(">>> New Inbound message detected from

```



```

Gateway: " + gateway.getGatewayId());
        else if (msgType == MessageTypes.STATUSREPORT)
            System.out.println(">>> New Inbound Status Report message detected
from Gateway: "
                                + gateway.getGatewayId());
        notiEnt = new NotificacionesEntrantes();
        notiEnt.setContenido(msg.getText());
        notiEnt.setOriginador(msg.getSmscNumber());
        notiEnt.setContenido(msg.getText());
        notiEnt.setFechaRegistro(msg.getDate());
        notiEnt.setEstado(EstadoEnum.REG.getCodigo());
        notiEnt.setObservaciones("");
        notiEnt.persist();
        try {
            Service.getInstance().deleteMessage(msg);
        } catch (TimeoutException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (GatewayException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

// Método para leer el mensaje recibido
public void Read Message() throws Exception {
    List<InboundMessage> msgList;
    try {
        msgList = new ArrayList<InboundMessage>();
        for (InboundMessage msg : msgList) {
            notiEnt = new NotificacionesEntrantes();
            notiEnt.setContenido(msg.getText());
            notiEnt.setOriginador(msg.getSmscNumber());
            notiEnt.setContenido(msg.getText());
            notiEnt.setFechaRegistro(msg.getDate());
            notiEnt.setEstado(EstadoEnum.REG.getCodigo());
            notiEnt.setObservaciones("");
            notiEnt.persist();
        }
        msgList.clear();
    } catch (Exception e) {
        pararServicio();
        e.printStackTrace();
    }
}
}

```

Figura No. 3.52: Ejemplo Utilidad SMS.

También se presenta la utilidad UtilTwitter.java. Se puede observar en la *figura No. 3.53*.

```

UtilTwitter.java

// Método para el envío de mensajes Twitter
public class UtilTwitter {

    /**
     * Envía mensaje vía Twitter.
     *
     * @param cuenta
     * @param mensaje
     * @throws TwitterException
     */
    public void send(String cuenta, final String mensaje) throws TwitterException {
        ConfigurationBuilder cb = new ConfigurationBuilder();
        cb.setDebugEnabled(true);
        cb.setOAuthConsumerKey("KZLXdda26ybb1AeasDG0Fg");
        cb.setOAuthConsumerSecret("a5sRg79zYP3U0Uua92PncPeu595DV03ssA");
        cb.setOAuthAccessToken("11156ast544586-
txxJrOZsdsdXsKv6SHM48dtAJanZyYx8bUL0");

        cb.setOAuthAccessTokenSecret("ECwLUM5y5EbSXas49DrUwSJvTWPXasasXnOlxQxo55M
");
        TwitterFactory tf = new TwitterFactory(cb.build());
        Twitter = tf.getInstance();
        User = twitter.showUser(cuenta);
        twitter.sendDirectMessage(user.getId(), mensaje);
    }
}

```

Figura No. 3.53: Ejemplo Utilidad Twitter.

Archivo POM

POM responde a las siglas de Project Object Model, es un fichero XML, que es la “unidad” principal de un proyecto Maven.

Contiene información acerca del proyecto, fuentes, test, dependencias, plugins, versiones.

Aquí se tiene definidos los artefactos utilizados, como su versión y fabrica. Ver *figura No. 3.54*:

```

pom.xml

<dependency>
  <groupId>postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <version>9.1-901.jdbc3</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>3.6.9.Final</version>
</dependency>
<dependency>
  <groupId>javax.mail</groupId>
  <artifactId>mail</artifactId>
  <version>1.4.3</version>
</dependency>

```

Figura No. 3.54: Ejemplo POM.

Configuraciones Adicionales

Aquí se demuestra la estructura de directorios de las diferentes configuraciones realizadas en el desarrollo del Servicio de Comunicación. Ver *figura No. 3.55*.

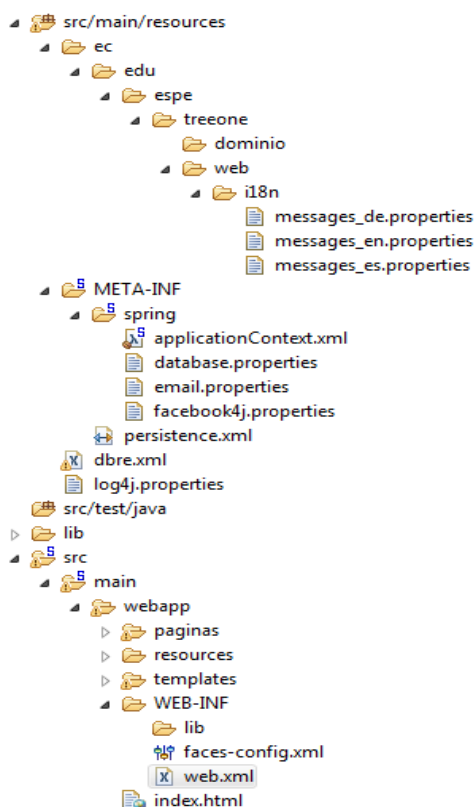


Figura No. 3.55: Estructura de directorios Configuraciones Adicionales.

➤ **applicationContext.xml**

El fichero de configuración básico de Spring es el contexto de aplicación (application context). Consiste en un fichero XML donde se añadirán todos los objetos que deberán existir en la aplicación al inicializarse la misma.

En donde se encuentra creado el JOB QUARTZ y la configuración del MAIL.

➤ **database.properties**

Aquí se encuentran los datos de la conexión a la base de datos.

➤ **messages_es.properties**

Aquí se encuentra detallada la internacionalidad de los mensajes y los label's que pueden ser utilizados para el Servicio de Comunicación, para el sistema solo se utilizó el lenguaje español.

➤ **web.xml**

Aquí se detalla todas las configuraciones referentes al aplicativo web, aquí se detalla cual es la página con la que arrancará el Servicio de Comunicación, también se detalla el tiempo de sesión, los temas del PRIMEFACE, se detallan los framework de desarrollo como JSF y PRIMEFACE, etc.

3.6. FASE DE TRANSICIÓN

Lo que se pretende lograr con esta fase es que el usuario final pruebe las funciones con las que cuenta el Servicio de Comunicación, estas se encuentran distribuidas por los perfiles creados y por las acciones o procesos automáticos del Servicio de Comunicación como:

- Perfil de Alumno o Estudiante
- Perfil de Docente o Profesor

- Perfil de Personal
- Perfil de Docente y Personal
- Perfil de Administrador del Servicio de Comunicación.
- Procesos y Validaciones del Servicio de Comunicación.

3.6.1 TESTS o PRUEBAS

Para la ejecución de las pruebas funcionales del Servicio de Comunicación, se tomó en base a la información de los usuarios entregados por la unidad TICS de la ESPE.

Con esta información de referencia se crea la base de datos pruebas Banner, con el fin de simular o recrear el proceso de gestión del Sistema Banner.

Los Test se harán en referencia a los escenarios de los casos de usos de cada funcionalidad.

3.6.1.1 Test de funcionalidad de Acceso y Suscripción al Servicio de Comunicación

✓ Accesos o Autenticación

Para las pruebas de Acceso o Autenticación, se realizan de las siguientes 2 maneras:

- Por el **personal universitario**: La ejecución de la aplicación se la realizará con el usuario del personal universitario creado en la base de datos de pruebas Banner, con la contraseña "1234", esta validación la realiza el servicio web.*

En la tabla No. 3.33 se presentan los escenarios y resultados:

Tabla No. 3.33: Escenarios - Acceso al Servicio de Comunicación

Escenarios	Resultados Esperados	Evaluación de la Prueba	Observaciones
<ol style="list-style-type: none"> 1. Se introduce el "id banner" en el campo usuario. 2. Se introduce "1234" en el campo contraseña. 3. Se pulsa el botón "aceptar" de la aplicación. 4. La interfaz que se aparecerá será la de bienvenida. 	<ol style="list-style-type: none"> 1. El servicio valida usuario y contraseña 2. El servicio acepta al usuario y permite el ingreso al servicio. 3. El servicio entrega mensaje de que no existe el usuario en la base de datos de pruebas. 4. El servicio presenta los accesos del usuario. 	Prueba exitosa.	- No se subió la base de datos de pruebas en Mysql, para probar la funcionalidad de conexión por medio de los servicios web.

b. Por el **Usuario Administrador del Servicio de Comunicación:** La ejecución de la aplicación se la realizará mediante el usuario administrador del Servicio de comunicación, este usuario no será validado con los usuarios de la base de datos de pruebas Banner.

En la tabla No. 3.34 se presentan los escenarios y resultados:

Tabla No. 3.34: Escenarios - Acceso al Servicio de Comunicación 2

Escenario	Resultados Esperados	Evaluación de la Prueba	Observaciones
<ol style="list-style-type: none"> 1. Se introduce el usuario administrador en el campo usuario. 2. Se introduce la contraseña. 3. Se pulsa en el botón "aceptar" de la aplicación. 4. La interfaz que se aparecerá será la de administración del servicio. 	<ol style="list-style-type: none"> 1. El servicio valida usuario y contraseña 2. El servicio acepta al usuario y permite el ingreso al servicio. 3. El servicio presenta los accesos del usuario. 	Prueba exitosa.	.

✓ Suscripción

Para las pruebas de Suscripción, se realizan de las siguientes 2 maneras:

a. **Usuario No Suscripto:** cuando el usuario necesita suscribirse al servicio de comunicación.

En la tabla No. 3.35 se presentan los escenarios:

Tabla No. 3.35: Escenarios Suscripción - Usuario No Suscripto.

Escenarios	Resultados Esperados	Evaluación de la Prueba	Observaciones
1. El Personal Universitario se autentifica en el Servicio.	<ul style="list-style-type: none"> El usuario registra los contactos y los guarda. 	Prueba exitosa.	- Que el contacto del correo ESPE venga nulo.
2. La interfaz que se presentará será de bienvenida y se exponen los diferentes servicios de comunicación	<ul style="list-style-type: none"> Se envía un correo de confirmación de la suscripción. 		
3. Se pulsa el botón "suscribirse" de la pantalla.	<ul style="list-style-type: none"> El personal universitario se crea y accede directamente al Servicio con el rol correspondiente. 		
4. Se despliega el dialogo de registro de contactos.			

b. **Usuario Suscripto:** cuando el usuario se encuentra suscripto en el servicio de comunicación.

En la tabla No. 3.36 se presentan los escenarios y resultados:

Tabla No. 3.36: Escenarios Suscripción - Usuario Suscripto.


Escenario	Resultados Esperados	Evaluación de la Prueba	Observaciones
1. El Personal Universitario se autentifica en el Servicio.	<ul style="list-style-type: none"> Se desplegará la pantalla principal del Sistema de Comunicación. SE presentaran los accesos asociados al rol del suscriptor. 	Prueba exitosa.	N/A

✓ Cancelar Suscripción

Se presentan los escenarios para el proceso de cancelación de la suscripción.

En la *tabla No. 3.37* se presentan los escenarios y resultados:

Tabla No. 3.37: Escenarios Cancelar Suscripción.

Entrada	Resultado Esperado	Evaluación de la Prueba	Observaciones
1. El Personal accede a la opción de  Cancelar Suscripción. 2. Aparecerá un diálogo de confirmación de cancelación de la suscripción.	<ul style="list-style-type: none"> Se confirmación la cancelación de la suscripción. Se cancela la sesión y se elimina el suscriptor. 	Prueba exitosa.	N/A

3.6.1.2 Test de Administración y configuración

✓ Gestión de Suscriptores Administradores

La ejecución de estos escenarios lo realiza el suscriptor administrador del Servicio de Comunicación.

En la *tabla No. 3.38* y *tabla No. 3.39* se presentan los escenarios y resultados:

Tabla No. 3.38: Escenarios Suscriptores Administradores - Crear.

Entrada	Resultado Esperado	Evaluación de la Prueba
1. Se accede a la opción. 2. Se selecciona el suscriptor. 3. Se selecciona la opción de agregar como administrador. 4. Se despliega la confirmación de la creación del nuevo administrador. 5. Se selecciona el botón aceptar. 6. Se presenta un mensaje dialogo de Suscriptor Administrador creado correctamente.	El servicio acepta crea el nuevo perfil del suscriptor.	Prueba exitosa.

Tabla No. 3.39: Escenarios Suscriptores Administradores - Eliminación.

<i>Entrada</i>	<i>Resultado Esperado</i>	<i>Evaluación de la Prueba</i>
1. Se accede a la opción. 2. Se selecciona el suscriptor. 3. Se selecciona la opción de eliminar como administrador. 4. Se despliega la confirmación de la eliminación del rol de administración	El servicio elimina lógicamente el nuevo perfil del suscriptor.	Prueba exitosa.

✓ Gestión de Parámetros

La ejecución de estos escenarios lo realiza el suscriptor administrador. En la *tabla No. 3.40* se presentan los escenarios y resultados:

Tabla No. 3.40: Escenarios Gestión Parametrización.

<i>Entrada</i>	<i>Resultado Esperado</i>	<i>Evaluación de la Prueba</i>
1. Se accede a la opción. 2. Se selecciona el parámetro a configurar. 3. Se configuran los valores del parámetro. 4. Se despliega la confirmación del parámetros configurado.	Nueva configuración del servicio.	Prueba exitosa.

✓ Gestión de Periodos

La ejecución de estos escenarios lo realiza el suscriptor administrador. En la *tabla No. 3.41* se presentan los escenarios y resultados:

Tabla No. 3.41: Escenarios Gestión de Periodos.

<i>Entrada</i>	<i>Resultado Esperado</i>	<i>Evaluación de la Prueba</i>	<i>Observaciones</i>
1. Se accede a la opción. 2. Se verifica si no existe periodo. 3. Se accede al botón "Nuevo" y se crea el nuevo periodo. 4. Se despliega la confirmación del periodo creado.	Nuevo periodo creado.	Prueba exitosa.	N/A

✓ **Gestión de Configuración Activación/Desactivación Módem**

La ejecución de estos escenarios lo realiza el suscriptor administrador. En la *tabla No. 3.42* se presentan los escenarios y resultados:

Tabla No. 3.42: Escenarios Gestión Módem.

<i>Entrada</i>	<i>Resultado Esperado</i>	<i>Evaluación de la Prueba</i>	<i>Observaciones</i>
<ol style="list-style-type: none"> 1. <i>Se accede a la opción.</i> 2. <i>Se puede configurar con los parámetros que se necesita para el uso del módem, como puerto, baudios, fabrica</i> 3. <i>Además se puede activar y desactivar el módem.</i> 4. <i>Se despliega la confirmación de la configuración del módem</i> 	<ol style="list-style-type: none"> 1. <i>Configuración es creada.</i> 2. <i>Activación/Desactivación Módem realizados.</i> 	<i>Prueba exitosa.</i>	<i>- el módem de estar correctamente configurado a nivel de los puertos COM</i>

✓ **Gestión de Cambio Contraseña**

La ejecución de estos escenarios lo realiza el suscriptor administrador. En la *tabla No. 3.43* se presentan los escenarios y resultados:

Tabla No. 3.43: Escenarios Cambio de Contraseñas.

<i>Entrada</i>	<i>Resultado Esperado</i>	<i>Evaluación de la Prueba</i>
<ol style="list-style-type: none"> 1. <i>Se accede a la opción.</i> 2. <i>Procede a registrar la contraseña antigua.</i> 3. <i>Procede a ingresar la nueva contraseña.</i> 4. <i>Se despliega la confirmación del cambio de contraseña.</i> 	<i>Cambio de contraseña exitoso.</i>	<i>Prueba exitosa</i>

3.6.1.3 Notificación Entrante y Saliente.

✓ **Gestión de Notificación Entrante de gestiones académicas.**

La ejecución de esta opción se realiza por medio del proceso interno, ya que obtiene el código de consulta del mensaje de texto-SMS enviado por el estudiante suscriptor.

En la *tabla No 3.44* se presentan los escenarios y resultados:

Tabla No. 3.44: Escenarios Notificación Entrante.

Entrada	Resultado Esperado	Evaluación de la Prueba	Observaciones
1. El estudiante envía un mensaje SMS al número configurado y con la palabra configurada	<ol style="list-style-type: none"> Se verifica el envío de la gestión académica consultada por medio de mensajes de texto SMS. Se verifica el registro de mensaje enviado correctamente a través de la opción Bandeja de Mensajes Los mensajes de los destinatarios no suscriptos en el servicio, no serán enviados. Los mensajes no enviados fueron registrados como fallidos. 	Prueba exitosa.	- Si el mensaje de texto no llegó a sus destinatarios, se debe revisar en las tablas de soporte de Notificaciones.

✓ **Gestión de Notificación Saliente de gestiones académicas.**

La ejecución de esta opción se realiza de manera automática ya que permite obtener las gestiones académicas registradas o actualizadas en la base de datos de pruebas Banner.

En la *tabla No. 3.45* se presentan los escenarios y resultados:

Tabla No. 3.45: Escenarios Notificación Saliente.

Entrada	Resultado Esperado	Evaluación de la Prueba	Observaciones
1. Se registran eventos de registro o actualización de las gestiones académicas sobre la base de datos de prueba Banner.	<ol style="list-style-type: none"> Verifica si el suscriptor está activo. Se entrega la gestión Académica por mensaje de texto SMS, correo y cuenta social. Se registra el evento en la bandeja de mensajes Si el número receptor no se encuentra registrado en el servicio, el mensaje no se responde. Se guarda en notificaciones entrantes como fallido. 	Prueba exitosa.	- Si el mensaje de texto no llegó a sus destinatarios, se debe revisar en las tablas de soporte de Notificaciones.

3.6.1.4 Anuncios Académicos

✓ Gestión de Anuncios Docente.

La ejecución de esta opción lo hace el suscriptor DOCENT. En la *tabla* No. 3.46 se presentan los escenarios y resultados:

Tabla No. 3.46: Escenarios Anuncios Docente.

Entrada	Resultado Esperado	Evaluación de la Prueba	Observaciones
1. <i>DOCENTE</i> accede a la opción. 2. Selecciona las Materias que da en el periodo configurado. 3. Ingresa la información del asunto y Detalle del anuncio. 4. Envía anuncio.	1. Anuncio llega a sus destinatarios ESTUDIANTES según la materia seleccionada por el docente. 2. El anuncio llega a las cuentas de la ESPE, cuenta personal, cuenta Facebook y cuenta Twitter	Prueba exitosa. Se envían todos los correos hacia sus destinatarios.	- verificar que los usuarios hayan registrado correctamente las cuentas. - verificar que el servidor de correos no encole los comunicados.

✓ Gestión de Anuncios Docente.

La ejecución de esta opción lo hace el suscriptor PERSONAL. En la *tabla* No. 3.47 se presentan los escenarios y resultados:

Tabla No. 3.47: Escenarios Anuncios Personal.

Entrada	Resultado Esperado	Evaluación de la Prueba	Observaciones
1. <i>PERSONAL</i> accede a la opción 2. Ingresa la información de Asunto y Detalle del anuncio 3. Envía anuncio	1. Anuncio llega a sus destinatarios PERSONAL según la materia seleccionada por el docente. 2. El anuncio llega a las cuentas de la ESPE, cuenta personal, cuenta Facebook y cuenta Twitter	Prueba exitosa. Se envían todos los correos hacia sus destinatarios.	- verificar que los usuarios hayan registrado correctamente las cuentas. - verificar que el servidor de correos no encole los comunicados.

3.6.2 Pruebas de Caja Blanca

En estas pruebas se validaron los logs y los valores de salida de cada ejecución, con este se pudo comprobar la lógica o el camino de cada iteración. En la *figura No. 3.56* se presenta el log de la activación del módem.

```

ACTIVA MÓDEM
This software is distributed under the terms of the Apache v2.0 License.
Web Site: http://smslib.org
15:03:55,037 INFO [smslib] (http--0.0.0.0-8080-1) Version: 3.5.2
15:03:55,037 INFO [smslib] (http--0.0.0.0-8080-1) JRE Version: 1.6.0_32
15:03:55,039 INFO [smslib] (http--0.0.0.0-8080-1) O/S: Windows 7 / amd64 / 6.1
15:03:55,051 INFO [stdout] (http--0.0.0.0-8080-1) 00.....Comenzar Servicio
15:03:55,052 INFO [stdout] (http--0.0.0.0-8080-1) 01.....Iniciando MODEM
15:03:55,053 INFO [stdout] (http--0.0.0.0-8080-1) 03.....Estado MODEM -->
STOPPED
15:03:55,055 INFO [stdout] (http--0.0.0.0-8080-1) 04.....IF (MODEM STOPPED)
15:03:55,056 INFO [stdout] (http--0.0.0.0-8080-1) 05.....MANDA A STARSERVICE
15:03:58,062 INFO [smslib] (Thread-70) GTW: modem.com1: Starting gateway, using Generic
AT Handler.

ENVIAR SMS
15:03:58,064 INFO [smslib] (Thread-70) GTW: modem.com1: Opening: COM13 @115200
15:06:25,307 INFO [stdout] (Thread-81) Inicio...
15:06:25,309 INFO [stdout] (Thread-81) Date: Tue Jul 01 15:06:21 COT 2014
15:06:25,317 INFO [stdout] (Thread-81) Failure Cause: NO_ERROR
15:06:25,322 INFO [stdout] (Thread-81) Message Status: SENT
15:06:25,323 INFO [stdout] (Thread-81) Recipient: 0996097310
15:06:25,326 INFO [stdout] (Thread-81) fin...

ENVIANDO SMS Y VERIFICANDO ESTADO MÓDEM
15:09:25,164 INFO [stdout] (http--0.0.0.0-8080-1) .....SEND
15:09:25,165 INFO [stdout] (http--0.0.0.0-8080-1) .....ENTRO HILO ISALIVE
15:09:25,166 INFO [stdout] (http--0.0.0.0-8080-1) .....HILO STOP Y COMEMZAR
SERVICIO
15:09:25,167 INFO [stdout] (http--0.0.0.0-8080-1) .....Comenzar Servicio
15:09:25,168 INFO [stdout] (http--0.0.0.0-8080-1) .....Iniciando MODEM
15:09:25,169 INFO [stdout] (http--0.0.0.0-8080-1) .....Estado MODEM --> STARTED
15:09:25,170 INFO [stdout] (http--0.0.0.0-8080-1) .....ELSE (MODEM NOT STOPPED)
15:09:28,371 INFO [smslib] (http--0.0.0.0-8080-1) GTW: modem.com1: Stopping
gateway...
15:09:28,419 INFO [smslib] (http--0.0.0.0-8080-1) GTW: modem.com1: Closing: COM13
@115200
15:09:28,420 INFO [smslib] (http--0.0.0.0-8080-1) GTW: modem.com1: Gateway stopped.
15:09:28,477 INFO [stdout] (NotifyQueueManager) >>> Gateway Status change for
modem.com1, OLD: STOPPING -> NEW: STOPPED
15:09:28,620 INFO [stdout] (http--0.0.0.0-8080-1) .....STOPSERVICE INICIADO
15:09:43,625 INFO [stdout] (http--0.0.0.0-8080-1) .....MANDA A STARSERVICE
15:09:46,630 INFO [smslib] (Thread-99) GTW: modem.com1: Starting gateway, using
Generic AT Handler.
15:09:46,631 INFO [smslib] (Thread-99) GTW: modem.com1: Opening: COM13 @115200

```

Figura No. 3.56: Conexión Módem.

3.6.2.1 Validación de mensajes salientes.

Se verifica el envío de mensajes automáticos por el registro de las notas académicas. Ver *figura No. 3.57*.

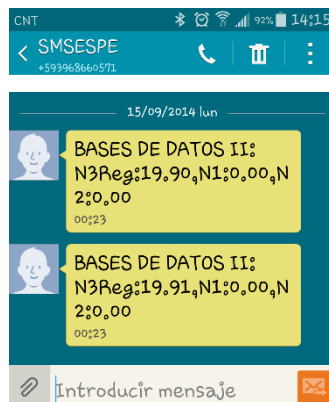


Figura No. 3.57: Pruebas - SMS Saliente.

3.6.2.2 Validación de mensajes entrantes.

Se verifica envío de mensajes automático por medio de la consulta realizada por el suscriptor estudiante.

En el siguiente ejemplo envío un mensaje de texto – SMS al número configurado por el administrador y con el código correspondiente, para el caso de consultar las notas académicas, el código es: notas. Ver *figura No. 3.58*.

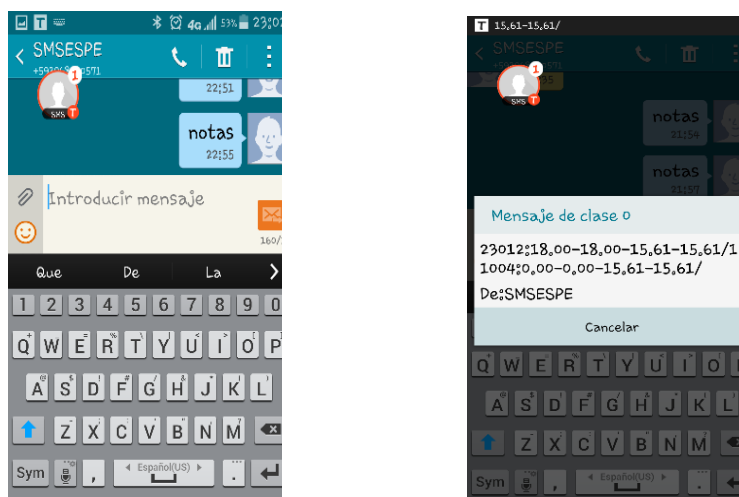


Figura No. 3.58: Pruebas - SMS Entrante y Saliente Notas.

En el siguiente ejemplo es para el caso de consultar los impedimentos académicos y saldos pendientes, los códigos son: impedimentos y cuentas. Ver los mensajes de respuesta en la *figura No. 3.59*.

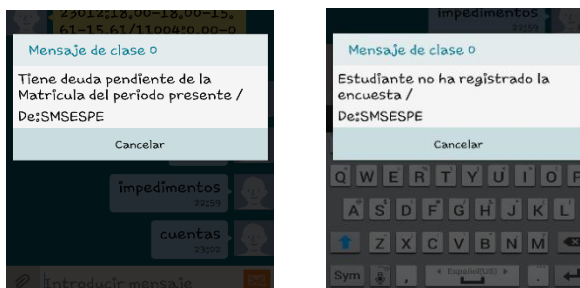


Figura No. 3.59: Pruebas - SMS Entrante y Saliente Impedimentos - Cuentas.

3.6.2.3 Validación de anuncios enviados a las cuentas sociales.

El proceso de generación de anuncios académicos esta para notificar tanto a los estudiantes como al personal administrativo ESPE sobre los diferentes informativo, sucesos o noticias académicas, para lo cual se utilizan los siguientes medios para la notificación.

Por Twitter: El usuario debe seguir a la cuenta Twitter de la ESPE **@serviciosmsespe**, como se indica en la *figura No. 3.60*.



Figura No. 3.60: Pruebas - Seguimiento Twitter.

En la *figura No. 3.61* se presentan los diferentes mensajes enviados por el anuncio registrado. La cuenta **@serviciosmsespe** es el encargado de notificar al usuario **@leomolinar**.

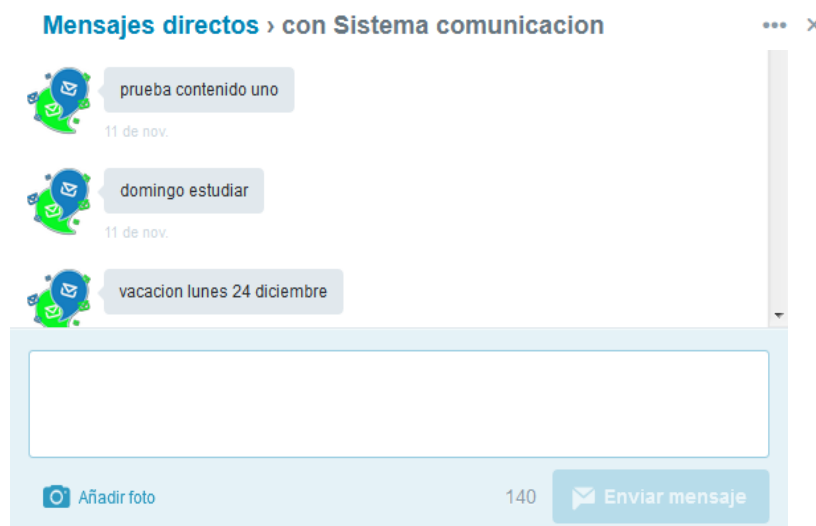


Figura No. 3.61: Pruebas - Anuncios Mensaje Twitter.

En las *figura No. 3.62* y *No. 3.63* se presentan los correos registrados en el buzón de la cuenta de correo ESPE **lemolina3@espe.edu.ec**. Estos correos fueron enviados por el anuncio registrado.

Para las pruebas se configuró la cuenta de correo origen **leo.molina.noboa@gmail**.

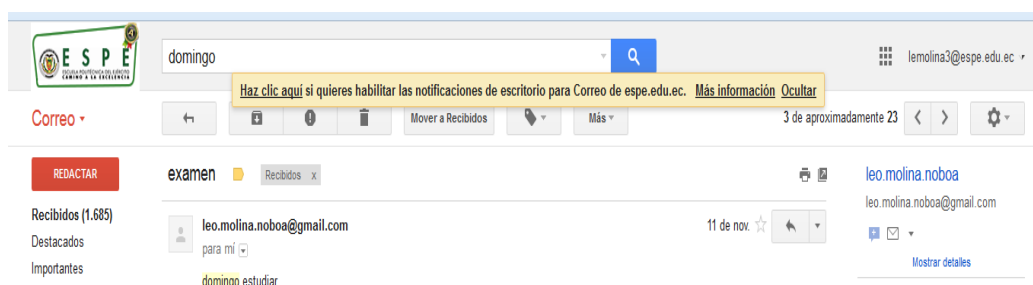


Figura No. 3.62: Pruebas - Anuncios Mensaje Correo ESPE No1.

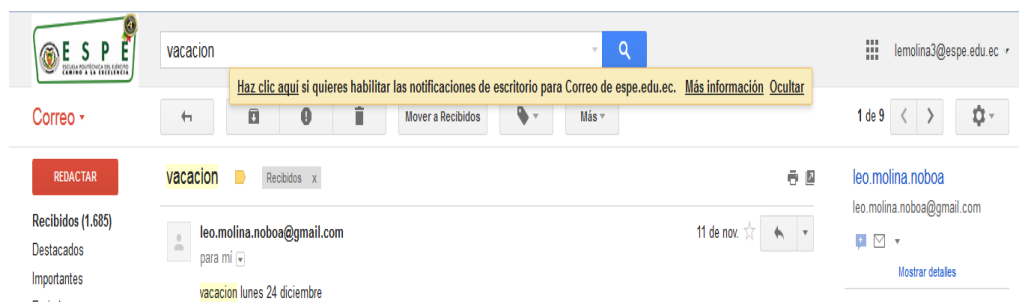


Figura No. 3.63: Pruebas - Anuncios Mensaje Correo ESPE No2.

En las *figuras No. 3.64 y No. 3.65* se presentan los correos registrados en el buzón de la cuenta personal ***k_leonardo_molina@hotmail.com***. Estos correos fueron enviados por el anuncio registrado.

Para las pruebas se configuró la cuenta de correo origen ***leo.molina.noboa@gmail.com***.

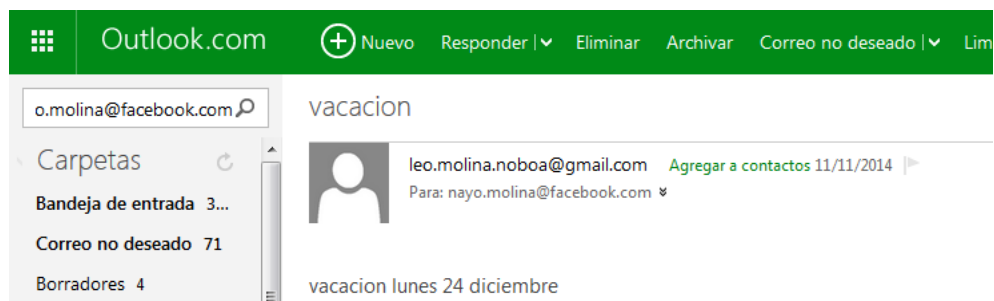


Figura No. 3.64: Pruebas - Anuncios Mensaje Correo Personal No1.

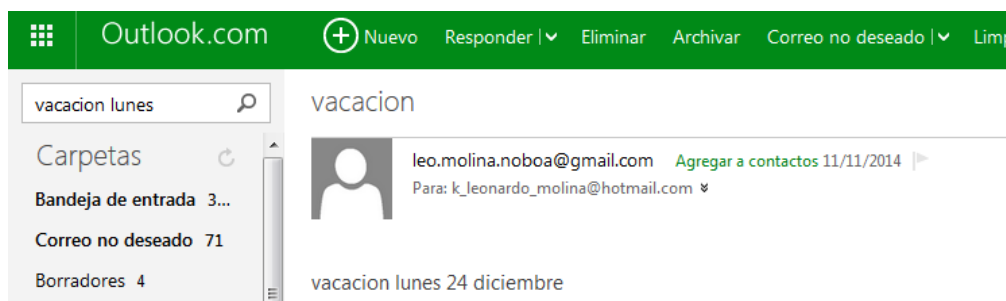


Figura No. 3.65: Pruebas - Anuncios Mensaje Correo Personal No2.

En las *figura No. 3.66 y No. 3.67* se presentan los correos registrados en el buzón de la cuenta personal ***nayo.molina@facebook.com***. Estos correos fueron enviados por el anuncio registrado.

Para las pruebas se configuró la cuenta de correo origen ***leo.molina.noboa@gmail.com***.

Para el caso de Facebook y por políticas propias de seguridad de la red social, se eliminaron todo envío de mensajes privados desde aplicativos web, por tal razón aplicaron la regla para que cualquier mensaje llegue desde aplicativos web se direccionen al correo configurado en la red social.

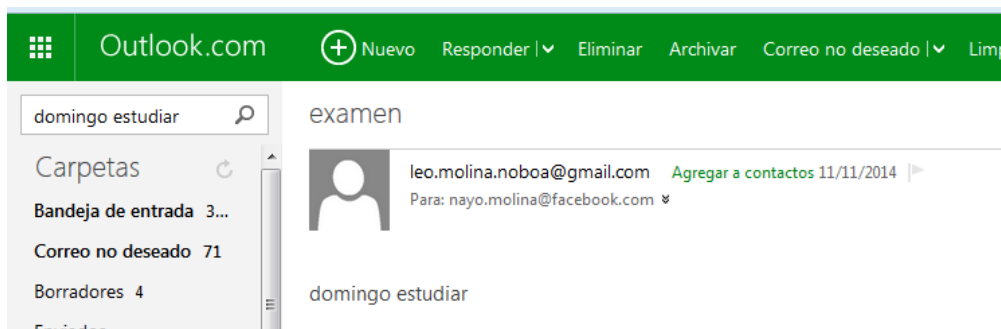


Figura No. 3.66: Pruebas - Anuncios Mensaje Correo Facebook No1.

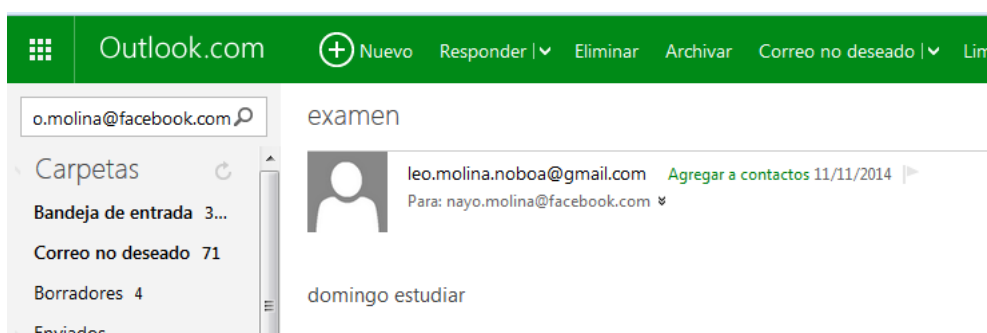


Figura No. 3.67: Pruebas - Anuncios Mensaje Correo Facebook No2.

3.6.2.4 Validación del correo informativo de suscripción.

En las figura No. 3.68 se presentan los correos registrados en el buzón de la cuenta de correo ESPE **lemolina3@espe.edu.ec**.

Estos correos fueron enviados en el momento de la suscripción para informar al usuario de la suscripción al servicio de comunicación.

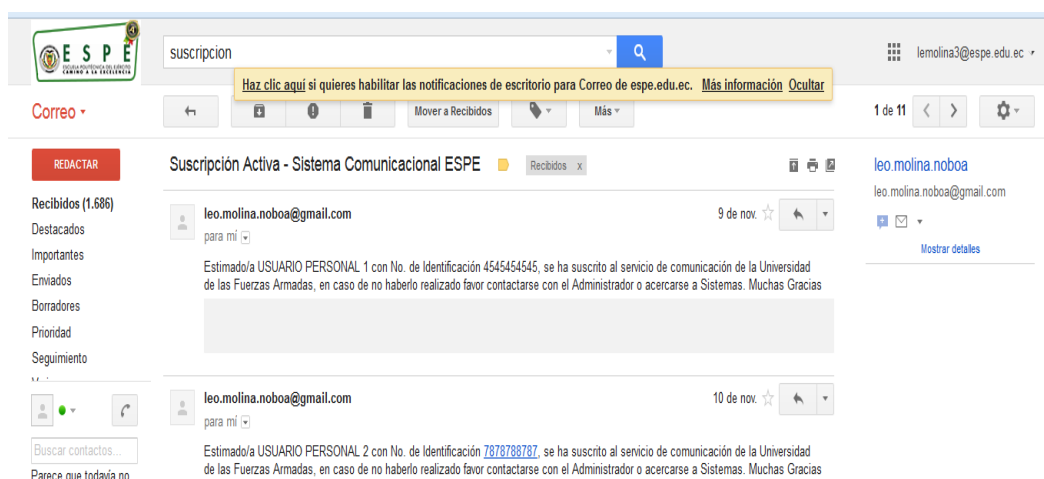


Figura No. 3.68: Pruebas - Anuncios Mensaje Suscripción.

3.6.3 Pruebas de Interfaces

También se realizó pruebas de interfaces, se probó en 3 navegadores diferentes, a continuación se presentan los resultados. Ver *tabla No. 3.48*.

Tabla No. 3.48: Pruebas de Interfaces.

Navegador	Prueba
Mozilla 31.0	Óptimo en el tiempo de respuesta y ningún tipo de problema
Explorer 11.0	Óptimo en el tiempo de respuesta y ningún tipo de problema
Chrome 39	Óptimo en el tiempo de respuesta y ningún tipo de problema

3.6.4 Pruebas de Validación

El objetivo de estas pruebas es certificar el envío de: correos electrónicos, mensajes por número celular, correos a cuentas Facebook y mensajes a la red social Twitter. Entre las diferentes pruebas se destacan las siguientes:

- Validación en el registro de los correos, que cumpla con el estándar de un correo.
- Validación en el registro del número celular, que permita el registro de un número móvil que contenga 9 dígitos y sin el cero "0" al inicio del número.
- Validación del registro de los campos requeridos.

3.6.4.1 Manuales

a) *Instalación del Aplicativo y configuración de Cuentas*

Se referencia al Anexo 1.

b) *Manual de Usuarios.*

Se referencia al Anexo 2.

c) *Manual Administrativo del Servicio de Comunicación.*

Se referencia al Anexo 3.

d) *Manual de Usuario Desarrollador Twitter.*

Se referencia al Anexo 4.

e) *Diccionario de Datos.*

Se referencia al Anexo 5.

CAPITULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- El manejo de la metodología RUP para el desarrollo y análisis del servicio de comunicación ha sido de gran ayuda, ya que permitió tener una idea clara del desarrollo del proyecto mediante el uso de las diferentes representaciones gráficas.
- Con las aplicaciones Enriquecidas RIA (Rich Internet Applications), mejoran las aplicaciones web convencionales, permitiendo ofrecer una interfaz de usuario enriquecido y de mejor rendimiento, conservando así las mismas características que las aplicaciones web tradicionales.
- Gracias a la herramienta AJAX, se mejora el desempeño de la aplicación minimizando la comunicación cliente-servidor mediante comunicaciones asíncronas.
- El patrón modelo vista controlador (MVC) se adapta muy fácilmente a la arquitectura Cliente-Servidor, facilitando la creación de aplicaciones Web escalables y flexibles.
- El Servicio de Comunicación brindará al personal universitario de una nueva alternativa de comunicación e información, facilitando las consultas y notificaciones de las diferentes gestiones académicas, aprovechando así las bondades tecnologías con las que actualmente se cuenta en beneficio de la comunidad universitaria.
- Los servicios de notificación inmediata a través de SMS's, serán de gran ayuda y apoyo para el estudiante, permitiendo la toma de decisiones inmediatas sobre sus nuevos registros notificados.

- Con el registro de los Anuncios Académicos, la comunidad universitaria se mantendrá informada o comunicada sobre la ejecución de los diferentes eventos académicos, mediante los medios de comunicación predominantes del momento, como los correos electrónicos, los mensajes Twitter y los correos Facebook.

4.2. RECOMENDACIONES

- Con el beneficio de los servicios de comunicación, la Universidad de las Fuerzas Armadas ESPE, puede brindar en un futuro medidas de seguridad sobre sus Sistemas, generando nuevos servicios de confirmación por el acceso o ingresos a los diferentes sistemas de información.
- El Servicio de Comunicación puede ser tomado y usado como base para el desarrollo de nuevos productos de comunicación en beneficio de la Universidad de las Fuerzas Armadas ESPE.
- Para la adquisición del módem, se sugiere que el dispositivo sea de marca HUAWEI bajo tecnología GSM o 3G, además que sea de uso exclusivo para el envío y recepción de mensajes escritos.
- Establecer convenios con las operadoras de telefonía móvil o celular, para la contratación de paquetes de mensajes o SMS.
- Para el correcto funcionamiento de los servicios de comunicación, los suscriptores deberán registrar de modo correcto la cuenta de correo ESPE, la cuenta de correo personal, las cuentas sociales y su número móvil, ya que de esto depende que la notificación sea entregada correctamente.
- Para el manejo de envíos de mensajes de texto o SMS, se recomienda utilizar la librería de código abierto SMSLib, escrita en código Java y que permite enviar y recibir SMS mediante un módem GSM/3G o teléfono GSM compatible.
- El servidor de correo SMTP de la Universidad de las Fuerzas Armadas ESPE, deberá controlar el número máximo de correos inválidos, para evitar posibles encolamientos de entrega de correos.

4.3. REFERENCIAS BIBLIOGRÁFICAS

Alvarez, C. (2012). *Arquitectura java solida*.

Brito, K. (2009). *SELECCIÓN DE METODOLOGÍAS DE DESARROLLO PARA APLICACIONES WEB*.
Obtenido de <http://www.eumed.net/libros-gratis/2009c/584/Por%20que%20utilizar%20RUP%20para%20desarrollar%20aplicaciones%20web.htm>

Calderón Villavicencio, J. L. (Agosto de 2012). *Servicio de mensajería instantánea para la comunidad del Instituto Técnico Salesiano*. Obtenido de <http://dspace.ups.edu.ec/handle/123456789/2958>

Campana, D. (2009). *Especificación de Interfaces y Patrones RIA*. La Plata: Universidad Nacional de La Plata.

Cataluña, G. d. (Octubre de 2013). *Guía de redes sociales*. Obtenido de http://www.gencat.cat/web/meugencat/documents/guia_usos_xarxa_es.pdf

Cédric Simon. (s.f.). *Curso de Java Server Faces 2 con Hibernate 3*. Obtenido de http://www.solucionjava.com/pdf/Curso_JS2_Hibernate3.pdf

Çivici, Ç. (s.f.). *Primeface USER'S GUIDE*. Obtenido de Primeface USER'S GUIDE: <http://www.primefaces.org/documentation>

Commons, C. (mayo de 2013). *Redes Sociales*. Obtenido de [www.gobcan.es: http://www.gobcan.es/identidadgrafica/manual_identidad_grafica/manual_desglosado/redes_sociales_guia_estilo.pdf](http://www.gobcan.es/identidadgrafica/manual_identidad_grafica/manual_desglosado/redes_sociales_guia_estilo.pdf)

CRUZ, J. A. (2008). Tesis DESARROLLO DE UN PORTAL WEB PARA LA FEDERACIÓN. el salvador. Obtenido de http://rd.udb.edu.sv:8080/jspui/bitstream/123456789/233/1/43061_tesis.pdf

Deitel, P. J. (2008). *JAVA como programar 7ma Edición*. Estado de México: Pearson Education.

DUHL, J. (Noviembre de 2003). Rich Internet Applications. *Rich Internet Applications*. Global Headquarters. Obtenido de http://www.adobe.com/platform/whitepapers/idc_impact_of_rias.pdf

Eguiluz, J. (Agosto de 2013). *Introducción a AJAX*. Obtenido de Creative Commons No comercial - Atribución - Compartir igual (CC BY-NC-SA) 3.0: <http://librosweb.es/ajax/>

EuropeMarketext. (s.f.). *www.1-2-Text.com*. Obtenido de [www.1-2-Text.com: http://www.1-2-text.com/v5/index_porque_sms.php](http://www.1-2-Text.com)

- Fundación Wikimedia, I. (22 de Septiembre de 2014). *Rich Internet application*. Obtenido de Rich Internet application: http://es.wikipedia.org/wiki/Rich_Internet_application
- Graells, D. P. (23 de 03 de 2008). *LAS TIC Y SUS APORTACIONES A LA SOCIEDAD*. Obtenido de <https://docs.google.com/document/d/1rKWgUcP2MkUfrYAQm1j6pWeuSfan3xCPvEUt4vfxQJE/edit?>
- http://es.wikipedia.org/wiki/Java_Persistence_API. (s.f.). Obtenido de http://es.wikipedia.org/wiki/Java_Persistence_API
- <http://smslib.org/>. (2015). <http://smslib.org/>.
- <http://www.cantabria.ccoo.es/>. (s.f.). Obtenido de <http://www.cantabria.ccoo.es/>: http://www.cantabria.ccoo.es/comunes/recursos/5/doc165200_Guia_de_usos_de_redes_sociales_de_CCOO_de_Cantabria.pdf
- <http://www.discapnet.es/>. (Diciembre de 2010). Obtenido de <http://www.discapnet.es/>: http://www.discapnet.es/Observatorio/Observatorio_Accesibilidad_redessociales_VersionReducida.pdf
- James Rumbaugh, I. J. (2000). En G. B. Ivar Jacobson, *El Lenguaje Unificado de Modelado. Manual de referencia* (pág. 11). España: Addison Wesley.
- LibrosWeb. (12 de Agosto de 2013). *Introducción a AJAX*. Obtenido de *Introducción a AJAX*: http://librosweb.es/ajax/capitulo_1.html
- Méndez, G. (28 de Octubre de 2008). *Especificación de Requisitos según el estándar de IEEE 830*. Obtenido de IEEE Std. 830-1998: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
- Olmedo, G. (2009). *Implementación de un prototipo para gestión de productos y pedidos en distribuidoras utilizando SMS de tecnología GSM*. Obtenido de <http://repositorio.espe.edu.ec/handle/21000/470>
- Pisco, R. S. (12 de Mayo de 2013). Obtenido de <http://www.dspace.uce.edu.ec/bitstream/25000/1634/1/T-UCE-0011-55.pdf>
- Romero, L. A. (s.f.). Obtenido de http://sistemas.unach.edu.ec/documentos/20130424095108_AC3.pdf
- Tolmo, D. G. (2012). *SISTEMA DE TELECONSULTA MÉDICO*. Obtenido de SISTEMA DE TELECONSULTA MÉDICO: http://www.rcim.sld.cu/revista_24/articulo_pdf/teleconsulta.pdf
- Tórrez, C. L. (Mayo de 2009). *WEB 2.0, PORTAFOLIOS ELECTRÓNICO Y DISEÑO DE ACTIVIDADES EN LÍNEA*. Obtenido de http://issuu.com/carlalopeztorrez/docs/disen__o_de_actividades_en_li__nea

Twitter Inc. (2014). *Twitter for Websites*. Obtenido de Twitter Inc:
<https://dev.twitter.com/overview/documentation>

twitter4j. (2014). *Twitter4J* . Obtenido de Twitter4J : <http://twitter4j.org/en/index.html>

Wikipedia. (17 de Octubre de 2014). *Proceso Unificado de Rational*. Obtenido de
http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational

Wikipedia. (22 de 09 de 2014). *Rich Internet application*. Obtenido de
http://es.wikipedia.org/wiki/Rich_Internet_application

ANEXOS

HOJA DE LEGALIZACIÓN DE FIRMAS

**ELABORADO POR
LEONARDO EDWARD MOLINA NOBOA**



Sr. Leonardo Edward Molina Noboa

**DIRECTOR DE LA CARRERA
ING. MAURICIO CAMPANA**



Sr. Ing. Mauricio Campaña

Sangolquí, 05 Marzo del 2015