

**Universidad de las Ciencias Informáticas**

**Facultad 1**



**Trabajo de Diploma para optar por el título de Ingeniero en  
Ciencias Informáticas**

**Título:** Sistema de Gestión de Notificaciones para el Sistema Único de  
Identificación Nacional de la República de Cuba

**Autores:**

Yanisbel Álvarez Prieto

Arístides Gutiérrez Pina

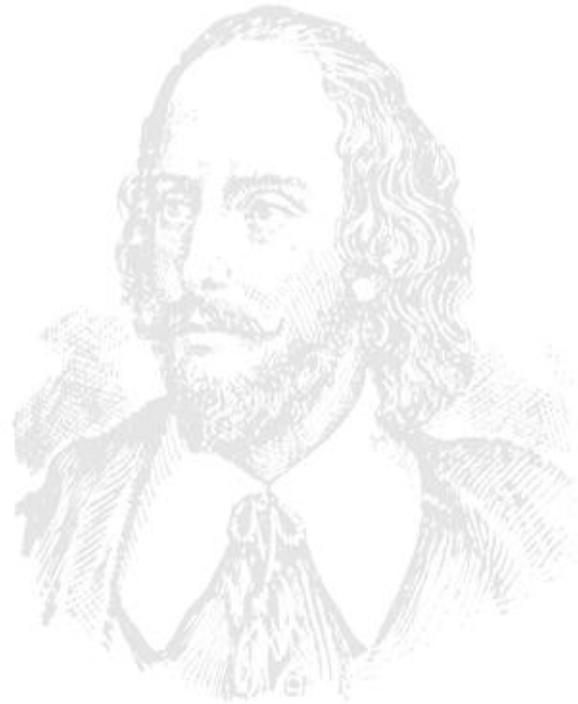
**Tutores:**

Msc. Erik de la Vega García

Ing. Reisbel Machado Rosabal

**Ciudad de la Habana 2012**

*“Sabemos lo que somos, pero no en lo que podemos convertirnos”*



Declaramos que somos los únicos autores de este trabajo y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes \_\_\_\_ del año \_\_\_\_.

---

Firma del Autor  
Yanisbel Álvarez Prieto

---

Firma del Autor  
Arístides Gutiérrez Pina

---

Firma del Tutor  
Msc. Erik de la Vega García

---

Firma del Tutor  
Ing. Reisbel Machado Rosabal

**Yanisbel Álvarez Prieto**

**Correo:** yprieto@estudiantes.uci.cu

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba

**Arístides Gutiérrez Pina**

**Correo:** agpina@estudiantes.uci.cu

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba

**Msc. Erik de la Vega García**

**Correo:** edelavega@uci.cu

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba

**Ing. Reisbel Machado Rosabal**

**Correo:** rmachado@uci.cu

Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba

*A mis padres por ser las personas más importante en mi vida y por darme tanto amor.*

*A mi hermana y mis sobrinos por tanto apoyo.*

*A mi esposo por tanta confianza y cariño.*

*A todas las personas que creyeron en mí.*

*A todos gracias de corazón.*

*Yanisbel.*

*Dedicado de manera especial a las cuatro personas más importantes en mi vida,  
mis padres que son mi inspiración, a mi hermano y a mi esposa.*

*Arístides.*

*A mi mamá Doris por darme tanto cariño y por ser la mejor madre que cualquiera quisiera tener y apoyarme en todas mis decisiones y darme tanto apoyo siempre.*

*A mi papa Gilberto por ser el mejor padre del mundo y por compartir conmigo tanto conocimiento y tanto respeto.*

*A mi hermana Yanisel que aunque no está conmigo físicamente me ha dado mucho apoyo.*

*A mi cuñado que lo quiero mucho.*

*A mis sobrinos Lorena y Marcel David que son lo más lindo que tengo en la vida y me hacen sentir tan feliz de vivirla.*

*Sobre todo a mi esposo por ser la persona que más creyó en mí en estos años de carrera y por darme todo su amor y su cariño, te amo.*

*A mi tío Juan Carlos, a Fabio pero le agradezco mucho por tanto cariño y tanta dedicación para conmigo a Ivón, por todo lo que ha hecho por mí desde que me conoció, por tratarme como una hija y sobre todo por todos los momentos bonitos que viví con ella gracias.*

*A toda mi familia por tanto interés en mi carrera y en mi preparación.*

*A mis suegros por tener un hijo tan maravilloso.*

*A mis tutores por todo su apoyo.*

*A todas mis amistades Karelia, Yanet, José, Mailin, Yeimy, Adrián, Michel, Sacha, Anisley, Julio, Yisel en fin a todos aquellos que desde el primer año en la universidad compartieron conmigo tantos momentos lindos.*

*A todos los que fueron a nuestra boda que de verdad fue el mejor regalo que me pudieron hacer porque fue un momento que compartí con ellos y nunca olvidaré.*

*Yanisbel.*

*A mis padres Nancy y Arístides pues sin ellos nada de esto habría sido posible.*

*A mi hermano.*

*A mi esposa por estar a mi lado en los buenos y malos momentos.*

*A mis suegros, gracias por acogerme casi como un hijo.*

*A mi cuñada.*

*A la gente de la pelota, la gente del básquet.*

*A mis compañeros del proyecto que me ayudaron mucho.*

*A mis amigos, los de la vieja y la nueva guardia.*

*Gracias a todos los que de una forma u otra hicieron posible este momento, muchas gracias.*

*Arístides.*

El Sistema Único de Identificación Nacional (SUIN) es una plataforma desarrollada para la modernización de la gestión de los procesos de identificación de personas en Cuba. Comúnmente su ejecución puede implicar la necesidad de comunicación de informaciones críticas relativas a la ocurrencia de eventos y hechos en los procesos que se ejecutan, tanto para clientes como los funcionarios que allí laboran, lo cual actualmente, se realiza de forma manual.

A partir de esta problemática el presente trabajo basa su objetivo en la creación de un sistema que, integrado al SUIN, permita la comunicación con los clientes y entidades interesadas en los procesos que se desarrollan, basado en el envío de notificaciones por correo electrónico, SMS y mensajería instantánea, reduciendo el tiempo de respuesta en la comunicación.

Para la realización de la solución se estudiaron los sistemas de notificaciones existentes, las aplicaciones orientadas a mensajes y la tecnología *Windows Workflow Foundation 3.5* como modelo de programación. Se utilizó la herramienta *Visual Studio 2010* para implementar el servicio de notificaciones, utilizando el lenguaje C# y el *Framework .NET 4.0* y como gestor de base de datos *Oracle Database Enterprise Edition 11gR2*.

**Palabras Clave:**

Mensajería, notificaciones, servicio, sistema de notificaciones.

## Índice de contenido

Introducción .....	1
Capítulo1: Fundamentación teórica .....	3
1.1 Conceptos asociados .....	3
1.2 Sistema de notificaciones .....	4
1.3 Los Middleware .....	6
1.3.1 Java Message Service .....	6
1.3.2 Amazon SQS .....	7
1.3.3 RabbitMQ.....	9
1.4 Técnica para la creación de mensajes.....	11
1.5 Ambiente de desarrollo.....	12
1.4.1 Metodología de desarrollo MSF for CMMI ProcessImprovement.....	12
1.4.2 Herramientas de modelado .....	13
1.4.3 Acceso a datos .....	15
1.4.4 Plataforma de desarrollo: .NET .....	17
1.4.5 Lenguaje de implementación .....	18
1.4.6 Microsoft Visual Studio 2010.....	19
1.4.7 Modelo de programación .....	19
1.4.8 Técnica de desarrollo webAJAX.....	22
1.4.9 Control de versiones TeamFoundation Server2010 .....	22
Conclusiones parciales .....	23
Capítulo 2: Características del sistema .....	24
2.1. Modelo de dominio .....	24
2.2. Propuesta de la solución .....	25
2.3. Requerimientos del servicio y la aplicación.....	27
2.4. Arquitectura de la solución .....	32
2.5. Patrones de diseño.....	35
2.6. Diagrama de clases.....	37
2.7. Diseño del Workflow.....	38
2.8. Modelo de datos.....	39
Conclusiones parciales .....	42
Capítulo 3: Implementación y prueba.....	43
3.1. Estándares de Codificación .....	43
3.2. Tratamiento de Errores.....	44
3.3. Diagrama de despliegue.....	45
3.4. Diagramas de componentes.....	45
3.5. Pruebas unitarias.....	47
3.6. Pruebas de caja negra realizadas a la Aplicación web .....	47
3.7. Resultado de las pruebas .....	50
3.8. Pruebas de carga del sistema .....	51

Conclusiones parciales .....	54
Conclusiones generales.....	55
Recomendaciones .....	56
Bibliografía referenciada y Bibliografía citada .....	57
Anexos.....	61
Anexo I    Catálogo de requisitos .....	61
Anexo II    Descripción de requisitos funcionales .....	63
Anexo III    Descripción de entidades del servicio y la aplicación .....	74
Anexo IV    Diseño de casos de pruebas.....	78
Anexo V    Procesos del sistema.....	87

## Índice de tablas

Tabla 1: Descripción de los roles del sistema y el servicio .....	27
Tabla 2: Requisitos funcionales del sistema y el servicio .....	28
Tabla 3: Descripción del requisito funcional Modificar conexión .....	29
Tabla 4: Descripción de la entidad DMENSAJENOTIFICACION .....	40
Tabla 5: Descripción de la entidad NTIPOMENSAJENOTIFICACION .....	41
Tabla 6: Diseño de caso de prueba del requisito Adicionar recurso XML .....	48
Tabla 7: Pruebas realizadas a cada requisito funcional por cada iteración del desarrollo .....	50
Tabla 8: Características del Servidor .....	51
Tabla 9: Comparación entre el SUIN y el sistema de notificaciones para el SUIN .....	54
Tabla 10: Requisito funcional Enviar mensaje .....	63
Tabla 11: Requisito funcional Obtener mensaje por usuario .....	64
Tabla 12: Requisito funcional Cambiar estado del mensaje .....	65
Tabla 13: Requisito funcional Gestionar recurso XML .....	66
Tabla 14: Requisito funcional Gestionar tipo de mensaje .....	68
Tabla 15: Requisito funcional Monitorear mensajes .....	70
Tabla 16: Requisito funciona Realizar suscripción a notificaciones .....	72
Tabla 17: Descripción de la entidad NESTADOMENSAJENOTIFICACION .....	74
Tabla 18: Descripción de la entidad DADJUNTONOTIFICACION .....	75
Tabla 19: Descripción de la entidad DRECURSOXML .....	75
Tabla 20: Descripción de la entidad NTIPORECURSOXML .....	76
Tabla 21: Descripción de la entidad DNOTIFICACIONPERSONAINST .....	76
Tabla 22: Descripción de la entidad DDATOMENSAJENOTIFICACION .....	77
Tabla 23: Descripción de la entidad NTIPOADJUNTONOTIFICACION .....	77
Tabla 24: Diseño de caso de prueba del requisito Modificar recurso Xml .....	78
Tabla 25: Diseño de caso de prueba del requisito Eliminar recurso Xml .....	79
Tabla 26: Diseño de caso de prueba del requisito Adicionar mensaje de notificación .....	80
Tabla 27: Diseño de caso de prueba del requisito Modificar mensaje de notificación .....	81
Tabla 28: Diseño de caso de prueba del requisito Eliminar mensaje de notificación .....	83
Tabla 29: Diseño de caso de prueba del requisito Monitorear mensajes .....	84
Tabla 30: Diseño de caso de prueba del requisito Modificar conexión .....	84
Tabla 31: Diseño de caso de prueba del requisito Realizar suscripción a notificaciones .....	86

## Índice de figuras

FIGURA 1: TRANSFORMACIÓN DE UN XML A TRAVÉS DE UN XSLT .....	11
FIGURA 2: METODOLOGÍA MSF FOR CMMI .....	13
FIGURA 3: MODELO DE DOMINIO .....	24
FIGURA 4: PROPUESTA DE SOLUCIÓN.....	26
FIGURA 5: EJEMPLO DE UN PROCESO DEL SUIN .....	27
FIGURA 6: ARQUITECTURA DE LA APLICACIÓN WEB .....	32
FIGURA 7: ARQUITECTURA DE LA ACTIVIDAD DE WORKFLOW .....	34
FIGURA 8: ARQUITECTURA DEL SERVICIO NOTIFICATIONSERVICE.....	35
FIGURA 9: DIAGRAMA DE CLASES DEL SISTEMA.....	37
FIGURA 10: WORKFLOW DEL SERVICIO.....	39
FIGURA 11: MODELO DE DATOS .....	40
FIGURA 12: DIAGRAMA DE DESPLIEGUE.....	45
FIGURA 13: DIAGRAMA DE COMPONENTES DEL SERVICIO.....	46
FIGURA 14: DIAGRAMA DE COMPONENTES DE LA APLICACIÓN .....	46
FIGURA 15: RESULTADO DE LA PRUEBA UNITARIA AL MÉTODO SENDJABBERMESSAGETEXT .....	47
FIGURA 16: REPRESENTACIÓN GRÁFICA DE LAS PRUEBAS REALIZADAS A CADA FUNCIONALIDAD POR ITERACIÓN	51
FIGURA 17: TIEMPO DE RESPUESTA DE LA PRUEBA .....	52
FIGURA 18: TIEMPO DE LLEGADA DE UNA NOTIFICACIÓN SMS.....	52
FIGURA 19: TIEMPO DE LLEGADA DE UNA NOTIFICACIÓN JABBER.....	53
FIGURA 20: TIEMPO DE LLEGADA DE UNA NOTIFICACIÓN CORREO ELECTRÓNICO .....	53
FIGURA 21: TIEMPO PROMEDIO DE LLEGADA DE UNA NOTIFICACIÓN (JABBER, CORREO, SMS) .....	53
FIGURA 22: PROCESO GESTIONAR RECURSO XML.....	87
FIGURA 23: PROCESO GESTIONAR TIPO DE MENSAJE .....	88
FIGURA 24: PROCESO DE ENVÍO DE NOTIFICACIONES .....	88

## Introducción

La evolución de las Tecnologías de la Información y las Comunicaciones (TIC) es un hecho que no pasa desapercibido producto a los apremiantes cambios que han ocurrido, y que continúan desarrollándose, para mejorar e incrementar los servicios que brindan. Debido a los avances de las tecnologías se impuso la necesidad de utilizar formas de comunicación que fuesen más rápidas que las ya existentes, como los correos y teléfonos tradicionales, los que a pesar de su utilidad no satisfacían las necesidades precisas de la sociedad.

Internet permite reducir el costo de la comunicación a cualquier parte del mundo y minimiza la demora de los correos, mediante tal novedad informática, la transferencia del origen al destino es casi inmediata. A raíz de esta necesidad de inmediatez sin importar la distancia, surgen los servicios de mensajería como: el correo electrónico, servicios de mensajes cortos (del inglés SMS) y los innumerables clientes de mensajería instantánea, que detallan una serie de mensajes que contienen información relacionada con alguna suscripción que se haya realizado.

Los clientes de mensajería instantánea, más conocidos como *chats*, han alcanzado un gran auge con el decursar del tiempo, desarrollándose gracias a las infinitas posibilidades que brinda Internet. Representan una forma de comunicación en tiempo real entre dos o más personas que en la actualidad ha experimentado un aumento importante en cuanto a aceptación de los usuarios (1). Internet contribuyó al desarrollo del correo electrónico, servicio de gran utilidad en la actualidad, que posibilita el envío de mensajes haciendo uso de un ordenador o dispositivo conectado a la red, actualmente conquistan a los usuarios permitiéndoles enviar textos y mensajes con posibilidad de adjuntar imágenes, archivos de datos y otros elementos multimedia digitalizados (2).

El servicio de mensajes cortos surge como un nuevo canal de comunicación para complemento de los existentes, su uso está limitado a situaciones urgentes e inmediatas ya que su costo global puede ser considerable. Permite el envío de mensajes entre móviles, teléfonos fijos y otros dispositivos de mano, agilizando así la comunicación entre usuarios de diferentes regiones del planeta (3).

Desde el año 2009 la Universidad de las Ciencias Informáticas (UCI), en colaboración con el Ministerio del Interior (MININT), ha venido desarrollando el Sistema Único de Identificación Nacional de la República de Cuba (SUIN), plataforma concebida con un enfoque a procesos, que permite la gestión segura y eficiente de trámites de identificación, incluyendo la búsqueda en bases de datos, la captura de datos

alfanuméricos, documentales y biométricos, la validación de la identidad por comparación biométrica y la obtención de reportes para este nivel.

A partir de la ocurrencia de eventos y hechos en los procesos que se ejecutan en este sistema, tanto los clientes como los funcionarios que allí laboran, requieren mantenerse informados para ejecutar las acciones pertinentes de acuerdo con la información recibida, lo cual, con el sistema existente, se realiza de forma manual. En este sistema se evidencia retrasos en la comunicación a los usuarios y clientes sobre los eventos que ocurren en los procesos que se desarrollan en las Oficinas de la Dirección Identificación y Registro (DIR) de la población y Dirección de Inmigración y Extranjería (DIE) de la República de Cuba. Los clientes no poseen mecanismos para conocer el estado de sus trámites, de la misma forma que las entidades no cuentan con los métodos para recibir las notificaciones pertinentes ante las irregularidades ocurridas. Existe por parte de los funcionarios dependencia de acciones manuales a la hora de informar cualquier evento ocurrido a lo largo del proceso, lo que conlleva a demoras en los tiempos de ejecución de dichos procesos.

Teniendo en cuenta los argumentos anteriormente expuestos se define como **problema de investigación** a resolver: *¿Cómo lograr mayor agilidad en la comunicación a las personas y entidades interesadas sobre los eventos que ocurren en los procesos del Sistema Único de Identificación Nacional de la República de Cuba?*

El diseño de la investigación permite establecer como **objeto de estudio**: *Procesos de gestión de sistemas de notificaciones*. Para dar solución al problema planteado se precisa como **objetivo general**: *Desarrollar una solución de software que permita gestionar las notificaciones en el Sistema Único de Identificación Nacional de la República de Cuba garantizando mayor agilidad en su ejecución*.

Teniendo como **campo de acción**: *Procesos de gestión de sistemas de notificaciones mediante soluciones web, basadas en Windows Workflow Foundation*.

Como posibles **resultados** de la investigación se espera:

- ✓ Un sistema que permita la gestión de notificaciones en el Sistema Único de identificación Nacional de la República de Cuba.
- ✓ Especificación de requisitos funcionales
- ✓ Diseño de casos de prueba

Para garantizar el cumplimiento del objetivo general se identifican los siguientes **objetivos específicos**:

- ✓ Realizar el estudio referente a los sistemas de notificaciones existentes en la actualidad.

- ✓ Desarrollar el análisis y el diseño del sistema de notificaciones del Sistema Único de Identificación Nacional.
- ✓ Implementar el sistema de notificaciones del Sistema Único de Identificación Nacional.
- ✓ Validar la implementación de la solución.

Para asistir las tareas de la investigación se emplearon los siguientes **métodos científicos**:

**Como métodos teóricos:**

**Histórico-lógico:** Utilizado para estudiar los diferentes sistemas de notificaciones y los software de negociación de mensajes existentes en la actualidad, las ventajas y desventajas, así como su desarrollo a niveles internacionales.

**Analítico-sintético:** Utilizado para recopilar toda la información relacionada al tema de las notificaciones de diferentes medios bibliográficos, para el posterior avance de la solución.

**Modelación:** Utilizado para la creación de modelos (manual de usuario) que visualizan una reproducción resumida de la realidad, relacionados con el objeto de estudio y la solución que se propone.

La presente investigación se encuentra estructurada en capítulos enumerados del 1 al 3, los cuales se desarrollan a partir de los estudios realizados y de la observación de las tareas específicas de los procesos que se despliegan. A continuación se describe el contenido que se aborda en cada uno:

- ✓ **Capítulo 1 Fundamentación Teórica:** Se refleja el estudio del arte del tema a desarrollar de acuerdo con lo más ajustado en materia de notificaciones. Se describen los conceptos fundamentales asociados al dominio del problema realizándose un análisis de la situación actual. Se define el ambiente de desarrollo necesario para el análisis y la implementación del sistema.
- ✓ **Capítulo 2 Características del sistema:** Se explican los conceptos existentes a través del modelo de dominio y la solución que se propone, así como el levantamiento de requisitos que incluye la definición de requisitos funcionales y no funcionales del sistema. Se realiza el modelo de diseño del sistema que se va a implementar. Se describe la arquitectura a utilizar, así como los patrones de diseño puestos en práctica en el desarrollo del sistema y el modelo de datos usado.
- ✓ **Capítulo 3 Implementación y Pruebas:** Se modela la estructura y partes físicas del proceso de envío de notificaciones a través del diagrama de despliegue y de componentes. Se establecen los elementos asociados a las pruebas de calidad a las que se someterá el sistema, con el fin de validar que las funcionalidades implementadas respondan a las necesidades establecidas.

## Capítulo 1: Fundamentación teórica

El objetivo fundamental del capítulo es el análisis de soluciones existentes que basan su funcionamiento en realizar notificaciones mediante los diferentes servicios de envío de mensajes. Se abordan los conceptos fundamentales relacionados a las notificaciones y se describen las herramientas y metodologías utilizadas durante el desarrollo de la solución.

### 1.1 Conceptos asociados

Según la Real Academia Española, “*las **notificaciones** son acciones o efectos de notificar que constituye comunicar formalmente una resolución o dar una noticia con propósito cierto*” (4).

Los **sistemas de notificaciones** son mensajes asíncronos enviados a un usuario por un canal específico, manteniéndolos informados en todo momento. Brindan la posibilidad de enviar información a través de una confirmación (5).

La comunicación de información de forma inmediata se define como **comunicación síncrona**, por otra parte, la **comunicación asíncrona** no es más que la posibilidad de postergar la comunicación de la información hasta que se estime conveniente.

El **correo electrónico** es un servicio de red que permite a los usuarios enviar y recibir mensajes. Los protocolos utilizados para el envío y recepción de correo electrónico varían según los servidores, siendo los más comunes: el Protocolo Simple de Transmisión de Correo (del inglés SMTP) para el envío, protocolo que dispondrá la solución para manejar las notificaciones de este tipo, el Protocolo de Oficinas en su tercera versión (del inglés POP3) o el Protocolo de Acceso a Mensajes de Internet (del inglés IMAP) para la recepción (2).

Varios autores plantea que la **mensajería instantánea**, permite el intercambio de mensajes entre diferentes personas en tiempo real, enviados desde cualquier parte del mundo y recibidos al mismo tiempo; de igual forma permite enviar mensajes aunque el usuario destino no esté conectado en ese instante. La mensajería instantánea marcó sus primeros inicios con la implementación del sistema de PLATO<sup>1</sup>, más tarde el sistema *talk* desarrollado en Unix/Linux y finalmente el sistema Te Busco (del inglés ICQ), que fue el primer sistema de mensajería instantánea para ordenadores con sistemas operativos diferentes del Unix/Linux (6), (1).

---

<sup>1</sup> Lenguaje de programación lógica para la enseñanza de operaciones automáticas.

La **telefonía móvil** o como también es conocida telefonía celular, es un servicio que desempeña en la actualidad un papel fundamental para la sociedad, permitiendo gracias a su portabilidad la comunicación entre personas de distintas partes del mundo, teniendo como su principal función la comunicación de voz y el envío de textos aunque con una cifra precisa de caracteres. Los **SMS** permiten el envío de mensajes de texto entre celulares, sin importar la compañía a la que pertenecen. La característica de mayor importancia que presenta es que a pesar de que el teléfono al que se envíe el SMS esté apagado o fuera del área de cobertura, el mensaje permanece en la red haciéndolo llegar al móvil en cuanto se conecte nuevamente (7).

El **XML** es un conjunto de reglas que sirven para definir etiquetas semánticas<sup>2</sup>, es un metalenguaje que te permite diseñar tu propio lenguaje de etiquetas y se utiliza para contener los datos que conforman el mensaje de notificación que será enviado, debido a la facilidad de implementación que posee y lo aplicable que resulta para los diferentes sistemas.

Los **esquemas XML** (del inglés XSD) proporcionan un medio para definir la estructura, contenido y la semántica de los documentos XML (8).

Los **XSLT** son un estándar que presenta una forma de transformar documentos XML en otros, incluso a formatos que no son XML. Las hojas de estilo XSLT realizan la transformación del documento, utilizando una o varias reglas de plantilla unidas al documento fuente a transformar (9). Mediante los XSLT se transformarán los datos contenidos en el XML y se construirá el mensaje de notificación a enviar posteriormente.

## 1.2 Sistema de notificaciones

Los sistemas de notificaciones son soluciones que permiten el envío de mensajes o notificaciones a partir de suscripciones realizadas sobre algún evento específico. Para dar solución al problema planteado sobre la necesidad de mantener notificados a los usuarios y entidades involucradas con el SUIN, se estudiaron varios sistemas de notificaciones que permiten la comunicación por diferentes servicios.

La red social *Facebook* contiene un sistema de notificaciones que permite a los usuarios conocer la mayoría de las informaciones que se mueven alrededor de sus cuentas, tanto por el correo electrónico,

---

<sup>2</sup> Está vinculada al significado, sentido e interpretación de palabras, expresiones o símbolos.

Jabber<sup>3</sup> o mediante los mensajes enviados directamente al teléfono móvil. Según puntualiza Randy Alonso<sup>4</sup> el sistema de notificación de Cubadebate envía a los usuarios expresamente suscritos por correo electrónico notificaciones con un resumen diario de las noticias más relevantes de la actualidad (10).

### **Servicio Administrativo Identificación Migración y Extranjería en la República Bolivariana de Venezuela**

El Servicio Administrativo Identificación Migración y Extranjería (SAIME), almacena información sobre los datos personales de los ciudadanos venezolanos y extranjeros, así como de cada uno de los trámites realizados en las diferentes direcciones de la entidad, dígame los movimientos migratorios, condición migratoria de los extranjeros, información sobre cédula, pasaporte, visa, prohibiciones, órdenes de busca y captura, antecedentes penales, solicitud y registros de refugiados, control de aprehendidos e indocumentados. El SAIME cuenta con un servicio de notificaciones el cual posee un mecanismo de integración e intercambio de información entre sistemas, además de otras aplicaciones como la notificación de mensajes a personas o entidad ante eventos o cambios de estados en los procesos que se desarrollan en determinado sistema (11).

El estudio del SAIME es de gran interés debido a que su negocio es similar al propuesto para la solución y fue desarrollado en la UCI específicamente en el Centro de identificación y Seguridad Digital (CISED). Según sus creadores el servicio de notificaciones del SAIME propone un sistema que permita a las entidades externas obtener información referente a los procesos de identificación de la población mediante las notificaciones de correo electrónico y SMS (11).

Los sistemas de notificaciones mencionados anteriormente se basan para su creación en características que corresponden al ambiente donde se desarrollan. En el caso concreto del SAIME una limitante es la alta concurrencia que genera el sistema en la base de datos, producto al cúmulo de mensajes que maneja. Debido a que el ambiente de desarrollo de estos sistemas es diferente del propuesto no cumple objetivo la utilización de los mismos, aunque muchas de las soluciones han sido factibles para estudiar como notificar a los implicados en el sistema y los requisitos que debe cumplir su implementación.

---

<sup>3</sup>Servicio de mensajería instantánea.

<sup>4</sup>Licenciado en Periodismo en la Universidad de La Habana. Director del programa de la Televisión Cubana "Mesa Redonda" y del sitio Cubadebate.

### 1.3 Los *Middleware*

A partir del análisis realizado por Roger Pressman<sup>5</sup> en su libro Ingeniería de Software, “los *middleware*<sup>6</sup> son software que se utilizan como intermediarios entre aplicaciones distribuidas sobre plataformas heterogéneas. Estos servicios funcionan como una capa de abstracción distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red)”. El *middleware* puede verse como una capa API<sup>7</sup>, que sirve como base a los programadores para que puedan desarrollar aplicaciones que trabajen en diferentes entornos sin preocuparse de los protocolos de red y comunicaciones en que se ejecutarán (12). Para el desarrollo de la solución se utilizaron los *middleware* orientados a mensaje que brinda la posibilidad de agilizar el proceso de envío de mensaje a través del mecanismo de colas con prioridad que implementan.

#### ***Middleware* orientados a mensajes (MOM)**

Los *middleware* orientados a mensajes son aplicaciones que utilizan los mensajes como método de integración y provee mecanismos para crear, manipular, almacenar y transmitir esos mensajes (13), (14). Permiten que las aplicaciones intercambien información en forma de mensajes compuestos por cabeceras y datos. Los mensajes son enviados de forma asíncrona, el encargado de enviar el mensaje no tiene que esperar una respuesta, envía el mensaje y sigue procesando. Los mensajes son tratados como unidades auto-contenidas las cuales contienen todos los datos necesarios para que puedan ser procesados.

#### 1.3.1 *Java Message Service*

El Servicio de Mensajes de *Java* (del inglés JMS) es un software de negociación de mensajes de código abierto y entra dentro de la categoría de *middleware* de mensajería para el envío entre dos o más clientes. JMS es una parte de la plataforma *Java Enterprise Edition* y está definido por una especificación desarrollada bajo el *Java Community Process*<sup>8</sup>. Se trata de un estándar de mensajería que permite a los componentes de la aplicación basada en la edición JEE crear, enviar, recibir y leer mensajes. Permite la comunicación entre los diferentes componentes de una aplicación distribuida para ser imprecisa, fiable y

---

<sup>5</sup> Ingeniero de software estadounidense, autor y consultor. Presidente de RS (Pressman&Associates) y miembro de la IEEE.

<sup>6</sup> Software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones.

<sup>7</sup> Utilizado por un programa para activar e interactuar con las funciones de otro programa o de un equipo físico. Asegura la conectividad entre todos los componentes de una infraestructura informática. Es la estructura para enlazar todas las aplicaciones en forma integrada.

<sup>8</sup> Permite a las partes interesadas involucrarse en la definición de futuras versiones y características de la plataforma *Java*.

asíncrona. JMS soporta dos modelos de mensajería: punto a punto (cola) y publicación/suscripción (15), (16).

- ✓ **Punto a punto (destino de cola):** En este modelo, se entrega un mensaje de un productor a un consumidor. Los mensajes se envían al destino, que es una cola y luego entregado a uno de los consumidores registrados para la cola. Mientras que cualquier número de productores pueden enviar mensajes a la cola, cada mensaje está garantizado para ser entregado y consumido por un consumidor. Si no están registrados los consumidores para consumir los mensajes, estos se mantienen en cola hasta que un consumidor se registre (16).
- ✓ **Publicación / suscripción (destino tema):** En este modelo se entrega un mensaje de un productor a cualquier número de consumidores. Los mensajes se envían al destino y luego a todos los consumidores activos suscritos. Además, cualquier número de productores pueden enviar mensajes a un destino y cada mensaje puede ser enviado a cualquier número de suscriptores. Si no hay consumidores registrados, el destino no tiene mensajes a menos que tenga suscripción duradera para los consumidores inactivos (16).

Los **suscriptores** pueden registrar su interés en recibir mensajes sobre un tema determinado. En este modelo, ni el editor ni el suscriptor se conocen. Existe una dependencia de tiempo entre los editores y suscriptores. El editor tiene que crear un tema de mensaje para los clientes que se suscriban. El suscriptor tiene que seguir siendo continuamente activo para recibir mensajes, a menos que haya establecido una suscripción duradera. En ese caso, los mensajes publicados mientras que el abonado no está conectado se redistribuyen cuando se vuelve a conectar (15), (16).

El software JMS no es el adecuado para la implementación de la solución debido a que las colas que contienen los mensajes que han sido enviados y están a la espera de ser leídos, contrariamente a lo que su nombre indica, los mensajes no son entregados en el mismo orden en el que fueron enviados y solo garantiza que cada mensaje se procese una vez.

### 1.3.2 Amazon SQS

Cristian Baun<sup>9</sup> en colaboración con otros autores en el libro *Cloud Computing: Web-based Dynamic IT Services* plantean que, “Amazon Simple Queue Service (Amazon SQS) ofrece un sistema de gestión de colas fiable y ampliable para almacenar mensajes a medida que se transfieren entre sistemas” (17).

---

<sup>9</sup>Estudiante de doctorado en el Instituto de Tecnología de Karlsruhe (KIT).

Mediante *Amazon SQS*, los desarrolladores pueden transferir datos entre componentes distribuidos de aplicaciones que realizan distintas tareas, sin perder mensajes y sin necesidad de que cada componente esté siempre disponible. Funciona utilizando la infraestructura de gestión de mensajes a escala web como un servicio web. Cualquier sistema de Internet puede añadir o leer mensajes sin tener instalado ningún software ni configuración de cortafuegos especial. Incluye dentro de sus principales funcionalidades (18):

- ✓ Los desarrolladores pueden crear un número ilimitado de colas con el número de mensajes que deseen.
- ✓ El texto del mensaje puede contener un máximo de 64 KB de texto en cualquier formato.
- ✓ Los mensajes se pueden retener en colas durante un máximo de 14 días.
- ✓ Los mensajes se pueden enviar y leer simultáneamente.
- ✓ Los desarrolladores pueden compartir de forma segura las colas con otros usuarios.
- ✓ Cuando se recibe un mensaje, queda "bloqueado" mientras está procesando. Esto impide que otros sistemas procesen el mensaje simultáneamente. Si el mensaje no se procesa correctamente, el bloqueo vence y vuelve a estar disponible.

*Amazon SQS* ofrece ventajas significativas sobre la complejidad de los sistemas de gestión de mensajes tradicionales (18):

- ✓ **Fiable:** Se ejecuta dentro de centros de datos de alta disponibilidad, lo que garantiza que las colas siempre están disponibles cuando las aplicaciones las necesitan. Para evitar que los mensajes se pierdan o dejen de estar disponibles, todos los mensajes se almacenan en varios servidores y centros de datos.
- ✓ **Ampliable:** Diseñado para permitir que un número ilimitado de sistemas puedan leer y grabar un número ilimitado de mensajes en cualquier momento.
- ✓ **Seguro:** Se proporcionan mecanismos de autenticación para garantizar que los mensajes almacenados en colas estén protegidos frente a un acceso no autorizado.

El uso de la aplicación *Amazon SQS* no es adecuado para la implementación de la solución, debido a que no garantiza que el envío de los mensajes mediante las colas sea del tipo FIFO<sup>10</sup> y cuando un mensaje es eliminado existe la probabilidad que vuelva a ser enviado debido a una de las operaciones que el sistema contiene, que no elimina todas las copias de un mensaje.

---

<sup>10</sup> Estructuras de datos para implementar colas. Se interpreta como; primero en llegar, primero en salir.

### 1.3.3 RabbitMQ

*RabbitMQ* es un software de negociación de mensajes de código abierto que se encuentra dentro de la categoría de *middleware* de mensajería. Implementa el estándar *Advance Message Queuing Protocol* (AMQP). El servidor *RabbitMQ* está escrito en *Erlang*<sup>11</sup> y utiliza el *framework*<sup>12</sup> *Open Telecom Platform* (OTP<sup>13</sup>) para construir sus capacidades de ejecución distribuida y conmutación ante errores. El código fuente está liberado bajo la licencia *Mozilla Public License*. A su vez *RabbitMQ* consta de diferentes partes; pasarelas para los protocolos HTTP, XMPP y STOMP<sup>14</sup>, un servidor de intercambio, librerías de clientes para *Java* y el *framework* *.NET* y el *plugin Shovel* que se encarga de copiar mensajes desde un corredor de mensajes a otros. Este software es utilizado por compañías de gran prestigio como *Google*, *Mozilla* y *AT&T* (19).

### Estándar AMQP

AMQP permite a las aplicaciones enviar y recibir mensajes. En este sentido, es como la mensajería instantánea o correo electrónico. Donde AMQP difiere enormemente es que permite especificar qué mensajes se reciben, dónde y cómo realizan las compensaciones con respecto a la seguridad, fiabilidad y rendimiento. El estándar AMQP es un protocolo abierto en la capa de aplicaciones de un sistema de comunicación. Las características que lo definen son la orientación a mensajes, encolamiento, enrutamiento (tanto punto-a-punto como publicación-suscripción), exactitud y seguridad. Utiliza una serie de entidades que le permite la transferencia de los mensajes (20).

AMQP es diferente de otros estándares de *middleware* por ser (20):

- ✓ **Interoperable:** Todos los clientes AMQP interoperan<sup>15</sup> con todos los servidores AMQP. Diversos lenguajes de programación se pueden comunicar fácilmente. Corredores heredados de mensajes se pueden adaptar para eliminar protocolos propietarios de la red.
- ✓ **Confiable:** Capaz de eliminar las brechas de comunicación y retrasos entre las diferentes plataformas, sistemas y componentes críticos de aplicaciones tanto dentro de la empresa y fuera de los sistemas y organizaciones externas.

<sup>11</sup>Lenguaje de programación altamente concurrente y adaptada a las nuevas tecnologías de sistemas multi-core.

<sup>12</sup>Marco de trabajo.

<sup>13</sup>Plataforma de desarrollo para crear aplicaciones de telecomunicaciones basadas en Erlan. Fue liberado como código abierto en 1998.

<sup>14</sup>Protocolo de colas de mensajes, se utiliza para la comunicación distribuida entre computadoras.

<sup>15</sup>Distintos software pueden operar de manera simultánea.

- ✓ **Unificada:** Proporciona un conjunto básico de patrones de mensajería a través de un protocolo único manejable.

El **corredor de mensaje** es un servidor al que los clientes AMQP se conectan usando el protocolo AMQP. Los corredores de mensajes pueden ejecutarse en un entorno distribuido, pero esta capacidad es específica de la implementación y no está cubierta por la especificación. El **usuario** es la entidad que, mediante la presentación de credenciales tales como una contraseña, puede ser autorizado (o puede no ser autorizado) a conectarse a un corredor. La **conexión** está ligada a un usuario usando por ejemplo TCP/IP o SCTP. EL **canal** es una conexión lógica unida a una conexión, los clientes que realicen operaciones concurrentes mediante una misma conexión deben mantener un canal distinto para cada una de ellas. Las entidades utilizadas para el envío y recepción de mensajes en concreto son todas declaradas dentro de un canal (20).

Las **colas** son las entidades que reciben los mensajes; tienen un nombre y propiedades, pero no tienen tipo. Los clientes pueden suscribirse a las colas, con el efecto de que el corredor de mensajes les entregará el contenido de la cola a los clientes. La cola garantiza que los mensajes son entregados en el mismo orden en que llegaron a la cola, ordenación que se conoce habitualmente como FIFO. Las propiedades de las colas incluyen (20):

- ✓ **Intercambiador alternativo:** Cuando un mensaje es rechazado por un suscriptor o queda huérfano debido a la destrucción de una cola, son reenrutados a un intercambiador alternativo y borrado de la cola.
- ✓ **Pasiva:** La cola no será declarada, pero ocurrirá un error si no existe.
- ✓ **Perdurable:** La cola sobrevivirá a un reinicio del corredor.
- ✓ **Exclusiva:** Solo puede haber un cliente para una cola específica.
- ✓ **Autoborrado:** La cola será borrada tan pronto como no queden suscripciones activas.

Por su parte los **mensajes** no tienen nombre y son publicados en un intercambiador, consisten solamente en un encabezado y un cuerpo de contenido, el encabezado contiene una serie de propiedades opcionales (20):

- ✓ **Inmediato:** El mensaje será tratado como imposible de enrutar si al menos una de las colas que deberían recibir el mensaje no tiene ninguna suscripción.

- ✓ **Modo de entrega:** Indica que un mensaje puede necesitar perdurabilidad. Solo para este tipo de mensajes puede el corredor hacer un intento para impedir la pérdida del mensaje antes de que sea consumido.
- ✓ **Prioridad:** Un indicador (en el rango entre 0 y 9) de que un mensaje tiene precedencia sobre otros.
- ✓ **Vencimiento:** La duración en milisegundos antes de que el corredor pueda tratar el mensaje como imposible de enrutar.

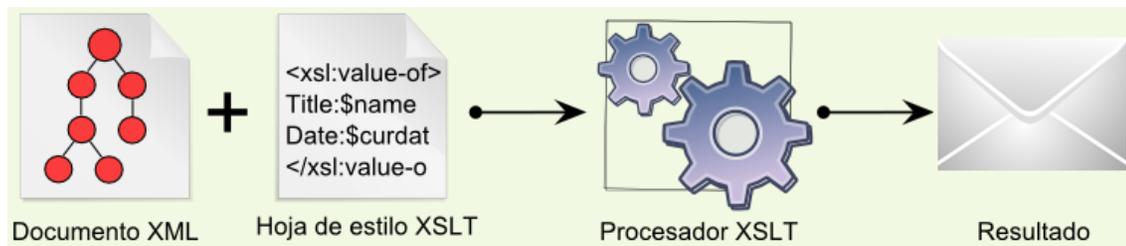
La utilización del software *RabbitMQ* que implementa el estándar AMQP, en la concepción del sistema de notificaciones, se fundamenta en:

- ✓ La capacidad que este posee para la gestión de colas de mensajes, logrando de esta forma disminuir la concurrencia en el servidor de base de datos.
- ✓ La garantía que brinda, haciendo posible que si existe algún problema con el servidor, los mensajes perduren en las colas hasta el restablecimiento del mismo, manteniendo su orden de prioridad.

#### 1.4 Técnica para la creación de mensajes

La creación de mensajes amigables para los usuarios finales, es un reto que conlleva el desarrollo de cualquier sistema. Los sistemas de notificaciones no están exentos de estos, si se tiene en cuenta que su razón de ser es la comunicación constante de información sobre un evento específico. Una de las técnicas más novedosas para la construcción de mensajes, es el uso de XSLT, aprovechando las facilidades que brinda para la transformación de documentos XML en textos planos o HTML.

La transformación se hace posible con la utilización de un procesador de XSLT, programa de ordenador que aplica al documento inicial las reglas de transformación incluidas en la hoja de estilo XSLT y genera el documento final como se muestra en la **Figura 1** (21).



**Figura 1: Transformación de un XML a través de un XSLT**

Los lenguajes de programación más utilizados en la actualidad, posibilitan la realización de dicho procesador mediante un conjunto de clases destinadas al trabajo con estas estructuras.

El correcto desarrollo de un sistema de notificaciones acorde a las exigencias actuales trae consigo la utilización de esta técnica de transformación para garantizar la creación de mensajes amigables para los usuarios finales.

### 1.5 Ambiente de desarrollo

En la Informática, obtener un software de calidad implica la utilización de metodologías, tecnologías y herramientas que permitan uniformar la filosofía de trabajo. Para dar solución al problema planteado, se utiliza el ambiente de desarrollo definido por el proyecto Identificación, Inmigración y Extranjería de la República de Cuba de la UCI.

#### 1.5.1 Metodología de desarrollo *MSF for CMMI Process Improvement*

Definida por el proyecto Identificación, Inmigración y Extranjería de la República de Cuba, *MSF for CMMI*<sup>16</sup> *Process Improvement*<sup>17</sup> es precisamente la metodología propuesta para el desarrollo de la solución, pues brinda un entorno flexible y perfectamente ajustable para su desarrollo, permitiendo a los miembros del equipo asumir uno o varios roles en función de las necesidades del ambiente de desarrollo.

*MSF por CMMI* define cinco fases para el ciclo de vida del proyecto que encapsulan un conjunto de secuencias de actividades de trabajo. Cada fase concluye con un punto de control y cada punto de control brinda una oportunidad para autorizar la continuación o suspensión del proyecto (22), (23). El trabajo en cada fase debe responder una serie de preguntas que van guiando el paso de una fase a otra como se muestra en la **Figura 2**.

---

<sup>16</sup>Del inglés *Capability Maturity Model Integration*. Es un modelo de calidad de software que clasifica las empresas productoras en alguno de los 5 niveles de madurez que define.

<sup>17</sup>Marco de trabajo de Microsoft para el Modelo de Integración de Madurez de las Capacidades.



Figura 2: Metodología MSF for CMMI

- ✓ **Visión:** En esta fase se crea el documento Visión del producto y se establecen los procesos del proyecto.
- ✓ **Planificación:** Se define la línea base de la gestión de configuración del proyecto, se crea el proyecto de base de datos, se conforman los requisitos de calidad del servicio, se crean y validan los escenarios definidos, se definen los requisitos funcionales y no funcionales de la aplicación y se validan estos requisitos. Durante el desarrollo de la fase de planificación se construye la arquitectura y se define tanto el ciclo de vida del proyecto como el plan de iteración.
- ✓ **Desarrollo:** Se realiza el análisis y diseño de la solución a partir de lo definido en la fase anterior, se realiza la implementación de la solución y las pruebas unitarias para la corrección de errores.
- ✓ **Estabilización:** En esta fase se actualizan las pruebas de corrección de errores realizadas en la fase anterior y se comprueban los requisitos del producto.
- ✓ **Implantación:** Se implementa la base de datos y se le realizan pruebas, se establece el entorno de prueba de aceptación de los usuarios y se genera la documentación requerida para el despliegue de la solución.

### 1.5.2 Herramientas de modelado

**Altova Umodel 2009:** Herramienta para el modelado y diseño de los procesos

Herramienta que permite el diseño visual de la solución, admitiendo de manera simple y asequible, dibujar en UML<sup>18</sup> y BPMN<sup>19</sup>. La herramienta *Case*<sup>20</sup> combina una intuitiva interfaz visual con funciones intuitivas, herramienta que permite la confección de los diagramas de despliegue, componentes y la descripción de procesos.

Las características de *UModel 2009* para el desarrollo de software basado en las capacidades de modelado avanzado son (24), (25):

- ✓ Soporte para los 14 tipos de diagramas UML.
- ✓ Generación de código fuente en lenguajes *Java*, *C#* y *VB.NET*.
- ✓ Creación de diagramas de secuencia desde el código fuente de la ingeniería inversa.
- ✓ Generación de documentación personalizable de proyecto.
- ✓ Integración con sistemas de control de versiones.
- ✓ Estrecha integración con *Visual Studio* y *Eclipse*<sup>21</sup>.

### Notación para el modelado de procesos de negocio

El estándar de modelado de procesos (del inglés BPMN) permite representar gráficamente las diferentes etapas del proceso. Diseñado para coordinar la secuencia de eventos y los mensajes que fluyen entre los diferentes procesos participantes, haciendo de BPMN un estándar capaz de expresar más patrones que los diagramas de actividad, pues es gráficamente más rico y con menos símbolos fundamentales, lo que facilita su comprensión por parte del personal no experto; además cubre casi totalmente los patrones de *workflow*<sup>22</sup> con lo cual se le supone una gran expresividad a la hora de especificar procesos (26).

### Lenguaje de modelado UML 2.0

Lenguaje Unificado de Modelado (del inglés UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, a partir de la metodología definida en la fase de Desarrollo se crean los diseños de los diferentes diagramas que conforman el diseño de las soluciones de software.

---

<sup>18</sup>Del inglés *Unified Model Language*. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

<sup>19</sup>Del inglés *Business Process Management Notation*.

<sup>20</sup>Del inglés *Computer Aided Software Engineering*.

<sup>21</sup>Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar proyectos.

<sup>22</sup>Plataforma que permite a los usuarios crear un flujo de trabajo en sus aplicaciones.

Según describe Craig Larman<sup>23</sup>, UML tiene las siguientes características (27):

- ✓ Permite modelar sistemas utilizando técnicas orientadas a objetos. Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así, modelos precisos, no ambiguos y completos.
- ✓ Puede conectarse con lenguajes de programación. Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas y versiones).
- ✓ Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.

### 1.5.3 Acceso a datos

#### **Oracle Database Enterprise Edition 11gR2: Sistema gestor de base de datos (SGBD<sup>24</sup>)**

Entre los sistemas propietarios de gestión de bases de datos más usados en el mundo se encuentra *Oracle*, considerado uno de los sistemas más completos, destacando su soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma mediante el uso de plataformas *grid*<sup>25</sup>, asegurando altos niveles de calidad de servicio e incremento de la flexibilidad de negocio reduciendo además los costes de explotación (28), (29).

La seguridad que proporciona a la información almacenada es uno de los elementos más importantes que hacen de *Oracle Database Enterprise Edition 11gR2* la opción escogida para el desarrollo de la solución de los procesos del SUIN, pues mejora de manera significativa las propiedades de XML utilizados para los mensajes y ofrece nuevas funcionalidades para mejorar los tiempos de respuesta de los procesos.

#### **Herramienta de modelado de la base de datos: ER/Studio**

*ER/Studio* es la herramienta usada para diseñar, generar y mantener el modelo datos de la solución, desde el modelo lógico de los requerimientos de información y reglas del negocio, hasta el modelo físico optimizado por las características específicas de su base de datos de destino, herramienta que se utiliza para modelar la base de datos de la solución.

*ER/Studio* permite visualizar la estructura adecuada, los elementos clave y un diseño optimizado de la base de datos. Incorpora nuevas capacidades colaborativas de modelado y administración, así como

---

<sup>23</sup> Consultor de gestión para la LSS de Nokia Siemens Networks, Alcatel-Lucent y UBS, y actualmente está trabajando con el equipo de liderazgo en Computer Associates en su adopción empresarial LSS.

<sup>24</sup> Los Sistemas de Gestión de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

<sup>25</sup> Infraestructura que permite la integración y el uso colectivo de ordenadores de alto rendimiento, redes y bases de datos.

mayor soporte en la integración de almacenamiento de datos diseñados para visualizar, documentar y compartir el conocimiento. Permite la creación y consolidación de modelos de proyecto en un modelo global. Para soportar la transición de lógico a físico, *ER/Studio* provee a los administradores de base de datos dos nuevas características: capacidad de planificación y modelado seguro. La capacidad de planificación permite a los diseñadores comunicar la forma en que esperan que la base de datos crezca y la funcionalidad de modelado seguro permite a los administradores de la información desarrollar roles de acceso en el modelo lógico, que pueden ser transferidos al modelo físico (30).

### Lenguaje de consultas integrado LINQ

El lenguaje LINQ<sup>26</sup> es utilizado por la solución para facilitar la implementación de la arquitectura para el acceso a datos y permite ejecutar las consultas de una manera fácil y uniforme, además de simplificar la implementación de acceso a los datos. A partir del estudio realizado por Miguel Katrib<sup>27</sup> plantea que las ventajas más importantes que aporta LINQ son (31):

- ✓ Permite reducir a una mínima expresión el desajuste de impedancias<sup>28</sup> en la programación actual.
- ✓ Permite elevar el nivel de abstracción y claridad de la programación y en gran medida indicar únicamente qué se desea obtener, dejando a un motor de evaluación de expresiones los detalles sobre cómo lograr ese objetivo.
- ✓ Seguridad mejorada gracias a la verificación de sintaxis y de tipos por parte de los compiladores.
- ✓ Disponibilidad de ayuda *IntelliSense*<sup>29</sup> desde *Visual Studio 2008*.
- ✓ Las aplicaciones están libres de dependencias de codificación rígida de un motor de datos o de un esquema de almacenamiento. Las asignaciones entre el modelo conceptual y el esquema específico de almacenamiento pueden cambiar sin tener que cambiar el código de la aplicación.
- ✓ Los desarrolladores pueden trabajar con un modelo de objeto de aplicación coherente que se puede asignar a diversos esquemas de almacenamiento, posiblemente implementados en sistemas de administración de base de datos diferentes.

---

<sup>26</sup>Del inglés *Language Integrated Query*.

<sup>27</sup>Dr. Profesor Titular jefe de la cátedra de Programación e Ingeniería de Software de la Facultad de Matemática y Computación de la Universidad de la Habana y miembro de la Academia de Ciencias de Cuba.

<sup>28</sup>Existencia de limitaciones al transmitir información de un medio a otro.

<sup>29</sup>Tecnología que permite completar el símbolo de los nombres que el programador está escribiendo. Sirve como documentación y evita ambigüedad de los nombres de variables, funciones y métodos.

- ✓ La compatibilidad con LINQ proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual.

### **ADO.NET Entity Framework**

*ADO.NET Entity Framework* mediante el mapeo a la base de datos, permite acceder como si fueran clases y permite evaluar el nivel de abstracción de la programación.

Dentro de las principales características del *Entity Framework* se encuentran (32), (33):

- ✓ Se trata de un *Framework* que permite manipular y acceder a los datos como clases *.NET*.
- ✓ Permite realizar un modelado nivel conceptual, es decir, habilita la separación de las aplicaciones y los esquemas de base de datos.
- ✓ Permite una mejor interacción de los datos con el entorno *.NET*.
- ✓ Proporciona una serie de servicios en tiempo de ejecución y en tiempo de diseño que permite a los desarrolladores describir los datos de la aplicación e interactuar con ellos a un nivel de abstracción conceptual.
- ✓ Se apoya en LINQ para definir las consultas a objetos del esquema Modelo Entidad-Relación (del inglés EDM).

#### **1.5.4 Plataforma de desarrollo: .NET**

##### ***.NET Framework 4.0***

Es el componente de *Windows* que se utiliza para la compilación y ejecución de la solución, contiene una biblioteca de clases que tiene una amplia colección orientada a objetos que se utiliza por la solución, así como el componente CLR<sup>30</sup> que permite la administración de memoria y la comunicación remota.

El *framework* brinda las siguientes características (34):

- ✓ Mejoras en *Common Language Runtime*, *ASP.NET* e innovaciones en los lenguajes *Visual Basic* y *CSharp*.
- ✓ *Entity Framework*, donde se permite a los desarrolladores programar con bases de datos relacionales usando objetos *.NET* y LINQ.
- ✓ Servicios de datos de *Windows Communication Foundation (WCF)*<sup>31</sup>, componente que permite crear servicios y aplicaciones que usen protocolo para exponer y usar datos a través de la Web.

---

<sup>30</sup>Del inglés *Common Language Runtime*. Es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma *Microsoft .NET*.

- ✓ Compatibilidad con formularios para nuevas mejoras en la librería AJAX.

### **ASP.NET**

*ASP.NET* es un *framework* para aplicaciones web desarrollado y comercializado por *Microsoft*. Permite a los programadores escribir código *ASP.NET* usando cualquier lenguaje admitido por el *.NET Framework*. Las páginas de *ASP.NET*, conocidas oficialmente como formularios web, son el principal medio de construcción para el desarrollo de aplicaciones.

Este *framework* contiene una serie de características nuevas en distintas áreas tecnológicas como son (31), (35):

- ✓ Integración total de LINQ y del reconocimiento de los datos. Esta nueva característica permite escribir código en idiomas habilitados para LINQ para filtrar, enumerar y crear proyecciones de varios tipos de datos SQL, colecciones, XML y conjuntos de datos usando la misma sintaxis.
- ✓ *ASP.NET AJAX* permite crear experiencias web más eficaces, más interactivas y con un gran índice de personalización que funcionan con los exploradores más usados.
- ✓ Nueva compatibilidad con el protocolo web para generar servicios WCF, como por ejemplo AJAX, JSON, REST, POX, RSS, ATOM.
- ✓ Compatibilidad total con las herramientas de *Visual Studio 2008* para WF, WCF y WPF, incluida la nueva tecnología de servicios habilitados para flujos de trabajo.
- ✓ Nuevas clases en la biblioteca de clases base de *Framework.NET 3.5* que tratan numerosas solicitudes de cliente comunes.

#### **1.5.5 Lenguaje de implementación**

##### **CSharp (C#)**

Es un lenguaje de programación orientado a objetos<sup>32</sup>, se utiliza en la solución para la implementación debido a la facilidad que proporciona su implementación y además ha sido diseñado específicamente para la plataforma *.NET*.

Las ventajas que ofrece C# frente a otros lenguajes de programación son (31):

- ✓ **Sencillez:** C# elimina elementos que otros lenguajes incluyen y que son innecesarios en *.NET*.

---

<sup>31</sup>Marco de trabajo que posibilita el envío de mensajes de un servicio a otro.

<sup>32</sup>Se basa en la idea de un objeto, que es una combinación de variables locales y procedimientos llamados métodos que juntos conforman una entidad de programación.

- ✓ **El código escrito:** C# es auto contenido, lo que significa que no necesita de ficheros adicionales tales como ficheros de cabecera.
- ✓ **Orientación a componentes:** Permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros).
- ✓ **Eficiente:** En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, a diferencia de *Java* en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros.

## XML

Algunos autores coinciden que XML es un conjunto de reglas que sirven para definir etiquetas semánticas para organizar un documento y es un metalenguaje que permite diseñar un propio lenguaje de etiquetas. Con el uso de los XML para la comunicación entre el SUIN y la solución, se logra un lenguaje común y entendible a través de sus etiquetas. Su principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Permite al programador dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas monótonas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello (36), (37).

### 1.5.6 Microsoft Visual Studio 2010

*Microsoft Corporation* define que, la herramienta *Visual Studio 2010* permite el desarrollo de las soluciones haciendo posible que la aplicación sea escalable y con la calidad requerida. Definida por el proyecto Identificación, Inmigración y Extranjería muestra novedosas herramientas que automatizan la realización de pruebas de la interfaz de usuario en aplicaciones web: pruebas manuales, pruebas de rendimiento, pruebas de carga, cobertura de código, generación de datos de pruebas. Herramienta que incluye una versión renovada del servidor *Team Foundation Server*, además de elementos nuevos como *LabManagement*<sup>33</sup> que permite la creación y gestión de entornos virtuales (38).

### 1.5.7 Modelo de programación

#### **Windows Workflow Foundation**

---

<sup>33</sup> Puede crear un entorno a partir de máquinas virtuales existentes e implementar su aplicación o ejecutar pruebas en el entorno.

*Windows Workflow Foundation 3.5* (WWF) posibilita modelar procesos en *workflows*, esta tecnología forma parte de la arquitectura del SUIN y facilita el modelado de los procesos de gestión de mensajes; los mismos están compuestos por una serie ordenada de pasos con un propósito definido de acuerdo a un conjunto de reglas. Los *workflows* proveen al desarrollador de una representación gráfica de las reglas de negocio; se componen por actividades, componentes reutilizables diseñados para cumplir un propósito definido. Esta tecnología brinda la posibilidad de crear actividades personalizadas y genéricas que permitan resolver problemas específicos (39), (40).

Sus principales características son:

- ✓ Posibilidad de comunicarse con otros software y sistemas.
- ✓ Construcción de sistemas escalables.
- ✓ Persistencia de los estados para una posible recuperación en caso de fallas.
- ✓ Posibilidad de realizar el desarrollo de una forma amigable y amena.

### ***Windows Communication Foundation***

La tecnología *Windows Communication Foundation* (WCF) es utilizada por *.NET* para realizar aplicaciones orientadas a servicios. Permite a los programadores generar soluciones de confianza que se integren en diferentes plataformas, intercambia mensajes en varios escenarios de comunicación y posibilita la ejecución de *workflows* a través de las operaciones en los servicios (41).

Es el marco de trabajo que posibilita el envío de mensajes de notificaciones para la solución y forma parte de la arquitectura definida por el SUIN que consta de un paquete de servicios y estándares implementados para la mensajería.

WCF incluye el siguiente conjunto de características (41), (42):

- ✓ **Varios modelos de mensajes:** Los mensajes se intercambian mediante uno de los distintos modelos. El más común es el de solicitud-respuesta, en que un extremo solicita datos de otro extremo y el otro extremo responde. Existen otros modelos, como un mensaje unidireccional, en que un único extremo envía un mensaje sin esperar ninguna respuesta. Un modelo más complejo es el modelo de intercambio de mensajes dúplex, en donde dos extremos establecen una conexión y se envían datos entre sí, como ocurre con un programa de mensajería instantánea.
- ✓ **Metadatos de servicios:** Admite la publicación de metadatos de servicios utilizando los formatos especificados en los estándares de la industria, como el Lenguaje de Descripción de Servicios

Web (del inglés WSDL), Esquemas XML y WS- *Policy*<sup>34</sup>. Estos metadatos pueden utilizarse para generar y configurar automáticamente clientes para el acceso a los servicios de WCF. Los metadatos se pueden publicar sobre el Protocolo de Transferencia de Hipertextos (del inglés HTTP) y el Protocolo Seguro de Transferencia de Hipertextos (del inglés HTTPS).

- ✓ **Contratos de datos:** Dado que WCF se basa en *framework.NET*, también incluye métodos con código sencillo para proporcionar los contratos que desea aplicar. Uno de los tipos de contrato universales es el contrato de datos. Básicamente, mientras se escribe el código del servicio utilizando *Visual C#* o *Visual Basic*, la forma más sencilla de controlar los datos consiste en crear clases que representan una entidad de datos con propiedades que pertenecen a la misma. Incluye un completo sistema para trabajar con los datos de esta manera tan sencilla. Cuando se crean las clases que representan los datos, el servicio genera automáticamente los metadatos que permiten a los clientes ajustarse a los tipos de datos diseñado.
- ✓ **Varios transportes y codificaciones:** Los mensajes pueden enviarse con cualquiera de los protocolos y codificaciones integrados. La combinación más frecuente de protocolo y codificación consiste en enviar mensajes SOAP<sup>35</sup> codificados de texto utilizando HTTP usado en la Red Informática Mundial (del inglés WWW).
- ✓ **Mensajes confiables y en cola:** Admite el intercambio de mensajes confiable utilizando sesiones confiables implementadas sobre *WS-Reliable Messaging*<sup>36</sup> y manejando el Servidor de Envío de Mensajes de *Microsoft* (del inglés MSMQ).
- ✓ **Mensajes duraderos:** Un mensaje duradero es aquel que nunca se pierde debido a una interrupción de la comunicación. Los mensajes que forman parte de un modelo de mensajes duraderos siempre se guardan en una base de datos. Si se produce una interrupción, la base de datos le permite reanudar el intercambio de mensajes cuando se restablezca la conexión. También puede crear un mensaje duradero utilizando WWF.
- ✓ **Compatibilidad con AJAX y REST:** REST es un ejemplo de una tecnología de web 2.0 en evolución. WCF se puede configurar para procesar datos XML sin formato que no se ajustan en un

---

<sup>34</sup> Especificaciones de tecnologías basadas en servicios web del W3C. Permite a los programadores de servicios web anunciar sus políticas relativas a seguridad y calidad de servicio.

<sup>35</sup> Del inglés Simple Object Access Protocol. Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

<sup>36</sup> Protocolo que permite que los mensajes que se entregarán de manera fiable entre aplicaciones distribuidas en la presencia de fallas de los componentes de software, sistema o red.

sobre SOAP. También se puede extender para admitir formatos XML concretos, como ATOM<sup>37</sup> e incluso formatos que no sean XML, como JSON<sup>38</sup>.

- ✓ **Extensibilidad:** La arquitectura de WCF tiene varios puntos de extensibilidad. Si se necesita una función adicional, existe una serie de puntos que le permiten personalizar el comportamiento de un servicio.

### 1.5.8 Técnica de desarrollo web AJAX

Se utiliza para realizar llamadas a los servicios sin necesidad de recargar las páginas así como para lograr mayor calidad visual. AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML". No es una tecnología en sí misma, sino que se trata de varias tecnologías independientes que se unen para conformarla. Las tecnologías que forman AJAX son (43):

- ✓ XHTML<sup>39</sup> y CSS<sup>40</sup>, para crear una presentación basada en estándares.
- ✓ DOM<sup>41</sup>, para la interacción y manipulación dinámica de la presentación.
- ✓ XML, XSLT y JSON, para el intercambio y la manipulación de información.
- ✓ XMLHttpRequest<sup>42</sup>, para el intercambio asíncrono de información.
- ✓ JavaScript, para unir todas las demás tecnologías.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página mediante la creación de un elemento intermedio entre el usuario y el servidor, ya que el intercambio de información con el servidor se produce en un segundo plano. Su capa intermedia mejora la respuesta del sistema ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando por la respuesta del servidor (31).

### 1.5.9 Control de versiones *Team Foundation Server 2010*

*Microsoft Visual Studio Team Foundation Server 2010* es la plataforma de colaboración central en la solución de administración del ciclo de vida de las aplicaciones de *Microsoft*, que automatiza el proceso de entrega de software y permite a las organizaciones administrar proyectos de desarrollo de software de

<sup>37</sup> Estándar popular de RSS.

<sup>38</sup> Del inglés *JavaScript Object Notation Formato ligero para el intercambio de datos*.

<sup>39</sup> Del inglés *Extensible HyperText Markup Language*. Es básicamente HTML expresado como XML válido.

<sup>40</sup> Del inglés *Cascading Style Sheets*. Lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

<sup>41</sup> Del inglés *Document Object Model*. Es una interfaz de la plataforma y del lenguaje neutro, que permite a los programas y scripts acceder y actualizar dinámicamente el contenido, estructura y estilo de los documentos.

<sup>42</sup> Es una API disponible en lenguajes web de scripting como JavaScript del navegador.

forma eficiente durante su ciclo de vida. *Team Foundation Server* 2010 permite al equipo ser más ágil, colaborar de un modo más eficaz y ofrecer software de más calidad, así como adquirir y compartir conocimiento institucional. Los datos y artefactos del proyecto de seguimiento de elementos de trabajo, control de código fuente, compilaciones y herramientas de pruebas se acumula en un almacén de datos y eficaces informes que proporcionan tendencias históricas, seguimiento completo y visibilidad en tiempo real de la calidad y el progreso respecto al propósito empresarial (44). Se utiliza para lograr mayor cohesión entre las soluciones propuestas por el equipo de desarrollo, para centralizar el almacenamiento del código fuente y para lograr mejor gestión en cuanto al control de versiones.

### **Conclusiones parciales**

El estudio de los diferentes sistemas de notificaciones arrojó la necesidad de realizar un sistema que permita la comunicación entre el Sistema Único de Identificación Nacional y los involucrados en los procesos que aquí se desarrollan.

El análisis referente a los software de negociación de mensajes posibilitó determinar, que el software *RabbitMQ* es la herramienta idónea a utilizar debido a las facilidades que brinda.

El análisis realizado a las diferentes herramientas y tecnologías permitió definir el ambiente para el desarrollo de la solución.

## Capítulo 2: Características del sistema

Durante el desarrollo del presente capítulo se especifica el modelo de dominio con sus relaciones y conceptos fundamentales y se definen los requisitos funcionales y no funcionales de la solución. Se describe la arquitectura utilizada por el servicio, la actividad y la aplicación, así como los patrones de diseño que permiten soluciones estándares para problemas comunes de programación. Se diseña el diagrama de clases, el cual representa las clases que intervienen en la solución y la descripción de cada una de ellas.

### 2.1. Modelo de dominio

Al realizar el análisis correspondiente, no se pudieron identificar procesos de negocios bien claros, solo elementos conceptuales; por lo que se propone un modelo de dominio para obtener una vista general de las entidades o conceptos que intervienen en el negocio. El modelo de dominio es un artefacto que incorpora comportamientos y datos que pueden ser empleados en los casos donde el negocio no comprende actividades demasiado cambiantes. La metodología definida para el desarrollo de la solución propone la creación de este modelo de dominio en la fase de Planificación. A continuación se observa en la **Figura 3** todos los conceptos y relaciones que forman parte de la solución así como las descripciones de cada uno de ellos.

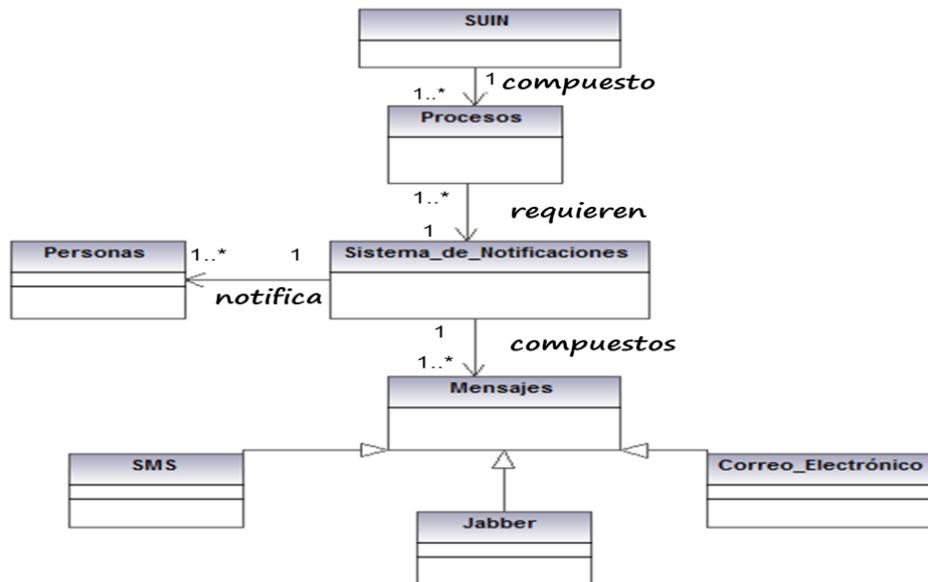


Figura 3: Modelo de dominio

- ✓ **SUIN:** Sistema Único de Identificación Nacional, sistema que se utiliza en las Oficinas de Carné de identidad y Registro de población, donde se realizan los procesos concernientes a la identificación de personas nacionales.
- ✓ **Procesos:** Conjunto de actividades que se desarrollan en el SUIN.
- ✓ **Sistema de notificación:** Mecanismo que posibilita el envío de mensajes utilizando diferentes medios de comunicación.
- ✓ **Personas:** Individuos implicados en los procesos que se desarrollan en el SUIN.
- ✓ **Mensajes:** Acciones que permiten la comunicación y a su vez están compuestos por un origen, un destino y el cuerpo del mensaje.
- ✓ **SMS:** Servicio de envío de mensajes por vía telefónica el cual requiere de un número y el texto a enviar.
- ✓ **Correo electrónico:** Servicio de envío de mensajes mediante una computadora u otro dispositivo conectado a una red que requiere un destino, asunto y el texto a enviar.
- ✓ **Jabber:** Servicio de envío de mensajes mediante una computadora u otro dispositivo conectado a una red que requiere del texto y un destino.

## 2.2. Propuesta de la solución

La arquitectura definida por el SUIN se encuentra basada en una solución cliente servidor, desarrollada en la plataforma *.NET* con el *Framework de .NET 4.0*. El núcleo principal del servidor de aplicaciones es un motor de procesos que utiliza la tecnología *WWF 3.5*, que centra su funcionamiento en una arquitectura orientada a servicios. La tecnología *WWF* permite la creación de actividades genéricas basadas en las necesidades propias de los sistemas. Los procesos en el SUIN se encuentran desarrollados con esta tecnología, haciendo posible que, la creación de una actividad de *workflow* personalizada que facilite la agregación de puntos de notificación en el SUIN, sea la mejor forma de lograr la comunicación entre el sistema y el servicio de notificaciones.

El SUIN contiene un conjunto de servicios que utilizan la tecnología *WCF 4.0* garantizando alta disponibilidad en cada uno de ellos. Para la confección del servicio de notificaciones se utiliza esta tecnología permitiendo la comunicación con la actividad del *workflow* anteriormente expuesta. En la creación de los mensajes de notificación se utiliza una estructura *XML* que almacena los datos necesarios para la construcción del mensaje a enviar. Los datos contenidos en esta estructura se validan a partir del *XSD* y se transforman en el mensaje a enviar a través del *XSLT* correspondiente al tipo de notificación

requerida, aprovechando las facilidades que brinda el *framework de .NET* para el trabajo con estas estructuras de datos. Para evitar la concurrencia en la base de datos los mensajes son enviados a las colas gestionadas por el software *RabbitMQ*, garantizando su posterior envío a través del servicio de mensajería instantánea, correo electrónico o SMS. Se propone la creación de una aplicación web que permita la gestión de los tipos de mensajes de notificación, la gestión de las estructuras utilizadas para la validación y creación de los mensajes a enviar y el monitoreo de las notificaciones enviadas como se muestra en la **Figura 4**.

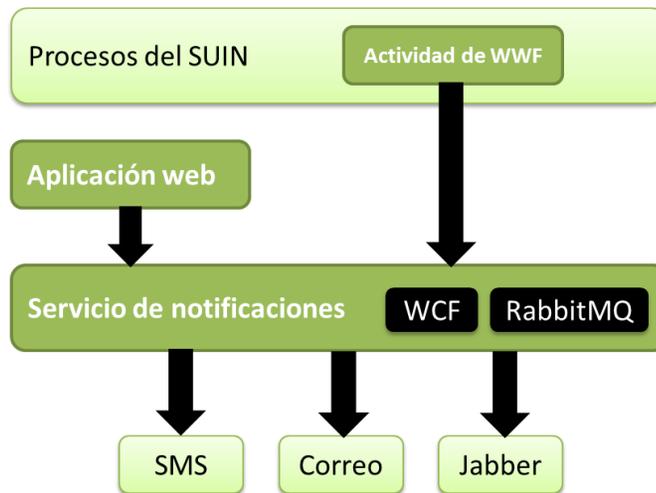


Figura 4: Propuesta de solución

### Ciclo de vida de las notificaciones

El sistema de notificaciones que se propone consta de 5 pasos para garantizar la correcta creación y envío de los mensajes de notificación. El proceso de las notificaciones comienza con la identificación de los puntos de notificación en los *workflows* definidos en el SUIN como se muestra en la **Figura 5**. La creación del XSD y XSLT correspondiente al tipo de notificación identificada en el proceso anterior es el segundo paso para lograr las notificaciones esperadas. El tercer paso comienza con la utilización de la aplicación web para gestionar las estructuras y los tipos de notificaciones creados a partir de estas, permitiendo definir el servicio de mensajería utilizado ya sea SMS, correo electrónico o Jabber para cada tipo de notificación a realizar.

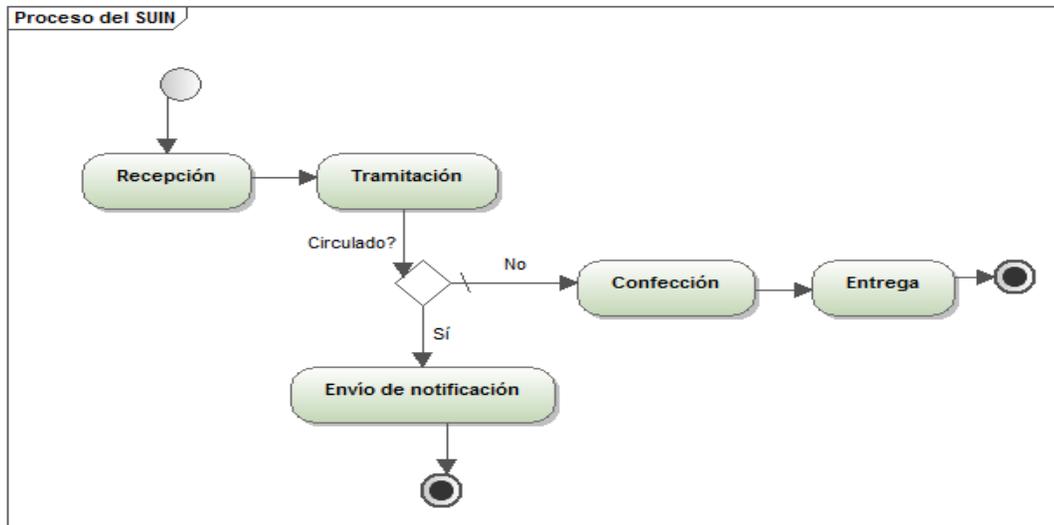


Figura 5: Ejemplo de un proceso del SUIN

Una vez creado el tipo de mensaje de notificación a enviar se coloca la actividad personalizada de *workflow* en el punto de notificación anteriormente identificado. Esta actividad facilita de forma genérica la configuración, creación del XML con los datos necesarios para construir el mensaje y la comunicación con el servicio de notificaciones, haciendo posible que se genere una notificación cuando el proceso modelado en el *workflow* donde esta se aloja, se ejecute. El último paso se desarrolla en el servicio de notificaciones; consiste en la validación y la transformación del XML con los datos en el mensaje a enviar y su posterior envío. Utilizando el XSD definido para la notificación generada se valida el XML, una vez validado se transforma en el mensaje a enviar a través del XSLT correspondiente al tipo de mensaje de notificación deseado. Esta transformación permite la realización de mensajes de una forma genérica, transparente a los desarrolladores, amigable y entendible por el usuario final. Posteriormente se procede al envío de la notificación utilizando el servicio de mensajería instantánea, correo electrónico o SMS.

### 2.3. Requerimientos del servicio y la aplicación

La metodología *MSF for CMMI*, establece que durante la fase de Planificación del producto se identifiquen las características que describen un cumplimiento parcial de los requisitos del cliente. Estos requisitos tienen dos clasificaciones: los requisitos funcionales para definir las funcionalidades finales del producto en cuanto a necesidades del negocio y los requisitos no funcionales que garantizan la usabilidad, seguridad y calidad del producto. A continuación se observa en la **Tabla 1** los roles fundamentales que intervienen en la solución.

**Tabla 1: Descripción de los roles del sistema y el servicio**

Rol	Objetivo
Desarrollador	Hace uso de la actividad creada con WWF y realiza las configuraciones requeridas para incluirla en su solución posibilitando el envío de las notificaciones.
Sistema	Realiza las acciones que sean necesarias de acuerdo con las configuraciones realizadas por el desarrollador, garantizando el correcto funcionamiento de la solución.
Administrador del Sistema	Es el encargado de monitorear todas las funcionalidades de la aplicación y realizar las configuraciones requeridas.

### Definición y especificación de requisitos funcionales

Los requerimientos funcionales de un sistema describen las funcionalidades o los servicios que debe cumplir según las especificaciones del cliente, buscando como resultado calidad y satisfacción. Un requisito funcional es la condición o capacidad que tiene que ser alcanzada o poseída por un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente (45). A continuación, en la **Tabla 2** se enuncian los requisitos funcionales con los que cuenta el sistema y el servicio.

**Tabla 2: Requisitos funcionales del sistema y el servicio**

Requisito funcional	Nombre del requisito funcional
RF.1	Enviar mensaje
RF.2	Obtener mensaje por usuario
RF.3	Cambiar estado del mensaje
RF.4	Gestionar recurso XML
RF.5	Gestionar tipo de mensaje
RF.6	Monitorear mensajes

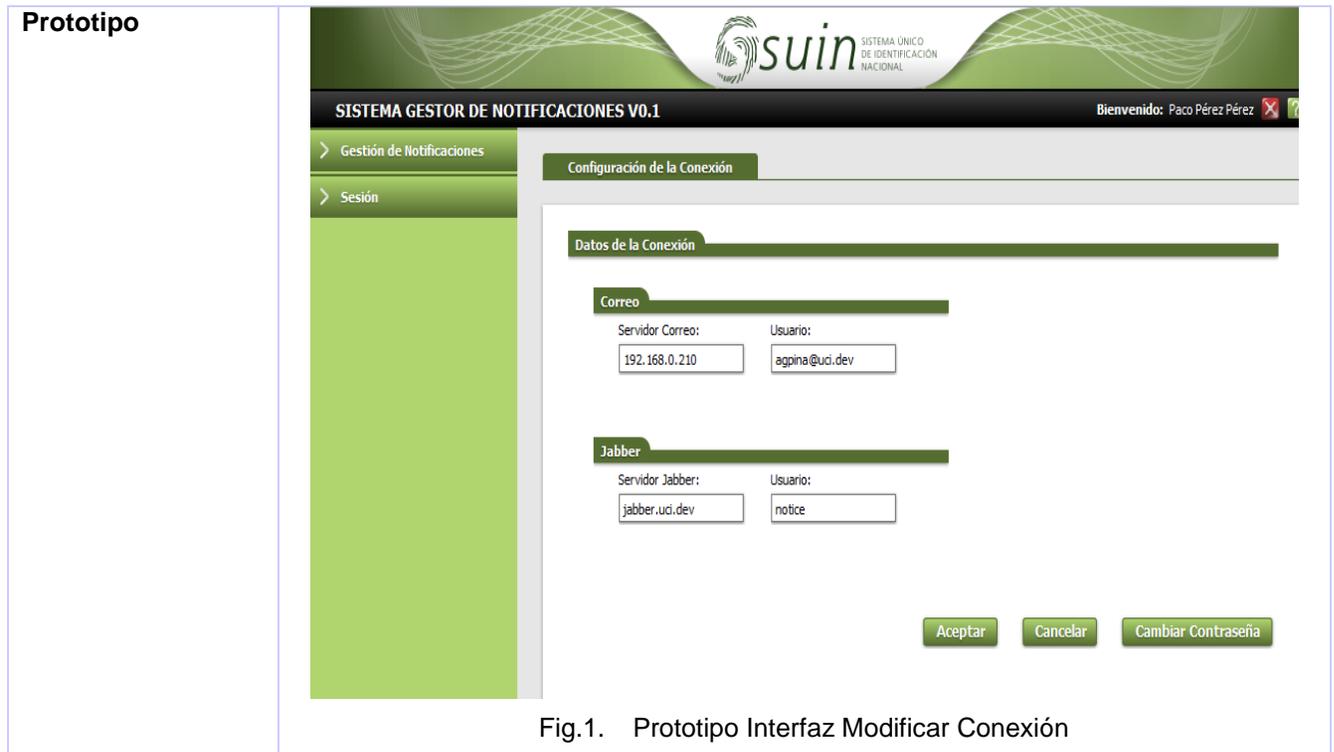
RF.7	Modificar conexión
RF.8	Realizar suscripción a notificaciones

A continuación, en la **Tabla 3** se describe el requisito funcional Modificar conexión que pertenece al *NotificationService*, el resto de la especificación de requerimientos se puede encontrar en el **Anexo II**, así como el catálogo de requisitos que se encuentra en el **Anexo I**.

**RF7 Modificar conexión**

**Tabla 3: Descripción del requisito funcional Modificar conexión**

<b>Propósito</b>	Modificar conexión
<b>Roles</b>	Administrador del sistema
<b>Precondiciones</b>	1. El administrador del sistema debe estar autenticado.
<b>Descripción</b>	<p>7.1 Mostrar los datos para la configuración de la conexión.</p> <ul style="list-style-type: none"> <li>a. Correo electrónico</li> <li>b. Jabber</li> </ul> <p>7.2 Mostrar los datos para la conexión de Correo.</p> <ul style="list-style-type: none"> <li>a. Servidor de correo</li> <li>b. Usuario</li> </ul> <p>7.3 Mostrar datos para la conexión de Jabber.</p> <ul style="list-style-type: none"> <li>a. Servidor Jabber</li> <li>b. Usuario</li> </ul> <p>Permitir introducir los datos para la configuración de la conexión.</p> <p>7.3.1 Se cambia la configuración existente, si se selecciona la opción Aceptar.</p> <p>7.3.2 Se muestra la página principal, si se selecciona la opción Cancelar.</p> <p>7.3.3 Se muestra un formulario, si se selecciona la opción Cambiar contraseña.</p> <p>7.4 Establecer los nuevos valores para la conexión.</p>
<b>Validaciones</b>	1. No pueden haber campos vacíos.
<b>Postcondiciones</b>	1. Se ha modificado la conexión.



### Definición y especificación de requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en las propiedades como las características que hacen al producto atractivo, usable, rápido y confiable. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto (45).

A continuación se detallan los requisitos no funcionales definidos para la solución, contenidos dentro de los requisitos no funcionales especificados por el proyecto Identificación, Inmigración y Extranjería de la República de Cuba.

#### Software

RnF 1. Utilizar Sistema Operativo: *Windows Server 2008 R2*.

RnF 2. Utilizar *.NET framework v3.5* o superior.

RnF 3. Utilizar Internet *InformationServer 7.5*.

#### Diseño e implementación

RnF 4. Utilizar las tecnologías que brinda el *framework .NET* sobre la programación orientada a objetos.

RnF 5. Garantizar la implementación usando el lenguaje C#.

RnF 6. Garantizar como sistema gestor de bases de datos, *Oracle Database Enterprise Edition 11gR2*.

RnF 7. Garantizar que el sistema debe desarrollarse usando el IDE *Visual Studio Team System 2010*.

RnF 8. Utilizar el *Microsoft Visual Studio Team Foundation Server 2010* como control de código fuente.

### **Seguridad**

RnF 9. Garantizar el tratamiento de excepciones.

RnF 10. Garantizar el control de acceso al sistema.

### **Usabilidad**

RnF 11. Garantizar que el servicio posea estructura y diseño homogéneo; los términos utilizados se establecerán acorde al negocio del servicio mejorando el entendimiento de los usuarios.

RnF 12. Garantizar que el sistema web podrá ser utilizado por cualquier usuario con las siguientes características:

- ✓ Conocimientos básicos relativos al uso de una computadora.
- ✓ Conocimientos básicos del sistema operativo *Windows*.

### **Disponibilidad**

RnF 13. Garantizar que el servicio esté disponible las 24 horas durante los 7 días de la semana.

### **Fiabilidad**

RnF 14. Garantizar la consistencia de los datos, realizándose comprobaciones y validaciones automáticas en todos los casos posibles.

### **Eficiencia**

RnF 15. Garantizar que el servicio sea capaz de soportar una cantidad escalable de peticiones y dar respuestas efectivas y rápidas.

### **Requisitos de licencia**

RnF 16. Para el desarrollo del sistema se necesitan un conjunto de aplicaciones, plataformas, sistemas operativos, gestores de bases de datos, herramientas, que son sistemas propietarios y necesitan de licencias para su buen desempeño y soporte, las cuales son:

- ✓ Visual Studio Team System 2010.
- ✓ PL/SQL Developer.
- ✓ Oracle Database Enterprise Edition.
- ✓ Windows Server 2008 R2.
- ✓ Embarcadero ER Studio 8.0.

### 2.4. Arquitectura de la solución

La arquitectura definida para la implementación del sistema de notificaciones está desarrollada siguiendo el mismo patrón de capas utilizadas por el SUIN, posibilitando reducir al máximo el acoplamiento y aumentar la reutilización entre cada una de las capas. El sistema de notificaciones consta de 3 componentes fundamentales:

- ✓ Aplicación web que permite gestionar los tipos de mensajes.
- ✓ Actividad que permite la comunicación entre el servicio y los procesos.
- ✓ Servicio que permite el envío de las notificaciones.

### Arquitectura de la aplicación web

La arquitectura se encuentra representada por 5 capas lógicas como se muestra en la **Figura 6**. La distribución de las capas permite que se realicen grandes cambios sin siquiera tener que realizar cambios en las demás capas. Una vez definida la comunicación entre ellas se realizará solo a nivel de interfaces que permiten trabajar de manera transparente a las instancias reales.

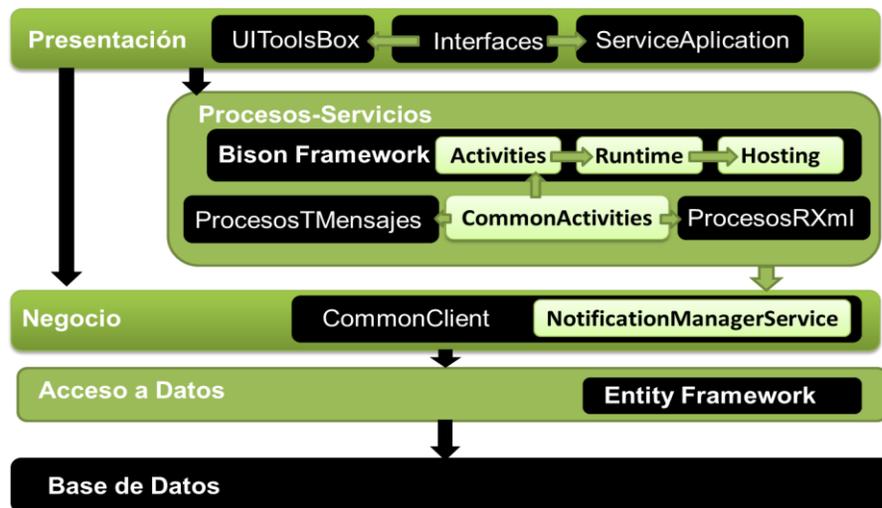


Figura 6: Arquitectura de la aplicación web

**Capa de Presentación:** Esta capa permite la comunicación entre los usuarios y la aplicación utilizando las clases interfaz *GestionarTMensajes*, *GestionarRXml*, *MonitoreoMensajes*, ver **Anexo II**, logrando igualdad en sus componentes visuales mediante el uso del ensamblado *UIToolBox*<sup>43</sup>. La capa presentación también posee el servicio ASMX<sup>44</sup> *ServiceApplication* el cual, permite cargar datos de forma rápida haciendo uso de AJAX. Está representada por el proyecto web de la aplicación y se comunica con la capa de Procesos y servicios y la capa de Negocio.

**Capa de Procesos y Servicios:** Se encuentra basada en el *framework Bison*<sup>45</sup>, compuesto por un conjunto de elementos y componentes complementarios para el desarrollo de *workflows* basados en la tecnología .NET 3.5. Contiene los *workflows* que permiten modelar los procesos de la solución utilizando la tecnología WWF 3.5, ver **Anexo V**. En esta capa se utiliza el componente *CommonActivities* el cual provee a las soluciones antes mencionadas de un conjunto de actividades personalizadas para definir comportamiento genérico y para garantizar la utilización de servicios específicos. Se relaciona con la capa de Presentación a la que brinda servicios y con la capa de Negocio de la que consume servicios.

**Capa de Negocio:** Realiza la lógica de negocio de la aplicación utilizando para esto el *NotificationManagerService* que se encuentra contenido dentro del paquete de servicios *CommonClient*. Este servicio tiene la responsabilidad de manejar las entidades del negocio y se comunica con el acceso a datos para obtener la información.

**Capa de Acceso a Datos:** Es el componente que da soporte a las funcionalidades de la capa de negocio que se encuentran relacionadas con una fuente de datos. Está directamente relacionada con los servicios definidos en el negocio haciendo uso de la fábrica de componentes y conectores, que implementan cada una de las funcionalidades definidas y el mecanismo de comunicación utilizado, estos conectores se enlazan con la fuente de datos mediante el componente *Entity Framework*. Su principal función es realizar una implementación de las interfaces definidas en la capa de negocio y al mismo tiempo trabajar directamente con la fuente de datos establecida.

---

<sup>43</sup> Componente con el objetivo de lograr unanimidad en las interfaces.

<sup>44</sup> Tecnología de .NET que brinda la posibilidad de publicar servicios web que permitan actualizar las interfaces en el navegador sin que el usuario los perciba a través de AJAX.

<sup>45</sup> Complemento para el motor de procesos que permite la implementación de workflow. Su desarrollo trae consigo una disminución en el tiempo de desarrollo y proporcionan una mayor aproximación a los usuarios de negocio.

**Capa de Base de datos:** Está constituida por tablas que permiten el almacenamiento de la información referente a los tipos de notificaciones, estructuras para la construcción de los mensajes y la información necesaria para garantizar el correcto funcionamiento del servicio de notificaciones.

**Arquitectura de la actividad *NotificationActivity***

La actividad de *workflow* tiene como objetivo posibilitar la comunicación entre el SUIN y el servicio de notificaciones. Creada utilizando la tecnología WWF 3.5 logrando la correcta integración con el componente *CommonActivities*, dicho componente forma parte de la capa de Procesos-Servicios de la arquitectura principal del SUIN permitiendo que la actividad pueda ser utilizada en los *workflows* que aquí se modelan. *NotificationActivity* hace posible que la configuración de las notificaciones se haga de una forma más fácil y realiza la creación del XML utilizado para contener los datos del mensaje a enviar como se observa en la **Figura 7**.

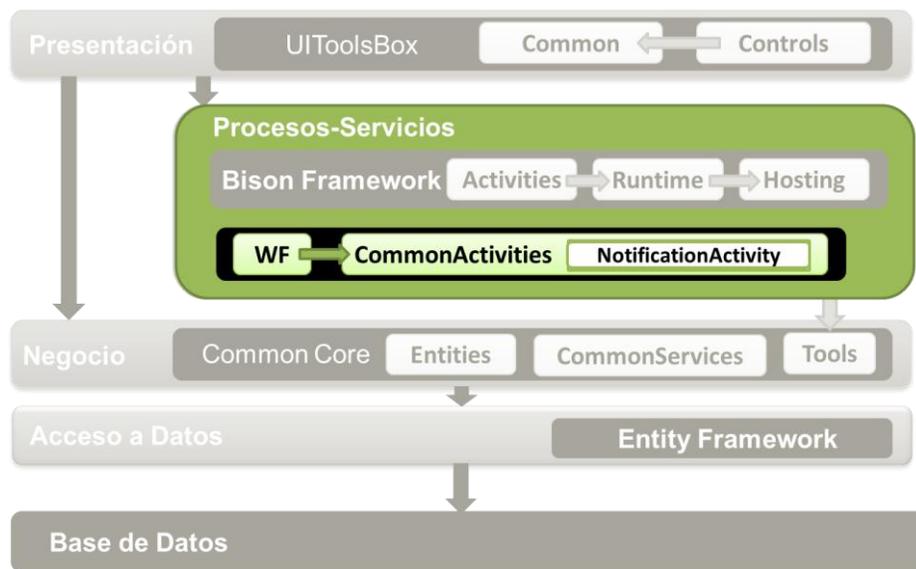


Figura 7: Arquitectura de la actividad de *workflow*

**Arquitectura del servicio de notificaciones**

La arquitectura del servicio se encuentra representada por 3 capas lógicas como se puede observar en la **Figura 8**. Está diseñada siguiendo un patrón de capas bien definidas para reducir al máximo el acoplamiento y aumentar la reutilización entre cada una de las capas.

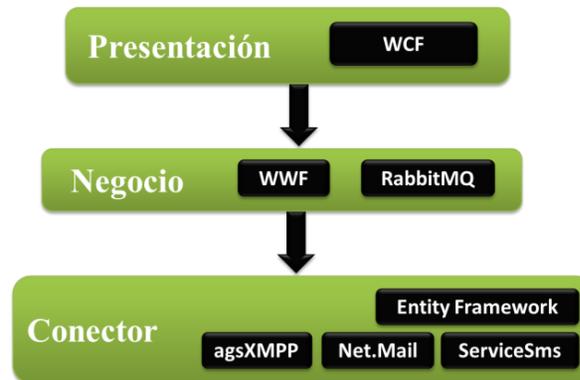


Figura 8: Arquitectura del servicio *NotificationService*

**Presentación:** Esta capa permite la comunicación con la actividad *NotificationActivity* para el envío de mensajes utilizando para esto la tecnología WCF. Posibilita también la definición del contrato del servicio, la publicación de sus entidades y brinda las interfaces de comunicación para las funcionalidades definidas.

**Negocio:** En esta capa se reciben las peticiones de la capa superior y se implementa la lógica de negocio del servicio realizando las peticiones a la capa conector y manejando los datos necesarios para el negocio del servicio. Es la intermediaria de la arquitectura del servicio, utiliza la tecnología WWF y el software de negociación de mensajes *RabbitMQ* para el manejo de las colas de mensajes.

**Conector:** La capa conector permite establecer una conexión entre las entidades del negocio y la base de datos interactuando con el modelo y las entidades generadas por el *Entity Framework*. En el conector se realizan las operaciones básicas de acceso a datos utilizando el lenguaje de consultas LINQ. En esta capa se utiliza el componente *agsXMPP* para enviar las notificaciones a través del servicio de mensajería instantánea, el componente *Net.Mail* para notificar mediante correo electrónico y el servicio *ServiceSms* el cual permite realizar el envío de SMS.

## 2.5. Patrones de diseño

Los patrones de diseño son soluciones simples y elegantes a problemas comunes del diseño orientado a objetos. Existen problemas durante el desarrollo de software que se repiten o que son análogos, que responden a un cierto patrón. Por lo que es necesario tener una colección de dichos patrones con las soluciones más óptimas.

Los patrones garantizan que la solución desarrollada siga las modernas técnicas orientadas a objetos facilitando mayor entendimiento en su construcción y posibilitando que sus componentes sean reutilizables.

Los patrones GoF se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (46). A continuación se exponen ejemplos de algunos de los patrones GoF utilizados para el desarrollo de la solución.

**Creacionales:** El objetivo de estos patrones es abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados (46).

- **Instancia única:** El uso de este patrón se evidencia en la solución con la creación de la instancia a la clase *Runtime* garantizando la existencia de una instancia única a la misma.

**Comportamiento:** Definen la comunicación e iteración entre los objetos del sistema (46).

- **Memento:** Este patrón permite volver a estados anteriores del sistema, su uso en la solución está ligado a la utilización del *framework Bison* el cual hace posible el almacenamiento y recuperación de los estados que componen los *workflows* que aquí se modelan.

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, algunos de los patrones GRASP que se evidencian en el desarrollo de la aplicación son (27):

- **Alta cohesión:** Detalla cómo cada elemento debe realizar una labor única dentro del sistema, y no desempeñada por el resto de los elementos. Las clases en la solución tendrán un número relativamente pequeño de métodos, con funcionalidades altamente relacionadas, logrando así una alta cohesión.
- **Bajo acoplamiento:** El patrón propone el diseño de clases que son más independientes, lo que reduce el impacto del cambio y facilita la reutilización en otros sistemas. Las clases del sistema serán por naturaleza genéricas y se podrán reutilizar fácilmente posibilitando tener un grado de acoplamiento bajo.
- **Experto:** El uso de este patrón se pone de manifiesto en la utilización de la clase *JabberConection*, la cual contiene exclusivamente los datos para garantizar la conexión y el envío de mensajes a través del Jabber, siendo esta la total responsable de realizar dicha acción; garantizando que para la realización de labores la responsabilidad sea de la clase que contiene los datos involucrados.

- Controlador:** Describe la asignación de la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. El uso de este patrón se evidencia en la clase *SendMessageRService* siendo esta la principal encargada de realizar la lógica de negocio del servicio de notificación.

### 2.6. Diagrama de clases

El diagrama de clases se utiliza para el proceso de análisis y diseño del sistema, donde se crea el diseño conceptual de la información que se manejará. Este artefacto se genera durante la fase de desarrollo definida por la metodología. En la **Figura 9** se observa el diagrama de clases genérico para la solución, donde están representadas las clases que intervienen en el proceso del servicio.



Figura 9: Diagrama de clases del sistema

### Descripción de las clases entidades

- ✓ **AppMessage:** Clase que posibilita el intercambio entre el cliente de la aplicación web y el servicio.
- ✓ **Message:** Clase que contiene los datos que conforman el mensaje a enviar.
- ✓ **MessageJabber:** Clase que contiene los datos de los mensajes que son enviados mediante el Jabber.
- ✓ **MessageMail:** Clase que contiene los datos de los mensajes que son enviados mediante correo electrónico.
- ✓ **MessageSms:** Clase que contiene los datos de los mensajes que son enviados mediante SMS.

### 2.7. Diseño del Workflow

Los *Workflows* constituyen una tecnología de modelado de procesos basados en flujos de trabajo, donde se encuentra la mayor parte de la lógica de negocio a partir de los procesos mejorados. La propuesta de solución está compuesta por el *workflows* que se muestra en la **Figura 10**, que representa el proceso de envío de mensajes.

A continuación se describen las principales actividades utilizadas (47):

- ✓ **ConfigurationActivity:** Permite configurar y registrar el *workflow* para acciones internas a través de las colas. Dentro de sus funcionalidades se encuentran gestionar las propiedades de búsqueda y el registro de las operaciones.
- ✓ **CallExternalMethodActivity:** Invoca el método especificado por las propiedades *InterfaceType* y *MethodName* quienes indicarán el lugar donde se aloja el servicio y método a invocar.
- ✓ **IfElseActivity:** Prueba una condición en cada bifurcación y realiza actividades en la primera bifurcación donde se cumple que la condición es verdadera.
- ✓ **CodeActivity:** Actividad que permite agregar al flujo de trabajo, código de los lenguajes de programación *Visual Basic* o *C#*.

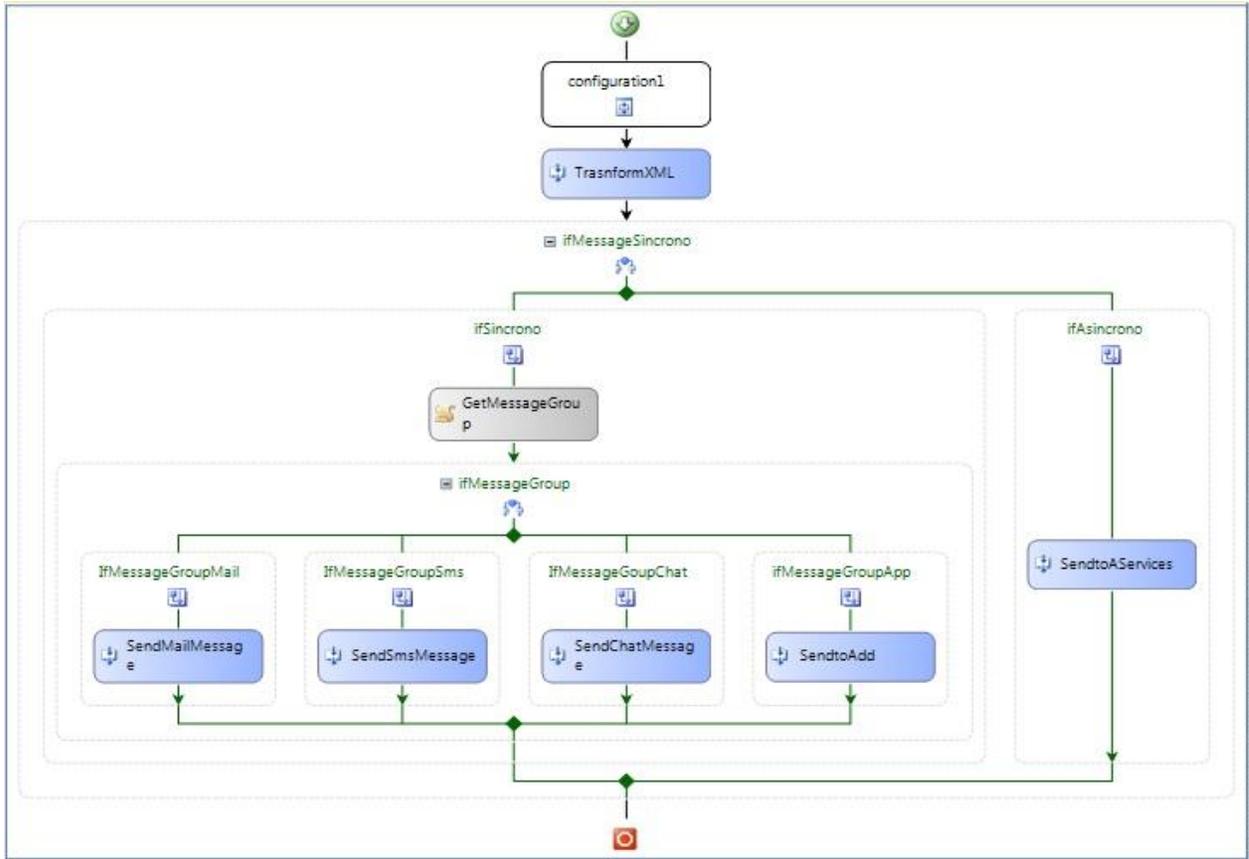


Figura 10: *Workflow* del servicio de notificaciones

### 2.8. Modelo de datos

A partir del análisis de las clases persistentes y el modelo de domino se genera el modelo de datos. El modelo relacional no representa clases, sino entidades que concluyen siendo esquemas relacionales, como se muestra en la **Figura 11**.

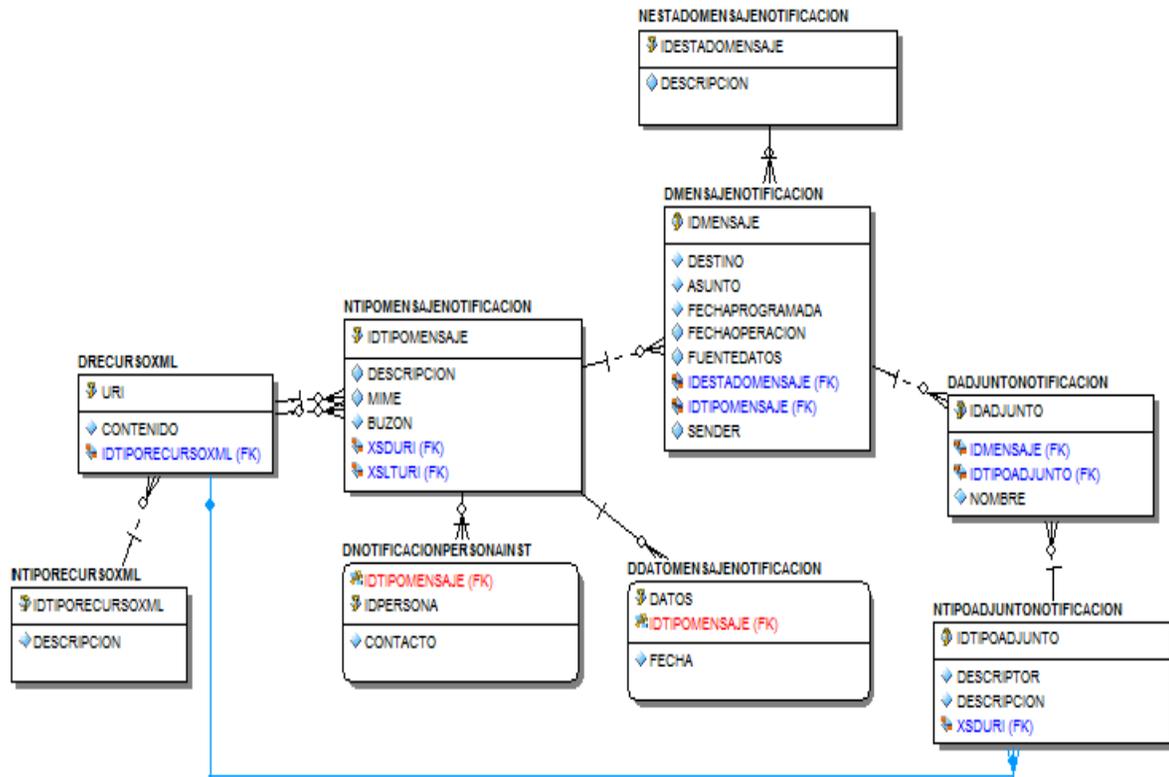


Figura 11: Modelo de datos

**Descripción de entidades**

En la **Tabla 4** y **Tabla 5** se describen las entidades principales *dmensajenotificacion* y *ntipomensajenotificacion* pertenecientes al modelo de datos utilizado en el SUIN, se nombran los atributos de ambas entidades especificando el tipo de datos, la descripción y si el atributo puede ser nulo o no.

**Tabla 4: Descripción de la entidad DMENSAJENOTIFICACION**

<b>Nombre de la Entidad:</b> DMENSAJENOTIFICACION			
<b>Descripción de la Entidad:</b>			
<b>Servicio:</b> <i>NotificationService</i>			
Atributo	Tipo de Dato	Nulo	Descripción

IDMENSAJE	Binary(16)	No	Identificador del mensaje
DESTINO	Varchar(100)	No	Destino del mensaje
ASUNTO	Varchar(100)	Si	El asunto del mensaje
FECHAPROGRAMADA	Date	No	Fecha se debe enviar el mensaje
FECHAOPERACION	Date	No	Fecha en que se envió el mensaje
FUENTEDATO	Varchar(1000)	No	El XML con los datos que conforman el mensaje
IDESTADOMENSAJE	Numeric(4,0)	No	El identificador del estado del mensaje
IDTIPOMENSAJE	Numeric(4,0)	No	El identificador del tipo de mensaje
SENDER	Varchar(500)	Si	Persona que envía el mensaje

Tabla 5: Descripción de la entidad NTIPOMENSAJENOTIFICACION

<b>Nombre de la Entidad:</b> NTIPOMENSAJENOTIFICACION			
<b>Descripción de la Entidad:</b>			
<b>Servicio:</b> NotificationService			
Atributo	Tipo de Dato	Nulo	Descripción
IDTIPOMENSAJE	Numeric(100)	No	Identificador del tipo de mensaje
DESCRIPCION	Varchar(100)	No	La descripción del mensaje
MIME	Varchar(100)	Si	Texto o HTML
BUZON	Varchar(100)	No	El tipo de protocolo
XSDURI	Varchar(500)	Si	Dirección del XSD
XSLTURI	Varchar(500)	No	Dirección del XSLT

Para ver las descripciones detallada de las entidades que conforman el modelo de datos para el servicio y la aplicación referirse a **Anexo III**.

### **Conclusiones parciales**

La obtención del modelo de dominio facilitó la comprensión de los conceptos asociados y permitió definir los componentes a desarrollar: la actividad de *workflow*, el servicio de notificaciones y la aplicación web.

Se definieron para el desarrollo de la solución, ocho requisitos funcionales y dieciséis requisitos no funcionales que permitieron una mejor descripción de las funcionalidades y cualidades del sistema.

La definición de la arquitectura orientada a servicio posibilita la integración de la solución con el Sistema Único de Identificación Nacional y permite flexibilidad entre los componentes que la conforman.

## Capítulo 3: Implementación y prueba

Durante el presente capítulo se describe la implementación del software a través de los diagramas de componentes y los estándares de codificación utilizados. Se define en el diagrama de despliegue la distribución de los nodos necesarios para el despliegue de aplicación. Se muestra el resultado de las pruebas unitarias, pruebas de caja negra y las pruebas de carga realizadas tanto al servicio como a la aplicación para comprobar el correcto funcionamiento de la solución.

### 3.1. Estándares de Codificación

Los estándares de codificación definidos por el proyecto Identificación, Inmigración y Extranjería permiten que la legibilidad del código fuente repercuta directamente en lo fácil que el programador comprenda el sistema de software. El mantenimiento del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores o mejorar el rendimiento. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad, es establecer un estándar de codificación sobre el que se efectuarán revisiones del código de rutinas.

#### Estilos para la capitalización:

**Pascal:** La primera letra en el identificador y la primera letra de cada subsiguiente palabra concatenada se capitalizan. Se pueden utilizar los identificadores de Pascal en caso de tres o más caracteres. Por ejemplo: *BackColor* (48).

**Camel Case:** La primera letra en el identificador se pone en minúscula y la primera letra de cada subsiguiente palabra concatenada en mayúscula. Por ejemplo: *backColor* (48).

**Mayúscula:** Todas las letras en el identificador se capitalizan. Convención que se utiliza solo para los identificadores que constan de dos o menos letras. Por ejemplo: *System.IO*, *System.Web.UI* (48).

**Sensibilidad a mayúsculas:** Para evitar confusiones y garantizar la interoperabilidad entre lenguajes, se siguieron las siguientes reglas sobre el uso de mayúsculas y minúsculas (48):

- ✓ No utilizar nombres o identificadores que requieran ser *case sensitivity*.

- ✓ No utilizar *namespaces* que solo se diferencien en el uso de las mayúsculas.
- ✓ No utilizar funciones con nombres de parámetros que se diferencian solo en el uso de la mayúscula.
- ✓ No utilizar *namespaces* con nombres de clases que se diferencian solo en el uso de las mayúsculas.
- ✓ No crear clases con propiedades que se diferencian solo en el uso de las mayúsculas.
- ✓ No crear clases con métodos que se diferencien solo en el uso de las mayúsculas.

**Evitando confusión de nombre y tipo:** Con el objetivo de incrementar la legibilidad del código se emplearon comentarios en todas las declaraciones de clases y funciones más complejas. También se organizó el código de forma estructurada, en bloques de código, para una mejor lectura del mismo.

**Nombrar los servicios del *workflow*:** Los nombres de servicios de negocio deben terminar en *Service* y los nombres de servicios de *Runtime* deben terminar en *RService*.

### 3.2. Tratamiento de Errores

Para la obtención de buenos resultados con el uso de la herramienta es preciso que el desarrollador este informado sobre los errores que cometa haciendo uso de la misma. Los mensajes de errores que emite el sistema se muestran en un lenguaje de fácil comprensión para los usuarios. Para obtener un control de los errores que se pueden emitir a lo largo de la ejecución de la solución, se crearon las clases *AppExcepciones* y *ServiceExcepciones* para la aplicación y el servicio respectivamente. En el interior de estas clases están contenidos los diferentes tipos de errores que pueden ocurrir a lo largo de la ejecución de la solución como por ejemplo: errores de conexión cuando se pretende acceder al servidor de *RabbitMQ* y al servidor de base de datos, errores en los permisos adquiridos por los usuarios para acceder a la base de datos y errores detectados durante la configuración del servicio. Los errores producidos por problemas con la conexión a los servicios de mensajería, traen consigo que los mensajes sean enviados nuevamente a las colas contenidas en el servidor *RabbitMQ* e intenten enviarse hasta el restablecimiento de los servicios.

### 3.3. Diagrama de despliegue

Según Jacobson<sup>46</sup> “un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación y los nodo<sup>47</sup> puede contener instancias de componentes de software, objetos y procesos”. Los diagramas de despliegue muestran las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes (49). A continuación, se muestra en la **Figura 12** el diagrama de despliegue por el que está contenida la solución.

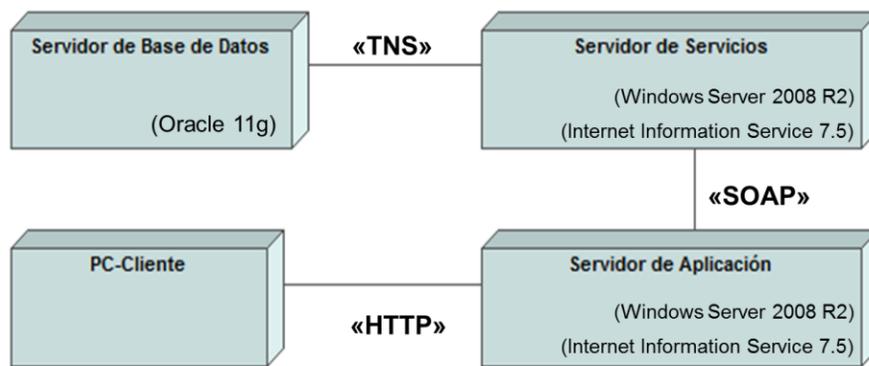


Figura 12: Diagrama de despliegue

El diagrama de despliegue para la solución, está compuesto por las PC-Clientes que se comunican con el Servidor de Aplicación mediante el protocolo “HTTP”, este Servidor de Aplicaciones que contiene la aplicación del SUIN, consume servicios a través de una comunicación por el protocolo de transporte “SOAP” con el Servidor de Servicios, este a su vez interactúa con el Servidor de Base de Datos a través del protocolo de *Oracle* “TNS” para realizar todas las funciones de acceso a datos.

### 3.4. Diagramas de componentes

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces, mostrando los elementos de diseño del software. Este diagrama modela cómo el sistema está desglosado en componentes y las dependencias entre ellos (49).

<sup>46</sup>Maestría en Ingeniería Eléctrica en la Universidad Tecnológica Chalmers. Doctorado lenguaje constructor para grandes sistemas en tiempo real en la Universidad Tecnológica Real (Kungliga Tekniskahögskolan) de Estocolmo en 1985.

<sup>47</sup>Objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento.

A continuación, se muestran los diagramas de componentes del servicio y la aplicación en la **Figura 13** y **Figura 14** respectivamente.

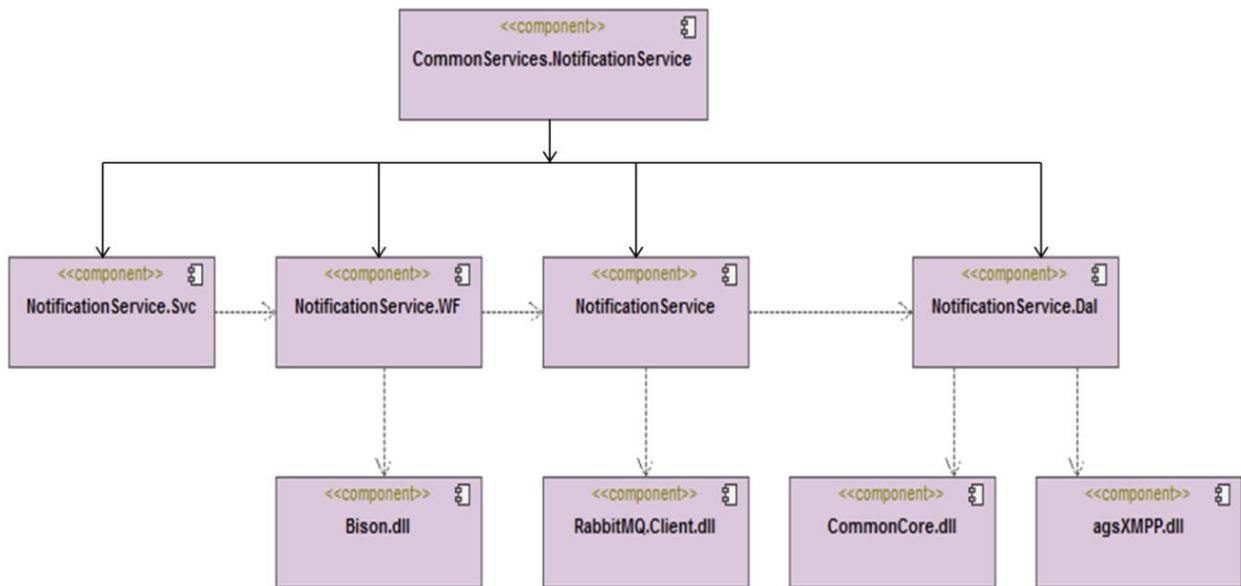


Figura 13: Diagrama de componentes del servicio

El *CommonServices.NotificationService* contiene cuatro componentes principales: *NotificationService.Svc* que incluye todo lo referente a la configuración y a la exposición del servicio. *NotificationService* que en conjunto con el *NotificationService.WF* manipulan todo lo relacionado a la lógica del servicio utilizando el *Basion.dll* y *RabbitMQ.dll* para la conexión con el servidor de *RabbitMQ*. *NotificationService.Dal* que abarca el trabajo de acceso a datos y utiliza el *CommonCore.dll* para el trabajo de seguimiento de error y la identificación de instancias y el *ags.XMPP.dll* para la conexión al servidor de Jabber.

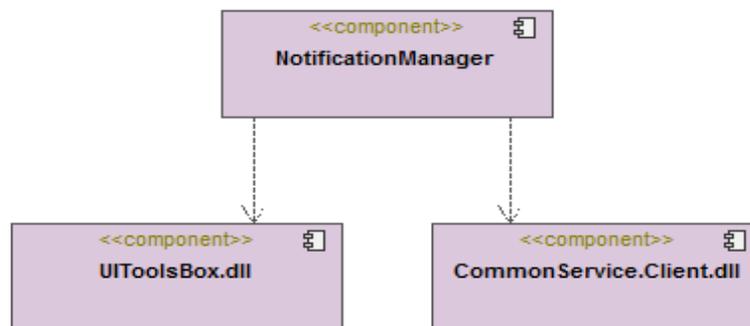


Figura 14: Diagrama de componentes de la aplicación

El *NotificationManager* contiene las interfaces y los servicios web necesarios para el desarrollo de la aplicación, utiliza el componente *UIToolsBox.dll* que posibilita unanimidad entre las interfaces y el *CommonService.Client.dll* que contiene la referencia de los servicios que son los que manejan el negocio de los componentes.

### 3.5. Pruebas unitarias

Las pruebas unitarias representan una forma de examinar un fragmento de código por lo que tiene gran importancia para los desarrolladores y el equipo de pruebas. La herramienta para pruebas del *Visual Studio 2010* permite generar pruebas unitarias específicas para métodos y funcionalidades (12).

Las pruebas unitarias se utilizan para ejecutar el código fuente llamando directamente a los métodos de una clase, pasando los parámetros apropiados y asignando variables para su posterior compilación. Estas pruebas se realizan durante la fase de desarrollo según establece la metodología utilizada, realizándose solo a las funcionalidades que se consideren críticas para la implementación de la solución, se aplicaron a los métodos *SendJabberGroupMessageText* y *SendEmailText* arrojando resultados satisfactorios. A continuación se muestran los resultados de las pruebas realizadas en la **Figura 15**.

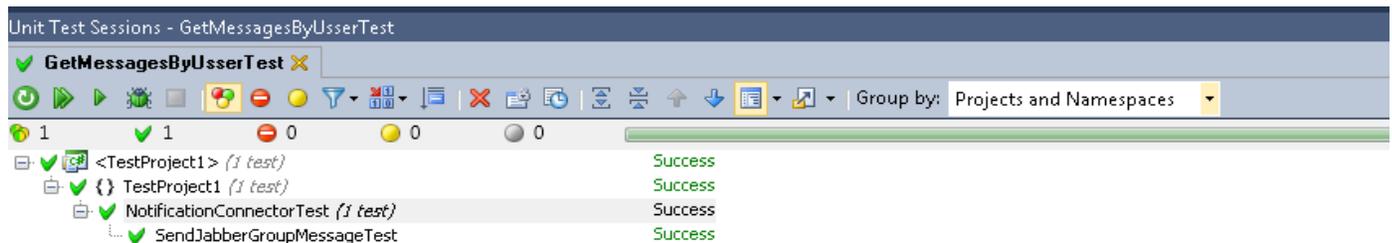


Figura 15: Resultado de la prueba unitaria al método *SendJabberMessageText*

### 3.6. Pruebas de caja negra realizadas a la aplicación web

En la ingeniería de software, los casos de pruebas son un conjunto de condiciones o variables mediante las cuales el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito. Lo que caracteriza un diseño de caso de

prueba es que hay una entrada conocida y una salida esperada, los cuales son formulados antes de que se ejecute la prueba (12).

A continuación se muestra en la **Tabla 6** el diseño del caso de prueba realizado al requisito funcional Gestionar recurso XML del escenario Adicionar recurso XML. Para observar en su totalidad los casos de pruebas realizados a los requerimientos del sistema referirse al **Anexo III**.

**Tabla 6: Diseño de caso de prueba del requisito Adicionar recurso XML**

Escenario	Descripción	Descripción	Tipo de recurso	URI	Respuesta del sistema	Flujo central
ESC 4.1 Adicionar recurso XML correctamente.	El sistema muestra una página con los campos requeridos para adicionar un recurso XML.	V/Este xslt se usa para notificar con respecto a un circulado.	V/XSLT	V/http://com monservices. betas.uci.dev /NotificationService/NoBorrar/message.xslt	El sistema muestra un mensaje "El recurso se adicionó correctamente".	1-Gestionar recurso XML. 2-Adicionar recurso XML. 3-Aceptar.
ESC 4.2 Adicionar recurso XML correctamente.	El sistema muestra una página con los campos requeridos para adicionar un recurso XML.	V/Este xsd es el que valida el xml que se usa para formar el mensaje utilizando xslt.	V/XSD	V/http://com monservices. betas.uci.dev /NotificationService/NoBorrar/message.xsd	El sistema muestra un mensaje "El recurso se adicionó correctamente".	1-Gestionar recurso XML. 2-Adicionar recurso XML. 3-Aceptar.
ESC 4.3 Adicionar recurso XML incorrectamente.	El sistema muestra una página que permite introducir datos para adicionar un recurso XML.	I/152\$.%/	V/XSLT	V/http://com monservices. betas.uci.dev /NotificationService/NoBorrar/message.xslt	El sistema muestra un mensaje "Existen datos incorrectos".	1-Gestionar recurso XML. 2-Adicionar recurso XML. 3-Aceptar.
ESC 4.4 Adicionar recurso XML incorrectamente.	El sistema muestra una página que permite introducir datos para adicionar un recurso XML.	V/Este xslt se usa para notificar con respecto a un circulado.	I/	V/http://com monservices. betas.uci.dev /NotificationService/NoBorrar/message.xslt	El sistema muestra un mensaje "Seleccione un tipo de recurso".	1-Gestionar recurso XML. 2-Adicionar recurso XML. 3-Aceptar.

ESC 4.5 Adicionar recurso XML incorrectamente.	El sistema muestra una página que permite introducir datos para adicionar un recurso XML.	V/Este xslt se usa para notificar con respecto a un circulado.	V/XSLT	/	El sistema muestra un mensaje "Cargue un archivo".	1-Gestionar recurso XML. 2-Adicionar recurso XML. 3-Aceptar.
ESC 4.6 Adicionar recurso XML dejando campos vacíos.	El sistema muestra una página que permite introducir datos para adicionar un recurso XML.	Vacío/	V/XSLT	V/http://comonservices.betas.uci.dev/NotificationService/NoBorrar/message.xslt	El sistema muestra un mensaje "Campo requerido".	1-Gestionar recurso XML. 2-Adicionar recurso XML. 3-Aceptar.
ESC 4.7 Adicionar recurso XML dejando campos vacíos.	El sistema muestra una página que permite introducir datos para adicionar un recurso XML.	V/Este xslt se usa para notificar con respecto a un circulado.	Vacío/	V/V/http://commonservices.betas.uci.dev/NotificationService/NoBorrar/message.xslt	El sistema muestra un mensaje "Campo requerido".	1-Gestionar recurso XML. 2-Adicionar recurso XML. 3-Aceptar.
ESC 4.8 Adicionar recurso XML dejando campos vacíos.	El sistema muestra una página que permite introducir datos para adicionar un recurso XML.	V/V/Este xslt se usa para notificar con respecto a un circulado.	V/XSLT	Vacío/	El sistema muestra un mensaje "Cargue un archivo".	1-Gestionar recurso XML. 2-Adicionar recurso XML. 3-Aceptar.
ESC 4.9 Opción Aceptar.	El sistema muestra la opción <b>Aceptar</b> para adicionar recurso XML.				El sistema adiciona el recurso al listado ya existente.	1-Gestionar recurso XML. 2-Aceptar.
ESC 4.10 Opción Cancelar.	El sistema muestra la opción <b>Cancelar</b> para cancelar la petición.				El sistema muestra la página principal.	1-Gestionar recurso XML. 2-Cancelar.

ESC 4.11 Opción Ver archivo.	El sistema muestra la opción <b>Ver archivo</b> para ver el archivo cargado.				El sistema muestra en la página la información del archivo que se cargó.	1-Gestionar recurso XML. 2-Ver Archivo.
------------------------------	--	--	--	--	--	--

### 3.7. Resultado de las pruebas

En la **Tabla 7** se muestran los resultados de las pruebas realizadas a las funcionalidades durante las 3 iteraciones de desarrollo y las 5 iteraciones de pruebas, obteniendo como resultado una solución de software que posee un correcto funcionamiento con relación a las funcionalidades definidas por el cliente como se observa en la **Figura 16**.

**Tabla 7: Pruebas realizadas a cada requisito funcional por cada iteración del desarrollo**

ITERACIONES	1	2	2.1	3	3.1
Requisitos No Implementados	5	3	3	0	0
Requisitos que No Pasaron	0	1	0	2	0
Requisitos que Pasaron	3	4	5	6	8

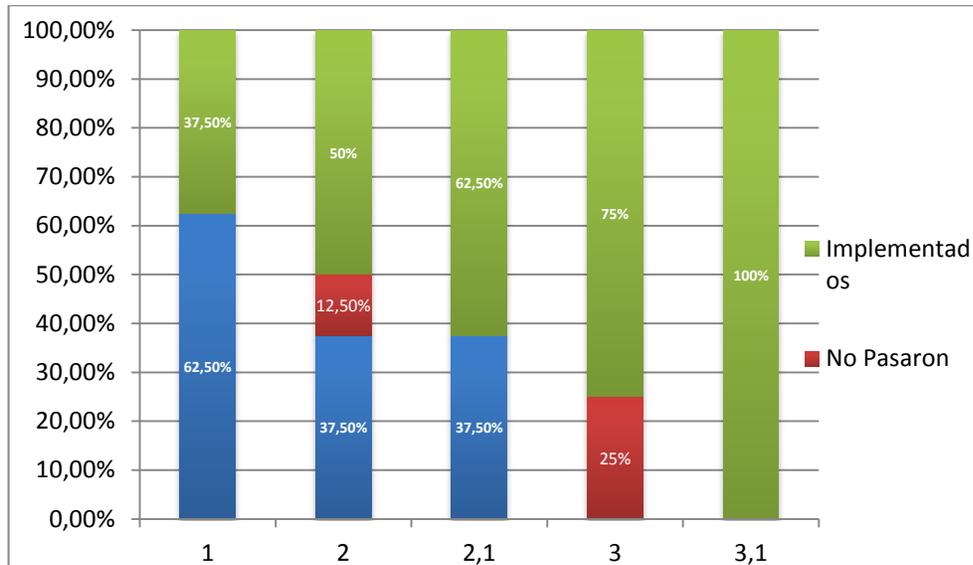


Figura 16: Representación gráfica de las pruebas realizadas a cada funcionalidad por iteración

### 3.8. Pruebas de carga del sistema

Para comprobar el rendimiento del servicio se efectuaron las pruebas de carga, publicándose en un servidor cuyas características se muestran en la **Tabla 8**. Se realizaron 200 hilos de peticiones constantes durante 5 minutos que representan el doble de las que debe atender el servicio. Se observó el buen rendimiento del servicio al ejecutar 5368 pruebas, a razón de 17,9 pruebas por segundo, a las que respondió cada 1.07 segundos como se muestra en la **Figura 17**.

Tabla 8: Características del Servidor

Hardware	Características
Microprocesador	Intel(R) Xeon(R) CPU 3.00 GHz (4 procesadores)
Memoria RAM	3 GB
Disco Duro	320 GB
Ancho de Banda	100 Mbps

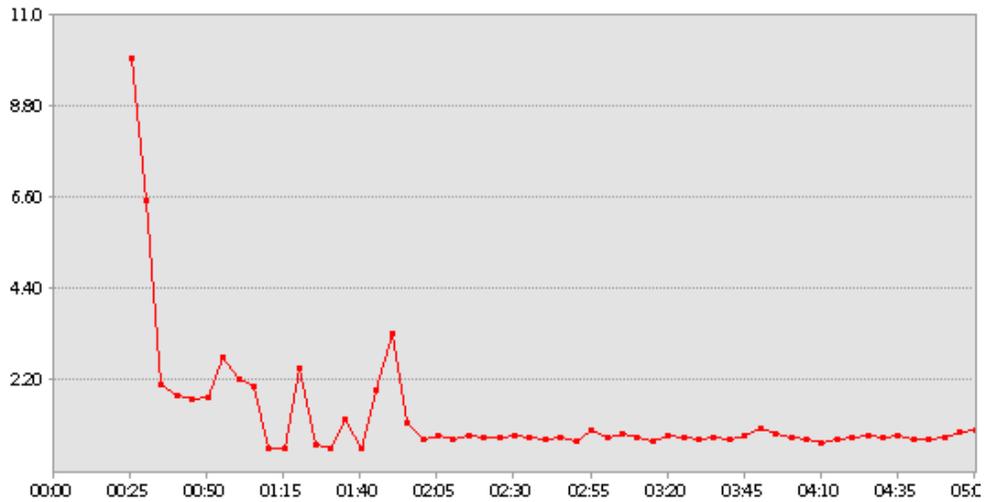


Figura 17: Tiempo de respuesta de la prueba

### Pruebas para comprobar agilidad

Para la realización de esta prueba se enviaron 50 notificaciones por cada vía de notificación para comprobar el tiempo promedio que demora una notificación en llegar a su destino como se muestra en la Figura 18, Figura 19, Figura 20, Figura 21.

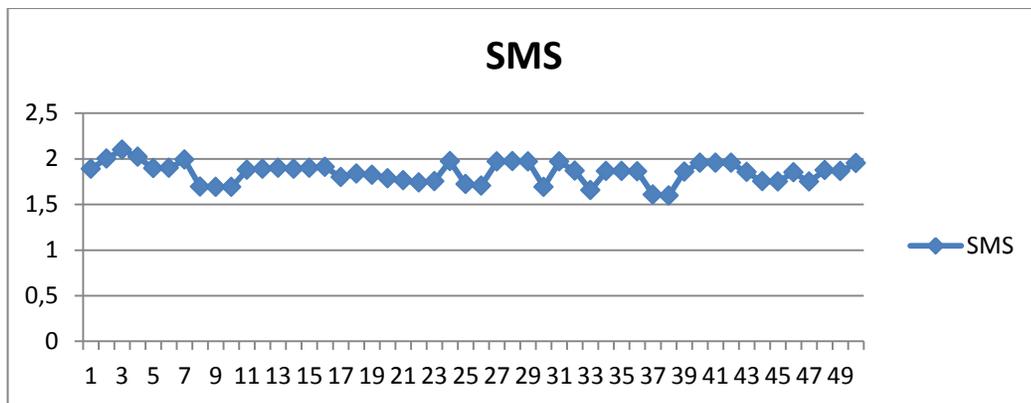


Figura 18: Tiempo de llegada de una notificación SMS

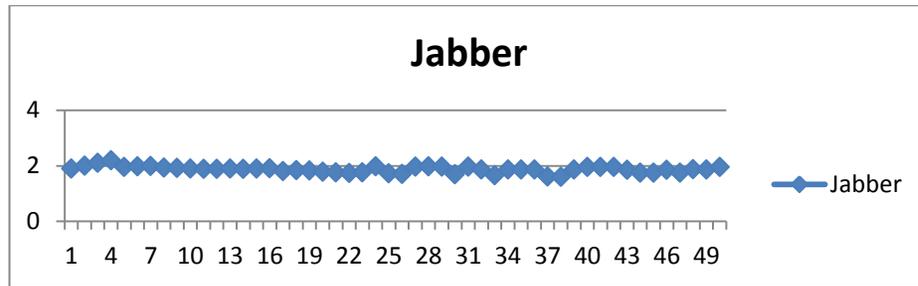


Figura 19: Tiempo de llegada de una notificación Jabber

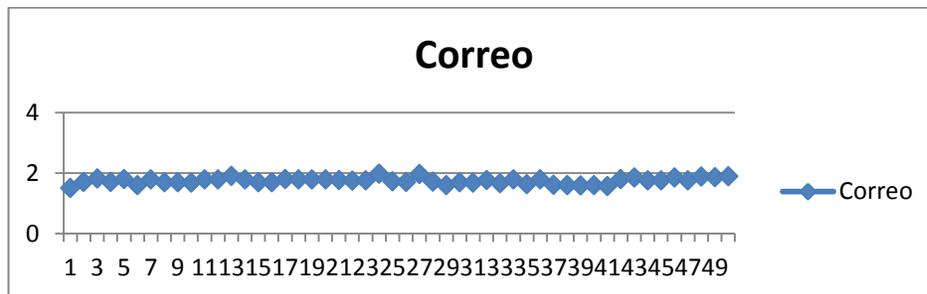


Figura 20: Tiempo de llegada de una notificación Correo electrónico



Figura 21: Tiempo promedio de llegada de una notificación (Jabber, Correo, SMS)

El retraso en la comunicación con los clientes y entidades interesadas en los procesos que se desarrollan en el SUIN se soluciona con el sistema de notificaciones implementado. Las notificaciones enviadas por el sistema al destinatario oscilan entre 1.07 y 1.09 segundos respectivamente, demostrando así que el sistema de notificaciones agiliza los tiempos de ejecución de los procesos como se muestra en la **Tabla 9**.

Tabla 9: Comparación entre el SUIN y el sistema de notificaciones para el SUIN

Características	SUIN	Sistema de notificaciones para el SUIN
Notificaciones	Manual	correo electrónico
		jabber
		sms
Tiempo promedio en notificar	Indefinido	1.07 seg
		1.09 seg
		1.08 seg

### Conclusiones parciales

El tratamiento de errores efectuado permitió garantizar una mayor integridad en los datos y la obtención de buenos resultados durante la utilización del sistema.

La obtención de la vista de despliegue estableció los componentes y protocolos de comunicación definidos para garantizar la distribución física de la solución.

La realización de pruebas al sistema permitió garantizar la calidad, demostrar el rendimiento y agilidad del sistema de notificaciones del Sistema Único de Identificación Nacional.

## **Conclusiones generales**

Con la realización del presente trabajo de diploma los autores concluyen que:

- ✓ El desarrollo del sistema de notificaciones permitió garantizar y agilizar la comunicación con los involucrados en los procesos que se desarrollan en el Sistema Único de Identificación Nacional.
- ✓ Las pruebas realizadas al sistema, permitieron comprobar su correcto funcionamiento y validar su agilidad en la comunicación con los clientes, realizándose en aproximadamente 1.08 segundos, lo cual representa un caso de éxito en el contexto enmarcado.
- ✓ La versión estable del producto obtenida, ha sido integrada y se encuentra en proceso de despliegue en las oficinas de Registro de la Población de todo el país, mejorando de esta forma los servicios que brinda el Sistema Único de Identificación Nacional.

## Recomendaciones

Con el objetivo de incorporar mejoras al sistema de notificaciones del SUIN, se recomienda:

- ✓ Crear una actividad de *workflow* que permita cargar automáticamente los datos desde el XSD correspondiente al tipo de mensaje a enviar, para de esta manera facilitar el trabajo de los desarrolladores al crear el XML con los datos.
- ✓ Crear un sistema de reportes para gestionar la información relacionada al sistema de notificaciones.

## Referencia bibliográfica

1. **Fernández, Marcelo Fidel.** *Mensajería Instantánea en Internet*. Argentina : s.n., 2001.
2. **Sivianes, Francisco, y otros, y otros.** *Servicios en red*. Madrid : Paraninfo, 2010. 9788497327657.
3. **Merlo, José Antonio.** *Información y Referencias en Entornos Digitales*. Murcia : Editum, 2009. 9788483719183.
4. **Real Academia Española.** Real Academia Española. *Real Academia Española*. [En línea] 2010. [Citado el: 14 de 11 de 2011.] [http://buscon.rae.es/drae/SrvltConsulta?TIPO\\_BUS=3&LEMA=notificaciones](http://buscon.rae.es/drae/SrvltConsulta?TIPO_BUS=3&LEMA=notificaciones).
5. **Charchabal, Barredo y Nereyda, Katia.** *Propuesta de diseño y arquitectura de la información de un sistema de notificaciones dentro de la gestión universitaria*. Ciudad de la Habana : s.n., 2010.
6. **Barceló, José M, Iñigo Griera, Jordi y LLorente Viejo, Silvia.** *Protocolos y Aplicaciones Internet*. Barcelona : Editorial UOC, 2008. 9788497887410.
7. **Ariencia, Jorge Luis.** El Observatorio de Internet Móvil. *El Observatorio de Internet Móvil*. [En línea] 25 de octubre de 2006. [Citado el: 14 de 11 de 2011.] <http://jlarienza.blogspot.com/2006/10/mensajeria-instantanea-el-fin-de-los.html>.
8. **w3c.** w3c. [En línea] 2007. [Citado el: 10 de febrero de 2011.] <http://www.w3.org/XML/Schema>.
9. **Mozilla Developer Network.** Mozilla Developer Network. *Mozilla Developer Network*. [En línea] 3 de febrero de 2007. [Citado el: 5 de febrero de 2011.] <https://developer.mozilla.org/es/XSLT>.
10. **Alonso, Randy Falcón.** Cubadebate. *Cubadebate*. [En línea] Consejo Editorial de Cubadebate, 9 de mayo de 2009. [Citado el: 2011 de diciembre de 15.] [www.cubadebate.cu](http://www.cubadebate.cu).
11. **Legón, Adonis Cesar y Landrián, Jorge García.** publicaciones.uci.cu. *publicaciones.uci.cu*. [En línea] 2008. [Citado el: 20 de noviembre de 2011.] <http://publicaciones.uci.cu/index.php/SC/article/view/46>.
12. **Pressman, Roger S.** *Ingeniería del software. Un enfoque práctico*. s.l. : McGraw-Hill, 2009.
13. **Rob, Peter y Coronel, Carlos.** *Sistema de bases de datos*. Mexico : s.n., 2006. 061906209.
14. **Centro de encuentro BPM.** *El libro del BPM*. Madrid : s.n., 2011. 9788461483679.
15. **Mahmoud, Qusay H.** Oracle. Getting Started with Java Message Service (JMS). *Oracle. Getting Started with Java Message Service (JMS)*. [En línea] noviembre de 2004. [Citado el: 2 de mayo de 2012.] <http://java.sun.com/developer/technicalArticles/Ecommerce/jms/>.
16. **Richards, Mark, Monson, Richard y Chappell, David A.** *Java Message Service*. s.l. : O'Reilly, 2009.

17. **Kunze, Marcel, y otros, y otros.** *Cloud Computing: Web-based Dynamic It Services*. London New York : Springer, 2011. 97836642209161.
18. **Amazon Web Services.** Amazon Web Services. *Amazon Web Services*. [En línea] 2012. [Citado el: 2 de mayo de 2012.] <http://aws.amazon.com/es/sqs/>.
19. **RabbitMQ.** RabbitMQ. *RabbitMQ*. [En línea] 2012. [Citado el: 2012 de abril de 22.] <http://www.rabbitmq.com/documentation.html>.
20. **AMQP.** AMQP.com. *AMQP.com*. [En línea] 2012. [Citado el: 10 de marzo de 2012.] <http://www.amqp.org/product/overview>.
21. **Sintes Marco, Bartolomé.** <http://www.mclibre.org>. [En línea] 3 de abril de 2012. [Citado el: 10 de mayo de 2012.] [http://www.mclibre.org/consultar/xml/lecciones/xml\\_xslt.html](http://www.mclibre.org/consultar/xml/lecciones/xml_xslt.html).
22. **Microsoft Corporation.** Microsoft. MSF for CMMI Process Improvement. *Microsoft. MSF for CMMI Process Improvement*. [En línea] 2012. [Citado el: 15 de noviembre de 2011.] <http://guides.brucejmack.biz/MSF%20for%20CMMI%20Process%20Improvement/Process%20Guidance/Supporting%20Files/ProcessGuidance.htm>.
23. **Woodward, Martin, Blankenship, Ed y Keller, Brian.** *Professional Team Foundation Server 2010*. 2010.
24. **Altova.** Altova. *Altova*. [En línea] 17 de enero de 2011. [Citado el: 15 de 11 de 2011.] <http://www.altova.com/umodel.html>.
25. **Altova Umodel.** *Altova® UModel® 2011 User & Reference Manual*. 2011. 978193321062.
26. **White, Stephen A y Miers, Derek.** *Guía de referencia y modelado BPMN*. Uruguay : s.n., 2009. 9780981987033.
27. **Larman, Craig.** *UML y Patrones.Introducción al análisis y diseño orientado a objetos*. Uruguay : Prentice Hall, 2001. 053681.
28. **Oracle Corporation.** Oracle Corporation. *Oracle Corporation*. [En línea] 2010. [Citado el: 15 de 11 de 2011.] <http://msdn.microsoft.com/en-us/library/ee958158.aspx>.
29. **Gabillaud, Jérôme .** *Oracle 11g: SQL, PL/SQL, SQL\*PLUS*. 2010. 978-2-7460-5360-1.
30. **Embarcadero Technologies.** Embarcadero Technologies, Inc. Embarcadero. E/R Studio Modelado de datos empresarial. Database Tools and Developer Software. *Embarcadero Technologies, Inc. Embarcadero. E/R Studio Modelado de datos empresarial. Database Tools and Developer Software*. [En línea] 2009. [Citado el: 14 de 12 de 2011.] <http://www.embarcadero.com>.

31. **Katrib, Miguel, y otros, y otros.** *Visual Studio 2008.Desafia todos los retos.* Madrid : Fondo editorial de Natalia, 2009. Visual Studio 2008.Desafia todos los retos.
32. **González, Martín y Juan Carlos.** Blog del Centro de Innovación en Integración de Cantabria. ADO.NET Entity. *Blog del Centro de Innovación en Integración de Cantabria. ADO.NET Entity.* [En línea] 25 de enero de 2008. [Citado el: 11 de 12 de 2011.] <http://geeks.ms/blogs/ciin/archive/2008/01/25/ado-net-entity-framework-linq-to-entities-entity-sql-y-entity-services-i.aspx>.
33. **Jennings, Roger.** *Professional ADO.NET 3.5 with Linq and the Entity Framework.* 2009. 9780470182611.
34. **Microsoft.** Microsoft .NET Framework 4.0 . Detalles de la Descarga: Microsoft .NET Framework 4.0 (instalador web). *Microsoft .NET Framework 4.0 . Detalles de la Descarga: Microsoft .NET Framework 4.0 (instalador web).* [En línea] 4 de abril de 2010. [Citado el: 13 de noviembre de 2011.] <http://www.microsoft.com/downloads/details.aspx?familyid=9CFB2D51-5FF4-4491-B0E5-B386F32C0992&displaylang=es..>
35. **Brice Arnaud, Guérin.** *ASP.NET con C#.* Barcelona : Pablo Narváez, 2008. 9782746037083.
36. **Brochard, Jhonny.** *XML:Conceptos e implementación.* Barcelona : s.n., 2010. 2746014025.
37. **Amelot, Michele.** *Recursos Informaticos VBA.* Barcelona : s.n., 2010. 9782746059573.
38. **MSDN.** MSDN.Introducción a Visual Studio. *MSDN. Introducción a Visual Studio.* [En línea] enero de 2011. [Citado el: 15 de noviembre de 2011.] <http://msdn.microsoft.com/es-es/library/6x6bk1f4.aspx>.
39. **Microsoft Corporation.** MSDN.com. *MSDN.com.* [En línea] 2012. [Citado el: 25 de noviembre de 2011.] <http://msdn.microsoft.com/library/ee210343.aspx>.
40. **Bukovics, Bruce.** *Pro WF: Windows Workflow in .NET 3.5.* New York : Apress, 2008. 9781430209751.
41. **Cibraro, Pablo, y otros, y otros.** *Professional WCF 4.* Indianapolis : Wiley Publishing, 2010. 9780470563144.
42. **Sharp, Jonh.** *Windows Communication Foundation 4.* California : Octal Publishing, 2010. 978-0-735-64556-1.
43. **Garrett, Jesse James.** Introduccion a AJAX. *Introduccion a AJAX.* [En línea] 18 de febrero de 2005. [Citado el: 15 de enero de 2012.] <http://www.maestrosdelweb.com/editorial/ajax>.
44. **Microsoft.** msdn.com. [En línea] 5 de agosto de 2010. <http://www.microsoft.com/downloads/es-es/details.aspx?FamilyID=3660cacf-f077-44d3-a9d9-97e801da2035>.

45. **Sommerville , Ian.** *Ingenieria Del Software.* 2006. 84-7829-074-5.
46. **McC Smith, Jason.** *Elemental Design Patterns.* 2012. 0-321-71192-0.
47. **Blanco Zambrano, Reinier.** *Guía del usuario Bison.* La Habana,Cuba : s.n., 2010.
48. **Proyecto identidad Cuba.** *Estándares de codificación.* Ciudad de la Habana : s.n., 2009.
49. **Jacobson, I. y Rumbaugh, G. Booch y J.** *El Proceso Unificado de Desarrollo de software.* 2000.

## Anexos

### Anexo I Catálogo de requisitos

#### Servicio:

#### **RF1 Enviar Mensaje**

RF1.1. Permitir mostrar los campos para el envío de mensaje.

- a) Asíncrono
- b) Asunto
- c) Tipo de mensaje
- d) Enviar
- e) Para
- f) Xml

RF1.2. Permitir enviar el mensaje.

#### **RF2 Obtener mensaje por usuario**

RF2.1. Permitir mostrar campos para obtener mensaje por usuario.

- a) Para

RF2.2. Permitir comprobar en la base de datos si hay mensaje para ese usuario.

#### **RF3 Cambiar estado del mensaje**

RF3.1. Permitir mostrar campo para el cambio de estado del mensaje.

- a) Identificador

RF3.2. Permitir comprobar que el mensaje existente en la base de datos.

RF3.3. Permitir cambiar el estado del mensaje.

#### Aplicación:

#### **RF4 Gestionar recurso XML**

RF4.1. Permitir mostrar un listado con los recursos existentes.

- a) Descripción
- b) Tipo
- c) URL

RF4.2. Permitir seleccionar un recurso XML.

RF4.3. Permitir mostrar campos para adicionar un recurso XML.

- a) Descripción
- b) Tipo

- c) URL

RF4.4. Permitir mostrar campos para modificar un recurso XML.

- a) Descripción
- b) URL

RF4.5. Permitir eliminar un recurso XML.

## **RF5 Gestionar tipo de mensaje de notificación**

RF5.1. Permitir mostrar el listado de los mensajes de notificación.

- a) Buzón
- b) Descripción
- c) Recursos

RF5.2. Permitir seleccionar un mensaje de notificación.

RF5.3. Permitir mostrar campos para adicionar un mensaje de notificación.

- a) Buzón
- b) Mime
- c) Descripción

RF5.4. Permitir mostrar campos para modificar un mensaje de notificación.

- a) Buzón
- b) Mime
- c) Descripción

RF5.5. Permitir eliminar un mensaje de notificación.

## **RF6 Monitorear mensajes**

RF6.1. Permitir mostrar el listado de los mensajes enviados.

- a) Remitente
- b) Fecha de envío
- c) Estado

RF6.2. Permitir seleccionar un mensaje del listado existente.

## **RF7 Modificar Conexión**

RF7.1. Permitir mostrar los campos para la configuración de la conexión.

- a) Correo electrónico
- b) Jabber

RF7.2. Permitir mostrar datos para la conexión de correo.

- a) Servidor de correo
- b) Usuario

RF7.3. Permitir mostrar datos para la conexión de Jabber.

- a) Servidor de jabber
- b) Usuario

## RF8 Realizar suscripción a notificaciones

RF8.1. RF8.2. Permitir mostrar datos para la búsqueda de la persona a notificar.

- c) Numero CI
- d) Primer nombre
- e) Segundo nombre
- f) Primer apellido
- g) Segundo apellido

RF8.3. Permitir seleccionar la persona a notificar.

RF8.4. Permitir mostrar datos para adicionar suscripción.

- a) Contacto

RF8.5. Permitir eliminar suscripciones.

## Anexo II Descripción de requisitos funcionales

### **RF1 Enviar mensaje**

**Tabla 10: Requisito funcional Enviar mensaje**

<b>Propósito</b>	Enviar Mensajes	
<b>Roles</b>	Funcionario Correspondiente	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado.	
<b>Conceptos tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NTIPOMENSAJENOTIFICACION	idtipomensaje descripcion mime buzon xsduri xslturi

<b>Descripción</b>	1.1 Mostrar campos para el envío de mensajes: <ul style="list-style-type: none"> <li>a Asíncrono</li> <li>b Asunto</li> <li>c Tipo de mensaje</li> <li>d Enviar</li> <li>e Para</li> <li>f Xml</li> </ul> 1.2 Enviar mensaje.
<b>Validaciones</b>	1. No puede haber ningún campo vacío.
<b>Postcondiciones</b>	1. Se ha enviado el mensaje requerido.
<b>Prototipo</b>	No procede

**RF2 Obtener mensaje por usuario**

**Tabla 11: Requisito funcional Obtener mensaje por usuario**

<b>Propósito</b>	Obtener mensaje por usuario	
<b>Roles</b>	Funcionario Correspondiente	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	DMENSAJENOTIFICACION	idmensaje destino asunto fechaprogramada fechaoperacion fuentedatos idestadomensaje idtipomensaje sender

<b>Descripción</b>	2.1 Mostrar campo para obtener mensaje por usuario: a Para 2.2 Comprobar si existen mensajes para ese usuario. 2.2.1 Mostrar el listado de mensajes, si existe el usuario. 2.2.2 Mostrar un mensaje de error, si el usuario no existe.
<b>Validaciones</b>	1. No puede haber ningún campo vacío.
<b>Postcondiciones</b>	2. Se obtienen todos los mensajes que tenga el usuario seleccionado.
<b>Prototipo</b>	No procede

### **RF3 Cambiar estado del mensaje**

**Tabla 12: Requisito funcional Cambiar estado del mensaje**

<b>Propósito</b>	Cambiar estado del mensaje	
<b>Roles</b>	Funcionario Correspondiente	
<b>Precondiciones</b>	1. El funcionario correspondiente debe estar autenticado en el sistema.	
<b>Concepto tratados</b>	<b>Concepto</b>	<b>Atributos</b>
	NESTADOMENSAJENOTIFICAICON	Idestadomensaje descripcion
<b>Descripción</b>	3.1 Mostrar campo para el cambio de estado del mensaje: a Identificador 3.2 Comprobar que el mensaje exista en BD. 3.3 Cambiar estado, si el mensaje existe.	
<b>Validaciones</b>	1. No puede haber ningún campo vacío. 2. Debe existir en identificador del mensaje en BD.	
<b>Postcondiciones</b>	1. Se ha cambiado el estado del mensaje requerido.	
<b>Prototipo</b>	No procede	

**RF4 Gestionar recurso XML**

Tabla 13: Requisito funcional Gestionar recurso XML

<b>Propósito</b>	Gestionar Recurso XML	
<b>Roles</b>	Administrador del sistema	
<b>Precondiciones</b>	1. El administrador del sistema debe estar autenticado.	
<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	NTIPORECURSOXML	Idtiporecursoxml descripcion
	DRECURSOXML	url contenido

<b>Descripción</b>	<p>4.1 Mostrar listado de recursos XML.</p> <ul style="list-style-type: none"> <li>a Descripción</li> <li>b Tipo</li> <li>c URL</li> </ul> <p>4.2 Seleccionar la opción Adicionar recurso XML.</p> <ul style="list-style-type: none"> <li>a Descripción</li> <li>b Tipo</li> <li>c URL</li> </ul> <p>4.2.1 Se adiciona el recurso XML al listado existente, si se selecciona la opción Aceptar.</p> <p>4.2.2 Se muestra la página principal del requisito gestionar recurso XML, si se selecciona la opción Cancelar.</p> <p>4.2.3 Se muestra el archivo que se seleccionó para subir, si se selecciona la opción Ver Archivo.</p> <p>4.2.4 Adicionar el recurso XML al listado ya existente.</p> <p>4.3 Seleccionar la opción de Modificar recurso XML:</p> <ul style="list-style-type: none"> <li>a Descripción</li> <li>b URL</li> </ul> <p>4.3.1 Se modifica el recurso XML del listado existente, si se selecciona la opción Aceptar.</p> <p>4.3.2 Se muestra la página del requisito gestionar recurso XML, si se selecciona la opción Cancelar.</p> <p>4.3.3 Se muestra el archivo, si se selecciona la opción Ver Archivo.</p> <p>4.3.4 Modificar el recurso XML seleccionado.</p> <p>4.4 Seleccionar la opción de Eliminar recurso XML.</p> <ul style="list-style-type: none"> <li>4.4.1 Se elimina el recurso XML del listado existente, si se selecciona la opción Aceptar.</li> <li>4.4.2 Se muestra la página principal del requisito gestionar recurso XML, si se selecciona la opción Cancelar.</li> <li>4.4.3 Eliminar el recurso XML del listado.</li> </ul>
<b>Validaciones</b>	<ol style="list-style-type: none"> <li>1. No puede haber ningún campo vacío.</li> <li>2. Hay que validar que se seleccione algún recurso XML.</li> </ol>

<b>Postcondiciones</b>	1. Se han adicionado los recursos XML.
<b>Prototipo</b>	

Fig.2. Gestionar Recurso XML

**RF5 Gestionar tipo de mensaje**

**Tabla 14: Requisito funcional Gestionar tipo de mensaje**

<b>Propósito</b>	Gestionar tipo de mensaje	
<b>Roles</b>	Administrador del sistema	
<b>Precondiciones</b>	1. El administrador del sistema debe estar autenticado.	
<b>Conceptos tratados</b>	Conceptos	Atributos
	NTIPIOMENSAJENOTIFICACION	idtipomensaje descripcion mime buzón xsdurl xslturl

<b>Descripción</b>	<p>5.1 Mostrar listado de mensajes.</p> <ul style="list-style-type: none"> <li>a Buzón</li> <li>b Descripción</li> <li>c recurso</li> </ul> <p>5.2 Seleccionar la opción Adicionar mensajes:</p> <ul style="list-style-type: none"> <li>a Buzón</li> <li>b Mime</li> <li>c Descripción</li> </ul> <p>5.2.1 Se adiciona el nuevo mensaje al listado ya existente, si se selecciona la opción Aceptar.</p> <p>5.2.2 Se muestra la página principal del requisito gestionar tipo de mensaje, si se selecciona la opción Cancelar.</p> <p>5.2.3 Adicionar el tipo de mensaje de notificación al listado ya existente.</p> <p>5.3 Seleccionar la opción Modificar mensaje:</p> <ul style="list-style-type: none"> <li>a Buzón</li> <li>b Mime</li> <li>c Descripción</li> </ul> <p>5.3.1 Se modifica el mensaje del listado ya existente, si se selecciona la opción Aceptar.</p> <p>5.3.2 Se muestra la página principal del requisito gestionar tipo de mensaje, si se selecciona la opción Cancelar.</p> <p>5.3.3 Modificar el tipo de mensaje de notificación.</p> <p>5.4 Seleccionar la opción Eliminar mensaje:</p> <p>5.4.1 Se elimina el mensaje del listado ya existente, si se selecciona la opción Aceptar.</p> <p>5.4.2 Se muestra la página principal del requisito gestionar tipo de mensaje, si se selecciona la opción Cancelar.</p> <p>5.4.3 Eliminar el tipo de mensaje de notificación.</p>
<b>Validaciones</b>	<ol style="list-style-type: none"> <li>1. No puede haber ningún campo vacío.</li> <li>2. Validar que se seleccione algún mensaje de notificación.</li> </ol>
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Se han adicionado los tipos de mensajes de notificación.</li> </ol>

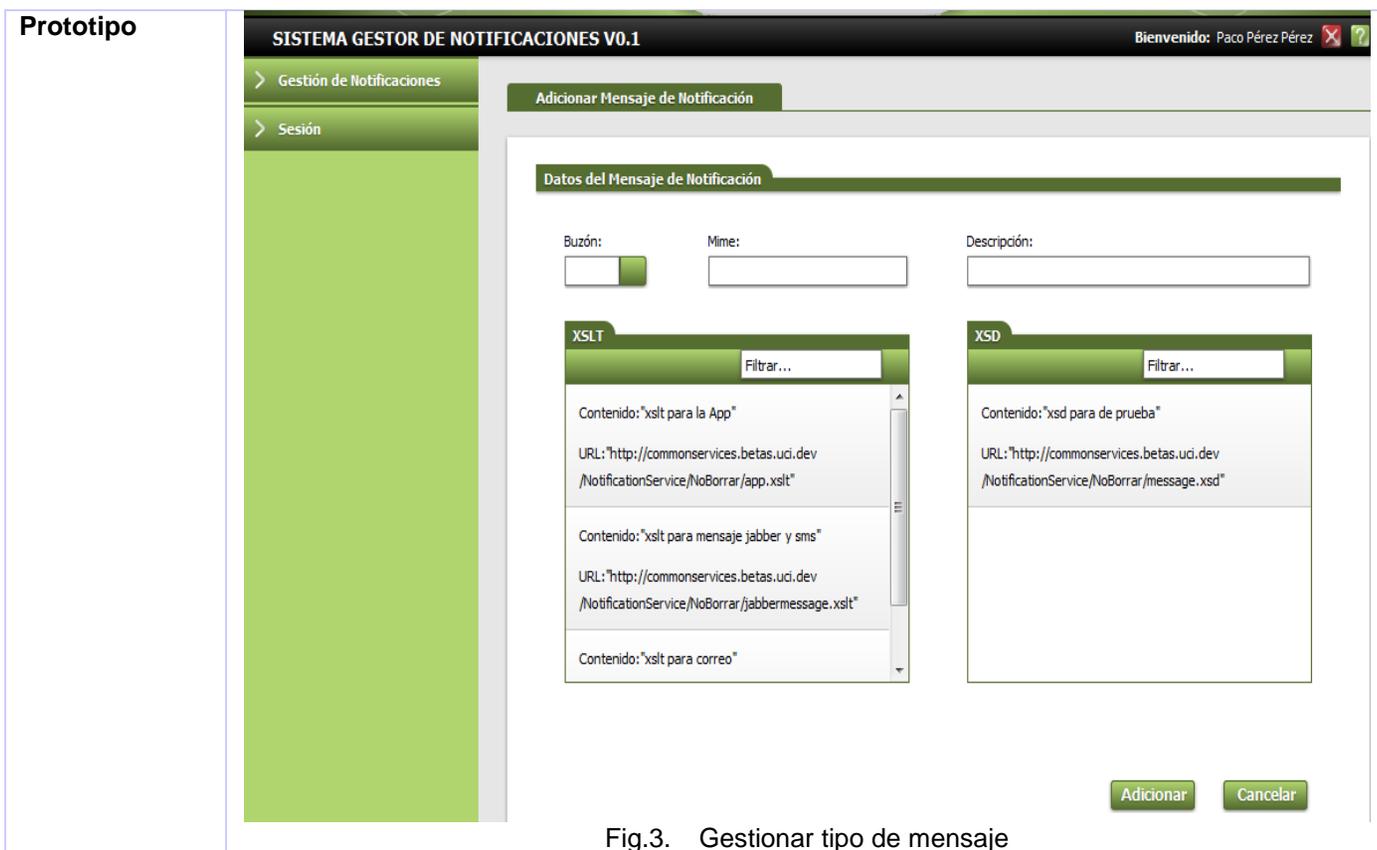


Fig.3. Gestionar tipo de mensaje

**RF6 Monitorear mensajes**

**Tabla 15: Requisito funcional Monitorear mensajes**

<b>Propósito</b>	Monitorear mensajes	
<b>Roles</b>	Administrador del sistema	
<b>Precondiciones</b>	2. El administrador del sistema debe estar autenticado.	
<b>Conceptos tratados</b>	Conceptos	Atributos

	DMENSAJENOTIFICACION	idmensaje destino asunto fechaprogramada fechaoperacion fuentedatos idestadomensaje idtipomensaje sender
<b>Descripción</b>	6.1 Mostrar el listado de los mensajes. <ul style="list-style-type: none"> <li>a Remitente</li> <li>b Fecha de envío</li> <li>c estado</li> </ul> 6.2 Seleccionar un mensaje. <ul style="list-style-type: none"> <li>6.2.1 Se elimina el mensaje del listado ya existente, si se selecciona la opción Aceptar.</li> <li>6.2.2 Se muestra la página principal, si se selecciona la opción Cancelar.</li> </ul> 6.3 Eliminar el mensaje del listado.	
<b>Validaciones</b>	1. Validar que se seleccione algún mensaje.	
<b>Postcondiciones</b>	1. Se han eliminado los mensajes enviados.	

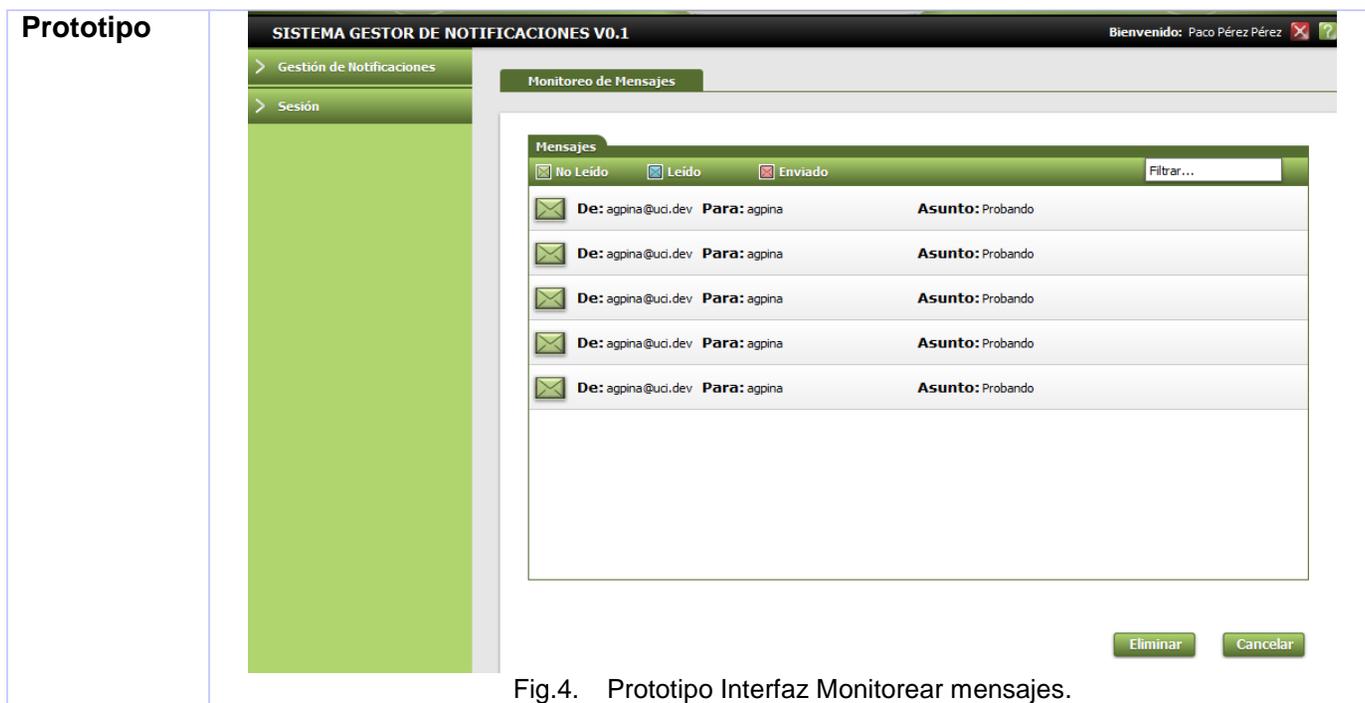


Fig.4. Prototipo Interfaz Monitorear mensajes.

**RF8 Realizar suscripción a notificaciones**

**Tabla 16: Requisito funciona Realizar suscripción a notificaciones**

<b>Propósito</b>	Suscripción a notificaciones	
<b>Roles</b>	Administrador del sistema	
<b>Precondiciones</b>	1. El administrador del sistema debe estar autenticado.	
<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	DNOTIFICACIONPERSONAINST	idtipomensaje idpersona contacto

<b>Descripción</b>	<p>8.1 Mostrar los datos para buscar la persona a notificar.</p> <ul style="list-style-type: none"> <li>a Numero CI</li> <li>b Primer nombre</li> <li>c Segundo nombre</li> <li>d Primer apellido</li> <li>e Segundo apellido</li> </ul> <p>8.1.1 Seleccionar la persona y seleccionar la opción Adicionar Suscripción.</p> <p>8.2 Mostrar datos para Adicionar Suscripción.</p> <ul style="list-style-type: none"> <li>a Contacto</li> </ul> <p>8.2.1 Seleccionar el tipo de notificación por la cual recibirá las notificaciones.</p> <p>8.2.2 Se muestra la página principal del requisito suscripción a notificaciones, si se selecciona la opción Aceptar.</p> <p>8.2.3 Se muestra la página principal, si se selecciona la opción Cancelar.</p> <p>8.3 Permitir seleccionar la persona para eliminar las notificaciones a las que está suscrito.</p> <p>8.3.1 Se elimina la suscripción, si se selecciona la opción Aceptar.</p> <p>8.3.2 Se muestra la página principal del requisito suscripción a notificaciones, si se selecciona la opción Cancelar.</p>
<b>Validaciones</b>	<ol style="list-style-type: none"> <li>1. No pueden haber campos vacíos.</li> <li>2. Validar que se seleccione alguna persona.</li> </ol>
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Se ha modificado la conexión.</li> </ol>

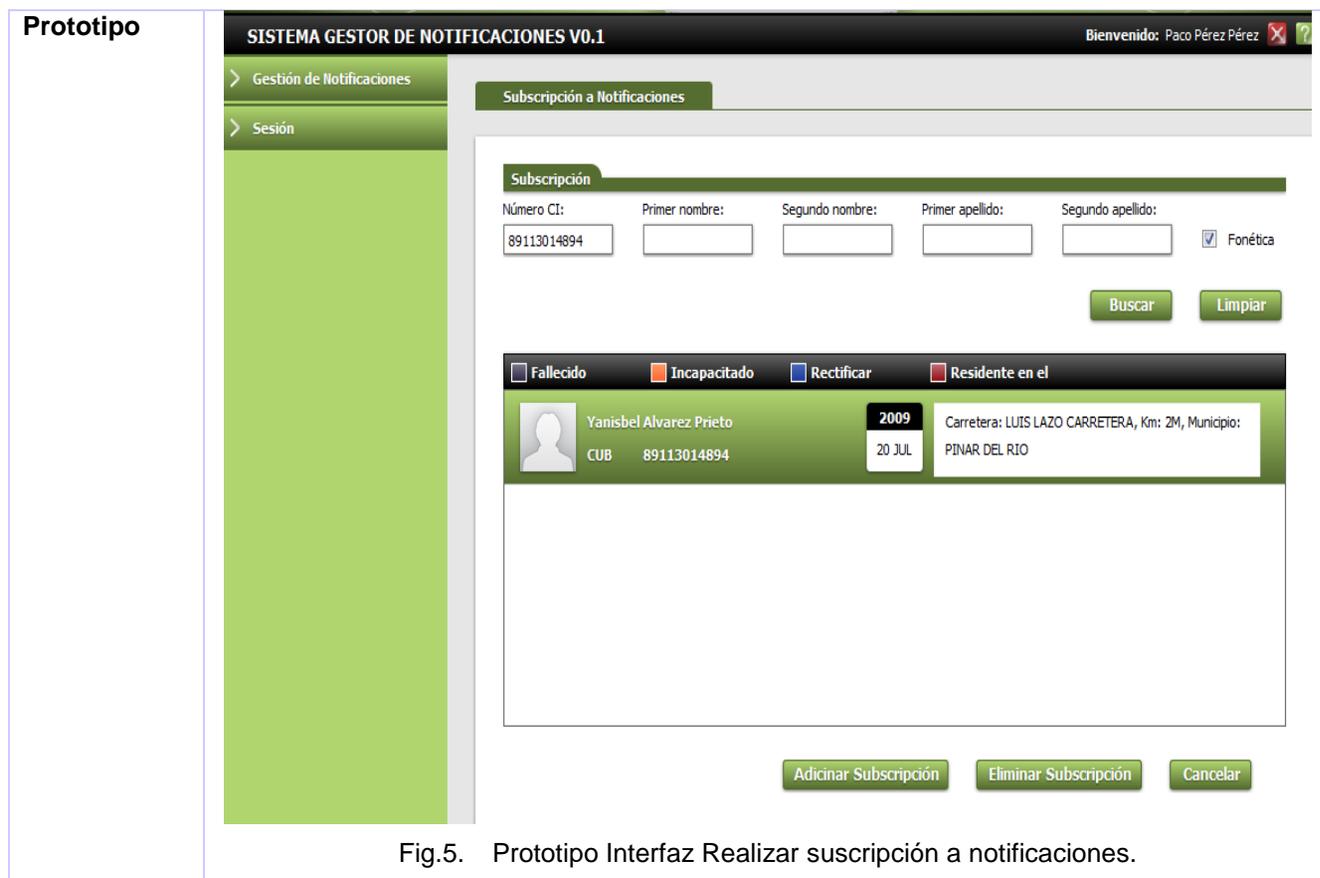


Fig.5. Prototipo Interfaz Realizar suscripción a notificaciones.

### Anexo III Descripción de entidades del servicio y la aplicación

Tabla 17: Descripción de la entidad NESTADOMENSAJENOTIFICACION

<b>Nombre de la Entidad:</b> NESTADOMENSAJENOTIFICACION			
<b>Descripción de la Entidad:</b>			
<b>Servicio:</b> NotificationService			
Atributo	Tipo de Dato	Nulo	Descripción
IDESTADOMENSAJE	Numeric(4,0)	No	Identificador del estado del mensaje
DESCRIPCION	Varchar(100)	No	Descripción del mensaje

Tabla 18: Descripción de la entidad DADJUNTONOTIFICACION

Nombre de la Entidad: DADJUNTONOTIFICACION			
Descripción de la Entidad:			
Servicio: <i>NotificationService</i>			
Atributo	Tipo de Dato	Nulo	Descripción
IDADJUNTO	<i>Numeric(10,0)</i>	No	Identificador del adjunto del mensaje
IDMENSAJE	<i>Binary(16)</i>	No	Identificador del mensaje
IDTIPOADJUNTO	<i>Numeric(4,0)</i>	Si	Identificador del tipo de adjunto
NOMBRE	<i>Varchar(200)</i>	No	Nombre del destino

Tabla 19: Descripción de la entidad DRECURSOXML

Nombre de la Entidad: DRECURSOXML			
Descripción de la Entidad:			
Servicio: <i>NotificationService</i>			
Atributo	Tipo de Dato	Nulo	Descripción
URI	<i>Varchar(500)</i>	No	Dirección del recurso
CONTENIDO	<i>Varchar(100)</i>	No	Contenido del recurso
IDTIPORECURSOXML	<i>Numeric(4,0)</i>	No	Identificador del tipo de recurso XML

Tabla 20: Descripción de la entidad NTIPORECURSOXML

<b>Nombre de la Entidad:</b> NTIPORECURSOXML			
<b>Descripción de la Entidad:</b>			
<b>Servicio:</b> <i>NotificationService</i>			
Atributo	Tipo de Dato	Nulo	Descripción
IDTIPORECURSOXML	<i>Numeric(4,0)</i>	No	Identificador del tipo recurso XML
DESCRIPCION	<i>Varchar(100)</i>	No	Descripción del recurso

Tabla 21: Descripción de la entidad DNOTIFICACIONPERSONAINST

<b>Nombre de la Entidad:</b> DNOTIFICACIONPERSONAINST			
<b>Descripción de la Entidad:</b>			
<b>Servicio:</b> <i>NotificationService</i>			
Atributo	Tipo de Dato	Nulo	Descripción
IDTIPOMENSAJE	<i>Numeric(4,0)</i>	No	Identificador del tipo de mensaje
IDPERSONA	<i>Binary(16)</i>	No	Identificador de la persona
CONTACTO	<i>Varchar(100)</i>	No	Nombre del contacto

Tabla 22: Descripción de la entidad DDATOMENSAJENOTIFICACION

Nombre de la Entidad: DDATOMENSAJENOTIFICACION			
Descripción de la Entidad:			
Servicio: <i>NotificationService</i>			
Atributo	Tipo de Dato	Nulo	Descripción
DATOS	<i>Varchar(200)</i>	No	Datos del mensaje
IDTIPOMENSAJE	<i>Numeric(4,0)</i>	No	Identificador del mensaje
FECHA	<i>Date</i>	No	Fecha del mensaje

Tabla 23: Descripción de la entidad NTIPOADJUNTONOTIFICACION

Nombre de la Entidad: NTIPOADJUNTONOTIFICACION			
Descripción de la Entidad:			
Servicio: <i>NotificationService</i>			
Atributo	Tipo de Dato	Nulo	Descripción
IDTIPOADJUNTO	<i>Numeric(4,0)</i>	No	Identificador del tipo de adjunto
DESCRIPTOR	<i>Varchar(30)</i>	No	Descriptor del adjunto de notificación
DESCRIPCION	<i>Varchar(100)</i>	No	Descripción del adjunto de notificación
XSDURI	<i>Varchar(500)</i>	Si	Dirección del xsd

## Anexo IV Diseño de casos de pruebas

**Modificar recurso XML**

Tabla 24: Diseño de caso de prueba del requisito Modificar recurso Xml

Escenario	Descripción	Descripción	URI	Respuesta del sistema	Flujo central
ESC 5.1 Modificar recurso xml correctamente.	El sistema muestra una página con los campos requeridos para modificar un recurso xml.	V/Este xslt se usa para notificar sobre una reunión urgente.	V/http://commons ervices.betas.uci. dev/NotificationService/NoBorrar/message.xslt	El sistema muestra un mensaje "El recurso se modificó correctamente".	1-Gestionar recurso XML. 2-Modificar recurso XML. 3-Aceptar.
ESC 5.2 Modificar recurso xml correctamente.	El sistema muestra una página con los campos requeridos para modificar un recurso xml.	V/Este xsd se usa validar el xml que conforma el xslt.	V/http://commons ervices.betas.uci. dev/NotificationService/NoBorrar/message.xsd	El sistema muestra un mensaje "El recurso se modificó correctamente".	1-Gestionar recurso XML. 2-Modificar recurso XML. 3-Aceptar.
ESC 5.3 Modificar recurso xml incorrectamente.	El sistema muestra una página que permite introducir datos para modificar un recurso xml.	l/154684!"\$."..% \$	V/http://commons ervices.betas.uci. dev/NotificationService/NoBorrar/message.xsd	El sistema muestra un mensaje "Existen datos incorrectos".	1-Gestionar recurso XML. 2-Modificar recurso XML. 3-Aceptar.
ESC 5.4 Modificar recurso xml incorrectamente.	El sistema muestra una página que permite introducir datos para modificar un recurso xml.	V/Este xsd se usa validar el xml que conforma el xslt.	V/	El sistema muestra un mensaje "Campo requerido".	1-Gestionar recurso XML. 2-Modificar recurso XML. 3-Aceptar.
ESC 5.5 Modificar recurso xml dejando campos vacíos.	El sistema muestra una página que permite introducir datos para modificar un recurso xml.	Vacío/	V/http://commons ervices.betas.uci. dev/NotificationService/NoBorrar/message.xsd	El sistema muestra un mensaje "Campo requerido".	1-Gestionar recurso XML. 2-Modificar recurso XML. 3-Aceptar.

ESC 5.6 Modificar recurso xml dejando campos vacíos.	El sistema muestra una página que permite introducir datos para modificar un recurso xml.	V/Este xsd se usa validar el xml que conforma el xslt.	Vacío/	El sistema muestra un mensaje "Campo requerido".	1-Gestionar recurso XML. 2-Modificar recurso XML. 3-Aceptar.
ESC 5.7 Opción Aceptar.	El sistema muestra la opción <b>Aceptar</b> para modificar el recurso xml.			El sistema modifica el recurso y lo adiciona al listado.	1-Gestionar recurso XML. 2-Aceptar.
ESC 5.8 Opción Cancelar.	El sistema muestra la opción <b>Cancelar</b> para cancelar la petición.			El sistema muestra la página principal.	1-Gestionar recurso XML. 2-Cancelar.
ESC 5.9 Opción Ver archivo	El sistema muestra la opción <b>Ver archivo</b> para ver el archivo cargado.			El sistema muestra en la página la información del archivo que se cargó.	1-Gestionar recurso XML. 2-Ver Archivo.

### **Eliminar recurso XML**

Tabla 25: Diseño de caso de prueba del requisito Eliminar recurso Xml

Escenario	Descripción	Respuesta del sistema	Flujo central
ESC 6.1 Eliminar recurso XML correctamente.	El sistema muestra una página con el listado de recursos XML.	El sistema muestra un mensaje "Esta seguro que desea eliminar".	1-Gestionar recurso XML. 2-Eliminar recurso XML 3-Aceptar.
ESC 6.2 Eliminar recurso XML incorrectamente.	El sistema muestra una página con el listado de recursos XML.	El sistema muestra un mensaje "Debe seleccionar un recurso a eliminar".	1-Gestionar recurso XML. 2-Eliminar recurso XML 3-Aceptar.
ESC 6.3 Opción Aceptar.	El sistema muestra la opción <b>Aceptar</b> para eliminar el recurso XML.	El muestra un mensaje "Se eliminó el recurso XML correctamente".	1-Gestionar recurso XML. 2-Aceptar.
ESC 6.4 Opción Cancelar.	El sistema muestra la opción <b>Cancelar</b> para cancelar la petición.	El sistema muestra la página principal.	1-Gestionar recurso XML. 2-Cancelar.

### **Adicionar mensaje notificación**

Tabla 26: Diseño de caso de prueba del requisito Adicionar mensaje de notificación

Escenario	Descripción	Buzón	Mime	Descripción	Respuesta del sistema	Flujo central
ESC 7.1 Adicionar mensaje de notificación correctamente.	El sistema muestra una página con los campos requeridos para adicionar un mensaje de notificación.	V/Correo	V/	V/Con este mensaje de notificación se le informa que puede pasar a recoger su carnet de identidad.	El sistema muestra un mensaje "Debe seleccionar un xslt y un xsd".	1-Gestionar mensaje de notificación. 2-Adicionar mensaje de notificación. 3-Aceptar.
ESC 7.2 Adicionar mensaje de notificación correctamente.	El sistema muestra una página con los campos requeridos para adicionar un mensaje de notificación.	V/Jabber	V/	V/Con este mensaje de notificación se le informa que puede pasar a recoger su carnet de identidad.	El sistema muestra un mensaje "Debe seleccionar un xslt y un xsd".	1-Gestionar mensaje de notificación. 2-Adicionar mensaje de notificación. 3-Aceptar.
ESC 7.3 Adicionar mensaje de notificación correctamente.	El sistema muestra una página con los campos requeridos para adicionar un mensaje de notificación.	V/SMS	V/	V/Con este mensaje de notificación se le informa que puede pasar a recoger su carnet de identidad.	El sistema muestra un mensaje "Debe seleccionar un xslt y un xsd".	1-Gestionar mensaje de notificación. 2-Adicionar mensaje de notificación. 3-Aceptar.
ESC 7.4 Adicionar mensaje de notificación correctamente.	El sistema muestra una página con los campos requeridos para adicionar un mensaje de notificación.	V/App	V/	V/Con este mensaje de notificación se le informa que puede pasar a recoger su carnet de identidad.	El sistema muestra un mensaje "Debe seleccionar un xslt y un xsd".	1-Gestionar mensaje de notificación. 2-Adicionar mensaje de notificación. 3-Aceptar.
ESC 7.5 Adicionar mensaje de notificación incorrectamente.	El sistema muestra una página con los campos requeridos para adicionar un mensaje de notificación.	I/	V/	V/Con este mensaje de notificación se le informa que puede pasar a recoger su carnet de identidad.	El sistema muestra un mensaje "Campo requerido".	1-Gestionar mensaje de notificación. 2-Adicionar mensaje de notificación. 3-Aceptar.
ESC 7.6 Adicionar mensaje de notificación incorrectamente.	El sistema muestra una página con los campos requeridos para adicionar un mensaje de notificación.	V/Jabber	I/	V/Con este mensaje de notificación se le informa que puede pasar a recoger su carnet de identidad.	El sistema muestra un mensaje "Existen datos incorrectos".	1-Gestionar mensaje de notificación. 2-Adicionar mensaje de notificación. 3-Aceptar.

ESC 7.7 Adicionar mensaje de notificación incorrectamente.	El sistema muestra una página con los campos requeridos para adicionar un mensaje de notificación.	V/Correo	V/	I/1654".\$%.\$%	El sistema muestra un mensaje "Existen datos incorrectos".	1-Gestionar mensaje de notificación. 2-Adicionar mensaje de notificación. 3-Aceptar.
ESC 7.8 Adicionar mensaje de notificación dejando campos vacíos.	El sistema muestra una página con los campos requeridos para adicionar un mensaje de notificación.	Vacío/	V/	V/Con este mensaje de notificación se le informa que puede pasar a recoger su carnet de identidad.	El sistema muestra un mensaje "Campo requerido".	1-Gestionar mensaje de notificación. 2-Adicionar mensaje de notificación. 3-Aceptar.
ESC 7.9 Adicionar mensaje de notificación dejando campos vacíos.	El sistema muestra una página con los campos requeridos para adicionar un mensaje de notificación.	V/App	Vacío/	V/Con este mensaje de notificación se le informa que puede pasar a recoger su carnet de identidad.	El sistema muestra un mensaje "Campo requerido".	1-Gestionar mensaje de notificación. 2-Adicionar mensaje de notificación. 3-Aceptar.
ESC 7.10 Adicionar mensaje de notificación dejando campos vacíos.	El sistema muestra una página con los campos requeridos para adicionar un mensaje de notificación.	V/SMS	V/	Vacío/	El sistema muestra un mensaje "Campo requerido".	1-Gestionar mensaje de notificación. 2-Adicionar mensaje de notificación. 3-Aceptar.
ESC 7.11 Opción Aceptar.	El sistema muestra la opción <b>Aceptar</b> para adicionar mensaje de notificación.				El sistema adiciona el mensaje al listado ya existente.	1-Gestionar mensaje de notificación. 3-Aceptar.
ESC 7.12 Opción Cancelar.	El sistema muestra la opción <b>Cancelar</b> para cancelar la petición.				El sistema muestra la página principal.	1-Gestionar mensaje de notificación. 2-Cancelar.

**Modificar mensaje notificación**

Tabla 27: Diseño de caso de prueba del requisito Modificar mensaje de notificación

Escenario	Descripción	Buzón	Mime	Descripción	Respuesta del sistema	Flujo central
-----------	-------------	-------	------	-------------	-----------------------	---------------

ESC 8.1 Modificar mensaje de notificación correctamente .	El sistema muestra un página con los campos requeridos para modificar el mensaje de notificación	V/Jabber	V/	V/Por medio de esta notificación le pedimos que se presente en nuestras oficina para culminar sus trámites.	El sistema muestra un mensaje "El mensaje se modificó correctamente".	1-Gestionar mensaje de notificación. 2-Modificar mensaje de notificación. 3-Aceptar.
ESC 8.2 Modificar mensaje de notificación correctamente .	El sistema muestra un página con los campos requeridos para modificar el mensaje de notificación	V/Correo	V/	V/Por medio de esta notificación le pedimos que se presente en nuestras oficina para culminar sus trámites.	El sistema muestra un mensaje "El mensaje se modificó correctamente".	1-Gestionar mensaje de notificación. 2-Modificar mensaje de notificación. 3-Aceptar.
ESC 8.3 Modificar mensaje de notificación correctamente .	El sistema muestra un página con los campos requeridos para modificar el mensaje de notificación	V/SMS	V/	V/Por medio de esta notificación le pedimos que se presente en nuestras oficina para culminar sus trámites.	El sistema muestra un mensaje "El mensaje se modificó correctamente".	1-Gestionar mensaje de notificación. 2-Modificar mensaje de notificación. 3-Aceptar.
ESC 8.4 Modificar mensaje de notificación correctamente .	El sistema muestra un página con los campos requeridos para modificar el mensaje de notificación	V/App	V/	V/Por medio de esta notificación le pedimos que se presente en nuestras oficina para culminar sus trámites.	El sistema muestra un mensaje "El mensaje se modificó correctamente".	1-Gestionar mensaje de notificación. 2-Modificar mensaje de notificación. 3-Aceptar.
ESC 8.5 Modificar mensaje de notificación incorrectamente.	El sistema muestra un página con los campos requeridos para modificar el mensaje de notificación	V/Jabber	I/1232xsds	V/Por medio de esta notificación le pedimos que se presente en nuestras oficina para culminar sus trámites.	El sistema muestra un mensaje "Existen datos incorrectos".	1-Gestionar mensaje de notificación. 2-Modificar mensaje de notificación. 3-Aceptar.
ESC 8.6 Modificar mensaje de notificación incorrectamente.	El sistema muestra un página con los campos requeridos para modificar el mensaje de notificación	V/Correo	V/	I/1232xsds	El sistema muestra un mensaje "Existen datos incorrectos".	1-Gestionar mensaje de notificación. 2-Modificar mensaje de notificación. 3-Aceptar.

ESC 8.7 Modificar mensaje de notificación dejando campos vacíos.	El sistema muestra un página con los campos requeridos para modificar el mensaje de notificación	V/App	Vacío/	V/Por medio de esta notificación le pedimos que se presente en nuestras oficina para culminar sus trámites.	El sistema muestra un mensaje "Campo requerido".	1-Gestionar mensaje de notificación. 2-Modificar mensaje de notificación. 3-Aceptar.
ESC 8.8 Modificar mensaje de notificación dejando campos vacíos.	El sistema muestra un página con los campos requeridos para modificar el mensaje de notificación	V/SMS	V/	Vacío/	El sistema muestra un mensaje "Campo requerido".	1-Gestionar mensaje de notificación. 2-Modificar mensaje de notificación. 3-Aceptar.
ESC 8.9 Opción Aceptar.	El sistema muestra la opción <b>Aceptar</b> para modificar el mensaje.				El sistema muestra un mensaje "El mensaje se modificó correctamente".	1-Gestionar mensaje de notificación. 2-Aceptar.
ESC 8.10 Opción Cancelar.	El sistema muestra la opción <b>Cancelar</b> para cancelar la petición.				El sistema muestra la página principal.	1-Gestionar mensaje de notificación. 2-Cancelar.

**Eliminar mensaje notificación**

**Tabla 28: Diseño de caso de prueba del requisito Eliminar mensaje de notificación**

<b>Escenario</b>	<b>Descripción</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
ESC 9.1 Eliminar mensaje de notificación correctamente.	El sistema muestra una página con el listado de mensajes.	El sistema muestra un mensaje "Esta seguro que desea eliminar el mensaje".	1-Gestionar mensaje de notificación. 2-Eliminar mensaje de notificación. 3-Aceptar.
ESC 9.2 Eliminar mensaje de notificación incorrectamente	El sistema muestra una página con el listado de mensajes.	El sistema muestra un mensaje "Debe seleccionar un tipo de mensaje a eliminar".	1-Gestionar mensaje de notificación. 2-Eliminar mensaje de notificación. 3-Aceptar.
ESC 9.3 Opción Aceptar.	El sistema muestra la opción <b>Aceptar</b> para eliminar el mensaje.	El muestra un mensaje " El mensaje se eliminó correctamente".	1-Gestionar mensaje de notificación. 2-Aceptar.

ESC 9.4 Opción Cancelar.	El sistema muestra la opción <b>Cancelar</b> para cancelar la petición.	El sistema muestra la página principal.	1-Gestionar mensaje de notificación. 2-Cancelar.
--------------------------	---	---	---

**Monitorear mensajes**

**Tabla 29: Diseño de caso de prueba del requisito Monitorear mensajes**

Escenario	Descripción	Respuesta del sistema	Flujo central
ESC 10.1 Mostrar listado de mensajes enviados.	El sistema muestra una página con el listado de mensajes que han sido enviados.	El sistema muestra un mensaje "El mensaje se eliminó correctamente".	1-Monitorear mensajes. 2-Aceptar.
ESC 10.2 Eliminar un mensaje enviado incorrectamente.	El sistema muestra una página con el listado de mensajes que han sido enviados.	El sistema muestra un mensaje "Debe seleccionar un mensaje para eliminar".	1-Monitorear mensajes. 2-Aceptar.
ESC 10.3 Opción Aceptar.	El sistema muestra la opción <b>Aceptar</b> para eliminar el mensaje.	El sistema muestra un mensaje "Esta seguro que desea eliminar el mensaje".	1-Gestionar mensaje de notificación. 2-Aceptar.
ESC 10.4 Opción Cancelar.	El sistema muestra la opción <b>Cancelar</b> para cancelar la petición.	El sistema muestra la página principal.	1-Gestionar mensaje de notificación. 2-Cancelar.

**Modificar conexión**

**Tabla 30: Diseño de caso de prueba del requisito Modificar conexión**

Escenario	Descripción	Servidor de Correo o Jabber	Usuario	Contraseña	Respuesta del sistema	Flujo central
ESC 11.1 Modificar conexión correctamente.	El sistema muestra una página con los campos requeridos para modificar la configuración de la conexión.	V/192.168.0.210	<a href="mailto:V/yprieto@uci.dev">V/yprieto@uci.dev</a>	V/●●●●●●●●	El sistema muestra un mensaje "La configuración de la conexión se modificó correctamente".	1-Modificar conexión. 2-Aceptar.
ESC 11.2 Modificar conexión correctamente.	El sistema muestra una página con los campos requeridos para modificar la configuración de la conexión.	V/jabber.uci.dev	<a href="mailto:V/yprieto@uci.dev">V/yprieto@uci.dev</a>	V/●●●●●●●●	El sistema muestra un mensaje "La configuración de la conexión se modificó correctamente".	1-Modificar conexión. 2-Aceptar.

ESC 11.3 Modificar conexión incorrectamente	El sistema muestra una página con los campos requeridos para modificar la configuración de la conexión.	I/165232 34.6756" 3	<a href="mailto://yprieto@uci.dev">//yprieto@uci. dev</a>	V//●●●●●●●●	El sistema muestra un mensaje "La configuración para la conexión tiene campos incorrectos".	1-Modificar conexión. 2-Aceptar.
ESC 11.4 Modificar conexión incorrectamente	El sistema muestra una página con los campos requeridos para modificar la configuración de la conexión.	V/jabber. uci.dev	<a href="mailto://lo1561.sifnj@fsdf">//lo1561.sifnj @.fsdf</a>	V//●●●●●●●●	El sistema muestra un mensaje "La configuración para la conexión tiene campos incorrectos".	1-Modificar conexión. 2-Aceptar.
ESC 11.5 Modificar conexión incorrectamente	El sistema muestra una página con los campos requeridos para modificar la configuración de la conexión.	V/192.16 8.0.210	<a href="mailto://yprieto@uci.dev">//yprieto@uci. dev</a>	//	El sistema muestra un mensaje "La configuración para la conexión tiene campos incorrectos".	1-Modificar conexión. 2-Aceptar.
ESC 11.6 Modificar mensaje de notificación dejando campos vacíos.	El sistema muestra una página con los campos requeridos para modificar la configuración de la conexión.	Vacío/	<a href="mailto://yprieto@uci.dev">//yprieto@uci. dev</a>	V//●●●●●●●●	El sistema muestra un mensaje "Campo requerido".	1-Modificar conexión. 2-Aceptar.
ESC 11.7 Modificar mensaje de notificación dejando campos vacíos.	El sistema muestra una página con los campos requeridos para modificar la configuración de la conexión.	V/192.16 8.0.210	Vacío/	V//●●●●●●●●	El sistema muestra un mensaje "Campo requerido".	1-Modificar conexión. 2-Aceptar.
ESC 11.8 Modificar mensaje de notificación dejando campos vacíos.	El sistema muestra una página con los campos requeridos para modificar la configuración de la conexión.	V/jabber. uci.dev	<a href="mailto://yprieto@uci.dev">//yprieto@uci. dev</a>	Vacío/	El sistema muestra un mensaje "Campo requerido".	1-Modificar conexión. 2-Aceptar.

ESC 11.9 Opción Aceptar.	El sistema muestra la opción <b>Aceptar</b> para modificar la conexión.				El sistema modifica la configuración para la conexión.	-Modificar conexión. 2-Aceptar.
ESC 11.10 Opción Cancelar.	El sistema muestra la opción <b>Cancelar</b> para cancelar la petición.				El sistema muestra la página principal.	1-Modificar conexión. 2-Aceptar.

### **Realizar suscripción a notificaciones**

Tabla 31: Diseño de caso de prueba del requisito Realizar suscripción a notificaciones

Escenario	Descripción	Contacto	Respuesta del sistema	Flujo central
ESC 12.1 Adicionar suscripción correctamente.	El sistema muestra los campos para la búsqueda de la persona a notificar. Luego de seleccionar la persona se muestra una página para adicionarle una notificación dicha persona.	V/yprieto	El sistema muestra el listado de los tipos de notificaciones existentes.	1-Suscripción a notificaciones. 2-Aceptar.
ESC 12.2 Opción Aceptar.	El sistema muestra el listado de notificaciones para que el usuario seleccione el tipo de notificación.		El sistema le adiciona esa suscripción y la elimina del listado de notificaciones.	1-Suscripción a notificaciones. 2-Aceptar.
ESC 12.3 Opción Cancelar.	El sistema muestra el listado de notificaciones para que el usuario seleccione el tipo de notificación.		El sistema muestra la página principal del requisito Suscripción a notificaciones.	1-Suscripción a notificaciones. 2-Cancelar.
ESC 12.4 Adicionar suscripción incorrectamente.	El sistema muestra los campos para la búsqueda de la persona a notificar. Luego de seleccionar la persona se muestra una página para adicionarle una notificación dicha persona.	V/-\$%\$&	El sistema muestra un mensaje de error "Existen datos incorrectos".	1-Suscripción a notificaciones. 2-Aceptar.
ESC 12.5 Adicionar suscripción dejando campos vacíos.	El sistema muestra los campos para la búsqueda de la persona a notificar. Luego de seleccionar la persona se muestra una página para adicionarle una notificación dicha persona.	Vacío/	El sistema muestra un mensaje de error "Campo requerido".	1-Suscripción a notificaciones. 2-Aceptar.
ESC 12.6 Eliminar suscripción correctamente.	El sistema muestra los campos para la búsqueda de la persona a notificar. Luego de seleccionar la persona se muestra una página para eliminar las notificaciones.	V/yprieto	El sistema muestra el listado de notificaciones con que cuenta esa persona.	1-Suscripción a notificaciones. 2-Aceptar.

ESC 12.7 Opción Aceptar.	El sistema muestra un listado con las notificaciones de la persona seleccionada.		El sistema muestra un mensaje "Usted desea eliminar esta suscripción".	1-Suscripción a notificaciones. 2-Aceptar.
ESC 12.8 Opción Aceptar.	El sistema muestra un listado con las notificaciones de la persona seleccionada.		El sistema muestra la página principal del requisito Suscripción a notificaciones.	1-Suscripción a notificaciones. 2-Aceptar.
ESC 12.9 Opción Cancelar	El sistema muestra los campos para la búsqueda de la persona a notificar.		El sistema muestra la página principal.	1-Suscripción a notificaciones. 2-Cancelar.

### Anexo V Procesos del sistema

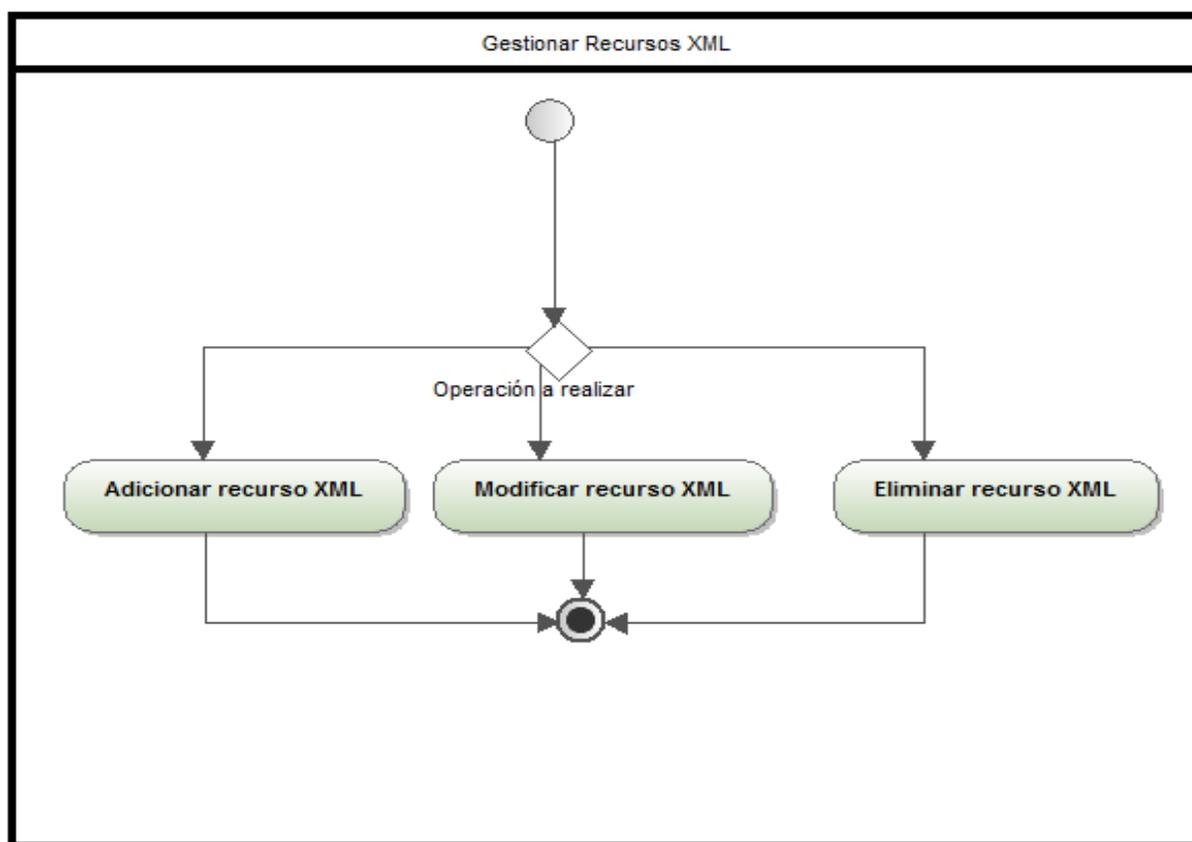


Figura 22: Proceso gestionar recurso XML

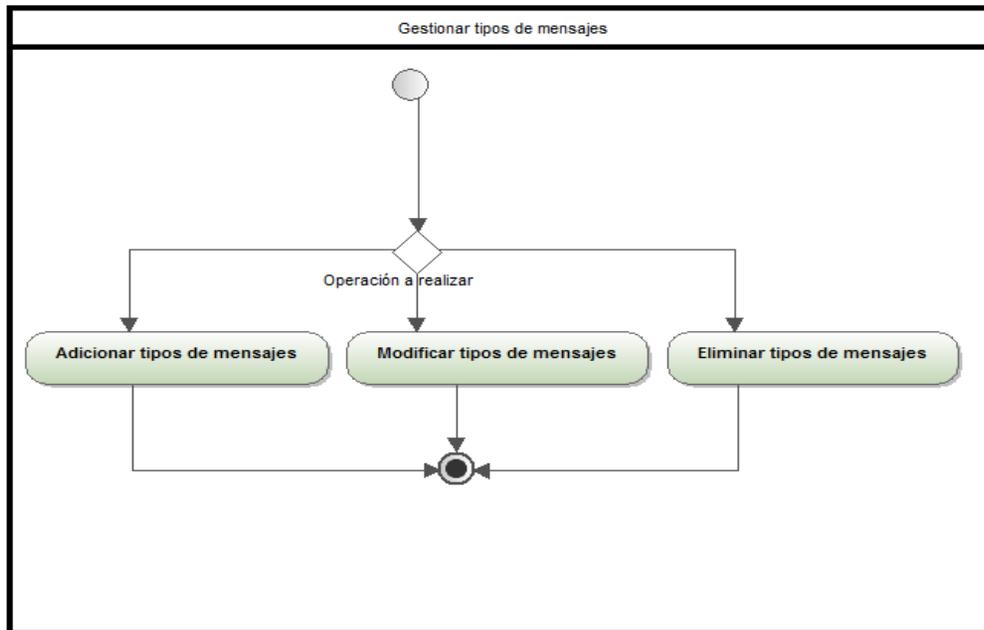


Figura 23: Proceso gestionar tipo de mensaje

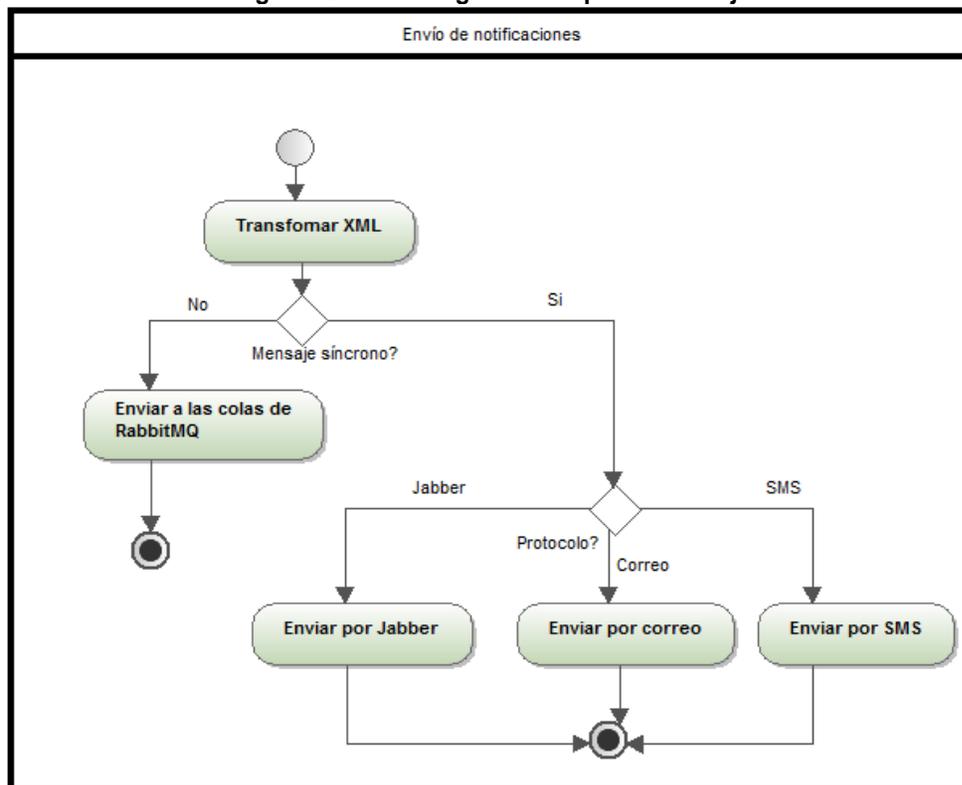


Figura 24: Proceso de envío de notificaciones