



Universitat Oberta
de Catalunya

www.uoc.edu

Desarrollo de Herramientas para la consulta de mensajería en integraciones realizadas con Mirth Connect.

Especialidad: Desarrollo de Aplicaciones de Software Libre.

Autor: Alberto Manso Iglesias.

Consultor: Gregorio Robles Martínez.

Tutor Externo: Eduardo Sierra Pérez.

Fecha: 09/06/2013.

Copyright (C) Alberto Manso Iglesias.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Índice de contenido

1.Introducción.....	4
1.1 Contextualización y Motivación.....	4
1.2 Tecnologías Relacionadas.....	7
1.3 Estructura del documento.....	8
2 Objetivos:.....	9
3 Desarrollo.....	11
3.1 Análisis.....	11
3.1.1- Toma de Requisitos:.....	11
3.1.2 Estudio de la gestión de mensajería en Mirth.....	12
3.1.3 Análisis de GWT, validez para la solución:.....	13
3.2 Diseño de la solución	14
3.2.1 Introducción:.....	14
3.2.2 Arquitectura:.....	14
3.2.3 Flujo Funcional:.....	15
3.2.4 Esquema de BBDD:.....	16
3.2.5 Pantalla:.....	17
3.2.6 Interfaces:.....	18
3.2.7 Consultas a BBDD:.....	20
3.3. Creación de Entorno:.....	20
3.4 Implementación de Servicios y Capa de Presentación:.....	22
3.5 Batería de Pruebas.....	25
4 Resultados.....	28
5 Conclusiones y Lecciones aprendidas.....	36
6 LICENCIA:.....	38
GNU Free Documentation License.....	38
0. PREAMBLE.....	38
1. APPLICABILITY AND DEFINITIONS.....	38
2. VERBATIM COPYING.....	39
3. COPYING IN QUANTITY.....	39
4. MODIFICATIONS.....	40
5. COMBINING DOCUMENTS.....	41
6. COLLECTIONS OF DOCUMENTS.....	42
7. AGGREGATION WITH INDEPENDENT WORKS.....	42
8. TRANSLATION.....	42
9. TERMINATION.....	42
10. FUTURE REVISIONS OF THIS LICENSE.....	43
11. RELICENSING.....	43

1.Introducción

1.1 Contextualización y Motivación

Las integraciones pretenden intercomunicar sistemas heterogeneos. En esta tarea el objetivo es que los sistemas a comunicar trasciendan del lenguaje, los protocolos de comunicación, estandares y las características de los otros. Así, gracias al motor de integración cada sistema proveerá/publicará la información de la manera más “natural” para él mismo y obviará toda gestión relacionada con el/los sistemas destino. Una integración coherente debe ser escalable y responsable de la entrega de la información a los destinos interesados despachando los mensajes que la contienen en el formato y mediante los protocolos de comunicación más óptimos para cada uno de los sistemas intervinientes. Debe por tanto tomar decisiones de enrutamiento e identificación de eventos y será el responsable de garantizar la entrega de los mensajes con la estrucutra más adecuada a cada sistema y mediante el protocolo que les sea más favorable.

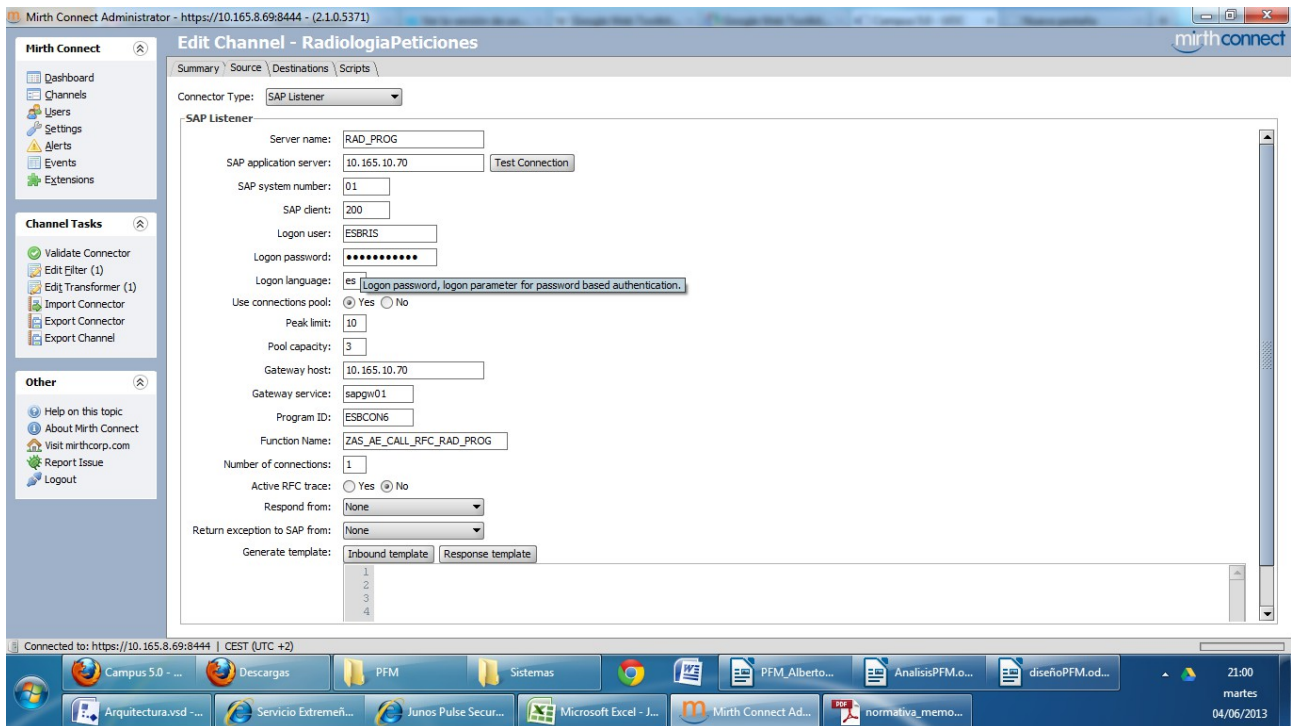
En este escenario es vital que la integración sea infalible en la garantía de entrega de la información a los sistemas implicados y se convierte en un punto nuclear de los servicios a la hora de detectar incidencias en los procesos que generan mensajería en los orígenes y destinos.

El principal objetivo de este proyecto es crear una herramienta que en el ámbito del proyecto JARA sea capaz de proveer al departamento de integración esa capacidad para identificar los puntos de fallo en los ciclos de mensajería entre sistemas de una manera, rápida, fiable y amigable, de manera que a la hora de resolver incidencias, el periodo de identificación y localización de la misma sea el mínimo posible. Así reduciremos los tiempos de respuesta y se facilitará la discriminación de responsabilidades.

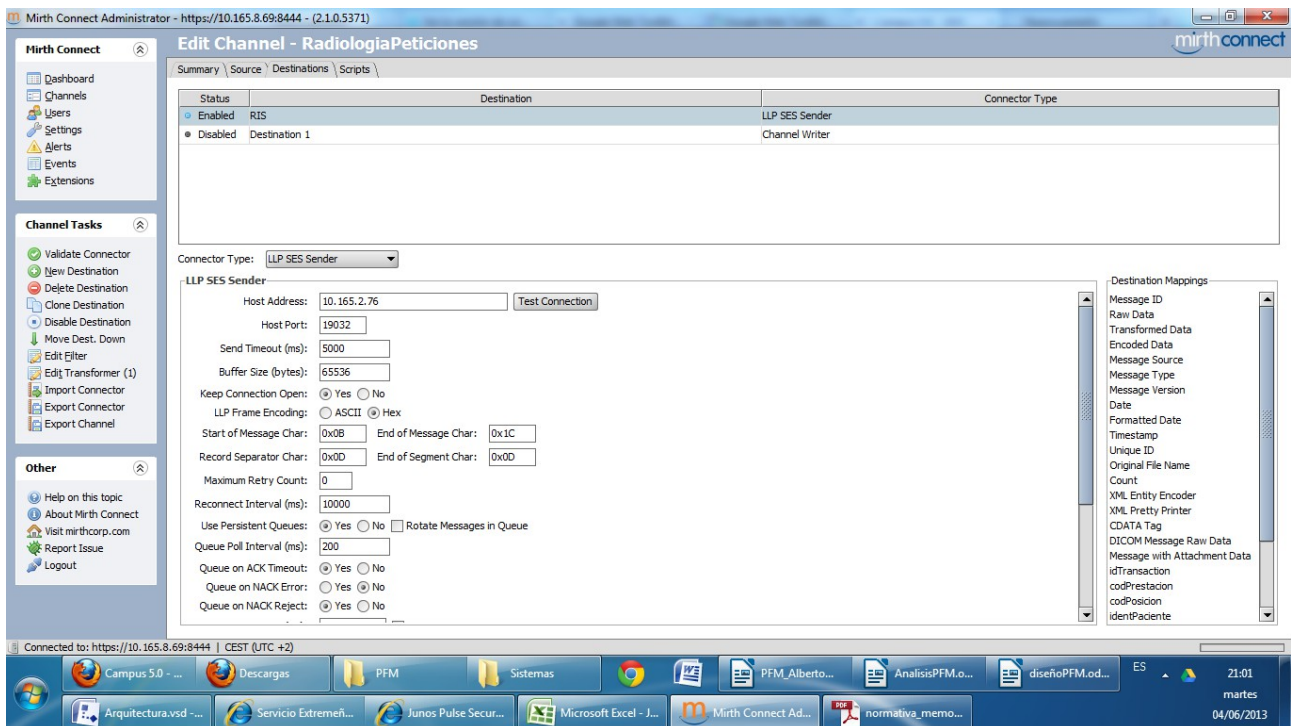
El proyecto JARA es un proyecto informatico-sanitario para el Sistema de Salud de Extremadura (SES). Es un proyecto ambicioso que intenta proveer una herramienta que abarque tanto los procesos asistenciales como la gestión del personal sanitario, el historial de los pacientes y prácticamente todo aquello relacionado con la realidad sanitaria en Extremadura. El núcleo central de este proyecto es una solución SAP customizada pero existen multitud de sistemas externos que son alimentados con información del sistema SAP y viceversa, o incluso, sistemas que no se comunican con SAP. Por tanto es necesaria la existencia de una integración que gestione ese intercambio de información entre sistemas y es necesario que esa integración disponga de una herramienta que permita diagnosticar los puntos de fallo de manera rápida y eficaz.

En el proyecto existen varios motores de integración pero actualmente el motor sobre el que más se desarrolla y al que se están migrando integraciones antiguas es Mirth Connect. Un motor de software libre desarrollado específicamente para integraciones sanitarias y que provee de manera nativa el estandar de mensajería HL7.

Las soluciones desarrolladas en Mirth Connect se llaman canales, de manera que, a grandes rasgos, un canal permite el envío de mensajería de un sólo origen a N destinos.



Img.1 Ejemplo de Origen en un canal

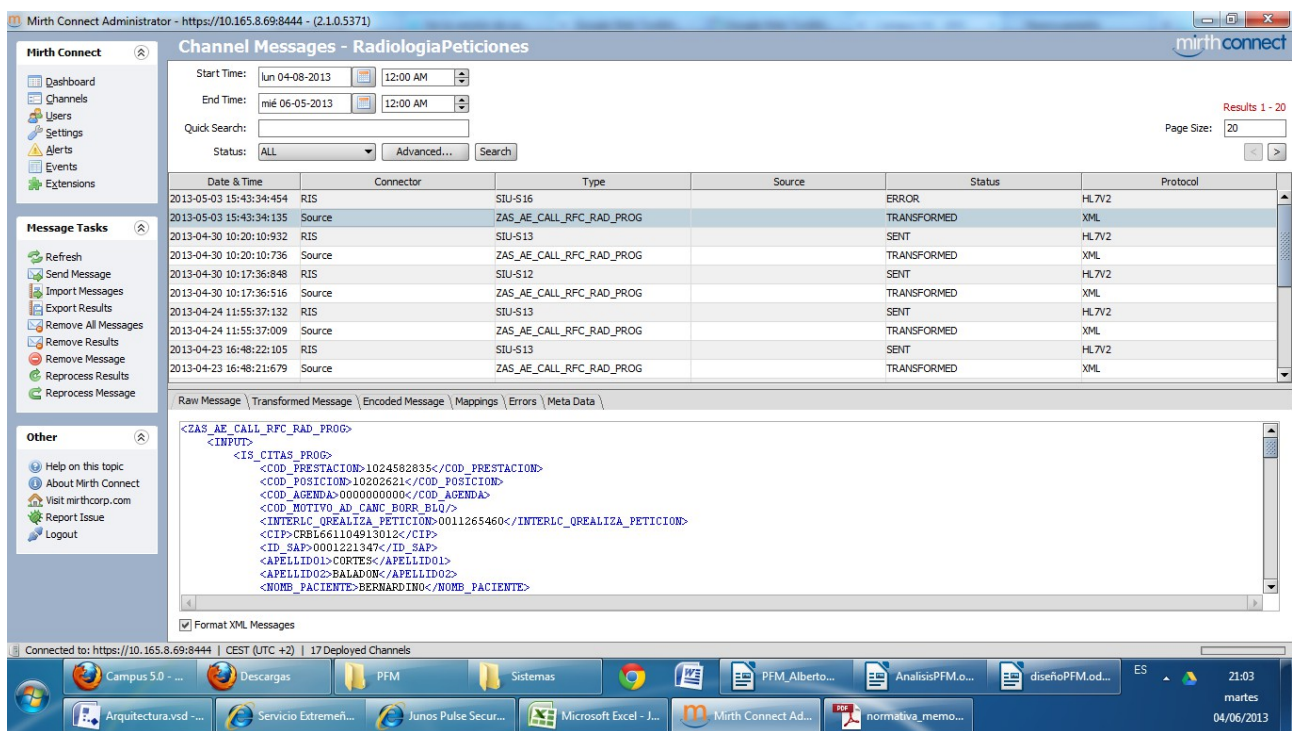


Img.2 Ejemplo de Destinos en un canal.

Este modelo de solución implica que cuando tengamos dos o más sistemas generadores y receptores de mensajería tendremos que crear más de un canal para proveer la integración de los mismos, ya que cada canal tan solo tiene un “source”, por tanto sólo podría recibir mensajes de un sistema.

Por regla general el intercambio de mensajería suele ser bidireccional y suele tener un flujo lógico y ordenado. Pongamos un ejemplo: Imaginemos dos sistemas, JARA como puesto clínico del médico y radiología como sistema utilizado por las máquinas y los profesionales de Radiodiagnóstico. Supongamos que desde JARA que se pretende dar una cita para realizarle una radiología a un paciente. Esa orden clínica viajaría desde JARA al sistema de Radiología a través de la integración. La solución en Mirth sería crear un canal que recibiera el mensaje de cita de JARA y lo entregara en Radiología. Pero el ciclo no acaba aquí, sino que una vez se registre la cita en Radiología y el paciente acuda a ella, Radiología deberá enviar un informe a JARA para que este disponga de los resultados de la prueba. Esta comunicación también se realizaría a través del motor de integración y requeriría la existencia de otro canal donde esta vez, Radiología sería el “source” y JARA el destino. Tendríamos por tanto una solución formada por dos canales Mirth. Esta puntualización es importante ya que en ella está la clave determina la necesidad del desarrollo de la herramienta que propone este proyecto.

Mirth Connect provee de una funcionalidad que permite realizar búsquedas de la mensajería que ha sido despachada por el motor. El problema de esta herramienta es que tan sólo permite realizar búsquedas por canal. Es decir, en el ejemplo anterior, si seguir el flujo de mensajería del caso anterior, deberíamos realizar dos búsquedas, una en el canal que recibe mensajes de JARA y lo entrega en Radiología y otra en el canal que recibe mensajes de Radiología y los entrega en JARA. Perdiendo además la visualización de la información extraída en la primera búsqueda al realizar la segunda.



Img.3 Ejemplo de consulta utilizando la herramienta que provee Mirth para tal fin.

La herramienta desarrollada en este proyecto permite vincular la mensajería entre canales. Gracias a ello se realizarán las búsquedas obviando la arquitectura de la solución y de manera transparente para el usuario de la herramienta. Tan sólo seleccionará la integración sobre la que quiere consultar y la aplicación automáticamente (previa configuración de metainformación en BBDD) identificará los canales que intervienen en esa integración y realizará la búsqueda en ellos,

mostrando posteriormente la información recuperada de manera ordenada y coherente. Esta solución permite que la herramienta sea escalable y se puedan vincular nuevos canales o destinos a una aplicación modificando la tabla de metainformación.

1.2 Tecnologías Relacionadas

La mensajería que pasa a través del motor se almacena en la base de datos que da soporte al motor. De ahí se recuperarán los mensajes que consultará la aplicación. La base de datos será instalada en nuestros motores será Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 – 64bits.

El motor de integración será la versión 2.5 de Mirth Connect con las extensiones desarrolladas por el departamento de integración del proyecto JARA para proveer otros protocolos de comunicación y algunas características relacionadas con la gestión de la respuesta. Mirth provee de la v2.5 en su web <http://www.mirthcorp.com/>

Con respecto a la aplicación se ha elegido crear una aplicación web que permita el acceso multiusuario a la misma y que sea desplegada sobre un servidor de aplicaciones. A tal efecto se ha valorado como solución el Framework de Google GWT. es un framework que nos permite utilizar las características de las Rich Internet Applications ejecutándose gran parte del procesamiento en cliente y evitando así la carga del servidor de la aplicación. La ventaja de GWT es que nos permite programar en JAVA sin necesidad de entrar en la complejidad de HTML+Javascript. Una vez compilado el código se transforma de manera transparente para el desarrollador en HTML y Javascript. Dispondremos así de un framework que trabajará con servicios asíncronos y nos permitirá crear una aplicación dinámica, amigable y poco pesada para el servidor. GWT provee de su propia API donde podemos encontrar todo tipo de objetos html enriquecidos ya disponibles para ser usados y con una alta capacidad de parametrización.

Todas estas características hacen de GWT un framework recomendable para la tarea que nos aborda, en la cual intentaremos no cargar las máquinas que contienen nuestros motores con el procesamiento de esta aplicación de consulta, ejecutando la mayor parte posible de código en el cliente.

Se puede encontrar el framework, documentación y todo tipo de información en el siguiente enlace <https://developers.google.com/web-toolkit/> .

Todo está montado sobre Sistemas AIX de 64 bits, aunque esto es transparente para el desarrollo ya que la aplicación se desplegará en un WAR sobre un servidor de aplicaciones Tomcat, por tanto no afecta al objetivo de este proyecto el sistema operativo que soporte los componentes.

1.3 Estructura del documento

El documento del que forma parte esta integración está compuesto por 4 apartados principales además de éste.

A esta introducción seguirían los siguientes apartados:

2.- **Objetivos.** Consistente en una descripción de los objetivos principales y las tareas y subobjetivos que se derivan de ellos, su planificación en tiempos y las características y justificaciones de los mismos.

3.- **Desarrollo.** Correspondería con todo el trabajo asociado a esas tareas para la consecución de los objetivos.

4.- **Resultados.** En este apartado mostraríamos el “producto” final del proyecto. Su explotación y las funcionalidades alcanzadas.

5.- **Conclusiones y Lecciones Aprendidas.** Apartado destinado a realizar un feedback del proyecto y los resultados obtenidos, análisis de los objetivos alcanzados, etc.

2 Objetivos:

El objetivo principal del proyecto es desarrollar una solución que nos permita consultar la mensajería que atraviesa las integraciones. El valor añadido que tiene esta solución sería la capacidad de aglutinar de manera ordenada todos los mensajes referentes a una integración.

Para alcanzar este objetivo principal hemos de ir alcanzando otros objetivos de manera secuencial hasta completar el ciclo de vida de un proyecto de desarrollo para garantizar así la fiabilidad de la solución.

Podemos secuenciar los objetivos de la siguiente manera:

1.- Análisis de la solución.

El objetivo sería la situación de partida y los objetivos funcionales del proyecto. Para ello será necesario conocer bien los componentes a tratar para que el análisis sea una base coherente para el proyecto. Se subdividirá este objetivo en tres subobjetivos.

1.1 toma de requisitos.

Consistente en identificar los requerimientos, los objetivos funcionales que queremos que abarque nuestra solución.

1.2 Estudio la gestión de mensajería de Mirth.

Consistente en un análisis de cómo el motor almacena la información de la mensajería en la BBDD. Así dispondremos del conocimiento necesario para poder “rescatar” la información al hacer una búsqueda.

1.3 Viabilidad de la solución en GWT.

Consiste en analizar si el Framework inicialmente propuesto para realizar el proyecto se ajusta a los requerimientos y es una solución adecuada a los mismos.

2.- Diseño de la solución.

La consecución de este objetivo nos permitirá tener un patrón técnico de la solución. Será una conceptualización técnica de la solución que consistirá en el esquema o modelo a seguir técnicamente para implementar la solución analizada en el punto 1. Definición de las interfaces. Consiste en identificar las clases y métodos que proveerán los servicios necesarios para la aplicación.

3.- Validación de la solución.

Una vez alcanzado este objetivo dispondremos de la aceptación por parte del tutor externo del modelo de la solución tanto a nivel funcional como a nivel técnico. Por tanto consiste en que el tutor externo de por válida la documentación generada para el análisis funcional y el diseño de la solución.

4.- Creación del Entorno. Consistente en preparar el entorno donde realizar el desarrollo y poder probar la aplicación.

5.- Implementación de los servicios.

Separando el desarrollo en 3 capas, una de presentación, otra de negocio y otra de acceso a BBDD. Podríamos implementar los servicios definidos en el objetivo anterior y probarlos de manera independiente, sin tener en cuenta la parte web o de presentación. Una vez conseguido este objetivo tendríamos la garantía de que la información recuperada y la funcionalidad definida de los procesos de negocio son correctos, más allá de la fiabilidad o calidad de la capa de presentación.

6.- Implementación de la capa de presentación.

Constituye el último objetivo del desarrollo y una vez alcanzado tendremos finalizada la

parte visual de la aplicación, es decir, la parte que interactúa con el usuario por un lado y con los servicios por otro.

7.- Batería de pruebas de la aplicación.

Éste objetivo podría definirse como la búsqueda de la garantía de calidad del producto. Nos aportará seguridad a la hora de confiar en el desarrollo.

8.- Validación de la aplicación.

Por último, el objetivo final será la validación de la aplicación por parte del tutor externo.

También podemos añadir otros objetivos que se espera ir adquiriendo con la consecución de los descritos anteriormente. Algunos ejemplos pueden ser:

- Conocimiento de la plataforma GWT.
- Desarrollo de aplicación en capas separadas de negocio, presentación y acceso a BBDD.
- Conocimiento de la gestión de mensajería en BBDD De Mirth.

3 Desarrollo

Este apartado aborda el desarrollo de todo el proyecto, desde las tareas de análisis de la solución hasta las pruebas de la aplicación. Por tanto, en él se podrá obtener el nivel de detalle preciso para conocer los motivos de cada una de las decisiones que se han tomado y valorar la idoneidad de las mismas.

3.1 Análisis

3.1.1- Toma de Requisitos:

La web final debe tener una serie de características:

- **Aplicacion:** Debe ser posible seleccionar la aplicación sobre la que se hará la consulta. Al referirnos a aplicación nos referimos a todos aquellos canales Mirth que estén relacionados funcionalmente. La motivación que genera este requisito es doble. Por un lado, al hacer una búsqueda es conveniente obtener sólo los resultados referidos a la funcionalidad que nos ocupe. Es posible que exista mensajería para el mismo paciente, profesional, hospital, etc, en distintas integraciones, así que es importante seleccionar los resultados que nos interesen. Por otro lado, seleccionar la aplicación implicará seleccionar una serie de canales y evitaremos así consultas demasiado pesadas que puedan ralentizar el funcionamiento de la web y afectar a la BBDD. Hay que tener en cuenta que la BBDD sobre la que se consulta es la BBDD en la que registra la mensajería el motor, por tanto es importante minimizar el impacto de las consultas para evitar caídas o incidencias en la BBDD que pudieran provocar pérdida de mensajería.

- **Entornos:** Se dispone de 3 entornos Desarrollo, Preproducción y Producción. En desarrollo hay una sola instancia de Mirth pero en PRE existen 2 instancias mientras que en PRO existen 4 instancias ejecutándose. La aplicación web debe permitir seleccionar el entorno donde se quiere buscar e identificar que aplicaciones están disponibles en ese entorno y en que instancia se encuentra. Este último dato es importante ya que la búsqueda debe realizarse sobre el esquema de BBDD que corresponde a la instancia donde se encuentran esos canales.

- **Fechas Búsqueda:** Es necesario realizar la búsqueda para un periodo de fechas determinado. Evitaremos latencia en la web, carga de la BBDD y resultados no deseados. Imaginemos el ejemplo de un paciente al que se le han hecho dos solicitudes de radiografías pero con un año de diferencia, si no filtramos bien por fechas, además de hacer una búsqueda excesivamente lenta obtendremos la mensajería tanto del año pasado como de éste, y tendremos que descartar los mensajes que no nos interesen.

- **Campos Claves:** en cada integración existen elementos de la mensajería que permiten identificar que un grupo de mensajes pertenecen a la misma actividad. Pongamos el caso de una orden clínica para unos análisis de laboratorio. En primer lugar llegaría la orden clínica que indica que pruebas se han de realizar, posteriormente un mensaje en sentido contrario indicando que las pruebas se han realizado correctamente y por último un tercer mensaje en el mismo sentido que el segundo con el informe elaborado a partir de los resultados obtenidos. Estos tres mensajes, que podrían pasar por 3 canales distintos, estarían relacionados por medio de un campo clave, que en

este caso sería el “id de orden clínica”.

Por tanto, para cada integración, debería estar identificado cual/cuales son los campos claves para que la búsqueda más óptima se haga a través de ellos.

- **Búsqueda Sin Campos Claves:** También es conveniente que se pueda hacer la búsqueda por otro texto que no sea un campo clave. El motivo es que en ocasiones, las incidencias que hay que gestionar no se crean informando los campos claves, por tanto, habría que realizar una búsqueda inicial con la información que nos provean en la incidencia y posteriormente, para obtener todo el ciclo de mensajería, consultar a partir de los campos claves que encontremos en la primera búsqueda.

- **Orden cronológico de resultados:** Los resultados que se obtengan de la búsqueda han de aparecer ordenados cronológicamente.

- **Resultados Búsqueda.** Cada fila de resultados debe contener los siguientes campos -

Canal: Nombre del canal al que pertenece el mensaje encontrado.

Paciente: Si existe paciente, el nombre y apellidos del paciente al que se refiere el mensaje.

CIP: El cip es el identificador único del paciente en el Servicio Extremeño de Saludo, es como el DNI sanitario. Si el cip está informado, aparecerá en el resultado.

Fecha: Fecha en la que se registró el mensaje en integración,

Tipo de Mensaje. Las mayoría de las integraciones siguen el estandar sanitario HL7, aún en los casos que no lo sigue, el tipo de mensaje nos ayuda a identificar la acción funcional que pretende realizar ese mensaje.

Destino. Sistema destino al que se pretende enviar el mensaje.

Estado: Estado en el que ha quedado ese mensaje tras su paso por ese punto. OK, error, mensaje enviado, mensaje recibido, mensaje filtrado, etc.

XML/ER7: Si el mensaje es XML, ER7, texto plano, etc.

Mensaje: el contenido del mensaje en sí.

- **Mensajes E/S:** Por cada mensaje que pase por integración es necesario mostrar en la búsqueda el mensaje recibido y el mensaje enviado a cada destino. Así dispondremos del mensaje que envían a integración y del mensaje/s que envía integración.

3.1.2 Estudio de la gestión de mensajería en Mirth

Mirth almacena los mensajes en la tabla MESSAGE con las siguientes columnas:

SEQUENCE_ID: id secuencial creciente.

ID: id del mensaje.

SERVER_ID: id del servidor, de la instancia de mirth.

CHANNEL_ID: id del canal al que pertenece el mensaje.

SOURCE: Origen del mensaje.

TYPE: Tipo de Mensaje. Ej: ADT_A08 , OUL_R24, etc.

DATE_CREATED: Fecha en la que se registra el mensaje.

VERSION: Versión de HL7 del mensaje si es HL7.

IS_ENCRYPTED: Si se ha registrado el mensaje encriptado.

STATUS: Enviado, Transformado o filtrado.

RAW_DATA: mensaje recibido sin tratar.

RAW_DATA_PROTOCOL: si el raw data es: XML, HL7, delimited (texto delimitado)

TRANSFORMED_DATA: Mensaje después de ser transformado.

TRANSFORMED_DATA_PROTOCOL: si el transformer data es: XML, HL7, delimited (texto delimitado)

ENCODED_DATA: Mensaje con la codificación.

ENCODED_DATA_PROTOCOL: si el encoded data es: XML,HL7, delimited.

CONNECTOR_MAP: Objeto con información del connector

CHANNEL_MAP: Objeto con información del canal

RESPONSE_MAP: Objeto con información de la respuesta

CONNECTOR_NAME: Source si es recibido por mirth, nombre del destino en caso de tratarse de un destino.

ERRORS: Error en caso de que se haya producido.

CORRELATION_ID: ID que relaciona la tupla del mensaje recibido con las tuplas del mensaje en los destinos que deba alcanzar.

ATTACHMENT: Adjunto si lo hubiera.

Mirth registra el mensaje en el Source y en cada uno de los destinos, por tanto se puede **obtener** con facilidad el mensaje recibido y el enviado y obtener la información precisa para mostrar los datos de la toma de requisitos.

3.1.3 Análisis de GWT, validez para la solución:

GWT es un framework de Google que nos permite utilizar las características de las Rich Internet Applications ejecutándose gran parte del procesamiento en cliente y evitando así la carga del servidor de la aplicación. La ventaja de GWT es que nos permite programar en JAVA sin necesidad de entrar en la complejidad de HTML+Javascript. Una vez compilado el código se transforma de manera transparente para el desarrollador en HTML y Javascript. Dispondremos así de un framework que trabajará con servicios asíncronos y nos permitirá crear una aplicación dinámica, amigable y poco pesada para el servidor. GWT provee de su propia API donde podemos encontrar todo tipo de objetos html enriquecidos ya disponibles para ser usados y con una alta capacidad de parametrización.

Todas estas características hacen de GWT un framework recomendable para la tarea que nos aborda, en la cual intentaremos no cargar las máquinas que contienen nuestros motores con el procesamiento de esta aplicación de consulta, ejecutando la mayor parte posible de código en el cliente.

3.2 Diseño de la solución

3.2.1 Introducción:

Este apartado pretende recoger el diseño técnico de la aplicación para la consulta de incidencias. A partir del análisis recogido en el punto 2 dónde se trataba el análisis . Recogeremos a continuación la información técnica para poder abordar la construcción del código en función de los requisitos identificados en el documento de análisis.

3.2.2 Arquitectura:

Los elementos de arquitectura serán:

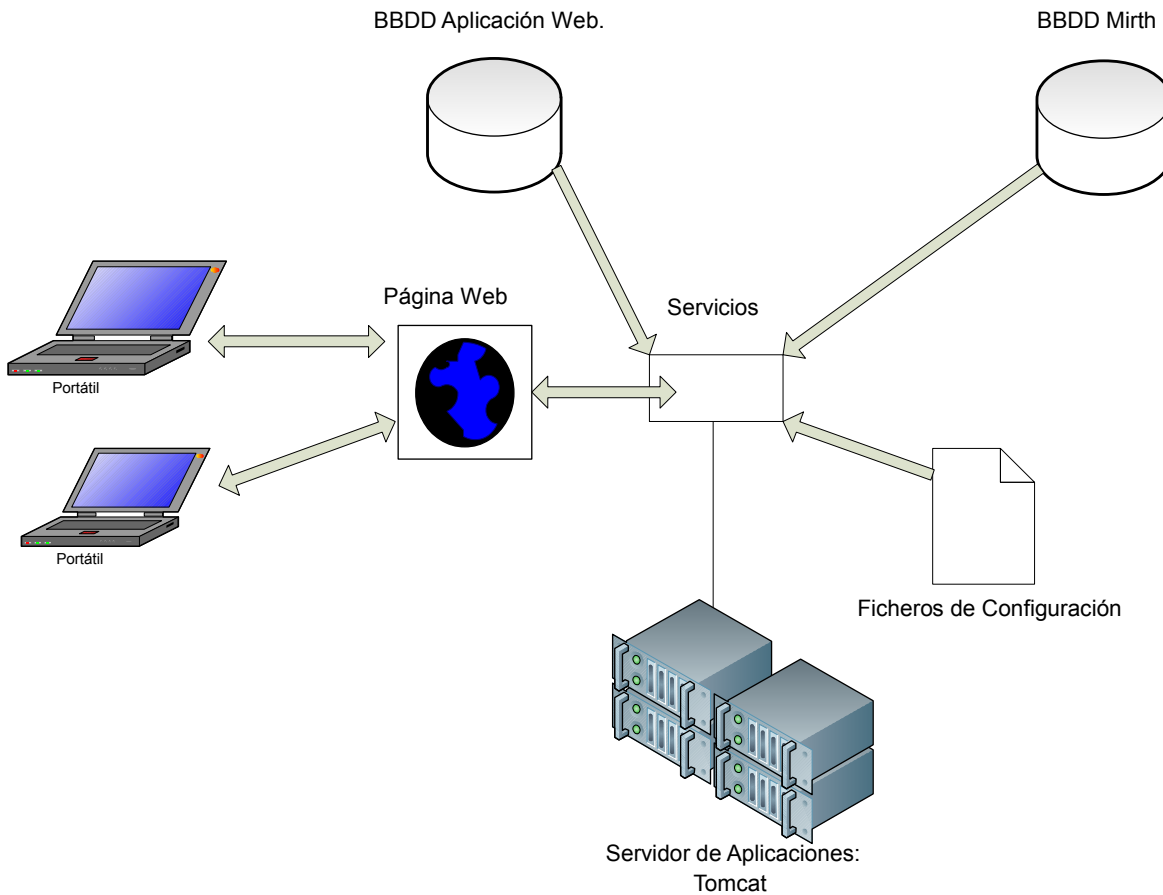
Servidor Tomcat 7.0.3 Será el servidor donde se aloje la aplicación.

Esquema BBDD Oracle 10g: BBDD utilizada por Mirth para almacenar mensajes.

Esquema BBDD Oracle 10g: BBDD utilizada por la aplicación para almacenar información propia.

Ficheros Configuración: Ficheros que utilizará la web para la configuración de la misma y para almacenar urls de conexión etc.

PCs clientes : Pcs que acceden a la aplicación.



3.2.3 Flujo Funcional:

Al cargar la página se realizará el siguiente proceso.

1.- A través del fichero MyConstants.properties cargamos los distintos entornos existentes en el combo a través de la propiedad del fichero entornos = DESARROLLO, PREPRODUCCION, PRODUCCION. Se cargaran esos valores en el combo de entornos.

2.- Se conecta a la BBDD de la aplicación y se recupera la información de las 3 tablas. APLICACIONES_INTEGRATOR_PRO, APLICACIONES_INTEGRATOR_DES, APLICACIONES_INTEGRATOR_PRE.

3.- Se selecciona por defecto el entorno de PRODUCCION en el combo de entornos.

4.- Se cargan en el combo de aplicaciones las aplicaciones del entorno de producción. Se selecciona la primera de ellas.

5.- Se cargan los textos y las cajas de textos referentes a los campos claves de la primera aplicación del entorno.

6.- Se cargan los textos y las cajas de textos de las fechas de inicio y fin de la búsqueda. También el texto y la caja de búsqueda por otro texto y el botón de buscar.

Nota*

Si se selecciona otro entorno en el combo de entornos se cargará de nuevo el combo de aplicaciones con la información referente a las aplicaciones del entorno seleccionado. Y se limpiaran los textos y las cajas de textos referentes a los campos claves anteriores y se cargarán los campos claves referentes a la primera aplicación que corresponda a ese entorno.

Si se cambia de aplicación dentro del mismo entorno, se limpiaran los textos y las cajas de textos referentes a los campos claves anteriores y se cargarán los campos claves referentes a la aplicación seleccionada.

Una vez elegidos los parámetros de la búsqueda y pulsado el botón buscar:

1.- Se conectará con la BBDD del entorno Mirth seleccionado. Se identificará la configuración de conexión gracias a al info existente en el fichero MyConstants.properties.

2.- Se realizará la consulta en la BBDD de Mirth entre las fechas seleccionadas para aquellos id de canal que correspondan con los ids relacionados con la aplicación en la tabla APLICACIONES_INTEGRATOR del entorno seleccionado.

3.- La consulta también devolverá los resultados para aquellas tuplas que contengan en el mensaje el valor especificado en un campo clave o los valores especificados en otros textos.

Nota* En la pantalla las cajas de campos claves son meramente informativas de que campos son los más óptimos para buscar. Sea cual sea el campo clave rellenado la búsqueda se hará en los mensajes de la misma manera.

4.- Una vez obtenidos los resultados se ordenará la información y se mostrará en pantalla.

3.2.4 Esquema de BBDD:

Existirá una tabla por cada entorno que utilizará la aplicación para funcionar contendrá las siguientes columnas:

APLICACION: Nombre de la aplicación a la que pertenece el canal.

NOMBRE_CANAL: Nombre del Canal.

ID_CANAL: ID que tiene el canal en la BBDD Mirth.

ORDEN: Posición en la que se mostrará la información referente a este canal. Supongamos que tenemos 3 canales asociados a una aplicación. Por tanto habrá 3 tuplas en esta tabla, una para cada canal. Una vez realizada una búsqueda en la BBDD de Mirth es necesario saber cual es el orden lógico o funcional en el que deberían haberse recibido los mensajes y por tanto mostrarse para

obtener la información de manera más clara. Esta información no tiene porqué corresponder con la fecha de recepción de la mensajería, ya que la incidencia puede venir determinada por un orden incorrecto de la mensajería.

TEXTO_FUNCIONAL: Texto que indica la funcionalidad de ese canal.

CANAL_ACKS: Indica si ese canal recoge ACKs.

ENTORNO: Entorno al que pertenece la aplicación. Desarrollo (DESA), preproducción1(PRE1), preproducción2(PRE2), producción1(PRO1), producción2(PRO2), producción3(PRO1), producción4(PRO1).

CAMPOCLAVE1..3: son columnas donde se indican los campos claves referentes a esa aplicación.

3.2.5 Pantalla:

El aspecto que debe tener la pantalla de la aplicación es el siguiente:

Entorno:

Aplicación: Campo Clave1:

Fecha Inicio: Campo Clave2:

Fecha Fin: Campo Clave3:

Buscar por otro texto

Paso1: NombreCanal1

Orden	Paciente	CIP	Fecha	Tipo Mensaje	Destino	Estado	Estado XML/ER7	MSG
Mirth								
1	Manso	MSG1324827	17-2 14:00:01.325	ORM_O01	INT	MSG Recibido	ok	ER7
2	Manso	MSG1324827	17-2 14:00:01.328	ORM_O01	APA	MSG En viado	Error	XML

Paso 2: NombreCanal2

Orden	Paciente	CIP	Fecha	Tipo Mensaje	Destino	Estado	Estado XML/ER7	MSG
Mirth								
3	Manso	MSG1324827	17-2 14:30:01.001	ORU_R01	INT	MSG Recibido	ok	XML
4	Manso	MSG1324827	17-2 14:30:01.001	ORU_R01	JARA	MSG En viado	ok	ER7

3.2.6 Interfaces:

Definiremos los siguientes objetos para intercambiar información:

Objeto Canales

```
private String aplicacion;  
private String nombreCanal;  
private String idCanal;  
private int orden;  
private String descripcionCanal;  
private String textoFuncional;  
private boolean canalACKs;
```

Objeto Aplicación

```
private String entorno;  
private String nombreAplicacion;  
private ArrayList<Canal> canales = new ArrayList<Canal>();  
private String campoRecomendado1;  
private String campoRecomendado2;  
private String campoRecomendado3;
```

Objeto Mensajes

```
private String paso;  
private String sequence_id;  
private String id;  
private String channel_id;  
private String type;  
private String status;  
private String date_created;  
private String raw_data;
```

Objeto FilaResultados

```
private String sequenceId;  
private String id;  
private String channelId;  
private String tipoMSG;  
private String fechaCreacion;  
private String status;  
private String raw_data;  
private String raw_data_protocol;  
private String transformed_data;  
private String transf_data_protocol;  
private String encoded_data;  
private String encoded_data_protocol;  
private String response;  
private String connectorName;  
private String errores;
```

```
private String correlationID;
private String textoResultadoMensaje;
private String paciente;
private String CIP;
```

ResultadosBusquedaCanal

```
private String nombCanal;
private String textoDescrip;
private ArrayList<FilaResultados> filas = new ArrayList<FilaResultados>();
private String idCanal;
private int orden;
private String textoFuncional;
private boolean canalRecepcionACKs;
```

Los servicios que publicará el servidor serán los dos siguientes:

1.- GetAplicaciones

No tendrá parámetros de entrada, devolverá un ArrayList de objetos Aplicacion con todas las aplicaciones de cada entorno.

```
public ArrayList<Aplicacion> getAplicaciones()
```

2.- GetResultadosBusqueda

Tendrá los siguientes parámetros de entrada:

- 2.1 entorno: Entorno en el cual se realizará la búsqueda de mensajería.
- 2.2 Campo: Array de String con los campos por el que se realizará la búsqueda dentro de los mensajes.
- 2.3 FechaInicio: Fecha a partir de la que se buscará en la BBDD Mirth correspondiente.
- 2.4 FechaFin: Fecha del último día del rango donde se buscarán en la BBDD Mirth correspondiente.

Este servicio devolverá un ArrayList de objetos de tipo FilaResultados ordenados.

```
public ArrayList<ResultadosBusquedaCanal>
getResultadosBusqueda(ArrayList<ResultadosBusquedaCanal> listaResultados, String
entorno, String campo, String fechaInicio, String fechaFin) {}
```

3.2.7 Consultas a BBDD:

Busqueda de aplicaciones

```
SELECT * FROM APLICACIONES_INTEGRATOR_DES ORDER BY APLICACION, ORDEN ASC
SELECT * FROM APLICACIONES_INTEGRATOR_PRE ORDER BY APLICACION, ORDEN ASC
SELECT * FROM APLICACIONES_INTEGRATOR_PRO ORDER BY APLICACION, ORDEN ASC
```

Busqueda de Resultados:

```
SELECT SEQUENCE_ID, CHANNEL_ID, SOURCE, TYPE, DATE_CREATED, STATUS, RAW_DATA,
TRANSFORMED_DATA, ENCODED_DATA, CONNECTOR_MAP, CONNECTOR_NAME, ERRORS,
CORRELATION_ID, RAW_DATA_PROTOCOL, TRANSFORMED_DATA_PROTOCOL,
ENCODED_DATA_PROTOCOL, RESPONSE_MAP FROM MESSAGE WHERE ( DATE_CREATED >='fechaInicio'
AND DATE_CREATED <='fechaFin') AND ( CHANNEL_ID ='idCanal1' or CHANNEL_ID ='idCanal2' ... or
CHANNEL_ID ='idCanalN') AND RAW_DATA LIKE '%campoBusqueda1%' OR TRANSFORMED_DATA like
'%campoBusqueda1%' and RAW_DATA LIKE '%campoBusqueda2%' OR TRANSFORMED_DATA like
'%campoBusqueda2%' and RAW_DATA LIKE '%campoBusqueda3%' OR TRANSFORMED_DATA like
'%campoBusqueda3%' AND ORDER BY DATE_CREATED ASC
```

eliminar aquellas partes de la sentencia en función de las características de la búsqueda.

3.3. Creación de Entorno:

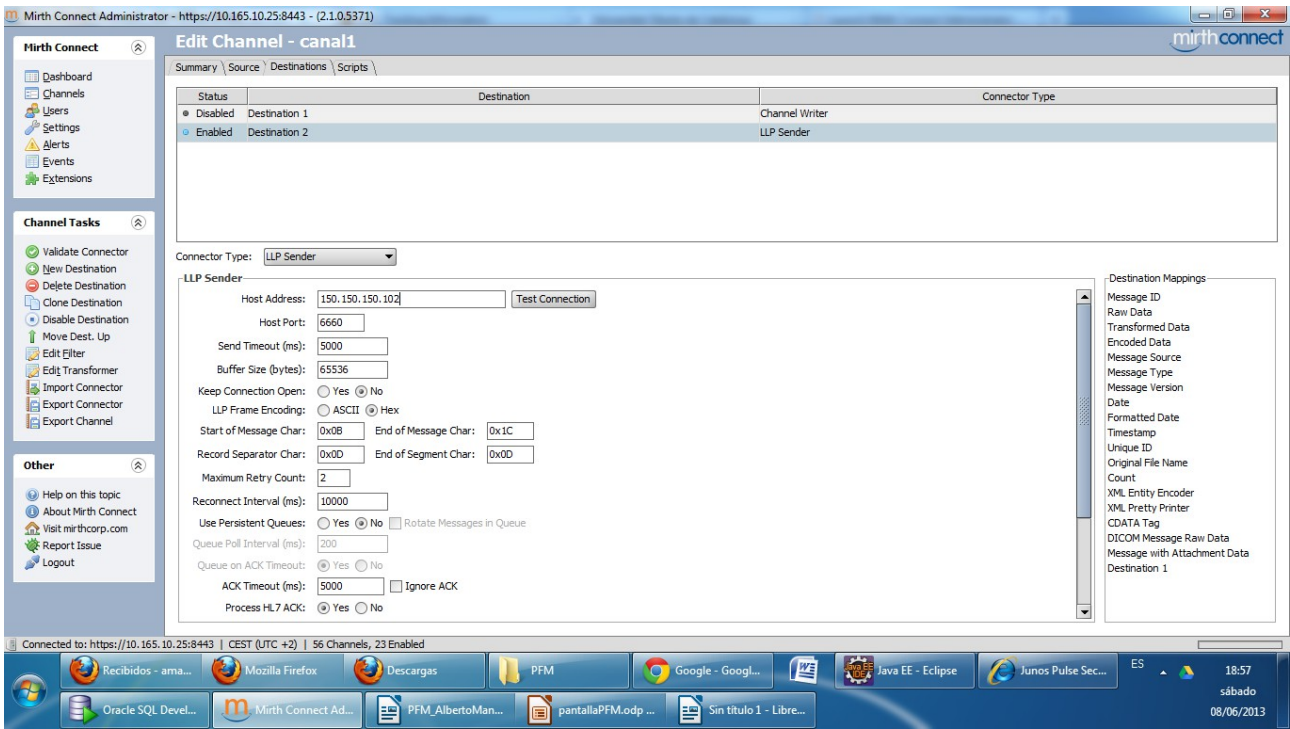
Para el entorno de desarrollo hemos utilizado un Eclipse versión Juno. A este eclipse le hemos instalado el plugin para GWT con la versión 2.5 del SDK y la 1.7 del AppEngine.

Para monitorizar la BBDD de Mirth y crear nuestras tablas con la metainformación hemos instalado Oracle SQL Developer v3.0.4.

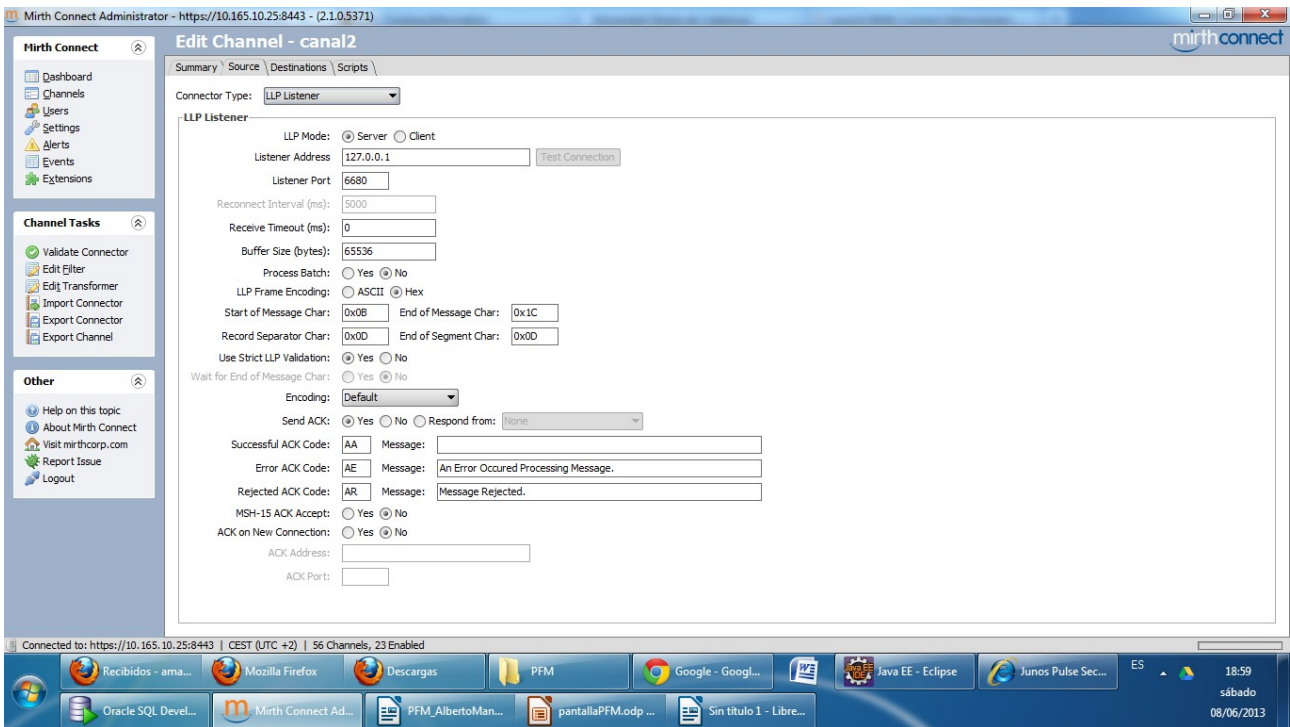
Hemos creado tres canales en Mirth para generar el entorno de pruebas. Cana 1 y Canal 2 que nos valdrán para ver el funcionamiento considerando ambos pertenecientes a la misma integración aplicaciónPrueba 1 y el Canal 3 que será un canal independiente, por tanto él será el único canal de la aplicaciónPrueba 2.

Los tres canales se han desarrollado para enviar y recibir h17.

Se ha creado un programa JAVA que pueda enviar mensajes vía socket para cargar de mensajería los canales.



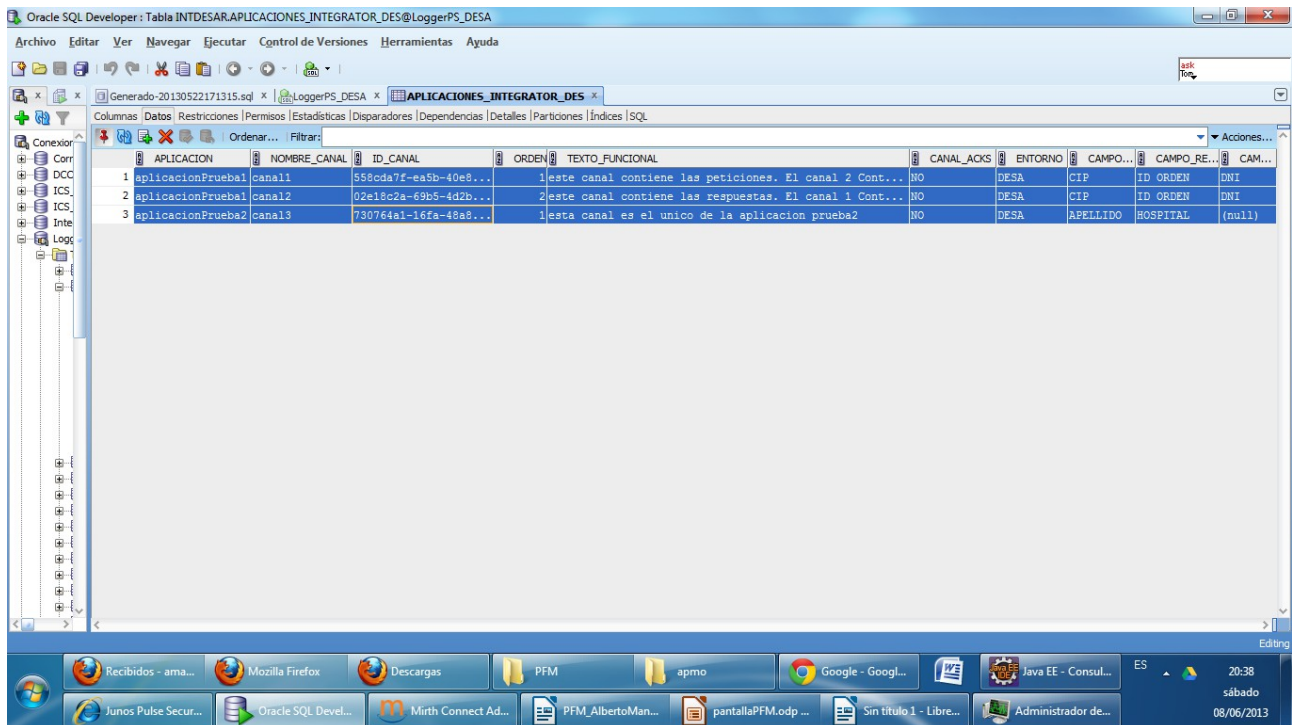
Img. Detalle del destino Socket del Canal1



Img Detalle del Source del Canal2.

Tabla con Metainformación de los canales:

Como hemos mencionado en la fase de definición de la solución, hemos de rellenar las tablas con la metainformación asociada a los canales Mirth. En primer lugar creamos las 3 tablas que se definen en el diseño técnico y posteriormente las rellenamos con los siguientes datos.



APLICACION	NOMBRE_CANAL	ID_CANAL	ORDEN	TEXTO_FUNCIONAL	CANAL_ACKS	ENTORNO	CAMPO...	CAMPO_RE...	CAM...
aplicacionPrueba1	canal1	558cda7f-ea5b-40e8...	1	este canal contiene las peticiones. El canal 2 Cont...	NO	DESA	CIP	ID ORDEN	DNI
aplicacionPrueba1	canal2	02e18c2a-69b5-4d2b...	2	este canal contiene las respuestas. El canal 1 Cont...	NO	DESA	CIP	ID ORDEN	DNI
aplicacionPrueba2	canal3	730764a1-16fa-48a8...	1	esta canal es el unico de la aplicacion prueba2	NO	DESA	APELLIDO	HOSPITAL	(null)

3.4 Implementación de Servicios y Capa de Presentación:

Estos dos objetivos se desarrollaron secuencialmente, en primer lugar se crearon los servicios a partir de las interfaces y una vez estuvieron construidos y con la certeza de que los procesos realizaban la funcionalidad definida se pasó a desarrollar la capa de presentación.

Pasamos a enumerar algunos de los métodos (obviando los métodos de los servicios que ya definimos en la definición de interfaces) más importantes que se han desarrollado en la capa de negocio y en la de BBDD en el lado del servidor.

```
public void connectEntornos(String[] accesoEntorno){}
```

Este método recibe un array de String de 3 elementos con la url de acceso a bbdd que corresponde con el entorno seleccionado para realizar la consulta y se comunica con la capa de BBDD para genera la conexión con ella ConnectDDBB.

Como se puede comprobar en la imagen se han creado 3 entradas, una para cada canal desarrollado para las pruebas con la metainformación. Así el canal1 y el canal2 están asociados a la misma aplicación, el orden para mostrar la información sería, en primer lugar los mensajes que pasen por el canal 1 y posteriormente los que pasen por el canal 2 y tienen los 3 mismos campos claves, ya que los campos claves definen valores de búsqueda para todo el ciclo de mensajería. En este caso se han definido como campos claves el DIP, el ID Orden y el DNI.

El canal 3 en cambio está asociado con la aplicación prueba 2.

En las 3 tuplas el entorno es DESA, ya que en desarrollo solo existe una instancia de Mirth y esa es la tabla correspondiente a la metainformación de desarrollo.

Como métodos para gestionar la conexión desde la capa de BBDD

```
public void connectionClose() throws Exception{  
public void connect() throws Exception{  
public void getConnection() throws Exception{
```

El método de la capa de BBDD dedicado a realizar la consulta es el siguiente:

```
public ArrayList<ResultadosBusquedaCanal>  
getResultadosBusquedaCanal(ArrayList<ResultadosBusquedaCanal> resultados, String[]  
campo, String fechaInicio, String fechaFin) throws Exception{
```

Otros métodos nos permiten identificar el CIP o el nombre y apellidos del paciente en los mensajes a partir del standar HL7 tanto en XML como en ER7. Como por ejemplo

```
public String getPacienteXMLType(String raw_data){  
public String getCIPXMLType(String raw_data){  
public String getPacienteER7Type(String[] contenidoEntreBarras){  
public String getCIPER7Type(String[] contenidoEntreBarras){
```

Con respecto a la capa de presentación. Tenemos una clases principal llamada ConsultaIncidencias.java donde es encuentra el código de la pantalla.

El método principal será el siguiente:

```
public void onModuleLoad() {
```

```

alturaNavegador = Window.getClientHeight();
anchoNavegador = Window.getClientWidth();
//RootPanel.get().add(cargando, (anchoNavegador/2)-
(cargando.getWidth()), alturaNavegador/2);
RootPanel.get().add(cargando, (anchoNavegador/2)-65, alturaNavegador/2);
cargando.setVisible(true);
this.service = (MessagesServiceAsync) GWT
.create(MessagesService.class);
ServiceDefTarget serviceDef = (ServiceDefTarget) service;
serviceDef.setServiceEntryPoint(GWT.getModuleBaseURL()
+ "messageService");
service.getAplicaciones(new AplicacionesCallback(){
public void onSuccess(java.util.ArrayList<Aplicacion> result) {

    aplicaciones = result;
    cargando.setVisible(false);

    // construimos la pagina una vez obtenidas las aplicaciones
    // creamos panel global
    panelVGlobal = new VerticalPanel();
    //panelVGlobal.setBorderWidth(5);
    panelVGlobal.setStylePrimaryName("panelGlobal");
    RootPanel.get().add(panelVGlobal, 10, 10);

    panelVGlobal.setSize("98%", "98%");

    panelHBusqueda();
    panelVResultadosBusqueda();

};
public void onFailure(Throwable caught) {

    Window.alert(caught.getMessage());
    cargando.setVisible(false);
}
});
}
}

```

Como resumen de su funcionamiento, secuencialmente:

- 1.- Define tamaño de la pantalla
- 2.- Muestra una barra de carga
- 3.- Llama al servicio que devuelve las aplicaciones y la información existente en las tablas de metainformación.
- 4.- Si el resultado es satisfactorio, desactiva la barra de carga
- 5.- Carga los dos paneles principales, el primero el de los campos para realizar la búsqueda y el segundo el de resultados con los siguientes métodos:

```

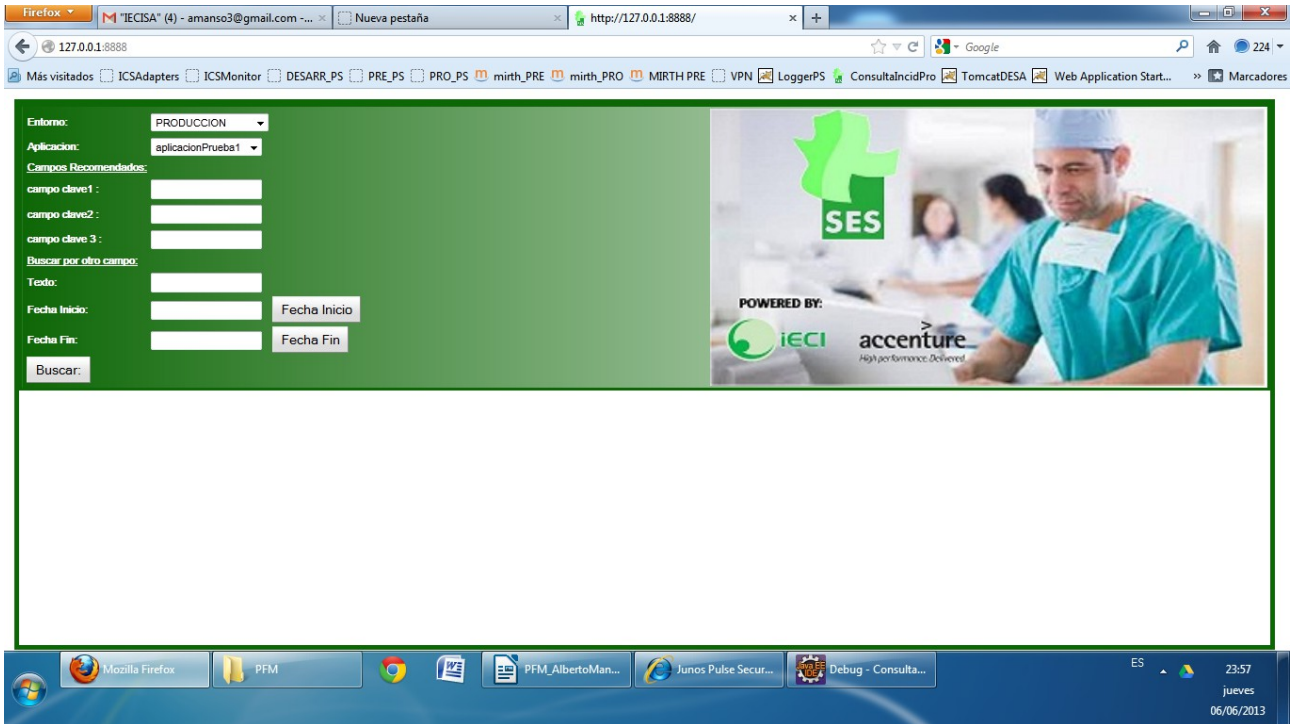
panelHBusqueda();
panelVResultadosBusqueda();

```

Nota* Una vez cargado el panel de búsqueda al pulsar el botón buscar llama al servicio que

devuelve los resultados de la búsqueda con la información que se haya rellenado para la consulta.

Una vez terminado el desarrollo la aplicación la pantalla se mostraba con la siguiente apariencia



3.5 Batería de Pruebas.

Se definieron las siguientes pruebas para comprobar la funcionalidad de la aplicación

1.- Envío de el mismo mensajes para el canal 1 (aplicación Prueba1) y el canal 3 (apliacionPrueba2)

Realizamos la búsqueda por CIP seleccionando la aplicación Prueba1, sólo aparece como resultado el mensaje del canal 1. No aparece ningún resultado para el canal 3 y aparece el canal2 pero sin ningún mensaje.

Realizamos la búsqueda por CIP seleccionando la aplicación Prueba2, sólo aparece como resultado el mensaje del canal 3. No aparece ningún resultado para el canal 1 ni canal2

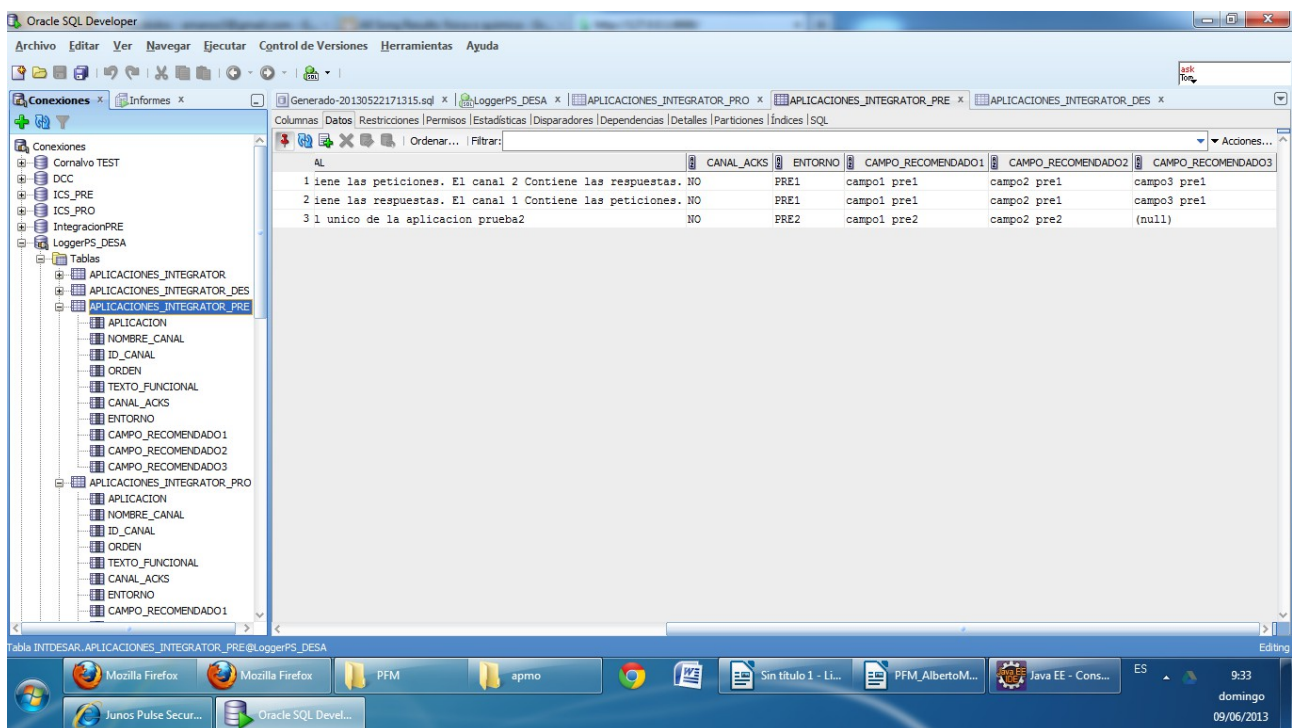
Por tanto concluimos que la aplicación realiza bien la identificación de la mensajería por aplicación y canal.

2.- Enviamos otro mensaje con el mismo CIP al canal 2.

Realizamos la búsqueda por CIP en la aplicación Prueba2 y obtenemos como resultado los mensajes del canal 1 y 2. Por tanto concluimos que la aplicación realiza bien la vinculación de los mensajes por los campos claves de varios canales para la misma aplicación.

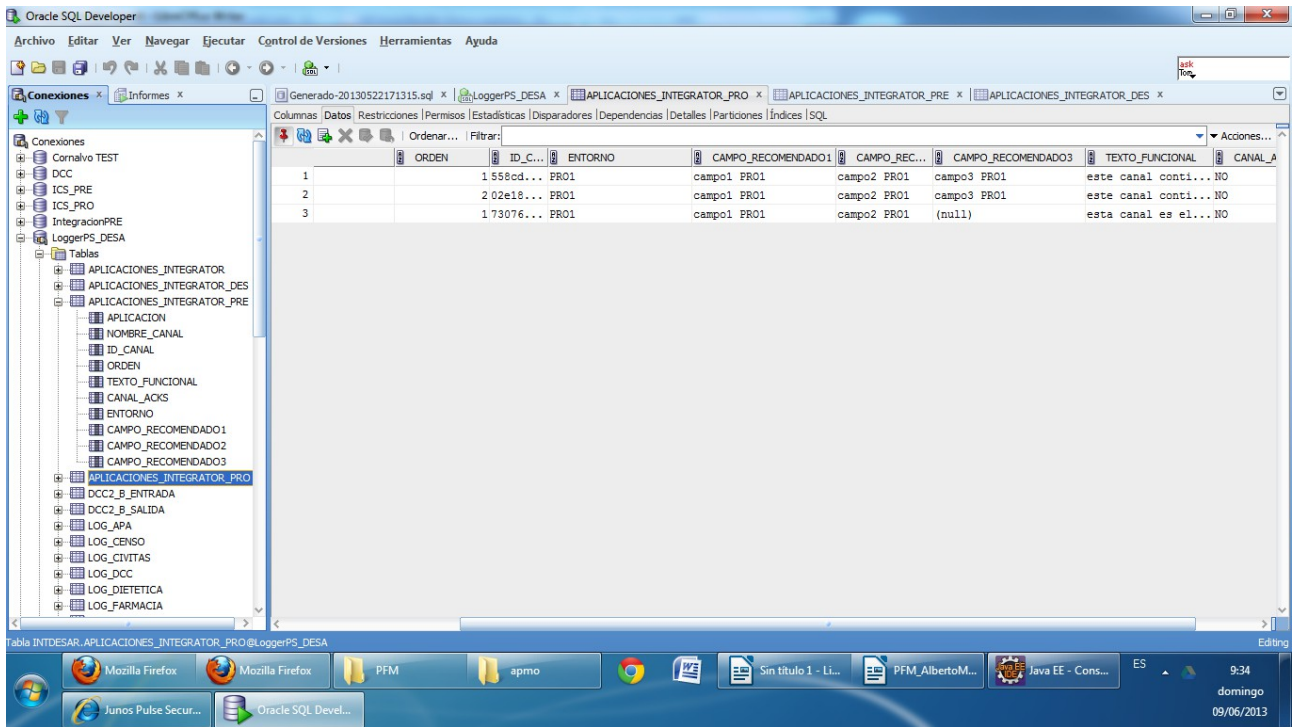
3.- Al no poder realizar pruebas en los entornos de producción y preproducción al ser sensibles y sólo disponer de una instancia de Mirth en Desarrollo simulamos los distintos entornos de la siguiente manera:

En los ficheros de configuración donde se indican las urls de los entornos ponemos en todos la url, usuario y password de Desarrollo. Así el programa identificará 3 entornos distintos con varias instancias pero que apuntan siempre a Desarrollo que es donde podemos probar. Para que sean visibles diferencias entre entornos, además de los nombres de las aplicaciones cambiamos el nombre con el que se mostrarán los campos claves, así seremos conscientes de que se ha recuperado la información correcta porque al cambiar de entorno en el combo de entorno y cargarse las aplicaciones, que siempre son las mismas, las que están en desarrollo, encontraremos nombres de campos clave distintos.



AL	CANAL_ACKS	ENTORNO	CAMPO_RECOMENDADO1	CAMPO_RECOMENDADO2	CAMPO_RECOMENDADO3
1 iene las peticiones. El canal 2 Contiene las respuestas.	NO	PRE1	campo1 pre1	campo2 pre1	campo3 pre1
2 iene las respuestas. El canal 1 Contiene las peticiones.	NO	PRE1	campo1 pre1	campo2 pre1	campo3 pre1
3 1 unico de la aplicacion prueba2	NO	PRE2	campo1 pre2	campo2 pre2	(null)

Img. Tabla metainformación entorno preproduccion.



Img tabla metainformacion entorno producción.

Cambiamos al entorno de producción y vemos que actualiza la información correctamente. Realizamos una búsqueda y el resultado es el correcto y el mismo que obtuvimos cuando teníamos el entorno de desarrollo seleccionado ya que los canales son los mismos y la mensajería también.

4.- Modificamos también el campo entorno de la BBDD de metainformación. Así comprobaremos que el combo entorno carga las aplicaciones de todas las instancias de Mirth en ese entorno. Comprobamos y los resultados son satisfactorios.

5.- Enviamos mensajería con distintos valores de id y buscamos por ellos. Comprobamos que sólo aparecen los resultados SQL correctos.

6.- Enviamos mensajería en distintos días para el mismo id y buscamos con un rango de fechas que abarque el envío de esa mensajería. Comprobamos que aparecen todos los mensajes, con el mismo id. ordenados correctamente.

7.- Buscamos ahora por un rango de fechas que sólo abarca un día. Comprobamos que sólo aparece la mensajería referente a ese día, a pesar de que en BBDD existen mensajes con el mismo id. Queda comprobado por tanto el funcionamiento de la búsqueda por fechas.

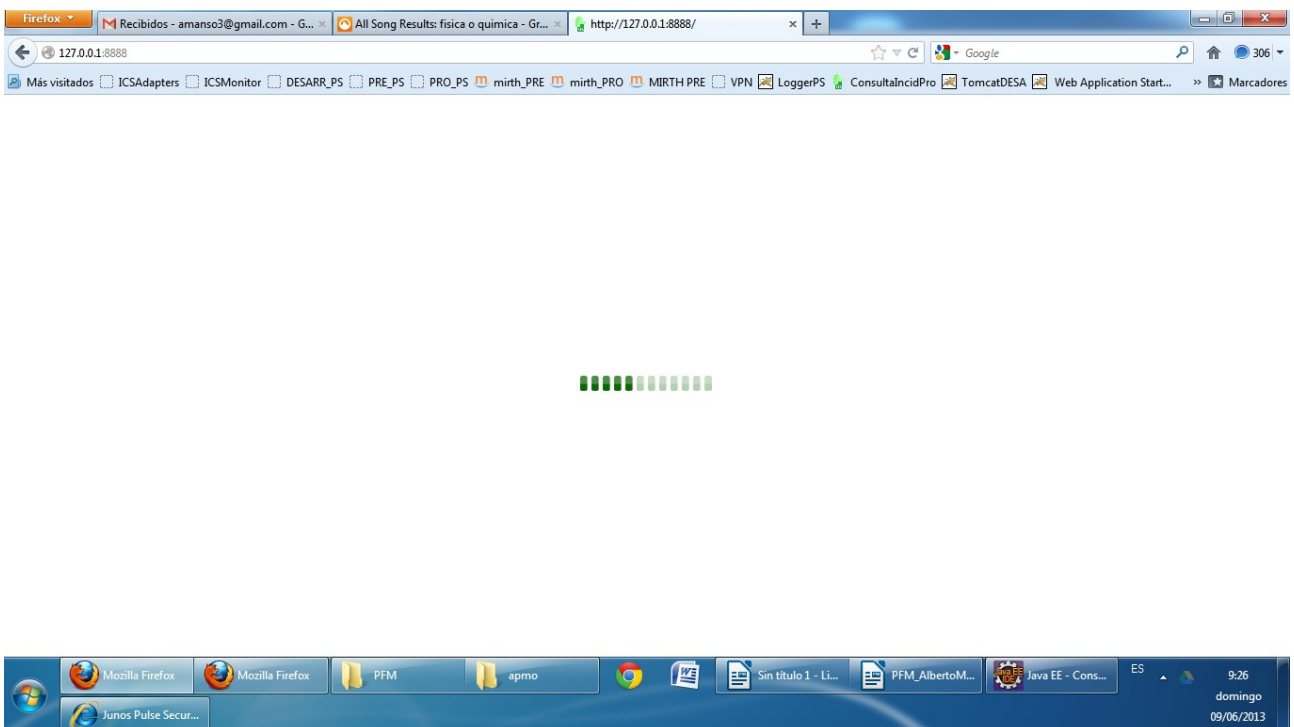
8.- Realizamos búsquedas por cada uno de los campos claves y por la caja “buscar por otro texto”.

9.- Enviamos mensajes con los destinos caídos para que provoquen error. Buscamos la mensajería y comprobamos que el estado que devuelve es Error.

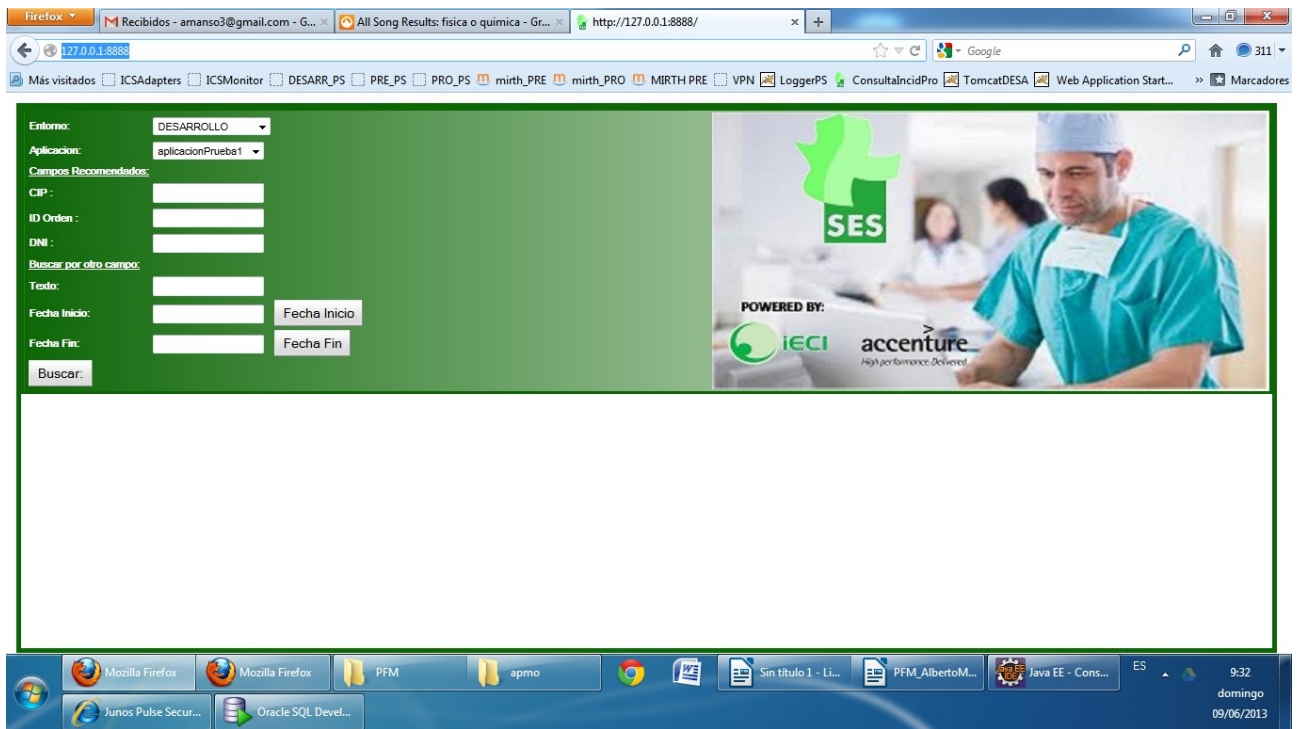
4 Resultados

El objetivo final de proyecto es conseguir una aplicación que nos provea de las funcionalidades identificadas en el apartado de requisitos. Esta aplicación debe ser utilizada por operadores que no tengan un excesivo conocimiento del funcionamiento de Mirth o las integraciones. Por tanto una buena forma de exponer los resultados es generar una pequeña guía de uso que muestre el comportamiento de la misma y el modo más adecuado de utilizarla.

1.- Acceso a la aplicación: Introducir en FireFox (la web está optimizada este navegador) la url de la aplicación. Encontraremos la siguiente pantalla de carga.

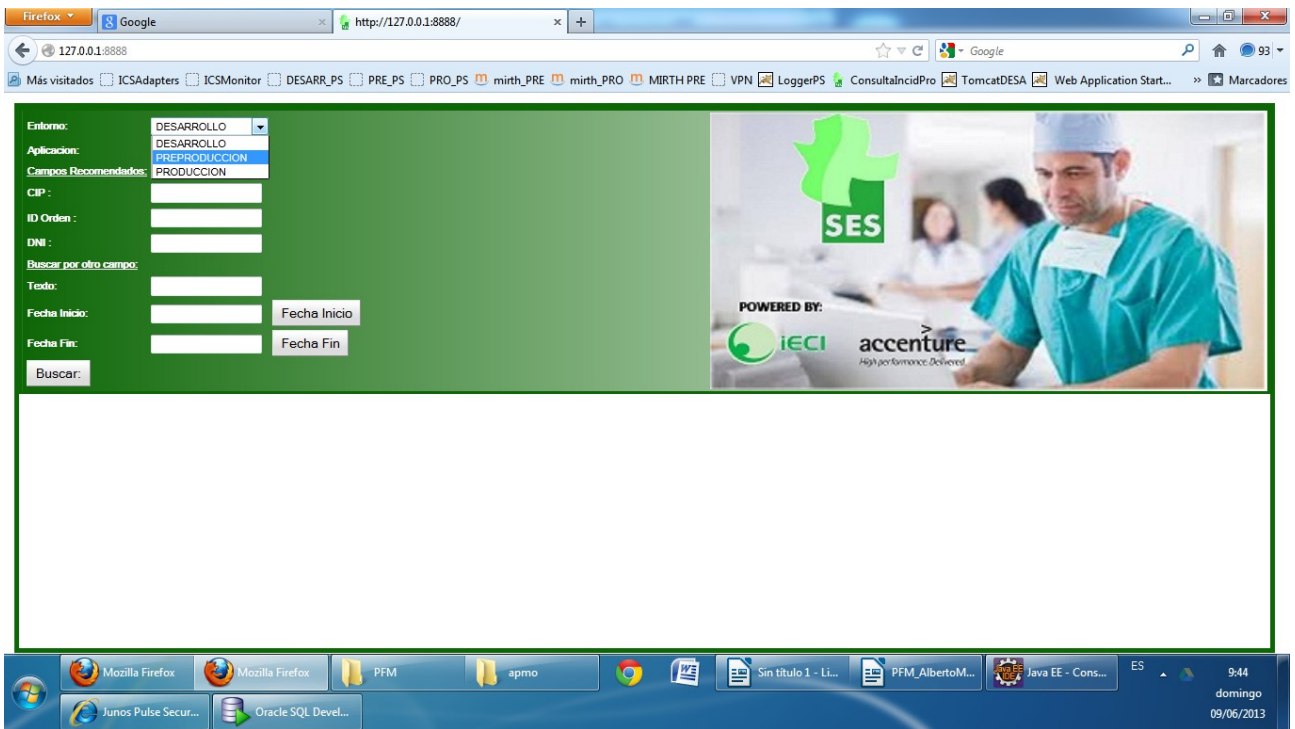


2.- Una vez recuperada la información de las aplicaciones obtendremos la siguiente pantalla

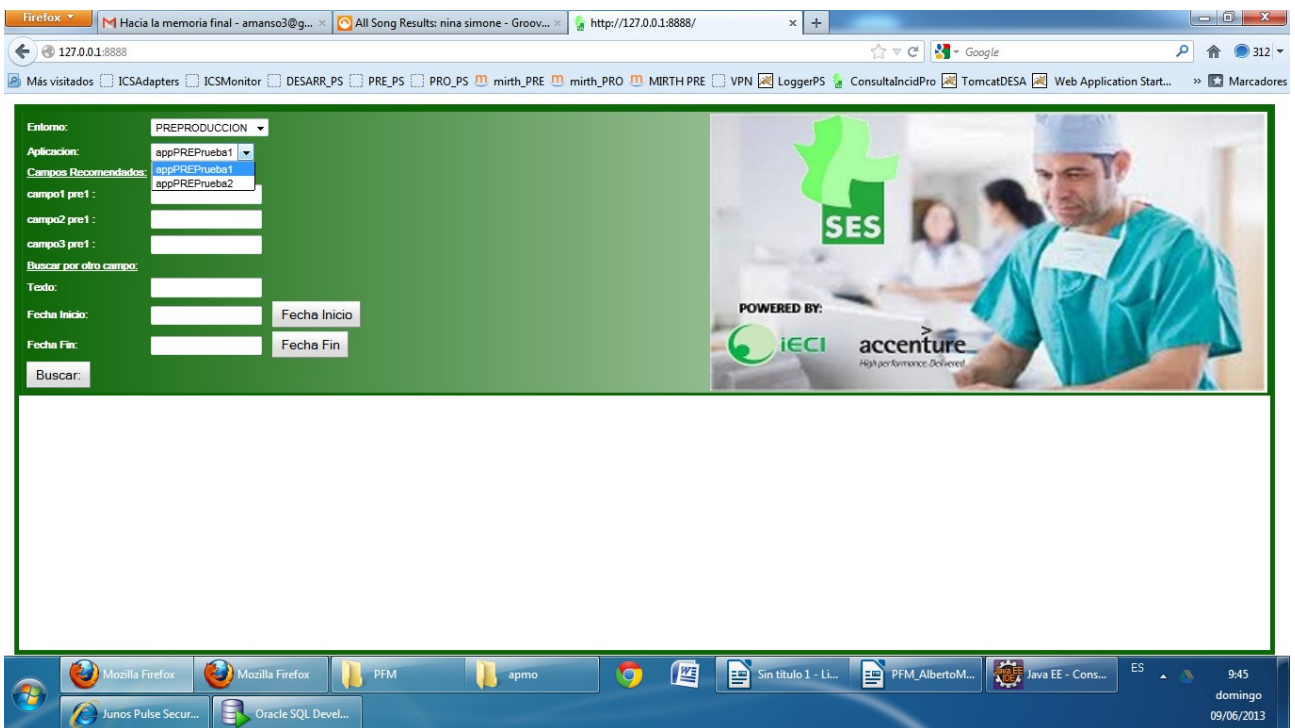


3.- Una vez estamos situados en la pantalla de búsqueda. Vemos que se carga el entorno de Desarrollo por defecto (en producción se cargará el entorno de producción por defecto). Y nos selecciona una aplicación por defecto.

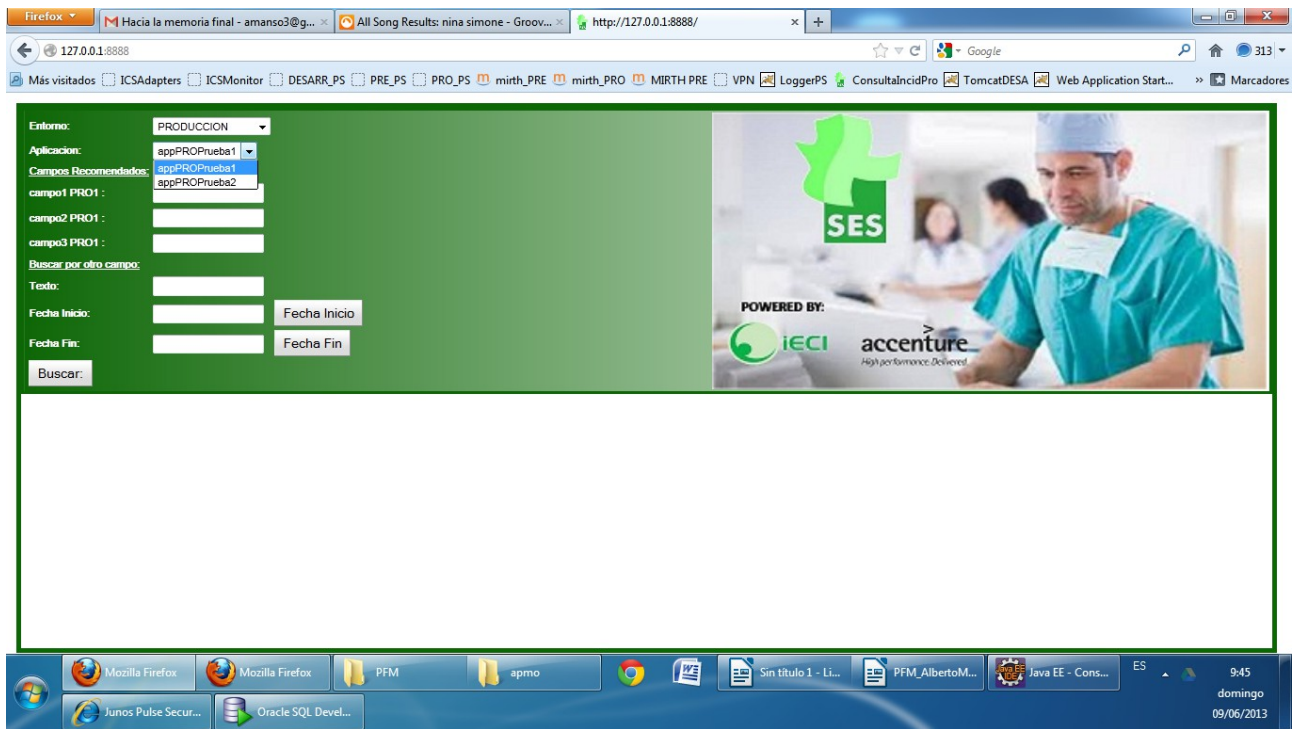
Jugando a cambiar en el combo de entorno veremos que el combo de aplicaciones se actualiza con las aplicaciones que están alojadas en cada entorno. Podremos observarlo desplegando el combo de aplicaciones una vez seleccionemos un entorno y mirando el valor de los campos claves.



Img. Opciones combo entorno.



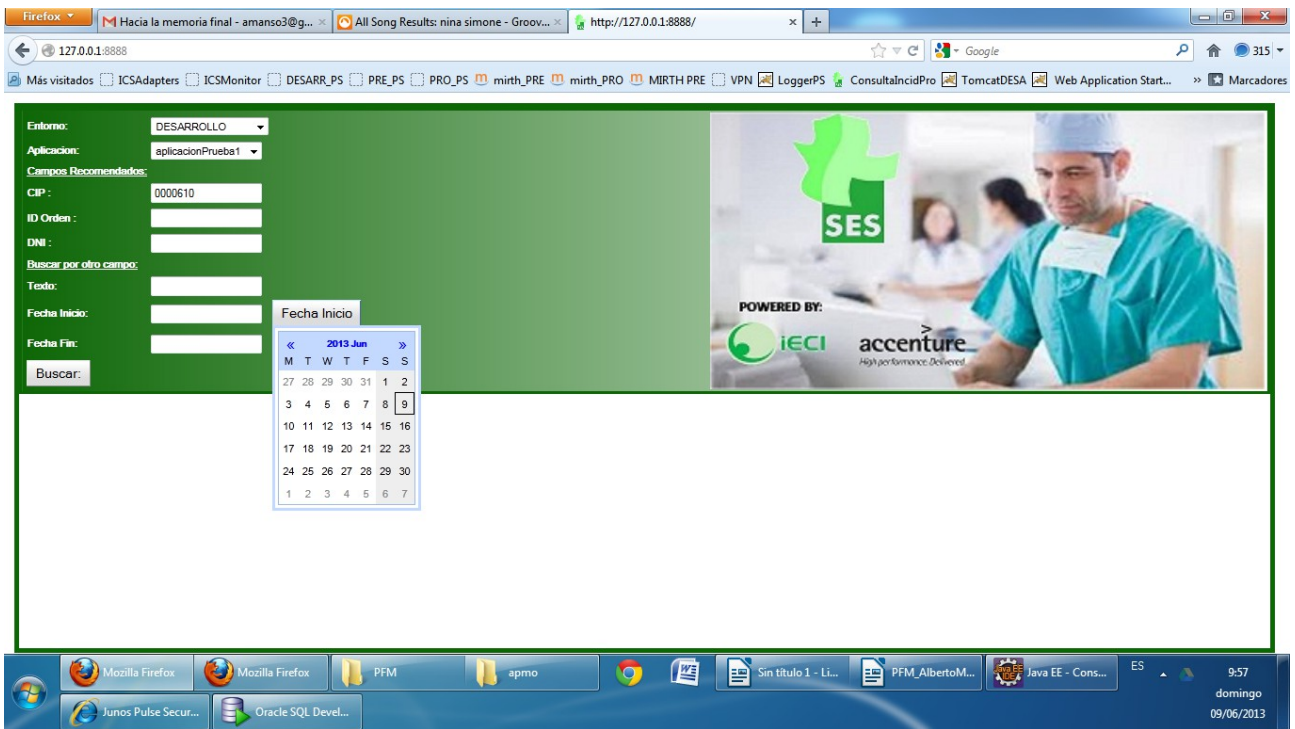
Img. Opciones combo aplicaciones.



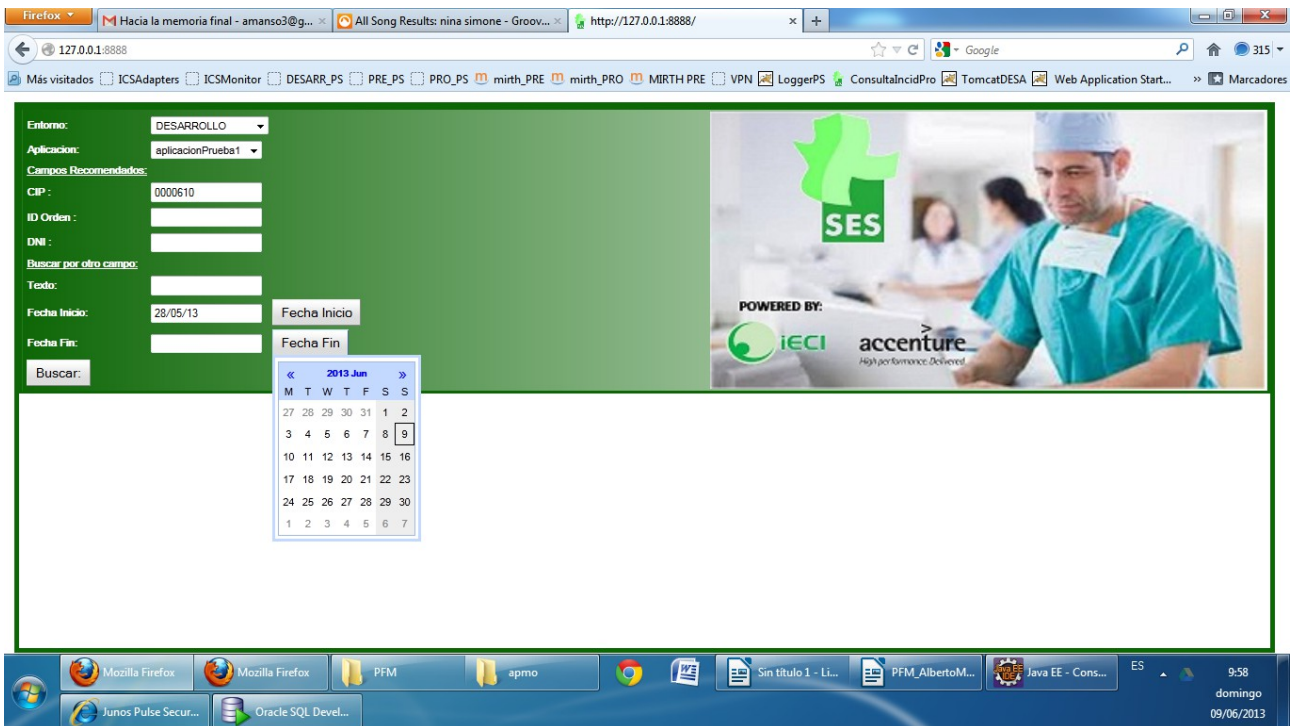
4.- Una vez seleccionados el entorno sobre el que queremos realizar la consulta y la aplicación de la que queremos rescatar información. Rellenaremos los campos de la búsqueda asociados a la mensajería. Dentro de los campos recomendados tenemos como máximo 3 campos claves. Estos campos nos orientan de por qué tipo de información es más recomendable realizar la búsqueda. Sabemos que si buscamos por esos campos encontraremos todo el flujo de mensajería a través de todos los canales que estén relacionados con la incidencia que pretendemos resolver. Si utilizamos otros valores que no sean claves para esa aplicación. No tenemos la garantía de recuperar los mensajes de todos los canales, ya que es posible que si por ejemplo, el apellido no sea un campo clave se informe un canal, pero en otro relacionado con la misma aplicación no se informe en el mensaje y por tanto no nos aparecerá ningún resultado relacionado con un mensaje que sí tiene importancia para el análisis de la incidencia.

5.- Si no tenemos valores de ningún campo clave para buscar, podemos utilizar la caja de texto de la sección “buscar por otro campo” . En ese caso nos aparecerá la mensajería que contenga esa información igualmente ordenada y coherente pero deberíamos realizar una segunda búsqueda. Una vez obtenido el primer resultado, examinar el/los mensajes, identificar un campo clave y realizar la búsqueda con ese valor para tener la garantía de que los resultados son fiables.

6.- Tan sólo nos quedaría acotar la búsqueda por fechas. Este paso es muy importante para evitar consultas pesadas y masivas en la bbdd donde Mirth almacena los mensajes.



Img. Boton calendario fecha inicio.



Img. Boton calendario Fecha Fin.

7.- Ya podemos pulsar el botón buscar para rescatar la información. Obtendremos algo así.

The screenshot displays a web application interface for Mirth Connect. The interface includes a search form on the left with fields for 'campo clave1', 'campo clave2', 'campo clave3', 'Buscar por otro campo', 'Trado', 'Fecha Inicio', and 'Fecha Fin'. The main content area shows two data tables, one for 'PASO: 1 canal1' and another for 'PASO: 2 canal2'. Both tables have columns for 'ORDEN', 'PACIENTE', 'CIP', 'FECHA CREACION', 'TIPO_MSG', 'DESTINO', 'ESTADO MIRTH', 'ESTADO', 'XML/ER7/...', and 'VER MENSAJES'. The status 'TRANSFORMED' is shown for the first step and 'SENT' for the second step. The interface also features a header with logos for SES, IECI, and accenture.

ORDEN	PACIENTE	CIP	FECHA CREACION	TIPO_MSG	DESTINO	ESTADO MIRTH	ESTADO	XML/ER7/...	VER MENSAJES
1	VICENTA RODRIGUEZ	RDPN400161913019	29/05/2013 10:15:37	ORM-001	INTEGRACION	TRANSFORMED	EL MENSAJE RECIBIDO CORRECTAMENTE POR INTEGRACION	HL7V2	ver mensaje
2	VICENTA RODRIGUEZ	RDPN400161913019	29/05/2013 10:15:37	ORM-001	Destination 1	SENT	OK: MENSAJE ENVIADO CORRECTAMENTE AL DESTINO	HL7V2	ver mensaje

ORDEN	PACIENTE	CIP	FECHA CREACION	TIPO_MSG	DESTINO	ESTADO MIRTH	ESTADO	XML/ER7/...	VER MENSAJES
1	IGLESIAS MANSO	MNRD690964913^^^	29/05/2013 10:07:25	ORU-R01	INTEGRACION	TRANSFORMED	EL MENSAJE RECIBIDO CORRECTAMENTE POR INTEGRACION	HL7V2	ver mensaje
2	IGLESIAS MANSO	MNRD690964913^^^	29/05/2013 10:07:25	ORU-R01	Destination 1	SENT	OK: MENSAJE ENVIADO CORRECTAMENTE AL DESTINO	HL7V2	ver mensaje

Para interpretar los resultados es necesario conocer lo siguiente:

Obtendremos la información agrupada por “pasos”. Eso indica que el orden correcto en el que se debe recibir mensajería es el dispuesto ahí. Primero se debería recibir mensajería por el canal x, después por el canal y etc. Así podremos detectar si ha habido un desfase en el orden en que los sistemas han enviado mensajería.

Podemos obtener información funcional sobre el paso pulsando sobre el texto “paso n: nombreCanal”.

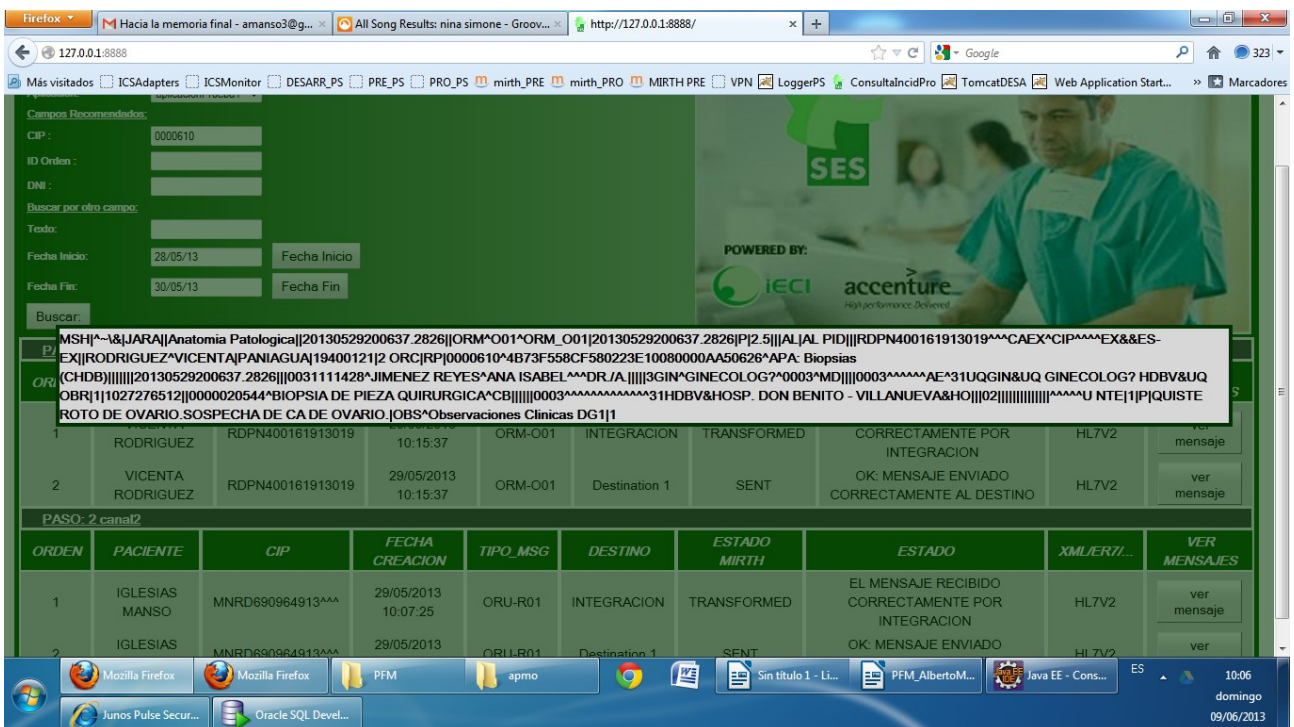


Img. Info sobre el paso1

Dentro de cada paso, los mensajes vendrán también ordenados por fechas y orden de recepción. Esta información nos la proveerán las columnas Orden y Fecha de Creación.

Si la mensajería sigue algún estándar como HL7 o el definido por SAP para este proyecto para la ubicación del nombre y apellidos del paciente y CIP, aparecerán estos valores en las columnas 2 y 3. Si no, aparecerá un mensaje que indica que no es posible extraer esa información y habrá que buscar esos valores en el mensaje.

El botón mensaje muestra el mensaje que se trata en ese instante.



Img. Muestra del mensaje.

Por cada mensaje que atraviese la integración debemos tener al menos dos filas:

- una fila que exprese la recepción del mensaje por parte de integración. Esta fila tendrá en la columna destino el valor “INTEGRACION”
- una fila por destino al que deba enviar el mensaje integración. Estas filas deben tener en la columna destino el nombre del destino al que se envía el mensaje.

ORDEN	PACIENTE	CIP	FECHA CREACION	TIPO_MSG	DESTINO	ESTADO MIRTH	ESTADO	XML/ER7...	VER MENSAJES
1	VICENTA RODRIGUEZ	RDPN400161913019	29/05/2013 10:15:37	ORM-001	INTEGRACION	TRANSFORMED	EL MENSAJE RECIBIDO CORRECTAMENTE POR INTEGRACION	HL7V2	ver mensaje
2	VICENTA RODRIGUEZ	RDPN400161913019	29/05/2013 10:15:37	ORM-001	Destination 1	SENT	OK: MENSAJE ENVIADO CORRECTAMENTE AL DESTINO	HL7V2	ver mensaje

La columna TIPO_MSG nos indica el tipo de mensaje en cada punto del flujo. Junto con la columna XML/ER7 que nos indica la forma en la que se comunica, como XML, ER7...

La columna ESTADO MIRTH. Tendrá varios posibles valores TRANSFORMED, FILTERED, SENT o ERROR en función del estado del mensaje en ese punto.

La columna ESTADO. Hace un análisis más descriptivo del estado del flujo en ese punto indicando si el mensaje ha sido recibido correctamente, enviado correctamente al destino, filtrado correctamente o una descripción del error en su caso.

1	VICENTA RODRIGUEZ	RDPN400161913019	09/06/2013 10:19:58	ORM-001	INTEGRACION	TRANSFORMED	EL MENSAJE RECIBIDO CORRECTAMENTE POR INTEGRACION	HL7V2	ver mensaje
2	VICENTA RODRIGUEZ	RDPN400161913019	09/06/2013 10:19:58	ORM-001	Destination 2	ERROR	ERROR: EL MENSAJE HA PROVOCADO UN ERROR	HL7V2	ver mensaje

Nota* No se pueden realizar búsquedas por más de un campo clave, la aplicación tan sólo recupera uno de los valores rellenos tanto en los campos claves como en la búsqueda por otro texto.

5 Conclusiones y Lecciones aprendidas

El resultado del proyecto ha sido en líneas generales satisfactorio. Los objetivos han sido siendo alcanzados y el resultado final de la aplicación consigue aportar el valor que pretendía.

Con respecto a las decisiones que se han ido tomando y la estructura de desarrollo del proyecto hemos aprendido la importancia de las validaciones y los puntos de retorno. La fase de validación porque pone el trabajo en conocimiento de más de un recurso y las aportaciones, las diferencias de perspectiva y de enfoque enriquecen el proyecto. En esta línea y en proyectos que implicaran a más recursos sería conveniente identificar procesos como revisiones entre iguales (peer reviews) en las que los recursos del mismo perfil revisaran el trabajo de sus compañeros. Así se podrían detectar fallos y se aportaría un punto de calidad al código. Con respecto a los puntos de retorno, tanto en la fase de diseño como en la de desarrollo se han detectado situaciones que invitaban a retomar la fase anterior. Así se ha reanalizado y rediseñado la aplicación a partir del conocimiento que ha aportado la fase siguiente, gracias a ello el proceso ha sido continuo y parte de un todo y creemos haber conseguido que la solución sea más robusta y coherente. En la mayoría de las ocasiones era posible corregir la situación en la fase donde se detectaban pero retornar al análisis o al diseño y reconceptualizar la funcionalidad mejoraba la sencillez y coherencia del desarrollo y evitó otros errores que pudieran haberse derivado de esos “apaños” que inicialmente eran viables en la fase donde nos encontrábamos. Este aspecto lo hemos visto claramente reflejado al plantearnos mejoras para un futuro como la posibilidad de mejorar el análisis de la información rescatada para mostrar resultados más transparentes. Por tanto entendemos que esta forma de conceptualizar el proyecto como algo abierto y cíclico ha mejorado la escalabilidad del mismo.

En cuanto al nivel de complejidad, el objetivo que menos esfuerzo ha supuesto ha sido el análisis ya que poseíamos ya bastante conocimiento del motor de integración. En cuanto al de mayor esfuerzo ha sido el diseño ya que implicaba formarnos sobre el framework y conceptualizar una aplicación cuando aún no habíamos desarrollado nunca en ese entorno, por tanto ha supuesto un nivel de abstracción más complejo del inicialmente planificado.

Con respecto a los objetivos tecnológicos, una de las motivaciones mayores a nivel personal era conseguir formación sobre el framework GWT. El objetivo se ha cumplido, a medias, debida al abanico de posibilidades que ofrece el framework y al alcance necesario para la parte web. Ha sido una toma de contacto gratificante pero nos quedamos con la sensación de que existe gran parte del framework que no hemos podido abarcar. En cualquier caso, valoramos lo aprendido como positivo ya que nos ha mostrado un entorno de desarrollo con muchas posibilidades, facilidad para el debugeo, la trazabilidad y unas librerías para la implementación de objetos relacionados con la programación web completa y atractiva para el usuario final.

Otro aspecto a mejorar podría ser la compatibilidad con distintos navegadores. La aplicación está optimizada para Firefox pero algunos componentes como la imagen o los marcos se descolocan en Explorer y Chrome.

En cuanto a la gestión de un proyecto en capas. Podríamos identificar algunas pegadas, ya que quizás hemos introducido demasiada lógica en los métodos de acceso a BBDD que deberían ir en los servicios. Como punto positivo consideramos hemos conseguido una separación bastante razonable entre la capa de presentación y la de negocio.

Como punto final comentar la evolución que pretendemos abordar para la aplicación. Hemos detectado que a pesar de intentar que la aplicación sea ágil, cuando las cargas de mensajería son

altas las consultas tardan bastante tiempo. El motivo es que la consulta hace un “like” dentro de los mensajes de la BBDD de Mirth. Esta consulta es pesada y lenta pero las soluciones utilizando la BBDD de Mirth son escasas ya que el registro en BBDD es transparente al desarrollo de canales y las implicaciones de tocar el código fuente de Mirth en esos aspectos no son aconsejables. Estamos conceptualizando una solución que permita utilizar la aplicación y elegir si se consulta en la BBDD de Mirth o en otra BBDD indexada que permita acceder rápidamente a la información.

Esto se conseguiría utilizando log4j en los canales Mirth para registrar la información en los distintos puntos del desarrollo extrayendo ya ahí los campos claves que interesan. Así tendríamos ficheros con la información ya estructurada. Estos ficheros se cargarían en la nueva BBDD a través de un canal Mirth encargado de leerlos y enviarlos a la BBDD. Mirth es bastante flexible a la hora de planificar tareas de lectura de ficheros y cargas.

Esta solución esperamos que en el futuro nos permita utilizar la búsqueda en la BBDD de Mirth para incidencias que se produzcan en tiempo prácticamente real y la búsqueda en la otra BBDD mucho más rápida para incidencias más comunes que sean de fechas pasadas.

6 LICENCIA:

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover

Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly

and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a

Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.