



Interoperabilidad de los Sistemas Informáticos en el Área Salud

Memoria para optar al grado de
INGENIERO CIVIL INFORMÁTICO

POR

Elvis Ríos Melo

Departamento de Ingeniería Informática y Ciencias de la
Computación

Universidad de Concepción

Profesor Guía: Marcela Varas Contreras

Agosto 2018

Resumen

La integración de datos es el proceso que permite combinar datos heterogéneos de muchas fuentes diferentes en la forma y estructura de una única aplicación. Este proceso de integración de datos facilita que diferentes tipos de datos, tales como matrices de datos, documentos, tablas y otros, sean fusionados por usuarios, organizaciones y aplicaciones para un uso personal, de procesos de negocio o de funciones, permitiendo generar información útil y valiosa para el organismo que lo utiliza.

En esta memoria de título se hace uso de la integración de datos sobre un motor de integración, es decir, el cual nos permite la comunicación entre distintos sistemas. Haciendo efectivo este último y con ayuda de servicios en la nube se obtienen resultados de intercomunicación entre sistemas de un centro de salud, dando paso a la interoperabilidad.

1. Introducción

Como cualquier persona que visita a su médico puede haber notado, se han ido los días en que los médicos registran sus notas en papel. Es más probable que los médicos ingresen a la sala de examen con una computadora portátil que con papel y bolígrafo. Este cambio es el subproducto de los esfuerzos para mejorar los resultados de los pacientes, aumentar la eficiencia e impulsar la salud de la población. Impulsar estas mejoras ha creado muchas nuevas oportunidades de datos y desafíos.

Cuando un paciente se realiza un examen físico, el consultorio de su médico tiene un cuadro electrónico con información sobre su demografía (nombre, fecha de nacimiento, dirección, etc.), historial de atención médica y visita actual.

Los proveedores registran la información del paciente en diferentes plataformas de software. Cada una de estas plataformas puede tener diferentes implementaciones de estándares complejos de datos de atención médica. Además, cada sistema necesita comunicarse con un repositorio central, llamado intercambio de información de salud (HIS) para construir un registro clínico central y completo para cada paciente.

A continuación, dentro de este informe habrá una discusión bibliográfica, donde se mencionará API, web services y motores integración. Seguido a esto una descripción de los motores de integración y la elección de uno de ellos, para finalmente llegar al montaje y la configuración del motor escogido.

1.1 Objetivo General

El objetivo de esta memoria es desarrollar un prototipo para un centro de salud haciendo uso de la interoperabilidad por medio de un integrador de datos, permitiendo establecer comunicación entre algunos sistemas.

1.2 Objetivos Específicos

- Mejorar la integración los datos entre dos o más sistemas utilizados dentro del centro hospitalario, permitiendo un mejor acoplamiento entre los mismos por medio de una correcta comunicación.

- Disminuir la redigitación a la hora de hacer ingresos de pacientes o traslado de estos entre áreas del centro de salud.
- Aumentar la velocidad de respuesta a la hora de tomar una decisión, esto se reduce a tener una mejor asociación entre los datos. Con lo último mencionado, se evitaría recopilar la información (sistema por sistema) para luego generar una decisión.

1.3 Realidad Actual

El lugar de referencia a tener en esta memoria es el Hospital Guillermo Grant Benavente, que hoy en día cuenta con una unidad informática que se encarga tanto de desarrollo interno como de otros sistemas. Por otro lado, dentro del centro hospitalario existen varios fragmentos o secciones:

- **Sistemas Internos:** Estos son de desarrollo propio por la UI (Unidad Informática) del establecimiento de salud.
- **Sistemas Externos:** Está proporcionado tanto por el gobierno (RRHH, FINANZAS, RED MINSAL), como por el SSC (Servicio de Salud Concepción), quien brinda apoyo a todo centro hospitalario ubicado en el gran Concepción, además se subdivide en sistemas internos y externos dando lugar a un software llamado Sinetsur.
- **Sistemas Externos Dependientes:** Estos últimos son proporcionados por las tecnologías que utiliza el hospital, donde alguna maquinaria o elemento tenga como uso específico un sistema o software determinado.

Dentro del establecimiento de salud de la región del Biobío esta el área clínica que se compone de urgencias, centro de atención ambulatoria e internado. El registro de los pacientes se rige por una FCE (Ficha Clínica Electrónica), la cual contempla datos personales, procedimientos, atenciones, historial, etc. Esta ficha se encuentra almacenada en un sistema HIS (DB Health Care) y un sistema llamado Galileo.

Como se mencionaba anteriormente, el centro hospitalario posee una parte clínica, pero además posee un sector administrativo. Ambas, para poder operar, utilizan una amplia y variada gama de sistemas, propios, pagados y otorgados por el Estado. Estos sistemas al ser parte de estas áreas necesitan comunicarse entre ellos para compartir información.

Para estándares nacionales el hospital posee una buena infraestructura y buen equipamiento técnico, pero carece de interacción entre sistemas, y las pocas que existen son insostenibles en el tiempo y poco eficaces. La carencia de interacción lleva a una serie de problemas: **duplicación, redigitación, mayor volumen de datos, menor calidad de información, disminución en la eficiencia y la toma de decisiones.**

2. Discusión Bibliográfica

La integración de datos dentro del área de salud es algo relativamente nuevo dentro del país, pero llevándolo a horizontes vecinos se puede ver que la integración de los datos no solo puede ser llevada a cabo por medio de un motor de integración, sino que también se puede lograr por medio del uso de **API** de forma combinada, como también lo permite la mezcla de varios **Web Services** según corresponda a cada sistema. [15]

2.1 API

Una API (interfaz de programación de aplicaciones), es aquella que representa la capacidad de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa una función para conseguir abstracción en la programación, generalmente entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en brindar un conjunto de funciones de uso general. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio. Las API asimismo son abstractas: el software que proporciona una cierta API generalmente es llamado la implementación de esa API.

Igual que una interfaz de usuario permite la interacción y comunicación entre un software y una persona, una API facilita la relación entre dos aplicaciones para el intercambio de mensajes o datos. Un conjunto de funciones y procedimientos que ofrece una biblioteca para que otro software la utilice como capa de abstracción, un espacio de acceso e intercambio de información adicional en la parte superior. Así una se sirve de la información de la otra sin dejar de ser independientes. [17]

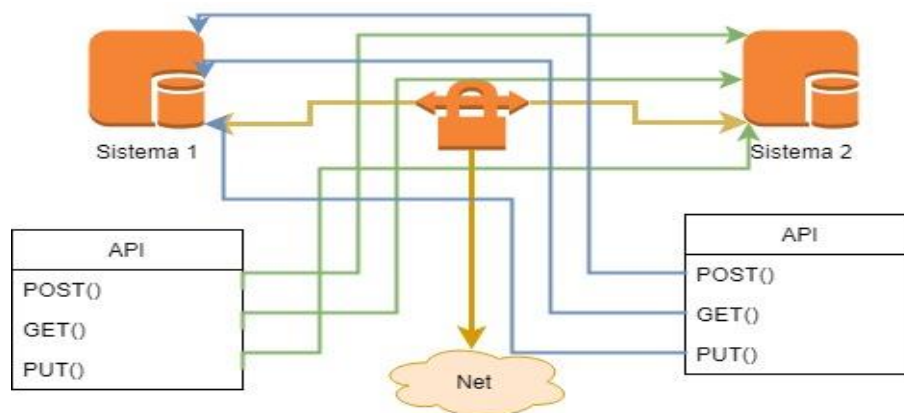


Figura 1: Representación de dos sistemas usando API entre sí.

En la **Figura 1** se tienen dos sistemas cada uno haciendo uso de API, ambos conectados a internet o red local y con la posibilidad de consumirse entre ellos por medio de operaciones CRUD (Create, Read, Update, Delete), las cuales permitirán ingresar, pedir, actualizar y borrar información entre los sistemas descritos.

2.2 Web Services

Un Servicio Web es una tecnología que utiliza una gama de protocolos y estándares, el cual sirve para intercambiar datos entre aplicaciones. Ya sea aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma (Multiplataforma), además pueden utilizar los servicios web para intercambiar datos en redes de ordenadores, como lo es Internet.

La interoperabilidad se consigue mediante la adopción de estándares abiertos. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

Como se menciona en el párrafo anterior y entrando en detalle se tiene que el WS-I, es la organización para la interoperabilidad de servicios web (Web Services Interoperability Organization), su objetivo es fomentar y promover la Interoperabilidad de Servicios Web [16].

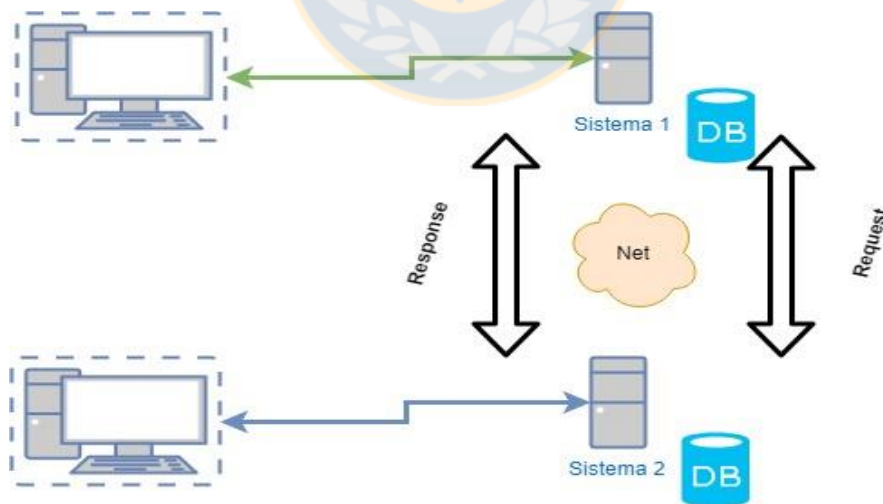


Figura 2: Representación de dos sistemas usando Web Services (response <-> request).

En la **Figura 2** se tienen dos sistemas que hacen uso de web services, donde estos por medio de internet o red local se comunican, para el intercambio de información entre el sistema 1

y 2 disponen de funciones particulares independientes, las que permitirán actualizar, mover, ingresar o borrar datos, según como se especifique en la misma.

2.3 Motores de Integración

Un Motor de Integración, es aquel como lo dice el nombre, un software, motor, aplicación que permite la integración de datos. Es decir, comunicación entre sistemas o bien dicho la interoperabilidad entre los sistemas. Además, es importante destacar que maneja varios estándares como protocolos de comunicación, y estos actúan como si fueran un intérprete que traduce y que hace que todos los sistemas se puedan comunicar entre sí.

Haciendo una analogía de lo planteado anteriormente. El descubrimiento de la piedra roseta permitió descifrar los jeroglíficos, es decir, el integrador de datos vendría siendo la piedra roseta y los sistemas los jeroglíficos. [17]

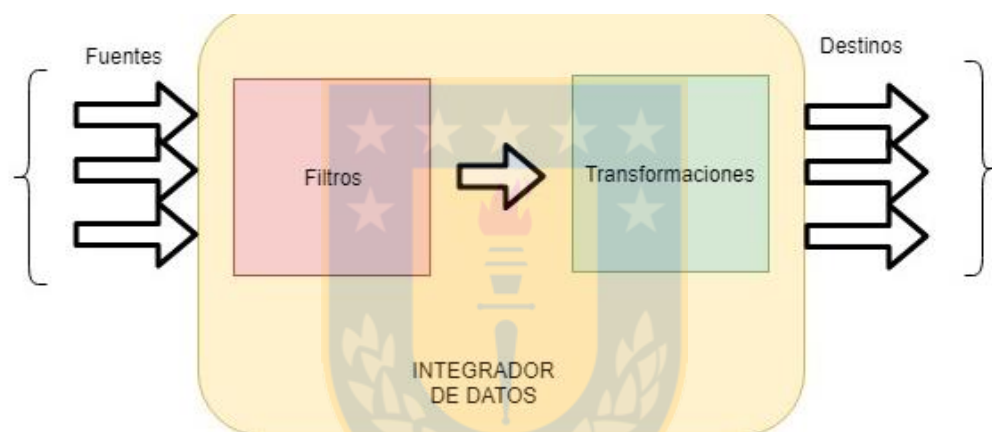


Figura 3: Representación de un motor de motor de integración.

En la **Figura 3** se tiene que un integrador de datos recibe información de una fuente (por ejemplo, sistema 1), luego esta información es filtrada, transformada a un formato aceptable para el sistema 2 (destino) y finalmente esta nueva codificación es comunicada por medio de algún protocolo de comunicación al destino respectivo.

2.4 Tabla Comparativa

Tomando la información recopilada y analizando cada descripción dada para API, Web Services y motores de integración en el punto anterior, se pueden comparar de acuerdo con las siguientes características para escoger una de las alternativas.

Característica	API	Web Services	Motor Integración
Permiten comunicación entre sistemas.	SI	SI	SI
Modificación código fuente de algún sistema.	SI	SI	NO
Soporte de múltiples formatos (nativa).	NO	NO	SI
Soporte de múltiples protocolos.	SI	SI	SI
Transformación de formatos de entrada por distinta salida.	NO	NO	SI

Tomando en cuenta la tabla anterior se puede decir que el **Motor de Integración** sería el ideal para el problema, ya que los formatos de cada sistema se pueden conocer, por ende, habrá que buscar un motor compatible con la gama de formatos que se requieran, para luego finalmente transformar, filtrar y lograr la intercomunicación.

Es importante recalcar que si bien API y Web Services no poseen el soporte a múltiples formatos es posible integrar librerías para permitir la interpretación de estos.

Ahora de lo anterior y llevándolo a distintos puntos de vista dentro de un centro hospitalario se puede decir lo siguiente:

Organización: Utilizando un integrador de datos en cada centro hospitalario de la región o del país se puede establecer una base de datos consolidada donde se encuentren todos los datos de los pacientes y al alcance de cada sistema que este dentro del centro de salud, permitiendo en un futuro una comunicación horizontal entre todos los sistemas a lo largo del país.

UCI (Unidad de cuidados intensivos): al implementar el integrador se obtendrá una mayor claridad de los datos, no existirá duplicación, no habrá duda a la hora de una asignación de un paciente, en conclusión, un servicio robusto y estable.

Usuarios: con el integrador existirá mayor despliegue de información, más estadísticas, parámetros, índices de mejora, mayor acceso a los datos y con mejor velocidad de obtención.

TI (Unidad de tecnologías de información): principales beneficiados pudiendo obtener información cruzada de distintas fuentes para elaborar estadísticas o Dashboard que permitan prever situaciones o eventualidades, como también la oportunidad de aplicar de buena forma el BI. [20]

Un ejemplo claro donde debería aplicarse la integración es a la hora de hacer el ingreso de un paciente y al ingresar una observación al mismo. Donde el sistema de ingreso de pacientes lleva varios años funcionando y posee un formato **X** y el sistema de ingreso de observaciones puede variar según la especialidad del médico tratante o el tipo de observación, por ejemplo, formato **A, B, C**. Entonces ocurrirá que se tiene la información del paciente para hacer el ingreso de dicha observación, pero no es legible para el sistema de ingreso de observaciones ya que esta en formato **X**, por ende, para integrar esos datos sin la acción de redigitar la información del paciente, es necesario el uso de un motor de integración que sea capaz de tomar esos datos en formato **X** y convertirlos al formato requerido, ya sea **A, B, C**. Finalmente se tendrá la información que es necesaria del formato **X** en el sistema de ingreso de observaciones, permitiéndonos la interoperabilidad entre los sistemas.

2.5 Formatos de datos del Área Salud

Como se dijo en el capítulo uno el lugar de estudio es un centro hospitalario tomando como referencia al Hospital Guillermo Grant Benavente, que como tal maneja una amplia variedad de sistemas y datos. También, se presentó la integración de datos y los motores de integración. Siendo los motores de integración una alternativa viable para una solución de la realidad actual del hospital.

Algo muy importante a destacar es que los centros hospitalarios no siempre han tenido sistemas, en sus inicios fueron métodos manuales, hoy en día software, es por esto que continuamente se tendrá una rotativa de sistemas a medida que transcurre el tiempo y las tecnologías. De lo mencionado se puede decir que la integración de sistemas es una solución más que acertada. [2]

2.5.1 Formatos de Mensajería

Como son diversos los sistemas que participan dentro de un centro de atención médica se pueden ver diversos tipos de datos [18], es decir, formatos de comunicación, entre ellos se tiene:

- **Texto Delimitado:** texto definido por software propio u otros para interpretar la información.
- **DICOM:** es un protocolo estándar de comunicación entre sistemas de información y a la vez un formato de almacenamiento de imágenes médicas.
- **EDI/X12:** Se usa para transferir documentos electrónicos o datos de negocios de un sistema computacional a otro.

- **HL7:** es un conjunto de estándares para facilitar el intercambio electrónico de información clínica; que utiliza una notación formal del lenguaje unificado de modelado (Unified Modeling Language, UML) y un metalenguaje extensible de marcado con etiquetas (Extensible Markup Language, XML).
 - **Mensajería HL7 Versión 2:** Estándar de mensajería para el intercambio electrónico de datos de salud.
 - **Mensajería HL7 Versión 3:** Estándar de mensajería para el intercambio electrónico de datos de salud basada en el RIM.
 - **CDA HL7:** (Clinical Document Architecture) Estándar de arquitectura de documentos clínicos electrónicos.
 - **SPL HL7:** (Structured Product Labeling) Estándar electrónico de etiquetado de medicamentos.
 - **HL7 Medical Records:** Estándar de administración de Registros Médicos.
 - **GELLO:** Estándar para la expresión de reglas de soporte de decisiones clínicas.
 - **Arden Syntax:** Es estándar sintáctico (if then) para compartir reglas de conocimiento clínico.
 - **CCOW:** Es un estándar para frameworks para compartir contexto entre aplicaciones.
 - **FHIR HL7:** FHIR son las siglas de Fast Healthcare Interoperability Resources (pronunciado como fire) y se trata del último estándar desarrollado y promovido por la organización internacional HL7 (Health Level Seven), responsable de algunos de los protocolos de comunicaciones más utilizados hoy en día en el ámbito sanitario. FHIR trata de combinar lo mejor de cada uno
- **JSON:** Es un formato de texto ligero para el intercambio de datos, además es un subconjunto de la notación literal de objetos de JavaScript.
- **NCPDP:** (National Council for Prescription Drug Programs, Consejo Nacional para los Programas de Prescripción de Medicamentos) exige que se cumplan los requisitos respecto de la manera estándar en que los reclamos de farmacia se deben transmitir.
- **Raw:** Es un formato de archivo digital de imágenes que contiene la totalidad de los datos de la imagen tal y como ha sido captada por el sensor digital de la cámara, ya sea fotográfica u otro tipo.
- **XML:** Se utiliza para el intercambio de datos estructurados. Más que un formato de archivos rígido, XML es un lenguaje que define los formatos aceptados que pueden utilizar los grupos para intercambiar información.

Del listado anterior es importante destacar HL7 y sus derivados ya que son un estándar característico del área clínica, el cual dentro del país está tomando cada vez más peso debido a los problemas de intercomunicación entre los sistemas, esto se puede reflejar en

la página del Minsal [14] donde en conjunto con Corfo se están otorgando becas para aprender a manipular los estándares de HL7 y poder mejorar la interoperabilidad dentro de los sistemas hospitalarios en Chile.

Cita:

“Cuando miramos los factores de riesgo, que son los más difícil de controlar, y el cambio de perfil demográfico y epidemiológico (del país), podemos darnos cuenta de que la forma en que entregamos salud no va a ser posible de continuar para responder a este nuevo cuadro. La cirugía mayor a la salud es necesaria aquí y ahora (...) para mejorar acceso y calidad. Con nuestro modelo fragmentado, con atención en silos, sin trazabilidad de los pacientes y sin existencia de autocuidado, es imposible entregar respuesta a las necesidades del futuro”, señaló Santelices, quien participó en un encuentro con expertos nacionales e internacionales en interoperabilidad a través de FHIR, un modelo de transferencia de datos en salud. [2]



Figura 4: Futuro aplicando en su totalidad la interoperabilidad en Chile. [2]

En la **Figura 4** se puede ver que un futuro se espera que la salud se encuentra interconectada, es decir, permite tener la información de un paciente ya sea en cesfam, hospitales o consultorios, dando paso a la referencia y contra referencia que trae un paciente, obteniendo con eso la información del mismo en todo lugar y momento.

2.5.2- Protocolos de Comunicación

Todos estos datos como tal deben de viajar a una base de datos o a otro sistema, si es que así fuera, lo harían por medio de algún transporte, el cual vendría siendo un protocolo de comunicación. Algunos de ellos son:

- TCP/MLLP
- Base de datos
- File (sistema de archivo local y de red)
- JMS
- Documentos RTF
- FTP/SFTP
- SMTP
- SOAP (usando HTTP Request)

Los listados anteriormente son los que mayormente se encuentran dentro de los motores de integración, destacando SOAP, TCP, FTP y bases de datos.



3. Elección de Motor a Implementar

Para esta memoria será necesario escoger un motor de integración de datos para luego montarlo y configurarlo, el cual debe tener como foco los formatos del área de la salud para poder trabajar con mayor comodidad.

3.1 Motores de integración Sector Salud

Cuando se habla de integradores de datos del sector o área salud se refiere a motores que integren dentro de su implementación la mayor cantidad de formatos que sean estándar hoy en día en los centros hospitalarios. Además de poder analizar el tipo de conversiones permitidas por estos integradores para lograr el intercambio de comunicación que se requiera. Algunos de los más importantes dentro del mercado son los siguientes:

- Ensemble [21]
- Mirth Connect [8]
- Cloverleaf [5]
- Corepoint [4]
- Rhapsody [6]

Los cinco nombrados anteriormente son los más relevantes para la comunidad médica internacional Core Health Technologies.

Core Health Technologies [11] es un equipo dedicado de expertos en tecnología de atención de la salud independientes del proveedor, que ayuda a hospitales, sistemas de atención médica y laboratorios a garantizar que su tecnología funcione para ellos, y no al revés.

3.1.1 Evaluación y Análisis

Dentro de la comparación general los elementos más relevantes a destacar son los que se mencionan en la tabla siguiente ya que ayuda a realizar una comparación más global de cómo se estructuran estos integradores.

Motores Integración	Formatos	Código Libre	Comunidad	Módulos
Mirth Connect	SI	SI	SI	SI
Clover Leaf	SI	NO	SI	SI
Rhapsody	SI	NO	SI	SI
Ensemble	SI	NO	SI	SI
Corepoint	SI	NO	SI	SI

A continuación, se despliega una serie de resultados, los cuales fueron obtenidos del Core healthTechnologies.

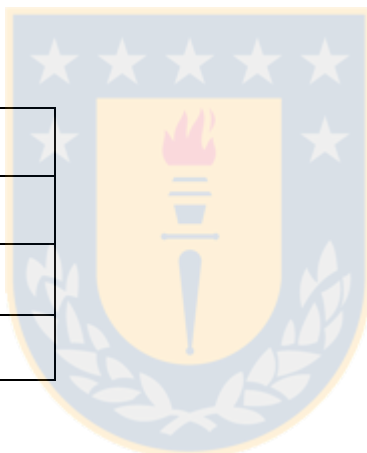
Desde el punto de vista de HL7 y encuestando a los siguientes trabajadores:

- Profesionales: Programadores, desarrolladores analistas, arquitectos de sistema, jefe proyecto, garantía de calidad y control de calidad.
- Manager: administrador interfaces de integración, jefes de equipo y manager aplicaciones clínicas.
- Ejecutiva: director de IT, CIO, CTO, CMIO, CNIO.

En la gráfica situada en **Anexo 1** que tiene relación con los encuestados mencionados anteriormente, se muestra ¿Que tecnología de soporte hl7 es el motor primario de su organización? [3].

Resultados:

CloverLeaf	21%
Rhapsody	20%
MirthConnect	15%
Otros	44%



En la gráfica situada en **Anexo 2** que tiene relación con los encuestados anteriormente, se muestra ¿Que tecnología de soporte hl7 utiliza actualmente? [7]. También se pueden encontrar resultados del 2012 [22].

Resultados:

MirthConnect	13,28%
CloverLeaf	12,99%
Rhapsody	10,73%
Otros	63%

Además, se consideraron otros aspectos para tener mas claridad sobre la elección del motor de integración a utilizar, son los siguientes:

- **Satisfacción de necesidades:** Muestra el porcentaje de satisfacción con respecto a las necesidades que tiene la gente que utiliza estos integradores de datos. Ver Anexo 3
- **Usabilidad y Características:** Muestra el porcentaje con respecto al conjunto de características completas y fáciles de usar dentro de estos integradores. Ver Anexo 4
- **Costo de Adquisición:** Muestran que tan costoso es adquirir estos integradores. Ver Anexo 5
- **Soporte:** indican el porcentaje con respecto al soporte que poseen estos integradores de datos. Ver Anexo 6

Considerando lo anterior se tiene que Mirth Connect:

- Es de código libre con posibilidad de escalar a una licencia pagada según los módulos que se requieran.
- Soporta diversos formatos, tiene como vías de comunicación diversos protocolos.
- Utiliza hl7 que es el formato clínico por excelencia y está tomando parte en nuestro país.
- Tiene algo muy particular como el FHIR.
- Realiza conversiones de HL7 V2 a V3 aún están vigente sus actualizaciones
- Según lo investigado en este estudio posee una gran comunidad hispana y de habla inglesa (se puede ver más información en los anexos y referencias).
- Presenta un log de transacciones realizadas.
- Los profesionales de hl7 lo consideran una gran carta para adoptar dentro de un centro hospitalario
- Los protocolos y formatos de mensajería que maneja el Mirth son más que suficientes para las tecnologías que se usan mayormente en la actualidad.
- Desde el punto de vista del montaje no se requieren mayores recursos ni sistemas operativos específicos (Multiplataforma).
- Cabe destacar que el Mirth se posiciona dentro de los 5 mejores del mundo según el Core health technologies donde compite con grandes como Corepoint y Cloverleaf que son de pago.

3.2 Mirth Connect

Es un motor de integración que permite la comunicación entre sistemas de información dispares diseñado para funcionar en instituciones de salud. Tiene disponible un conector de escucha FHIR "Fast Healthcare Interoperability Resources". Permite una solución de integración que nos deja manejar todo el trabajo necesario para transformar y enrutar mensajes de distintos orígenes.

El *Mirth* fue diseñado principalmente para la integración de mensajes HL7, pero soporta múltiples formatos. Algunos de estos formatos son: Texto Delimitado, DICOM, EDI/X12, HL7 v2.X, HL7 v3.X, JSON, NCPDP, Raw y XML. Para el traslado de los datos existen varios protocolos comunicación soportados: TCP/MLLP, DB, PDF, RTF, FTO/SFTP, HTTP, SMTP, SOAP.

Dentro del intercambio de mensajes se permite filtrar, transformar, extraer y enrutar la información.

Dentro de la página del Mirth se puede encontrar el siguiente esquema trata de clarificar lo que logra hacer utilizando la integración de los datos, reflejando como se integra dentro de los distintos sistemas dentro un centro hospitalario.

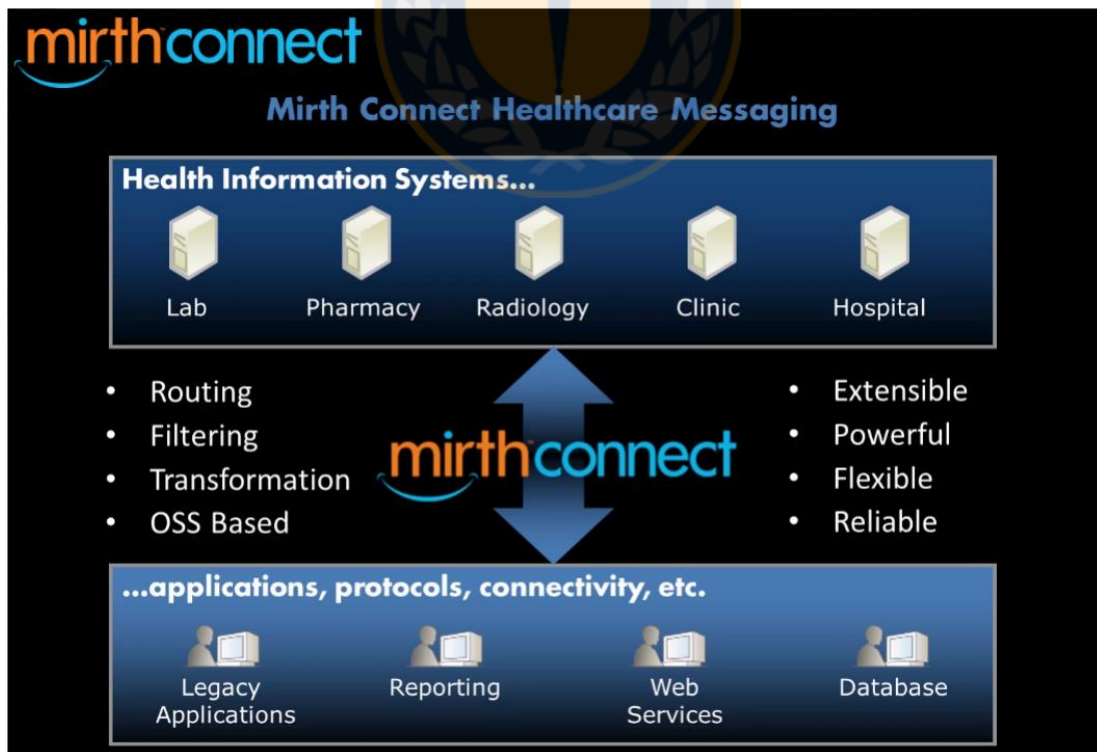


Figura 4: Funcionamiento de mirth connect. [8]

3.3 Infraestructura Planteada

A continuación, se verá una figura que representa la infraestructura que tendrá contemplada al motor de integración Mirth connect como foco central, cabe mencionar que se pensó en una infraestructura en la nube para tener acceso desde cualquier lugar como para aprovechar los servicios de este.

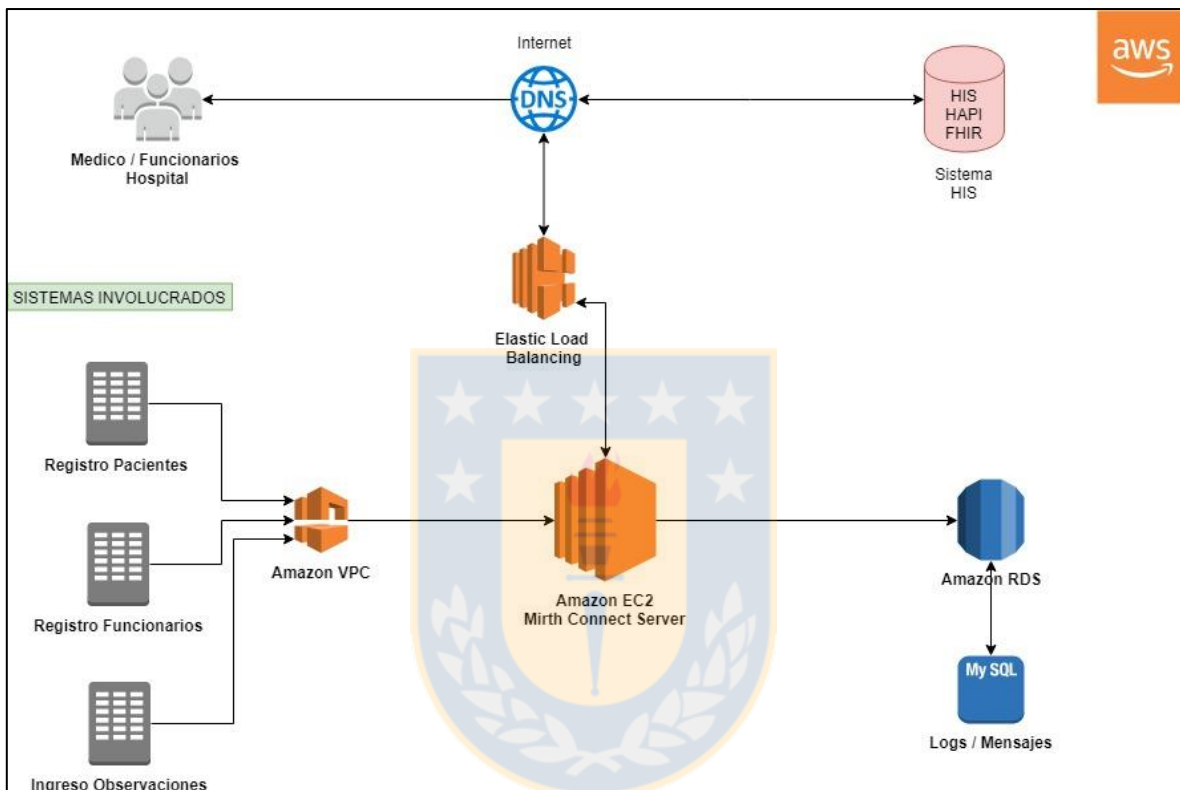


Figura 5: Infraestructura planteada para el montaje de Mirth Connect.

En la **Figura 5**, haciendo uso de los servicios en la nube por medio de Amazon Web Services, se tiene que **Elastic Load balancing** tiene salida a la red y nos permite crear una instancia de tipo **EC2** (Linux server) [12], donde se montará el **Mirth Connect Server**, adicional este **EC2** tendrá una conexión directa con una base de datos **MySql**, la cual fue creada por el Servicio **RDS** [13]. Además, Mirth Connect tendrá conexión vía HTTP Request con un servidor HAPI FHIR [1] (Actuará como sistema HIS), el cual almacenará todos los datos. Amazon VPC o red privada en la nube será el encargado de soportar la conexión con los distintos sistemas involucrados (Situación adaptable un centro Hospitalario).

Para el caso de esta memoria de título y considerando los sistemas mencionados en el esquema, estos fueron simulados sobre el servicio **EC2**, y no conectados vía AmazonVPC,

que para efectos de resultados no son relevantes, ya que simulados o no seguirían estando en una red local.

Finalmente, como Mirth se encuentra conectado a internet nos permite hacer una visualización macro sobre las transacciones que ocurren dentro del mismo, esto último se logra ingresando con una cuenta de funcionario.

3.4 Implementación y Montaje

De las consideraciones anteriores se tiene que el desarrollo será efectuado en la nube, utilizando los servicios de Amazon Web Services.

Lista de pasos a seguir:

- a) Configurar Servidor y AWS
- b) Configurar Seguridad
- c) Definir tipos de datos
- d) Configurar Mirth
- e) Definir Procesos en Mirth
 - a. Registro Pacientes
 - b. Ingreso Observaciones
 - c. Registro Personal Medico

3.4.1 Configurar Servidor y AWS

Para llevar a cabo la implementación se utilizó Amazon Web Services como proveedor de servicios donde se disponía de una cuenta de uso free, pero suficiente para el desarrollo de un prototipo.

Lo primero es crear una instancia de servidor, en este caso en particular se opta por un:

- Linux Server Ubuntu 16.04
- 1 Gb Ram
- Intel Xeon 1 Núcleo
- 30 Gb SSD
- Conexión Fibra Óptica
- Firewall Físico Controlado

Lo siguiente, es la configuración de este servidor para preparar la instalación del mirth connect, donde será necesario abrir los puertos 8080 y 8443 TCP tanto en el firewall físico como virtual además de agregar las reglas a iptables. También se debe de instalar el java jdk 8 para permitir la ejecución del mirth.

Es necesario instanciar un servicio de MySQL para luego asociarlo al mirth como integrador de datos, la versión de MySQL a utilizar será la 5.6 y será de tipo privado es decir solo el servidor en la misma Subnet tendrá acceso, en este caso el servidor Linux server que se mencionó al comienzo de la sección 3.4.1

Instalar los siguientes software sobre el sistema operativo Linux server:

- Apache Maven
- Npm
- No-ip
- Maven json

A continuación, se debe acceder a la web de nextgen [10] que es la encargada de la mantención del software mirth connect y descargar este servicio en su versión 3.4.2

Una vez descargado debe de instalarse el mirth y configurar en los puertos 8080 y 8443, a los cuales se dieron liberación antes, posteriormente deberá asociarse a una base de datos, como también deberán crearse transacciones para el traspaso de información según los servicios o sistemas que vallan a interactuar.

3.4.2 Configurar Seguridad

Considerando que el sistema está siendo utilizado en la nube es necesario tener resguardos tantos virtuales como físicos (configuración del firewall vía gestor digital online inbound, outbund, traffic) a nivel de seguridad de la información, ya que la que circula por estos sistemas es privada y debe de encapsularse.

A continuación, se muestran algunas medidas necesarias a nivel de servidor que se deben de tener en cuenta

Activación Firewall Linux

- ufw enable
- ufw allow 8080/tcp
- ufw allow 8443/tcp

Uso de DNS fijo por medio del servicio no-ip para evitar el cambio de ip pública por reinicio de servidores, con esto se logra encapsular la ip que genera el servidor en un nombre único, el cual se muestra a continuación:

DNS: mirthconnect.ddns.net

3.4.3 Definir Tipos de datos involucrados

Los ejemplos de esta memoria consideran los siguientes tipos de datos que representan la información disponible de una organización de atención médica típica:

- HL7 Versión 2
- CSV (variable separada por comas)
- HL7 FHIR

Los mensajes HL7 versión 2 definen tanto un formato de mensaje como un protocolo de comunicación para información de salud. Se dividen en diferentes tipos de mensajes según la información que transmiten.

Hay muchos tipos de mensajes disponibles, como ordenar laboratorios, dispensación de recetas, facturación y más. Durante una visita al médico de rutina, se crean numerosos mensajes para cada paciente. Esto proporciona mucha información, pero también un desafío en el almacenamiento y procesamiento. El tipo de dato utilizado para este prototipo es:

- ADT A01 (admisión del paciente)

Otro tipo de dato involucrado será el formato **CSV**, son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma en donde la coma es el separador decimal: Chile, Perú, Argentina, España, Brasil...) y las filas por saltos de línea. Donde este será utilizado para contener las observaciones del paciente ya sea presión, temperatura, niveles de colesterol, triglicéridos, etc

Finalmente, HL7 FHIR es el acrónimo de **Fast Healthcare Interoperability Resources** se incorpora al conjunto de estándares desarrollado y promovido por la organización Health Level 7, responsable de la mayoría de los protocolos mencionados anteriormente y destinados al intercambio de información en el entorno sanitario.

Un servidor HAPI FHIR [19], será el sistema HIS en este prototipo, es decir, un sistema central para almacenar la información total, donde los tipos de datos mencionados llegarán transformados para ser ingresados a este.

3.4.4 Configurar Mirth Connect

Mirth Connect está diseñado en torno al concepto de canales. Cada canal tiene una fuente y uno o más destinos. Además, los mensajes aceptados de la fuente pueden filtrarse (validar), modificarse (transformarse) para luego ser dirigidos a sus respectivos destinos.

Canales utilizados en el prototipo:

- **V2-ADT-A01 TO FHIR-PATIENT:** Procesa archivos HL7 V2 ADT A01 y los transforma a FHIR para ser ingresados al servidor HAPI FHIR.
- **OBS TO FHIR-OBSERVATION:** Procesa archivos CSV y los transforma a FHIR, luego los ingresa al servidor HAPI FHIR.
- **PRA TO FHIR-PRACTITIONER:** Procesa archivos en formato x y los transforman a FHIR para así ingresar al sistema HIS HAPI FHIR.

3.4.5 Definir Procesos en Mirth Connect para cada Canal

Las ilustraciones que serán desplegadas a continuación pertenecen al proceso que ocurre dentro del Mirth Connect cuando se realizan acciones en alguno de los sistemas

- Registro Paciente

El siguiente esquema despliega el proceso que al momento de hacer ingreso o registro de un paciente dentro de un centro hospitalario.

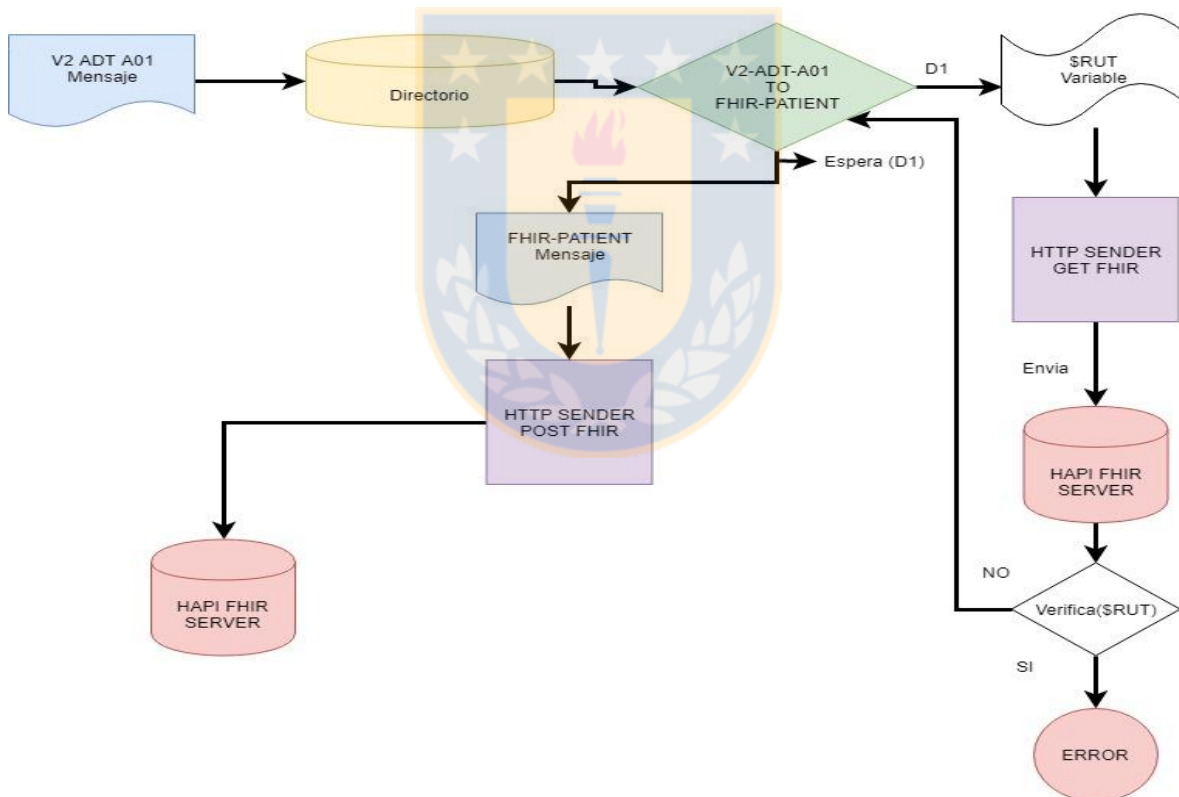


Figura 6: Proceso Registro Paciente sobre Mirth.

En la **Figura 6** se puede ver que una vez que se tiene el mensaje de ingreso de paciente en el formato **V2 ADT A01** se guarda en el directorio de input, luego el canal **V2-ADT-A01 TO FHIR-PATIENT**, toma el **RUT** y por medio de **HTTP SENDER** envía una petición **GET** al HIS para ver si existe dentro del servidor principal, en el caso de no estar se continúa donde luego el

mensaje se filtra y transforma para lograr el formato **FHIR-PATIENT**, que finalmente es enviado por **HTTP SENDER** vía POST al servidor **HAPI FHIR**.

- Ingreso Observaciones

El siguiente esquema representa el proceso que se ejecuta a la hora de tomar observaciones sobre el paciente, ya sea presión, nivel de triglicéridos, temperatura, pulso, etc.

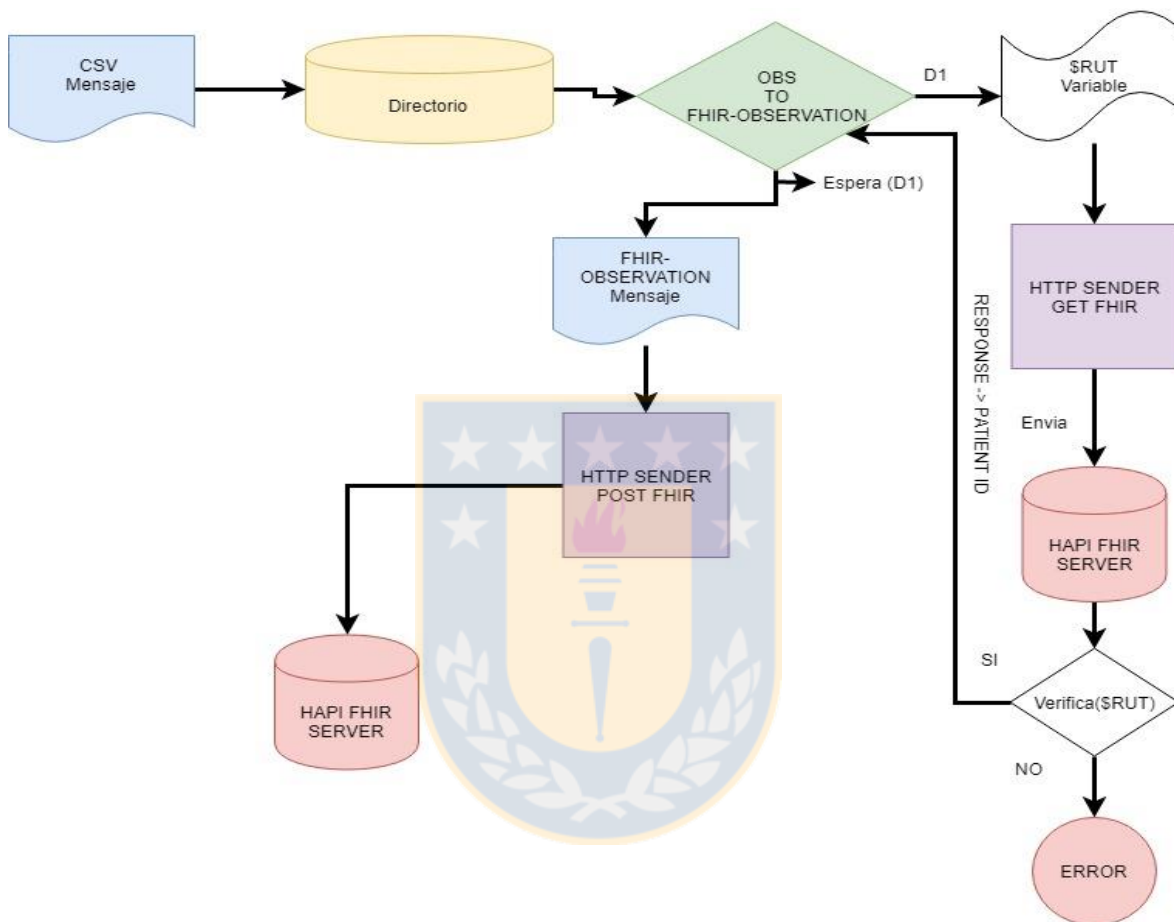


Figura 7: Proceso Ingreso Observaciones en Mirth.

En la **Figura 7**, cuando se tiene el mensaje **CSV** con las observaciones y se guarda en el directorio de input, el canal **OBS TO FHIR-OBSERVATION** toma este último y realiza las siguientes acciones por una parte toma la variable del Rut y verifica si existe en el sistema **HIS HAPI FHIR**, si la verificación resulta incorrecta da error y termina la transacción en el canal, y si es exitosa devuelve el **ID** de la variable **RUT** y se la pasa nuevamente al canal **OBS TO FHIR-OBSERVATION** para filtrar y transformar el mensaje al formato **FHIR-OBSERVATION**, donde luego es enviado al servidor **HAPI FHIR** por medio del **HTTP SENDER**.

- Registro Personal Médico

El siguiente esquema despliega el proceso que al momento de hacer ingreso o registro de un paciente dentro de un centro hospitalario.

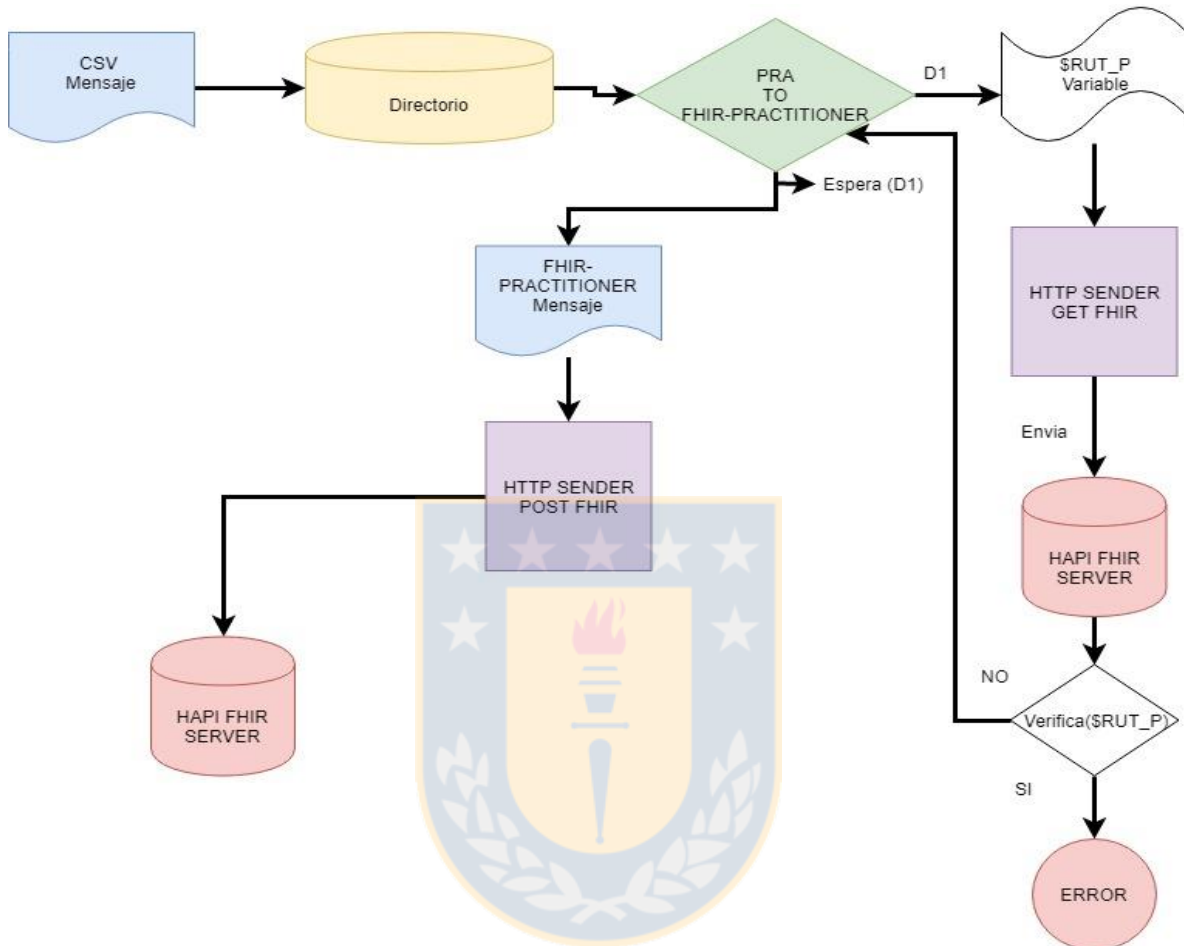


Figura 8: Proceso Registro Personal Médico en Mirth.

En la **Figura 8**, se puede ver que una vez que se tiene el mensaje de registro de personal médico en el formato **CSV** se guarda en el directorio de input, luego el canal **PRA TO FHIR-PRACTITIONER**, toma el **RUT** y por medio de **HTTP SENDER** envía una petición **GET** al HIS para ver si existe dentro del servidor principal, en el caso de no estar se continúa donde luego el mensaje se filtra y transforma para lograr el formato **FHIR-PRACTITIONER**, que finalmente es enviado por **POST** usando **HTTP SENDER** para así introducir la información en el servidor **HAPI FHIR HIS**.

4. Resultados

Canales

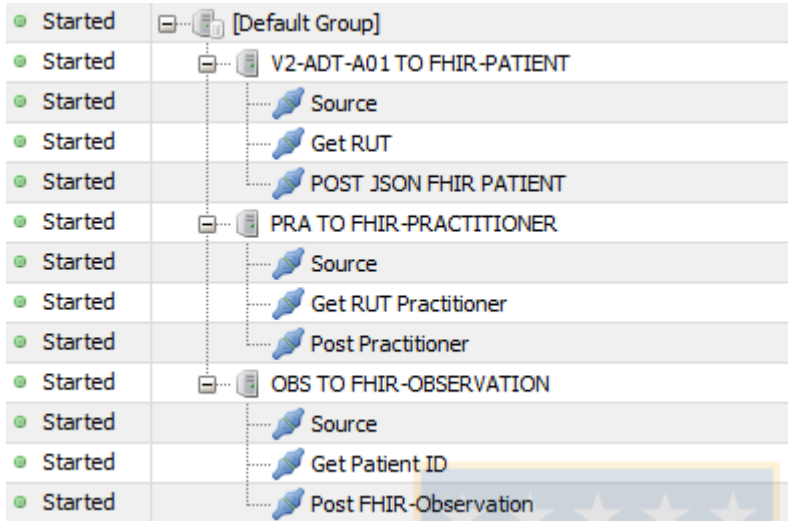


Figura 10: En esta figura se pueden ver los canales que están funcionando.

Como se ve en la **Figura 10** anterior el prototipo posee tres canales funcionando donde cada uno de ellos cuenta con una fuente (source) y dos destinos.

Los destinos de tipo GET proporcionan la información para verificar si un paciente o personal médico se encuentra dentro del HIS principal y no causar conflicto de datos.

Luego tiene los destinos de tipo POST, los cuales se encargan de cargar la información dentro del sistema central HIS HAPI FHIR.

CANAL V2-ADT-A01 TO FHIR-PATIENT:

Este canal recibe como input un mensaje en formato HL7 V2 ADT A01 el cual corresponde al ingreso de paciente en dicho formato. A continuación, se muestra su codificación:

INPUT

```
MSH|^~\&|ADT1|MCM|LABADT|MCM|198808181126|SECURITY|ADT^A01|$MsgId|P|2.6
EVN|A01|$Date
PID||||Perez^Juan^||19941103|male|||Calle1^^Ciudad1^Region1^^Chile|99998888||S||18000000-0|
NK1|$NkId|Jara^Camila^|$RelId^$RelDisplay|CalleV^^CiudadV^RegionV^^Chile|88889999|||||||female
```

Figura 11: En esta figura se puede ver la codificación de HL7 V2 ADT A01.

Dicho mensaje entra por la fuente(source) del canal mencionado y pasa por el primer destino de tipo GET, en donde este por medio del HTTP SENDER consulta al servidor principal HIS, si el RUT 18000000-0 se encuentra registrado, entregando como respuesta lo siguiente:

```
{
  "resourceType": "Bundle",
  "id": "d1ad76ba-f8b9-4d22-90c0-3be312a17207",
  "meta": {
    "lastUpdated": "2018-08-20T01:56:25.254-03:00"
  },
  "type": "searchset",
  "total": 0,
  "link": [
    {
      "relation": "self",
      "url": "http://152.74.29.160:8080/hapi-fhir-jpaserver-example/baseDstu3/Patient?identifier=RUT%7C18000000-0"
    }
  ]
}
```

Figura 12: En esta figura se puede ver la respuesta a la petición GET.

En la **Figura 12** se logra ver que la variable total, que es la que interesa indica un **0** lo cual significa que el Rut no está en el HIS por ende se podría ingresar, luego se continua al segundo destino del canal el cual posee un filtro, y consulta por la variable total donde esta continúa procesando solo si esta variable es igual a **0**. Después del filtrado se procede a las transformaciones pertinentes para lograr la conversión de datos y obtener el formato **FHIR-PATIENT** y enviarlo vía POST al servidor HIS.

OUTPUT

```
{
  "resourceType": "Patient",
  "id": "50",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2018-08-20T01:56:25.632-03:00"
  },
  "text": {
    "status": "generated",
    "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">paciente 3<a name=\"mm\"/></div>"
  },
  "identifier": [
    {
      "system": "MSG_ID",
      "value": "$MsgId"
    },
    {
      "system": "RUT",
      "value": "18000000-0"
    }
  ],
  "name": [
    {
      "family": "Perez",
      "given": [
        "Juan"
      ]
    }
  ],
  "telecom": [
    {
      "system": "phone",
      "value": "99998888",
      "use": "home"
    }
  ],
  "gender": "male",
  "birthDate": "1994-11-03",
  "address": [
    {
      "line": [
        "Calle1"
      ],
      "city": "Ciudad1",
      "state": "Region1",
      "country": "Chile"
    }
  ],
  "maritalStatus": {
    "coding": [
      {
        "system": "http://hl7.org/fhir/v3/MaritalStatus",
        "code": "S"
      }
    ]
  }
}
```



Figura 13: Fragmento del resultado en json del servidor FHIR.

En la **Figura 13** se logra ver el output o mensaje ya convertido, e ingresado dentro del HAPI FHIR, dando paso así a la interoperabilidad entre estos estándares V2 y FHIR.

CANAL OBS TO FHIR-OBSERVATION

Este canal recibe como input un mensaje en formato csv que corresponde a una observación a un paciente específico, que contiene Rut del paciente, código del examen según loinc.org, unidad de medida, valor, rango bajo y alto. EL mensaje se ve así:

INPUT

"18805406-4", "718-7", "g/L", "haemoglobin", "100", "90", "120"

Figura 14: En esta figura se puede la codificación del formato csv.

El mensaje de la **Figura 14** es recibido por la fuente (source) del canal y luego es llevado al primer destino donde toma el RUT 18805406-4 y verifica si está ese paciente dentro del sistemas HIS utilizando una petición GET al servidor HAPI, entregando el siguiente resultado:



```
{
  "resourceType": "Bundle",
  "id": "2caedf47-2ccf-4759-af59-ab8d1b12fa19",
  "meta": {
    "lastUpdated": "2018-08-20T03:35:55.302-03:00"
  },
  "type": "searchset",
  "total": 2,
  "link": [
    {
      "relation": "self",
      "url": "http://152.74.29.160:8080/hapi-fhir-jpaserver-example/baseDstu3/Patient?identifier=RUT%7C18805406-4"
    }
  ],
  "entry": [
    {
      "fullUrl": "http://152.74.29.160:8080/hapi-fhir-jpaserver-example/baseDstu3/Patient/39",
      "resource": {
        "resourceType": "Patient",
        "id": "39",

```

Figura 15: Fragmento de la respuesta del servidor HAPI para la petición GET.

Tomando la **Figura 15**, se puede ver que el id correspondiente al Rut es el 39, dando paso al siguiente destino que usa como parámetro el id rescatado, permitiendo de esta forma hacer las transformaciones necesarias para realizar una inserción de la observación vía POST.

OUTPUT

```
{
  "resourceType": "Observation",
  "id": "51",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2018-08-20T03:35:55.665-03:00"
  },
  "text": {
    "status": "generated",
    "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">Test
                                                    Units      Value
  },
  "status": "final",
  "code": {
    "coding": [
      {
        "system": "http://loinc.org",
        "code": "718-7",
        "display": "haemoglobin"
      }
    ]
  },
  "subject": {
    "reference": "Patient/39"
  },
  "valueQuantity": {
    "value": 100,
    "unit": "g/L",
    "system": "http://unitsofmeasure.org",
    "code": "g/L"
  },
  "referenceRange": [
    {
      "low": {
        "value": 90,
        "unit": "g/L",
        "system": "http://unitsofmeasure.org",
        "code": "g/L"
      },
      "high": {
        "value": 120,
        "unit": "g/L",
        "system": "http://unitsofmeasure.org",
        "code": "g/L"
      }
    }
  ]
}
```



Figura 16: Vista del servidor FHIR de la observación ingresada.

En la **Figura 16** se mantiene la información que se encontraba en el formato original pero esta vez en su nuevo formato de destino FHIR-OBSERVATION.

PRA TO FHIR-PRACTITIONER:

Este canal recibe como input un mensaje en formato csv que corresponde al registro de un personal médico el cual posee Rut, nombre, apellido, teléfono, dirección, ciudad, región, país, sexo y fecha de nacimiento. En la **Figura 17** se verá su codificación:

INPUT

```
"18805406-4","rios","elvis","987017217","sor vicenta 213","los angeles","biobio","chile","male","1994-11-03"
```

Figura 17: CSV codificado para registro de personal médico.

El mensaje anterior es recibido por la fuente (source) del canal para luego ir al primer destino en donde se verifica si el 18805406-4 es un médico registrado o no dentro del sistema central HIS HAPI FHIR.

```
{
  "resourceType": "Bundle",
  "id": "c8da78d8-8895-4e2b-ada3-b18364062a7d",
  "meta": {
    "lastUpdated": "2018-08-20T04:38:55.251-03:00"
  },
  "type": "searchset",
  "total": 0,
  "link": [
    {
      "relation": "self",

```



Figura 18: Fragmento de la respuesta del servidor HAPI para la petición GET.

Observando la **Figura 18** se ve que en la variable total el valor asociado es **0**, esto indica que no existe otro personal registrado con ese Rut, por ende, se podría registrar ese funcionario. Luego el siguiente destino que ya es válido para ejecutarse porque total es igual a **0** (filter), entonces se procede a realizar las transformaciones según corresponda para cada campo y finalmente ya convertido se envía una petición de tipo POST al servidor HAPI FHIR que funciona como sistema HIS.

OUTPUT

```
{
  "resourceType": "Practitioner",
  "id": "53",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2018-08-20T04:28:55.595-03:00"
  },
  "text": {
    "status": "generated",
    "div": "<div xmlns=\\"http://www.w3.org/1999/xhtml\\"><a name=\\"mm\\"/></div>"
  },
  "identifier": [
    {
      "system": "RUT",
      "value": "18805406-4"
    }
  ],
  "name": [
    {
      "use": "usual",
      "family": "rios",
      "given": [
        "elvis"
      ]
    }
  ],
  "telecom": [
    {
      "system": "phone",
      "value": "987017217",
      "use": "home"
    }
  ],
  "address": [
    {
      "line": [
        "sor vicenta 213"
      ],
      "city": "los angeles",
      "state": "biobio",
      "country": "chile"
    }
  ],
  "gender": "male",
  "birthDate": "1994-11-03"
}
```



Figura 19: Vista Practitioner desde servidor HAPI FHIR

Al ver la **Figura 19** se nota que al cambiar el formato los datos originales aún prevalecen, dando cuenta de que existió comunicación entre ellos y se logró integrar los datos.

4.1 Resolución

Considerando los resultados de las pruebas y la secuencia de pasos que se generaron a partir del funcionamiento de los canales en forma simultánea, se puede decir que, si bien en este prototipo fue posible manejar 3 procesos, albergando distintos sistemas (origen de formatos y/o destinos), utilizando una cierta cantidad de canales definidos, es posible deducir que se podrían agregar todos los procesos y sistemas que se consideren necesarios, siempre cuando se tenga en cuenta, que para cada interacción deberá generarse un canal que permita una fuente, filtro, transformación y / o destinos.

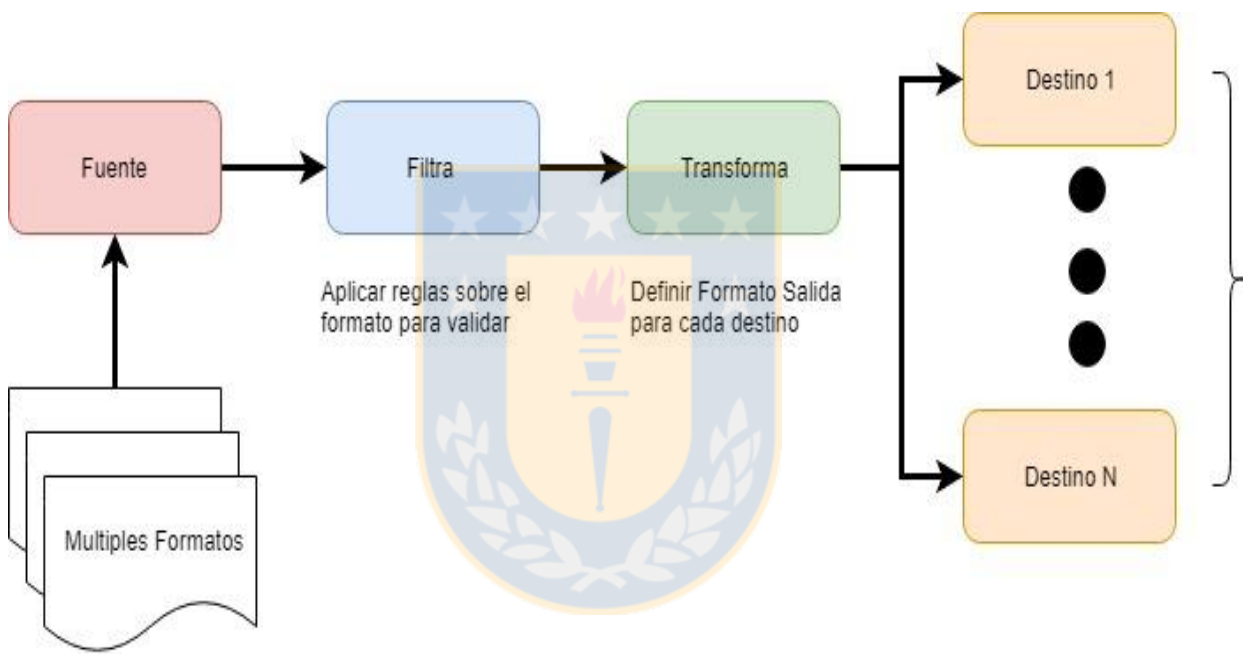


Figura 20: Representación general funcionamiento canal Mirth.

Considerando la **Figura 20** y el párrafo anterior, se interpreta que para poder agregar nuevos sistemas se debe de ser consciente de la **Fuente** de los datos y el formato que posee, dónde se quiere llegar (**Destinos**) y qué se quiere mandar (**Filtros**), como también tener en cuenta el cómo se quiere mandar (**Transformación**).

5. Conclusiones

El uso de estas tecnologías nos permite poder reducir tiempos de espera, evitar duplicación y redundancia en los datos además de evitar la redigitación de los mismos, por otro lado, poder implementar el uso de hl7 como una versión estandarizada sería de utilidad para agilizar aún más los procesos.

Este prototipo es la base para poder comenzar a construir un sistema hospitalario conjunto y distribuido, permitiendo dar facilidades a pacientes, tener un mejor servicio para la comunidad en general y otorgar calidad de servicio.

El uso de herramientas de código libre nos da el acceso a nuevos updates que se liberen por la comunidad los que con el pasar del tiempo podrán darle al Mirth aún más soporte [9] a nivel de formatos, permitiendo tratar con tecnologías posteriores.

Retomando los objetivos o expectativas planteadas a la hora del inicio de la memoria de título se tiene lo siguiente:

Objetivos Específicos

- A. Mejorar la integración los datos entre dos o más sistemas utilizados dentro del centro hospitalario, permitiendo un mejor acoplamiento entre los mismos por medio de una correcta comunicación.
- B. Trabajar para disminuir la redigitación a la hora de hacer ingresos de pacientes o traslado de estos entre áreas del centro de salud.
- C. Mejorar, agilizar, aumentar la velocidad de respuesta a la hora de tomar una decisión, con esto se tiene una mejor asociación entre los datos. Con lo último mencionado se evitaría recopilar la información (sistema por sistema) para luego generar una decisión.

El punto A prácticamente se abarca completo ya que en este prototipo se interactuó con tres sistemas, donde existe la relación de dos en dos, en la cual se comparten sus datos.

En el punto B, uno de los sistemas que se trabajó, es el ingreso de pacientes. La información de los pacientes fue ingresa a una base de datos, por lo tanto, esta información es ingresada sólo una vez y consultada cada vez que se necesite.

El punto C es un poco más delicado ya que este al ser un prototipo y trabajar con una mínima cantidad de datos no se puede llegar una conclusión clara, pero si se toma la escala de resultados y se lleva a una mayor cantidad de datos la tendencia sería agilizar la velocidad de respuesta.

6. Referencias

- [1] HAPI FHIR HIS SERVER <http://152.74.29.160:8080/hapi-fhir-jpaserver-example/>. Junio 2018
- [2] Corfo Minsal Estandares de Interoperabilidad https://www.corfo.cl/sites/cpp/sala_de_prensa/nacional/bcch_estandares_de_interoperabilidad_para_el_sector_salud. 2017
- [3] CoreHealth Technologies HL7 Ratings Results, <https://www.corehealthtechnologies.com/insights/2017-hl7-interface-engine-ratings-survey-results/support-and-services/>. 2017
- [4] Cloverleaf Portal, <https://www.infor.com/product-summary/healthcare/cloverleaf/>. 2018
- [5] CorePoint Portal, <https://corepointhealth.com/resource-center/datasheets/corepoint-integration-engine-product-overview>. 2018
- [6] Rhapsody Portal, <https://orionhealth.com/global/products/rhapsody/>
- [7] HL7 Professionals Opinion, <https://www.corehealthtechnologies.com/ensemble-hl7-interface-consultant/>. 2017
- [8] Mirth Connect Corporation, <http://www.mirthcorp.com/>. 2018
- [9] Mirth Connect Support Community, <http://www.mirth-support.com/>. 2018
- [10] NextGen HealthCare, <https://www.nextgen.com/>. 2018
- [11] Core Health Technologies, <https://www.corehealthtechnologies.com/>. 2018
- [12] Aws Amazon EC2 https://aws.amazon.com/es/ec2/?sc_channel=PS&sc_campaign=acquisition_CL&sc_publisher=google&sc_medium=english_ec2_b&sc_content=ec2_e&sc_detail=aws%20amazon%20ec2&sc_category=ec2&sc_segment=177714740769&sc_matchtype=e&sc_country=CL&skwid=AL!4422!3!177714740769!e!!g!!aws%20amazon%20ec2&ef_id=Wid2ygAAAVQknVEk:20180830165341:s 2018
- [13] Aws Amazon RDS, <https://aws.amazon.com/es/documentation/rds/> 2018
- [14] Minsal, <http://www.minsal.cl/> 2018
- [15] Power Data, <https://www.powerdata.es/integracion-de-datos/> 2018
- [16] Web Services Interoperability, <http://www.ws-i.org/> 2018

[17] *API's Interoperability* <https://www.computerweekly.com/feature/APIs-lead-the-way-to-better-interoperability/> 2018

[18] *HL7 Standard*, <http://www.hl7.org/> 2018

[19] *HAPI FHIR DOCS*, <http://hapifhir.io/> 2018

[20] *Business Intelligence*, https://www.sinnexus.com/business_intelligence/ 2017

[21] *Ensemble Health*, <https://ensemblehp.com/> 2018

[22] *Core Health Rating Results 2012* <http://www.corehealthtechnologies.com/wp-content/uploads/2017/11/2012-HL7-Interface-Technology-Survey-Results.pdf?submissionGuid=b85ef392-6ec8-44bc-b195-dff59d9edca2/>. 2012



7. Anexos

Anexo 1

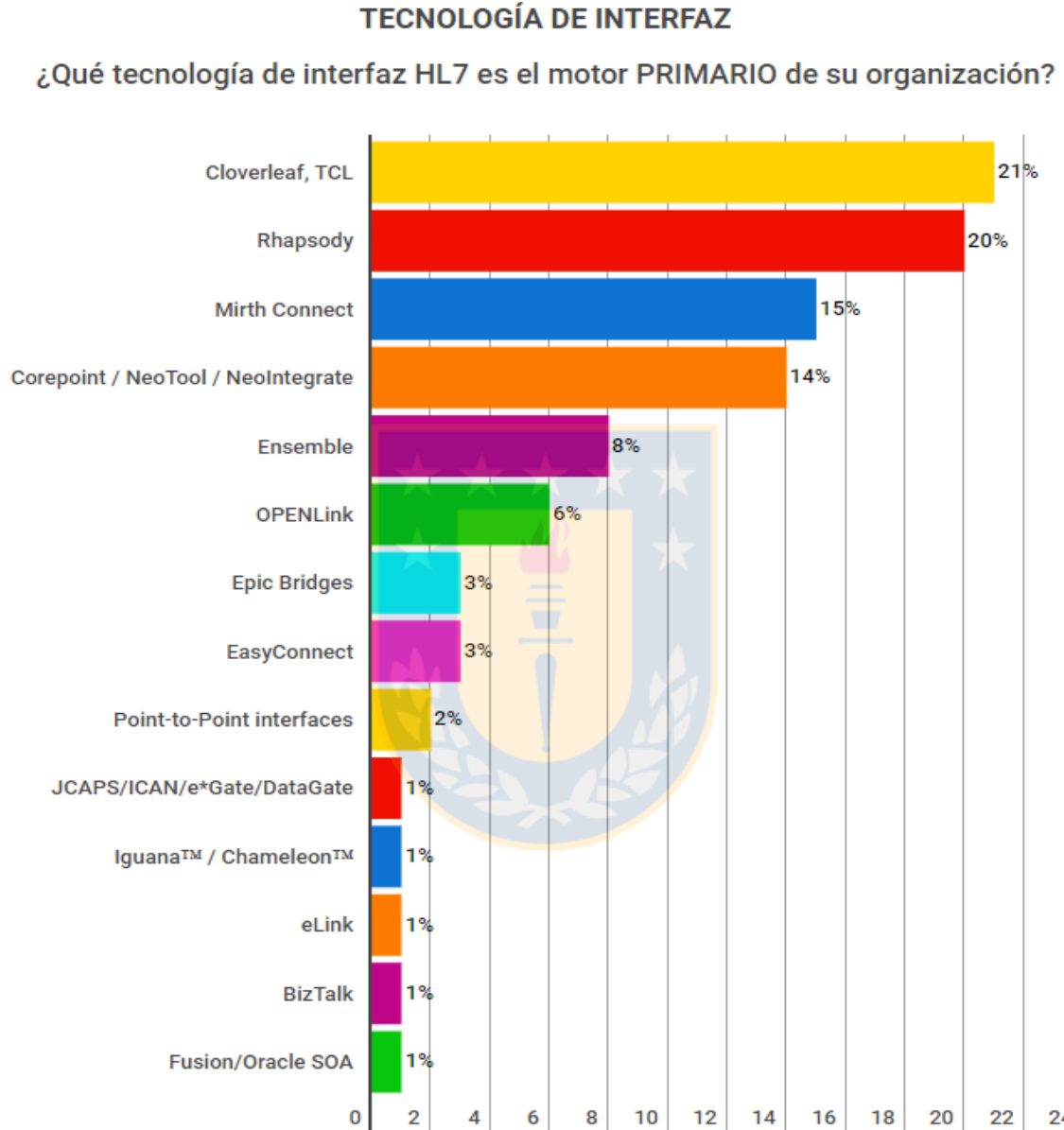


Figura 21, Fuente: CoreHealthTechnologies.

TECNOLOGÍA DE INTERFAZ

Tecnologías de interfaz HL7 que su organización utiliza ACTUALMENTE.

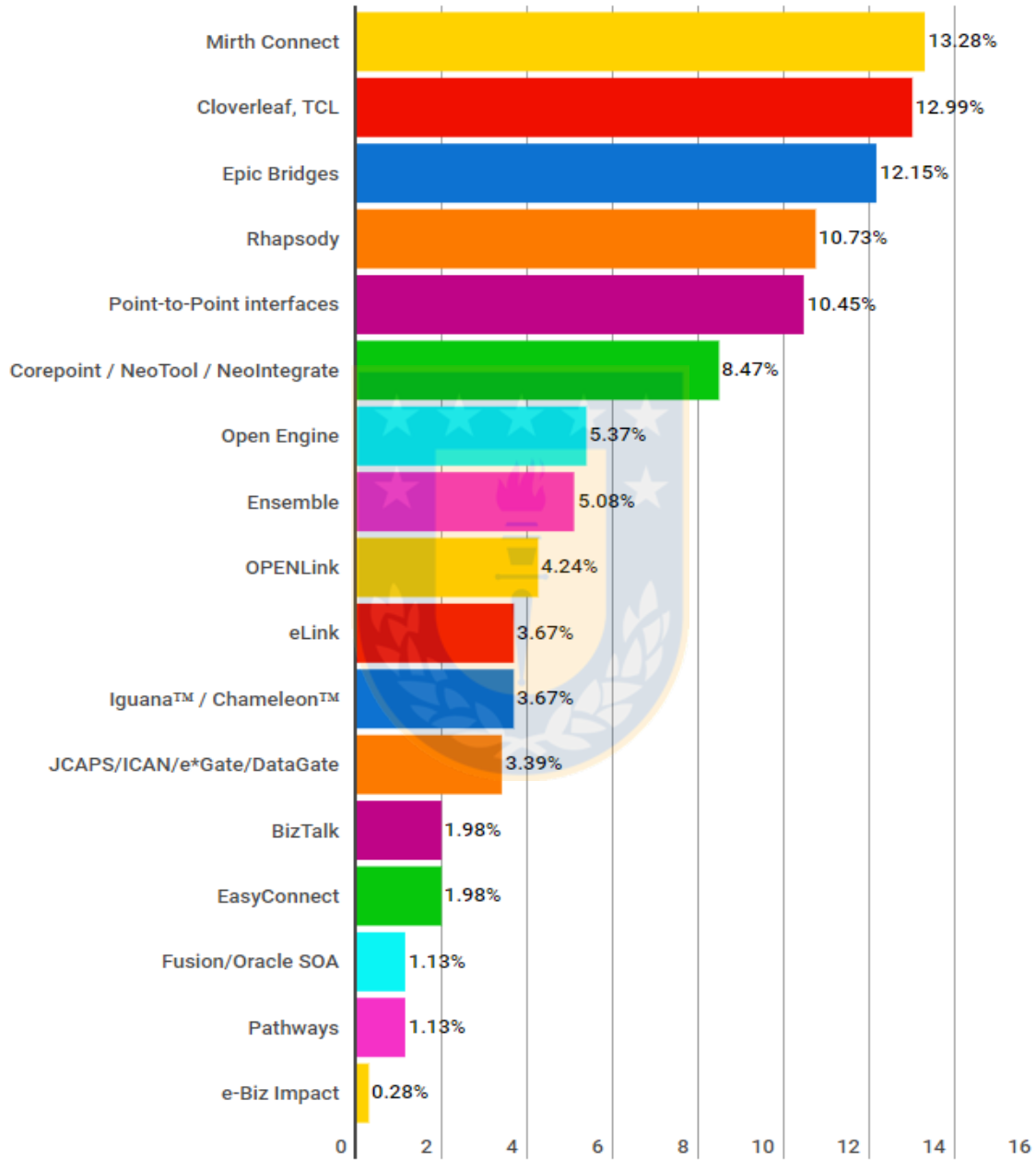


Figura 22, Fuente: CoreHealthTechnologies.

Anexo 3

Satisfacción de necesidades: los resultados que más abajo aparecen muestran el porcentaje de satisfacción con respecto a las necesidades que tiene la gente que utiliza estos integradores de datos.

Basado en la siguiente encuesta:

- ¿El producto cumplió con su necesidad declarada cuando lo seleccionó y lo compró? (es decir, ¿hizo lo que dijeron que haría?)
- Desde la selección inicial, ¿el motor continuó satisfaciendo todas sus necesidades organizativas?

Escala

0% - APENAS SATISFACE NUESTRAS NECESIDADES

100%: CUMPLE COMPLETAMENTE CON NUESTRAS NECESIDADES



Resolución: El Mirth logra un 84% en la satisfacción de necesidades lo que lo deja un 12% abajo del primero

Anexo 4

Usabilidad y Características: los resultados de la estadística dan el porcentaje con respecto al conjunto de características completas y fáciles de usar dentro de estos integradores.

Basado en la siguiente encuesta:

- ¿Cómo maneja el motor diferentes protocolos (TCP / IP, SFTP / SOAP, servicios web, etc. ...)?
- ¿Qué nivel de compatibilidad ofrece el motor con otro software y hardware?
- ¿Qué tan complejo / fácil es duplicar, reutilizar o copiar interfaces de servidor a servidor con su motor?
- ¿Cómo es el control y la documentación de la versión de la interfaz?
- ¿Tiene el motor los adaptadores de conectividad predefinidos?

Escala

0% - CONJUNTO DE CARACTERÍSTICAS INCOMPLETAS Y NO FÁCIL DE USAR

100% - CONJUNTO DE CARACTERÍSTICAS COMPLETO Y MUY FÁCIL DE USAR



Resolución: El Mirth al ser una herramienta de código libre enfocada en la funcionalidad, carece en lo amigable y fácil de usar, eso lo deja un 16% abajo.

Anexo 5

Costo de adquisición: los resultados arrojados a continuación muestran un porcentaje el cual indica que tan costoso es adquirir estos integradores

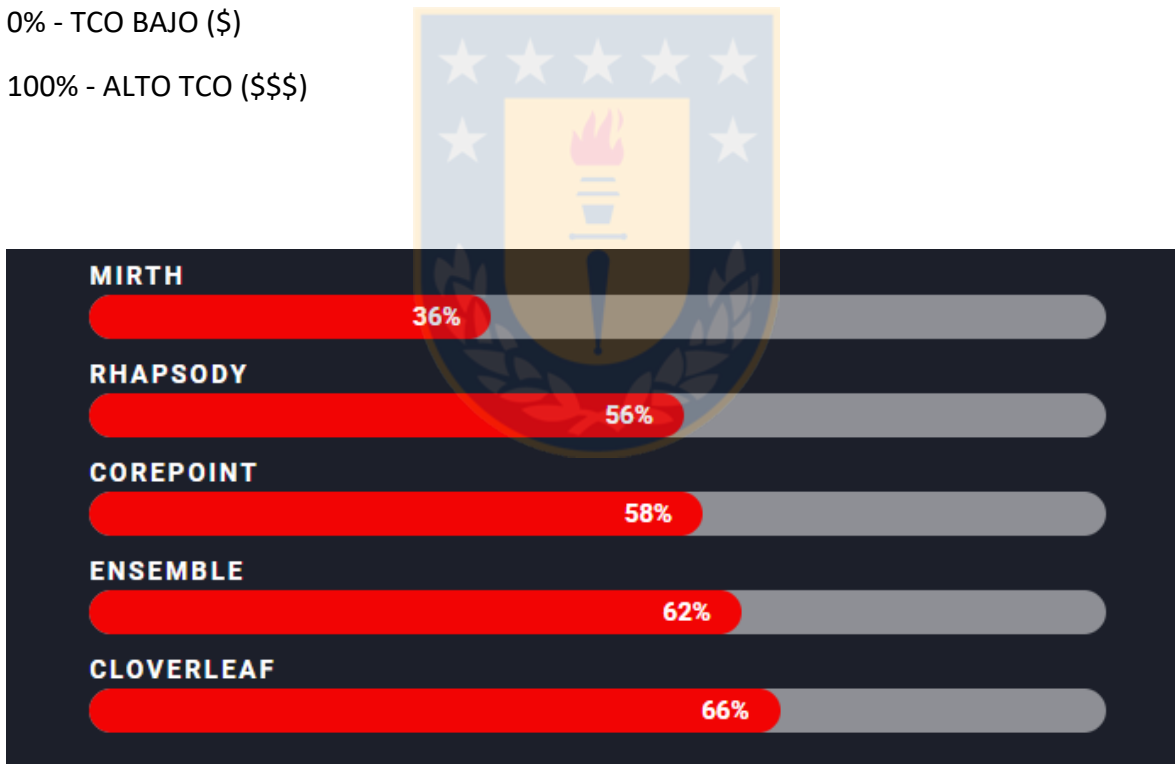
Basado en la siguiente encuesta:

- ¿Cómo fue el costo relativo del motor durante la compra / adquisición inicial?
- ¿Cómo es el costo relativo de propiedad de este motor 3-5 años después de la compra? (¿Hubo algo inesperado?)

Escala

0% - TCO BAJO (\$)

100% - ALTO TCO (\$\$\$)



Resolución: El Mirth resulta ser el de más bajo costo con relación al resto con un 20% de holgura.

Anexo 6

Soporte: Los siguientes resultados proporcionan el porcentaje con respecto al soporte que poseen estos integradores.

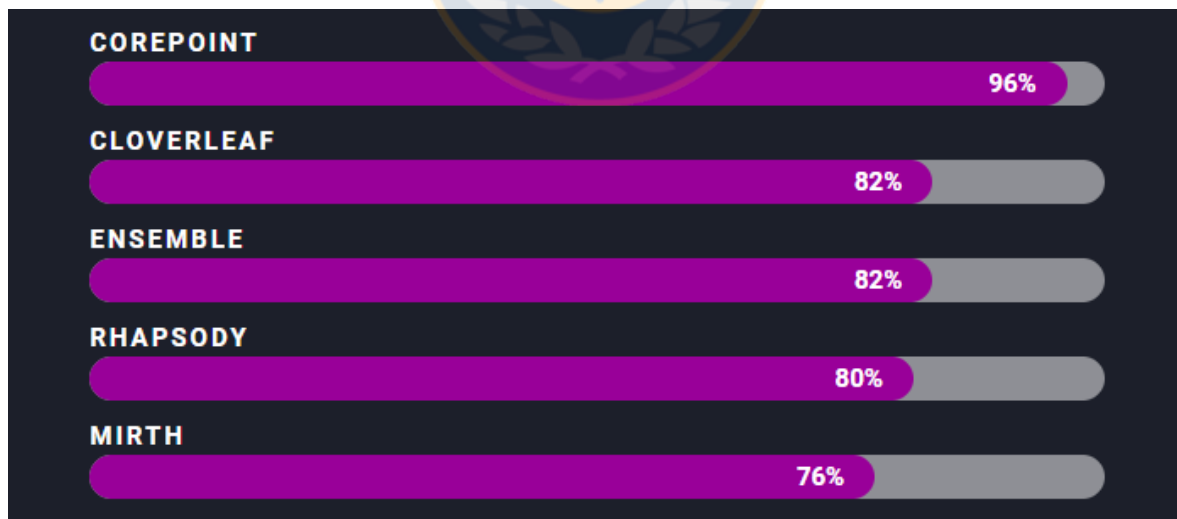
Basado en la siguiente encuesta:

- ¿Cómo están el soporte y los servicios del proveedor? ¿Su disponibilidad? ¿Costos?
- ¿Hay comunidades de usuarios para este motor? ¿Son valiosos?
- ¿Hay talento calificado en el mercado? (es decir, consultores calificados / certificados? ¿candidatos de Perm?)
- ¿Qué más, si hay alguno, puede decir sobre el soporte y los servicios disponibles para este motor?

Escala

0% - POBRE

100% - EXCELENTE



Resolución: El Mirth resulta tener una buena comunidad, pero carece de cursos oficiales de no pago, no así como el resto de las plataformas que son de pago y muy costosas, lo deja 20% más abajo.

