



Facultad de Ingeniería

Escuela de Ingeniería Civil Biomédica

Desarrollo de un Prototipo Informático para la Verificación de Mensajes HL7 2.5.1 de la Guía de implementación Chilena de Admisión de Urgencias

Por

Christian Alexis Vargas González

Trabajo de Título para optar al Título de
Ingeniero Civil Biomédico

Prof. Guía: M.Sc César Galindo

Diciembre 2014

Dedicatoria

Dedico este trabajo a mi familia, por todo el apoyo brindado durante estos largos y extenuantes años de sacrificio; a mis compañeros de universidad, por la grata compañía en los momentos difíciles; y a los profesores de la carrera, por compartir sus conocimientos y buenas aptitudes hacia la enseñanza.

“Los caminos de la vida pueden ser difíciles, pero con el apoyo de tu familia y amigos es más fácil recorrerlos”

Christian Vargas G.

Agradecimientos

Agradezco a mis padres, Luis y Verónica, por el apoyo constante que me han dado a lo largo de mi vida; a mi hermana Tamara, por los momentos divertidos que hemos pasado; a mi abuela Laura, por el cariño y preocupación; a mis primos, en especial a Rodrigo por compartir sus conocimientos; a mis amigos de universidad María José, Benjamín y Monserrat, por todos los momentos que compartimos juntos; y a mi profesor guía Cesar Galindo, por el tiempo dedicado para desarrollar este trabajo.

Christian Vargas G.

Resumen

Palabras claves: Interoperabilidad, Mensajes HL7 2.5.1, Guía de implementación Chilena de Admisión de Urgencia, Herramienta de verificación.

Resumen: *El presente trabajo de título tiene como objetivo el aplicar la “Guía de Implementación de Urgencias Chile” por medio del desarrollo de un prototipo de sistema informático en salud, capaz de verificar la composición de los mensajes HL7 V2.5.1 estrictamente definidos para los procesos de Admisión de Urgencia en Chile.*

En la actualidad los Sistemas informáticos en Salud implementados en Chile disponen la adopción de estándares de información en salud según lo especificado por cada fabricante. Internacionalmente cada país dispone del uso de Guías de Implementación donde se especifica el uso adecuado de los estándares para lograr la interoperabilidad de los sistemas. Por otro lado, en Chile solo se cuenta a la fecha con la Guía de Implementación de Urgencias, mientras que son escasas las exigencias respecto a la interoperabilidad entre sistemas.

La propuesta consiste en diseñar e implementar un prototipo de “Sistema de Admisión de Urgencias” conformado por tres módulos de pruebas capaces de codificar y decodificar mensajes HL7 2.5.1, además de enviar y recibirlos a través de sus respectivos servidores de prueba. La programación del sistema se llevó a cabo por medio de la utilización de un lenguaje web para la compatibilidad multi-plataforma del prototipo.

El alcance del prototipo es para ser usado a nivel académico o en instituciones estatales que presten apoyo en el desarrollo de Guías de Implementación para la validación de mensajería en la interoperabilidad de sistemas de información en salud.

Tabla de contenido

1.	Introducción.....	13
2.	Objetivos.....	14
3.	Estado del arte.....	15
3.1	Situación internacional sobre interoperabilidad en sistemas de información en salud.	15
3.2	Situación nacional sobre interoperabilidad en sistemas de información en salud.	16
4.	Análisis de la problemática.	20
5.	Estudio del marco teórico.	21
5.1	Plataformas informáticas en salud.	23
5.2	Estándar HL7 2.5.1 y mensajes definidos en guía de implementación de urgencias chile.	23
5.3	Guías de implementación.....	25
5.4	Modelo funcional.....	26
6.	Diseño de la propuesta.	27
7.	Metodología.....	28
8.	Resultados.....	31
8.1	Diseño general del prototipo del sistema informático de admisión de Urgencias.....	31
8.2	Modelo funcional de módulos de prueba.	33
8.3	Diseño del modelo de codificación y decodificación de mensajería HL7.	34
8.4	Diseño del verificador de errores de mensajería HL7 de Admisión de Urgencia.	35
8.5	Implementación del diseño general del sistema.....	36
8.6	Implementación de servidores de recepción y envío de mensajes.	37
8.7	Implementación de codificador y decodificador de mensajería HL7.	38
8.8	Implementación de la herramienta funcional verificadora de mensajería HL7.....	38
8.9	Implementación de bases de datos de módulos de prueba.	39
8.10	Implementación de funciones a los módulos del prototipo del Sistema de Admisión de Urgencias.	40
8.11	Funcionamiento general del prototipo y correcciones.	42
9.	Discusiones.....	42
10.	Conclusiones.	43
	Referencias Bibliográficas	45

Trabajo de Título II 2014	Glosario.....	46
	Anexos.....	48
	Anexo 1. “Aplicación modelo funcional para módulos de prueba del sistema de admisión de urgencias”.....	48
	Anexo 2. “Implementación cliente-servidor para módulos de prueba del sistema de admisión de urgencias”.....	58
10	Anexo 3. “Implementación codificación y decodificación de mensajería hl7 de admisión de urgencias chile”.....	59
	Anexo 4. “diseño de bases de datos para módulos de prueba del sistema de admision de urgencias”.....	79
	Anexo 5. Guía de instalación “prototipo informático para la verificación de mensajes hl7 2.5.1 de la guía de implementación chilena de admisión de urgencias”.	82

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 de la guía de implementación chilena de admisión de urgencias.

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

13

Christian Alexis Vargas González

Escuela de Ingeniería Civil Biomédica, Universidad de Valparaíso, Chile

Palabras claves: Interoperabilidad, Mensajes HL7 2.5.1, Guía de implementación Chilena de Admisión de Urgencia, Herramienta de verificación.

1. Introducción.

Los sistemas informáticos en salud han sido un pilar fundamental en la gestión de transmisión de datos clínicos, administrativos y técnicos dentro de una institución sanitaria. La inadecuada comunicación entre plataformas de salud puede generar datos erróneos conduciendo a un tratamiento equívoco que pone en riesgo al paciente (ECRI, 2013). La OMS, por su parte, hace hincapié en promover el desarrollo de sistemas informáticos en salud (SIS) interoperables, que sean capaces de transmitir información bajo el cumplimiento de estándares de comunicación en salud (OMS, 2012) con el fin de garantizar el intercambio seguro de datos clínicos.

El Institute of Medicine of the National Academies (IOM) define la Interoperabilidad como “la Habilidad de los sistemas para trabajar juntos, en general gracias a la adopción de estándares. La interoperabilidad no es solamente la habilidad de intercambiar información sanitaria, sino que requiere la habilidad de entender lo que se ha intercambiado” (Institute of Medicine, 2004). A su vez, existen dos modos de comprender el concepto de Interoperabilidad:

- a) Interoperabilidad sintáctica: Se refiere al medio o canal por donde se transmitirán los mensajes o información entre dos o más sistemas (López, 2008).
- b) Interoperabilidad semántica: Busca el logro de la correcta interpretación y el uso efectivo de la información intercambiada (Selene Indarte), es decir, permite la comprensión del significado de la información transmitida entre dos o más sistemas.

Si bien el intercambio de información puede ser abordado mediante la Interoperabilidad sintáctica, no asegura que los datos tengan un significado en concreto por el sistema que los recibe (López, 2008). Es por ello que la Interoperabilidad semántica se torna fundamental en el proceso de intercambio de datos clínicos por medio de la adopción de estándares que permitan estructurar e interpretar el significado de la información. Aunque los estándares de comunicación en salud definen las instrucciones de la arquitectura de mensajes y documentos clínicos, se hace necesario adoptar con mayor especificidad el estándar a nivel país, pues “sin un acuerdo previo es prácticamente imposible que dos o más sistemas intercambien mensajes y lleguen a interoperar semánticamente” (Pazos, 2009). Por lo tanto, es necesario establecer Guías de Implementación que especifiquen técnicamente la semántica de la información que se desea transmitir.

En Chile, desde el año 2008, se está trabajando en la informatización de procesos clínicos claves dentro de las instituciones sanitarias se ha comenzado a exigir interoperabilidad a los proveedores que desarrollan SIS por medio de la adopción de estándares de información en Salud. No se sabe con certeza si existe cumplimiento de protocolos de Interoperabilidad entre plataformas de diversas empresas, ya que -a la fecha- solo la Guía de implementación de Admisión de Urgencias está en espera de validación por HL7 Chile.

La propuesta de este proyecto consiste en aplicar la “Guía de implementación Chilena de Admisión de Urgencias” por medio del diseño e implementación de un prototipo de sistema informático de admisión de urgencia que permita la verificación de la correcta estructuración de los mensajes HL7. Su finalidad, por lo tanto, apunta a que este proyecto sea un precursor base en el desarrollo de SIS de apoyo en la monitorización y verificación de la aplicación de guías de implementaciones de mensajería HL7 en Chile.

El desarrollo del trabajo de título está conformado por una introducción y definición de objetivos que darán cuenta de una presentación del tema de proyecto y las metas a fines. Luego continúa con un “Análisis de la problemática” y “Estado del arte”; el primero analizará las causas y consecuencias a dar solución, y el segundo desarrollará la perspectiva del desarrollo local y global de la interoperabilidad de sistemas informáticos en salud. Posteriormente se abordará la propuesta de solución”, que estará conformado por un marco teórico (que expondrá conocimientos previos del tema expuesto), diseño de la propuesta y la metodología empleada en el diseño e implementación de los módulos del prototipo del sistema de Admisión de Urgencias. Por último se expondrán los resultados, discusiones y conclusiones, que reflejaran el desarrollo logrado y los trabajos futuros relacionados con este proyecto.

2. Objetivos.

Objetivo general.

Aplicar la “Guía de implementación de Admisión de Urgencias Chile”, por medio del diseño e implementación de un prototipo de sistema informático para verificar la composición de mensajería HL7 2.5.1 de Admisión de Urgencias.

Objetivo específicos.

- Establecer el diseño de los módulos de prueba del prototipo del sistema informático de admisión de Urgencias.
- Implementar el prototipo del sistema informático de admisión de Urgencias y su comunicación interoperable.
- Probar funcionamiento y aplicar correcciones al prototipo del sistema informático de admisión de Urgencias.

3. Estado del arte.

El avance tecnológico dentro de los recintos sanitarios ha ido de la mano con el desarrollo de sistemas informáticos en Salud (SIS) que facilitan la recolección, trasmisión y gestión de datos en los procesos administrativos, clínicos y de equipamiento tecnológico. Los SIS, al ser implementados correctamente en centros hospitalarios, otorgan beneficios como “la reducción de errores médicos (seguridad), mejor calidad de prestación, mejor documentación clínica, reducción en duplicación de estudios y disminución de errores en administración de medicamentos” (MINISTERIO DE SALUD, 2006). Sin embargo, son muchas las plataformas informáticas en salud que han sido desarrolladas bajo diversos diseños y estándares de comunicación, por lo que requieren una adecuada trasmisión que asegure la integridad de los datos; “la presencia de datos incorrectos en este tipo de sistemas, puede conducir al erróneo tratamiento, potencialmente resultando en daño al paciente” (Ecri, 2013). Por tanto, se requiere el uso de interoperabilidad para dar conformidad a la seguridad, trasmisión y semántica en la comunicación de SIS.

Abordaremos en esta sección, por lo tanto, la situación actual de la interoperabilidad y estándares de comunicación en salud, desde la perspectiva del contexto internacional y la situación en el plano nacional.

3.1 Situación internacional sobre interoperabilidad en sistemas de información en salud.

La interoperabilidad y la aplicación de estándares de comunicación en los sistemas informáticos en salud es un tema de relevancia internacional. La OMS (Organización Mundial de la Salud) en un foro realizado en el año 2012, que contó con la participación de más de 200 personas de 54 países, ha discutido el tema de la estandarización e interoperabilidad en los sistemas informáticos de salud. De las temáticas expuestas en el foro se destacan las discusiones acerca de la “contextualización de salud, la perspectiva de los países acerca de la aplicación de estándares sobre datos sanitarios a través de la discusión de aspectos críticos en la interoperabilidad, y el acceso libre a los datos sanitarios” (OMS, 2012).

El estándar de comunicación de datos y documentos clínicos con mayor uso a nivel mundial es Health Leven Seven, con más del 90% de presencia en los sistemas de información en salud (Health Leven Seven, 2014). Health Leven Seven (Nivel Siete) o en sus siglas HL7, es “una referencia al último nivel del modelo de comunicaciones para interconexión de sistemas abiertos (OSI Open Systems Interconnection) de la Organización Internacional para la Estandarización (ISO International Organization for Standardization)” (Health Leven Seven, 2014). El estándar HL7, en sus inicios (1987), comenzó siendo una guía para la comunicación de datos clínicos, pero la creciente necesidad de integración de sistemas dispuestos en múltiples hospitales, así como las nuevos Software en Salud, han marcado la creación de ampliar el estándar HL7 a nuevos campos como documentos clínicos, mensajes, reglas, modelos de referencia, entre otros. Existen otros estándares como OPENEHR o DICOM, además.

Para la correcta adopción de estándares internacionales de comunicación en salud “es necesario definir las llamadas “guías de implementación”, las cuales especifican acuerdos entre distintos actores para definir cómo implementarán uno o varios estándares” (Pazos, 2009). En el caso de HL7, este dispone de múltiples estándares donde se reúne gran información de los procesos clínicos con

una mirada internacional; es necesario, por tanto, especificar técnicamente el contenido a usar del estándar acorde a la realidad local.

Existe una organización llamada IHE (Integrating the Healthcare Enterprise) que define la utilización de los estándares mediante “perfiles de integración”. IHE es una organización sin ánimo de lucro, conformada por empresas, sociedades científicas, hospitales, etc. Su objetivo es desarrollar y promover la adopción coordinada de estándares ya existentes, para ofrecer soluciones a los problemas de interoperabilidad en el ámbito de la salud. Mediante la conformación de comisiones IHE, se establecen marcos técnicos para implementar “perfiles de integración” a la realidad de cada país. “IHE define los actores, transacciones y el contenido de la información necesaria para abordar el caso de uso clínico y la interoperabilidad de las plataformas en salud” (IHE, s.f.).

En lo relacionado a pruebas de interoperabilidad, IHE proporciona demostraciones internacionales llamadas “Connectathon”, cuyo propósito trata de “poner a prueba los sistemas de información en salud mediante el uso de validadores que comprueben la conformidad de la correcta aplicación de los perfiles IHE (guías de implementación)” (IHE Europe, 2014) para verificar la correcta interoperabilidad entre dos o más sistemas (Ver Figura 1). Lo anterior, por medio de actividades interactivas a gran escala en un entorno controlado y neutral.

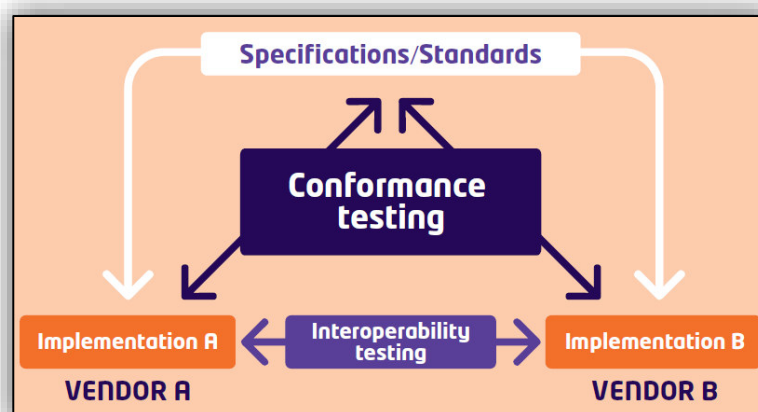


Figura 1: “Test Connectathon IHE”. Fuente: (IHE Europe, 2014)

3.2 Situación nacional sobre interoperabilidad en sistemas de información en salud.

Desde el año 2006 Chile se ha planteado como política de estado la informatización de procesos claves en las instituciones sanitarias, como estrategia para contrarrestar los cambios demográficos y epidemiológicos. En el Libro Azul (MINISTERIO DE SALUD, 2006) se plantea la agenda de proyectos de tecnologías de información para los próximos años.

En cumplimiento de los objetivos declarados en el “Libro Azul”, se crea la estrategia SIDRA en el año 2008. El “Proyecto SIDRA” o “Sistema de información de la red asistencial” establece como meta “habilitar el registro clínico electrónico en toda la red asistencial pública del país, desde hospitales de alta complejidad hasta los establecimientos de la APS, haciendo énfasis en la integración de los componentes de la red asistencial de salud” (Salud-e, s.f.). Para la implementación del proyecto, se licitó el convenio marco “CM-06-2008 de Software de Salud y Servicios Informáticos Asociados” en

diciembre de 2008 (MercadoPúblico, 2008), siendo adjudicado a tres empresas con experiencias en el extranjero. Dentro de los sistemas priorizados a informatizar tanto en Hospitales como en centros de atención primaria se encuentran: “Agenda”, “Referencia y Contra referencia”, “Registro de Población en Control”, “Dispensación de Fármacos” y “Urgencia”. Posteriormente en el año 2011 se incorporaron los procesos: “Gestión de Pabellones”, “Gestión de Medios de Diagnóstico”, “Gestión de Camas” y “Archivo (enlace entre ficha electrónica y en papel de un mismo paciente)”. La “Figura 2” presenta un resumen de los sistemas a informatizar donde se aprecia el proceso, el sub-proceso, la funcionalidad, la sigla y el ámbito asistencial donde será desarrollado.

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

Proceso	Sub-Proceso	Funcionalidad	SIGLA	Ámbito Asistencial
Soporte Administrativo	Agenda	Agenda, Admisión, Inscripción, Gestión de Citas	Agenda	Primaria y Especialidad
	Referencia	Derivación en la Red	RCR	Primaria y Especialidad
	Archivo	Gestión de Archivo	Archivo	Especialidad
Atención Clínica	Atención Ambulatoria	Atención Clínica	AA-AC	Primaria y Especialidad
		Solicitud de Medios Diagnósticos	AA-SMD	Especialidad
	Hospitalización	Gestión de Camas	H-GC	Especialidad
		Atención Clínica	H-AC	Especialidad
		Solicitud de Medios Diagnósticos	H-SMD	Especialidad
	Urgencia		URG	Primaria y Especialidad
		Solicitud de Medios Diagnósticos	URG-SMD	Especialidad
Soporte Clínico	Proceso Quirúrgico	Tabla Quirúrgica	PQ-TQ	Especialidad
		Intervención Quirúrgica	PQ-IQ	Especialidad
		Post Anestesia	PQ-PA	Especialidad
	Dispensación de Fármacos		DDF	Primaria y Especialidad

Figura 2: Procesos a informatizar. Fuente: (Ministerio de Salud, 2013).

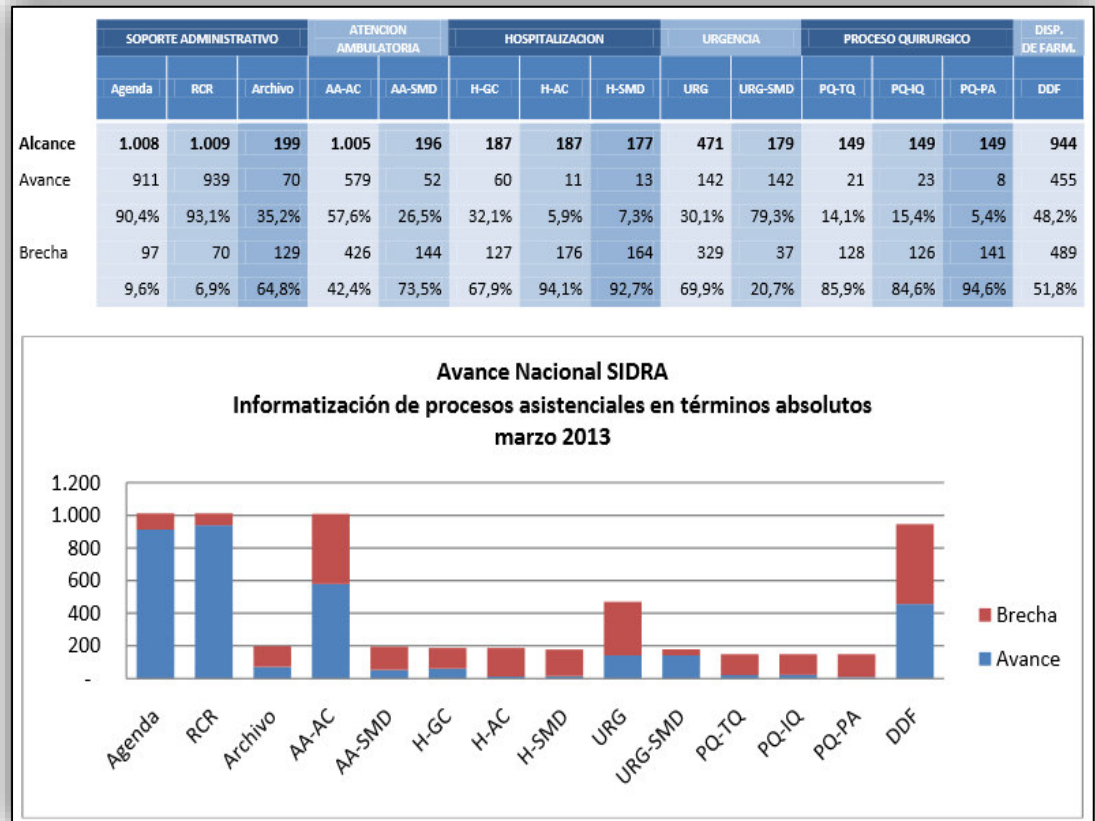


Figura 3: “Estado de avance Proyecto SIDRA. Fuente (Ministerio de Salud, 2013)”.

En el último informe emitido en marzo del 2013 referente al estado de implementación del Proyecto SIDRA, se registraron 1084 instituciones adheridas al proyecto con un porcentaje de mayor avance en los procesos de “Referencia contra referencia” con 93.1%, “Agenda” con un 90.4% y “Urgencias” (solicitud de medios diagnósticos) con un 79.3%, ver Figura 3. Dentro de las principales dificultades en la implementación del proyecto se menciona la “falta de estandarización de los procesos clínicos principalmente en los hospitales y la falta de estándares de información adecuados al propósito de contar con un registro clínico nacional” (Salud-e, s.f.).

A la fecha en el sitio web “Estrategia digital de salud” del Ministerio de Salud, se mencionan los estándares a ocupar, ver Tabla 1. No se ha encontrado ninguna resolución o decreto que respalde la decisión de adoptar estos estándares. Sin embargo, en el año 2010 la ACTI (Asociación Chilena de Empresas de Tecnología de Información A.G) emitió un documento llamado “ACTI Salud Estándares V1.0.6” (Hoffmann, 2011) donde recomienda los tipos de estándares internacionales a utilizar en los sistemas de información de salud en Chile.

ESTÁNDARES SISTEMAS INFORMÁTICOS EN SALUD.	
Identificación de los pacientes/prestadores.	Sexo, Id paciente, Comunas, tipos de establecimientos etc.
De comunicaciones	TCP/IP, HTTP, XML, SOAP, WDSL, UDDI, Web Services
De seguridad	<ul style="list-style-type: none"> – XMLDSIG: Firma digital, PKI, certificados – Kerberos: protocolo de comunicación segura en redes inseguras – SAML: Security Assertion markup language – SSL: Seguridad de la capa de transporte – HL7 RBAC (Role Based Access Control): define derechos de acceso según rol del usuario
De mensajería	<ul style="list-style-type: none"> – Datos clínicos (HL7) – Datos financieros y transacciones (X12N) – Imágenes (DICOM) – Prescripción de farmacias (NCPDP) – Instrumentos médicos (IEEE, ASTM)
De vocabularios o terminología médica	<ul style="list-style-type: none"> – Laboratorio: LOINC – Anatomía patológica: SNOMED CT
Morfologías, problemas, localizaciones.	<ul style="list-style-type: none"> – Hallazgos clínicos: SNOMED CT, UMLS – Diagnósticos, Motivos de consulta: SNOMED-CT, CIE10, CIAP – Procedimientos: LOINC, CIE9 – Enfermería: NANDA (diagnósticos), NIC (Intervenciones), NOC (Resultados).

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

19

Tabla 1: “Estándares de comunicación de sistemas informáticos en salud utilizados por el Ministerio de Salud”. Fuente: (Salud-e, s.f.)

En lo referente a la interoperabilidad de las plataformas desarrolladas en el Proyecto SIDRA, “se está consiente que la solución al problema de la interoperabilidad es de largo aliento y pasa por construir una entorno tecnológico favorable así como definiciones que van a tomar tiempo consensuar” (Salud-e, s.f.). Sin embargo, se priorizó la implementación de interoperabilidad en las siguientes plataformas:

- Registro Nacional de Inmunizaciones.
- Registro Nacional de Lista de Espera.
- Referencia.
- Contra Referencia.
- Certificador previsional Fonasa.
- Sistema de Gestión de Garantías – SIGGES

Dentro de las últimas líneas de acción del proyecto SIDRA, se ha estado trabajando en un nuevo convenio marco, que incluya directrices enfocadas a la interoperabilidad de sistemas para facilitar la automatización en los procesos clínicos. El nuevo convenio marco hace referencia a la interoperabilidad, exigiendo a los oferentes de la licitación que demuestren sus capacidades de comunicar los diversos sistemas dentro de la red asistencial, entre diferentes redes, con el nivel central y, eventualmente, con prestadores privados (Ministerio de Salud, 2011). Sin embargo, se

habla de soluciones de integración con los sistemas actuales, lo que no repercute en la aplicación de interoperabilidad por medio de la adopción de estándares a nivel país.

Por último, a la fecha se ha desarrollado solo una guía de implementación de mensajería HL7 para el proceso de “Admisión de Urgencias”. Esta guía fue elaborada por Juan Pablo Urzúa entre el año 2013-2014, y aborda desde el levantamiento de procesos, identificación y homologación de eventos, identificación de actores y roles del proceso, y definición del contenido y estructuración de la mensajería 2.5.1 para proceso de Admisión de Urgencias (Urzúa, 2014). La guía chilena de implementación de admisión de Urgencias se encuentra en etapa de corrección y aprobación por el capítulo Chileno de HL7, que ha sido nuevamente abierto el presente año.

4. Análisis de la problemática.

A nivel mundial, las tecnologías de información en salud han proporcionado una base sustancial en el proceso de captura y almacenamiento de datos clínicos en los diferentes servicios existentes al interior de los recintos sanitarios. Dentro de las causas que explican por qué razón los sistemas de información deben ser interoperables, se encuentra el artículo de la revista ECRI, “TOP 10 HEALTH TECHNOLOGY HAZARDS FOR 2014” (Top 10 de tecnologías peligrosas para la salud), el cual hace referencia al riesgo presente en el tratamiento de pacientes producto del aumento de sistemas de información en salud y la integridad de los datos.

Los avances nacionales de plataformas informáticas en salud han manifestado una inconsistencia en la implementación debido a la falta de regulación o adecuado uso de los estándares de comunicación en salud, lo cual es sin duda un problema a la hora de integrar la data. Para la correcta aplicación de estándares se requiere el uso de “Guías de implementación” (Pazos, Super estándares en informática médica, 2009) o el uso de “Perfiles técnicos” (IHE, s.f.). En Chile existen una escasez en el desarrollo de guías de implementación, siendo solo la “Guía de implementación de Urgencias” la única en vías de corrección y validación a la fecha (Urzúa, 2014). Por otro lado, las políticas de estado desde el 2006 han estado dirigidas hacia la informatización de procesos clínicos; no obstante, dentro de la información recopilada se ha avanzado muy poco en lo relacionado con interoperabilidad, a pesar de que los estándares a usar han sido mencionados (Salud-e, s.f.) y existen avances en la integración de plataformas (Ministerio de Salud, 2011). En efecto, la falta de políticas en esta materia se refleja en la poca normalización de procesos clínicos (Salud-e, s.f.) y de desarrollos de guías de implementación.

El último avance del proyecto SIDRA con fecha de marzo del 2013 (Ministerio de Salud, 2013), refleja un gran avance de implementación de plataformas informáticas en salud en más de 1084 instituciones públicas a lo largo de la red asistencial lográndose el mayor progreso en: “Referencia contra referencia” con más de 939 sistemas implementados de un total de 1009 instituciones (93.1% de avance), “Agenda” con más 911 sistemas implementados de un total de 1008 instituciones (90.4% de avance) y “Urgencias” (solicitud de medios diagnósticos) con 142 sistemas implementados de un total de 172 instituciones (79.3% de avance). Si bien se han implementado gran parte de los principales sistemas, el informe no hace referencia a pruebas de validación de interoperabilidad de los sistemas desarrollados. Además, las únicas exigencias sobre interoperabilidad son las emitidas en la propuesta del nuevo documento de convenio marco para el proyecto SIDRA 2 (Ministerio de Salud, 2011), que dejan a cargo a los proveedores de la implementación de Interoperabilidad de los SIS.

Las experiencias internacionales mencionadas en el estado del arte, establecen que el deber de homologar o especificar el uso de estándares de información de salud por medio de la elaboración de guías de implementación es del Estado. A la fecha existe solo una Guía de implementación construida en Chile, por lo que hay una gran brecha por cubrir donde el Estado debe ser líder en esta etapa que integre los requerimientos de información desde procesos clínicos a nivel país. Los sistemas informáticos en salud deben ser interoperables y el Estado debe ser la entidad que exija y regule esta capacidad a las empresas proveedoras de software en salud. De todas las plataformas implementadas en el proyecto SIDRA no se puede verificar si los sistemas satisfacen interoperabilidad debido a la falta de herramientas que aludan a lo definido en las “Guías de implementación”. Todo lo anterior repercute en una dificultad técnica que permita demostrar interoperabilidad en los sistemas de información implementados según los estándares de información en salud usados en Chile.

Los efectos de no validar la interoperabilidad entre las plataformas informáticas repercuten en una falta escasa capacidad para determinar si los proveedores de sistemas han aplicado las guías de implementación, produciendo un bajo control de la calidad de la transmisión de datos lo que aumenta las implementaciones ineficientes de interoperabilidad en salud. La escasez de interoperabilidad, en consecuencia, tiene su mayor efecto en el aumento del riesgo de eventos adversos en el paciente.

Todo lo anterior se ve reflejado en el diseño del árbol de problema de la Figura 4, donde se esquematiza las causas, el problema central y las consecuencias de la problemática estudiada.

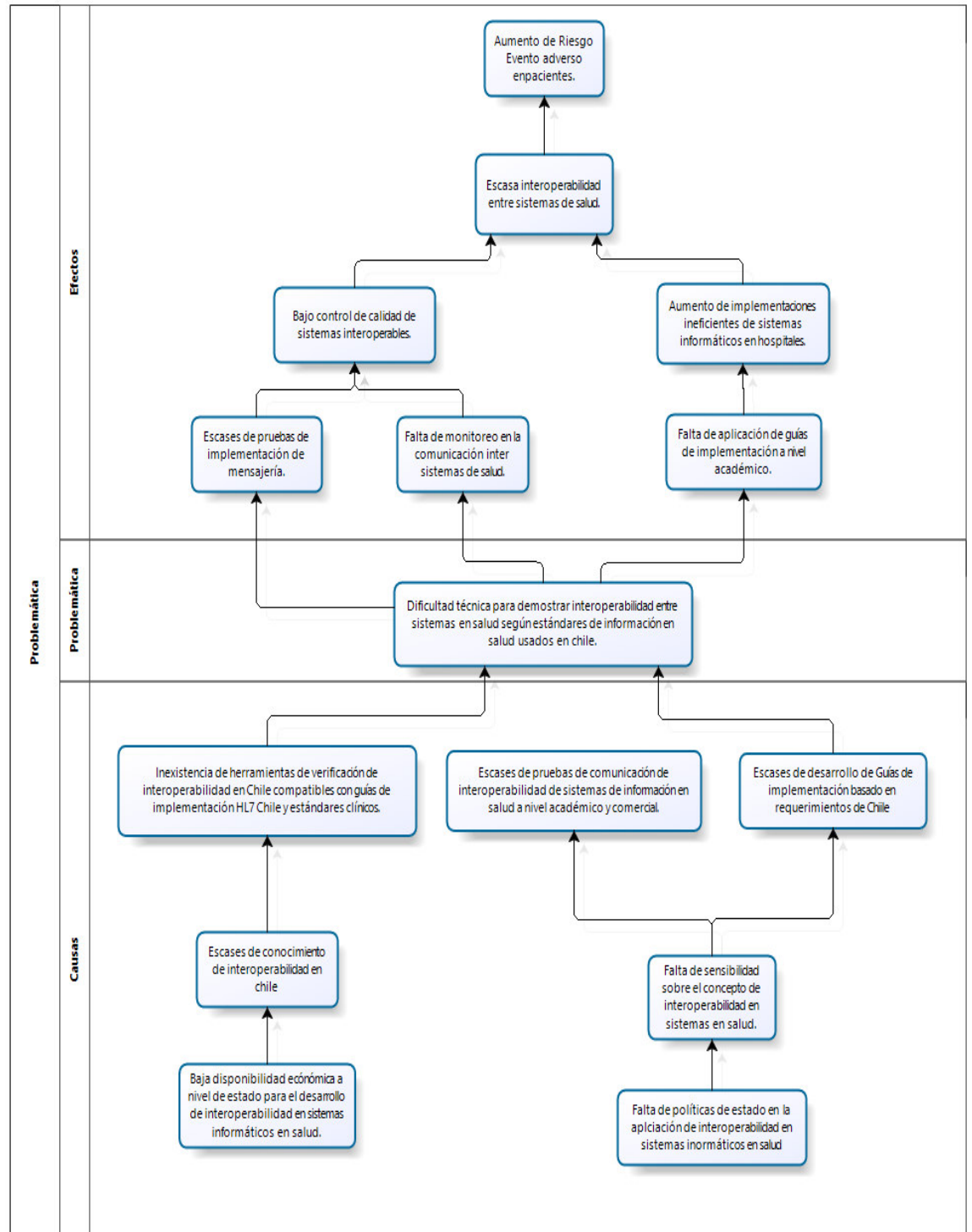
5. Estudio del marco teórico.

En el presente marco teórico, se establecerán los conceptos básicos para comprender el desarrollo de la propuesta de solución a la problemática identificada en el estado del arte, la cual identifica las dificultades e incapacidades actuales de entes reguladores públicos en demostrar si los sistemas desarrollados y comercializados en Chile son interoperables.

El marco teórico estará estructurado en cuatro secciones:

- Plataformas informáticas en salud.
- Estándar HL7 2.5.1 y mensajes definidos en Guía de Implementación de Urgencias Chile.
- Guías de implementación.
- Modelo Funcional.

Figura 4: "Árbol de problema problemática del proyecto"
Fuente: Diseño propio.



5.1 Plataformas informáticas en salud.

Las plataformas informáticas en salud son herramientas que permiten la recolección, procesamiento, análisis y transmisión de datos necesarios para organizar y operar en servicios de salud. También da soporte en la investigación y planificación del control de enfermedades.

La Sociedad británica de Informática Médica (British Medical Informatics Society, s.f.) define la informática médica como:

"... el conocimiento, habilidades y herramientas que permiten el intercambio y uso de la información para ofrecer atención sanitaria y promover la salud".

"... el nombre de una disciplina académica desarrollada y perseguida en las últimas décadas por una comunidad científica en todo el mundo, comprometida en el avance, promoción y enseñanza del conocimiento sobre la aplicación de tecnologías de información y a la asistencia sanitaria. - el lugar donde se entrecruzan la salud, las ciencias de la información y de la computación, psicología, epidemiología, y la ingeniería"

Una institución sanitaria, cuenta con múltiples sistemas para el apoyo de la operatividad hospitalaria y cuidado del paciente. Estas plataformas conforman lo que se conoce como HIS o Sistema de Información Hospitalaria y se pueden clasificar como (Arevalo):

- a) **Sistemas de apoyo a la gestión administrativa (ERP):** Son plataformas que dan soporte a procesos como contabilidad, finanzas, abastecimiento y gestión no clínica.
- b) **Sistemas de apoyo a la gestión clínica:** Son plataformas que se clasifican en dos tipos según el tipo de almacenamiento de información.
 - A) Sistema de gestión clínica: Sistemas compuestos por bases de datos más interfaz, cuya misión es el almacenamiento de información referente a los procesos clínicos involucrados en la asistencia del paciente; Registro clínico electrónico (RCE), Sistema de información radiológico (RIS), Sistema de información de laboratorio (LIS), Agenda electrónica, Gestión de camas, Registro de enfermería, etc.
 - B) PACS: Sistemas especializados en el almacenamiento y gestión de imágenes médicas.
- c) **Sistemas de apoyo a la gestión y mantenimiento de equipamiento:** Son plataformas enfocadas en el control del equipamiento médico e industrial por medio de módulos tales como: inventario, gestión de mantenimiento, gestión de insumos y accesorios, gestión de compras y trabajos, etc.

5.2 Estándar HL7 2.5.1 y mensajes definidos en guía de implementación de urgencias chile.

El estándar HL7 o "Health Level Seven" es una referencia al último nivel del modelo de comunicaciones para interconexión de sistemas abiertos (OSI Open Systems Interconnection) de la Organización Internacional para la Estandarización (ISO Internacional Organization for

Standardization) (Health Level Seven, 2014). El estándar define un marco referencial para la adopción de mensajes clínicos relacionados a ciertos eventos disparadores dentro del proceso hospitalario.

A continuación serán mostradas algunas definiciones recopiladas del documento “Glosario de términos de HL7 internacional” (Health Level Seven International, 2012):

- a) **Mensaje:** Un mensaje es la unidad central de los datos transferidos entre sistemas. Está compuesto por un grupo de segmentos en una secuencia definida. Cada mensaje HL7 está asociado a un tipo de mensaje en particular que define su propósito.
- b) **Segmento:** Un segmento HL7 es una agrupación lógica de campos de datos. Segmentos de un mensaje puede ser requerido u opcional. Pueden ocurrir sólo una vez en un mensaje o es factible que se repita. Cada segmento se identifica por un código de tres caracteres únicos conocidos como la ID del segmento.
- c) **Campo:** Un campo es una cadena de caracteres definida por uno de los tipos de datos dentro del segmento del mensaje de HL7.
- d) **ACK:** Mensaje de reconocimiento general. El mensaje ACK se utiliza para responder a un mensaje cuando se ha producido un error que impide procesamiento de la solicitud o en que la aplicación no define un tipo de mensaje especial para la respuesta.

Figura 5: Estructura mensaje HL7. Fuente: Elaboración propia.

```
MSH ^~&|||020046&061114||ADT^A02^ADT_A02|256|256^T|2.5.1<cr>
EVN |A02|020046&061114||08|||06^Valparaíso_San_Antonio|<cr>
PID ||3|17309623^2||Vargas^González^ChristianAlexis|<cr>
PV1 ||E^Urgencia|||001^López^Rodríguez^Antonio^AL||MED^Servicio Medicina|<cr>
ROL ||AD|PP^Profesional en atención primaria|01^Gomez^Vega^Luis|||025653215|<cr>
```

El ejemplo presentado en la “Figura 5”, muestra un mensaje HL7 donde se destaca la estructura principal que compone el mensaje. Destacado en amarillo se aprecia el tipo de evento disparador asociado al mensaje. En color rojo se aprecian los segmentos, iniciados con su propio ID, luego sus componentes separados por un delimitador “|”, donde en color verde se aprecia un ejemplo de campo, para finalizar con el delimitador de fin de segmento encerrado en un círculo rojo. Encerrado en un círculo azul se aprecia la declaración de uso de delimitadores los cuales se definen como separador de campos, separador de componentes, separador de repetición, carácter de escape y separador de subcomponentes respectivamente.

La definición de los eventos y estructura del mensaje HL7 con la adopción de segmentos y contenido de los campos fueron definidos en la “Guía de implementación de Urgencias Chile” (Urzúa, 2014). A continuación observaremos el extracto del resumen de la definición de eventos, su homologación en HL7 y la descripción general del contenido del mensaje.

Evento	Código	Homologación en HL7	Segmentos asociados
Atención.	A01	Notificación de Admisión/Visita	MSH, EVN, PID, PD1, NK1, PV1, ROL, DG1, GT1
Necesidad de atención.			
Derivación de paciente	A02	Transferencia de un Paciente	MSH, EVN, PID, PV1, ROL, DG1
Envío a Servicio Médico Legal o anatomía patológica			
Cierre de la atención	A03	Dar de Alta/ Fin de la Visita	MSH, EVN, PID, ROL, GT1
Alta de paciente			
Salida de paciente			
Cierre del caso	A04	Registro de un Paciente	MSH, EVN, PID, GT1
Abrir DAU (Datos de Admisión de Urgencia)			
Recepción de los afectados	A06	Cambiar un paciente externo a uno interno.	MSH, EVN, PID, DG1, GT1
Clasificación de la atención			
Realizar evaluación previa y priorización			
Validar previsión	A08	Actualización de la información del paciente	MSH, EVN, PID, PD1, NK1, ROL
Validar otros seguros			
Entrega antecedentes a médico regulador			
Asignar paciente a sala de espera	A09	Salida de un paciente - Seguimiento	MSH, EVN, PID, PV1
Entrega de Paciente			
Entrega de los afectados	A10	Llegada de un paciente – Seguimiento	MSH, EVN, PID, PV1
Entrega de Paciente			
Preparar la atención	A14	Admisión pendiente	MSH, EVN, PID, PV1
	A15	Transferencia pendiente	MSH, EVN, PID, NK1, PV1, ROL, DG1
	A17	Intercambio de Pacientes	MSH, EVN, PID, PD1, NK1, PV1, ROL, DG1, GT1
Atención inicial de los afectados	A28	Agregar Información de persona o paciente	MSH, EVN, PID, PV1, PR1
Evaluación médica			

Figura 6: Resumen de los eventos y su homologación a mensaje HL7 (determinación de segmentos) Fuente: (Urzúa, 2014)

5.3 Guías de implementación.

La implementación de los estándares de información no se desarrolla de forma directa, pues requiere de la especificación de las estructuras particulares y atributos en un determinado contexto acorde al proceso sanitario que se desea informatizar. En el caso del estándar HL7 su implementación “no es directa ni trivial, y requiere un esfuerzo de especificación extra sobre la especificación del propio estándar. Esto es debido a que el estándar de mensajería intenta abarcar tantos casos particulares de una sola forma genérica que los mensajes no son usables de forma directa, y debe especificarse cómo se usarán” (Pazos, Informática Médica).

Una “Guía de Implementación” es una especificación de la aplicación de un estándar de información en salud vinculado directamente a un proceso clínico de un país. En el caso de la mensajería HL7 se especifica el uso de la conformación de los mensajes mediante la homologación de eventos disparadores detectados en el resultado del levantamiento de un proceso clínico. Posteriormente, se seleccionan los segmentos más apropiados, se les otorga carácter de obligatoriedad, repetitividad, etc. Finalmente, dentro de los segmentos se especifica con mayor detalle la información contenida en cada campo otorgando mayor especialización según la forma de expresar los datos más recurrentes. Por ejemplo, el uso del RUT dentro del estándar de HL7 no existe, este debe ser adaptado de acuerdo a la realidad de Chile; lo mismo sucede con muchas estructuras semánticas, atributos y vocabularios específicos usados en diferentes estándares de información en Salud.

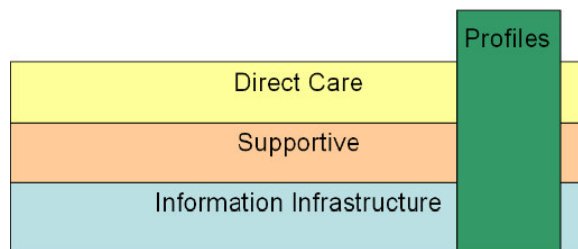
El proceso para diseñar una guía de implementación se resume en:

1. Levantamiento de proceso clínico.
2. Definición de flujo de datos, actores del proceso.
3. Identificar eventos disparadores.
4. Homologación de evento en HL7.
5. Definir diagrama de conectividad.
6. Definición de estructuras particulares para cada evento homologado como mensaje.
7. Validar propuesta referente Nacional.
8. Validar propuesta en capítulo HL7 Chile y luego HL7 internacional.

5.4 Modelo funcional.

Es un tipo de modelo que permite estructurar las funcionalidades de un sistema que se desea diseñar e implementar. Fue creado en el año 2004 por “Health Level Seven” para facilitar el desarrollo de sistemas informáticos en salud por medio de la simplificación de las funciones presentes en las diversas categorías en ámbito de la salud, (Health Level Seven, 2004). Este modelo describe tres categorías para la elección de funciones del diseño de un sistema informático en salud: funciones de cuidado del paciente, funciones de soporte y funciones de infraestructura del sistema; donde cada clasificación contiene funciones cada vez más especializadas, tal como se muestra en la “Figura 7”.

Este modelo permitirá diseñar y definir las funcionalidades del prototipo informático que se desea implementar, desde una perspectiva estandarizada de acuerdo a los diferentes tipos de funciones definidas, lo cual es una ventaja en el diseño de sistemas informáticos en Salud para satisfacer y asegurar el cumplimiento de especificaciones, y características propias dentro de los procesos que se desean dar soporte en un recinto sanitario.



Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

Direct Care	DC1.0	Care Management
	DC2.0	Clinical Decision Support
	DC3.0	Operations Management and Communication
Supportive	S1.0	Clinical Support
	S2.0	Measurement, Analysis, Research, Reporting
	S3.0	Administrative and Financial
Information Infrastructure	I 1.0	EHR Security
	I 2.0	EHR Information and Records Management
	I 3.0	Unique identity, registry, and directory services
	I 4.0	Support for Health Informatics & Terminology Standards
	I 5.0	Interoperability
	I 6.0	Manage business rules
	I 7.0	Workflow

Figura 7: “Modelo Funcional”.
Fuente: (Health Level Seven, 2004)

6. Diseño de la propuesta.

El análisis de la problemática y el estudio del estado del arte dan cuenta de las necesidades actuales para desarrollar mayores conocimientos y métodos de verificación de interoperabilidad en los sistemas informáticos de salud del país. Si bien, ha sido amplio el desarrollo en el diseño e implementación de plataformas desde la estrategia SIDRA, aún hay carencias a la hora de verificar si la propuesta de diferentes empresas está basada en un modelo de interoperabilidad por medio de los estándares establecidos por el Ministerio de Salud. Además, el desarrollo de guías de implementación de mensajería HL7 Chile permite estructurar y definir el contenido semántico de cada mensaje propio del país.

Ya se ha mencionado en capítulos anteriores que a la fecha se ha desarrollado solo una guía de implementación para la mensajería de Admisión de Urgencias. Es decir, se cuenta con la definición del tipo de eventos y mensajes a utilizar solo en dicho proceso. Es por ello que la propuesta del

presente proyecto consiste en aplicar la guía de implementación de mensajería HL7 2.5.1 de Admisión de Urgencias Chile, con el fin de permitir la verificación de interoperabilidad mediante el envío de mensajes HL7 codificados y decodificados bajo el estricto rigor de lo documentado en dicha guía. Por lo anterior, el prototipo del sistema generará una base de desarrollos futuros como modelo de aplicación y verificación de mensajería estructurada según lo normado por una guía de implementación HL7 nacional para la validación de interoperabilidad entre sistemas de información en salud.

El alcance del desarrollo de este proyecto es el diseño e implementación de un prototipo de sistema informático de Admisión de Urgencias junto con la elaboración de algunos mensajes definidos en la guía, que serán enviados y recibidos entre módulos de dicho sistema. En adhesión a lo anterior, se implementará una herramienta que permita la verificación de la estructura de cada mensaje para aceptar o no su contenido; también será aplicada a cada módulo del prototipo del sistema informático. El diseño del sistema será abordado por medio de la especificación de un “Modelo general” basado en el modelo de conectividad definido en la guía de implementación, donde se establecen los mensajes HL7 que se emiten y reciben por los diferentes actores que componen “Admisión de Urgencias”, mientras que las funciones de cada módulo del sistema serán definidas por medio del “Modelo Funcional”. La implementación se llevará a cabo a través del uso de programación web, con el fin de que el prototipo de sistema informático sea compatible con múltiples plataformas. Además, la emisión y recepción de mensajes se abordará por medio del uso de servidor web socket.

7. Metodología.

La metodología presentada en esta sección es elaboración propia. El diseño está orientado para dar cumplimiento de los objetivos inicialmente declarados. El desarrollo de cada uno de ellos será llevado a cabo por actividades respectivamente descritas, y de las cuales se espera obtener un resultado.

En la Tabla 2, se presenta la tabla con la descripción de la metodología empleada para el desarrollo del proyecto. En cada columna se encuentra el objetivo, sus actividades, la descripción y el resultado esperado.

Tabla 2: “Metodología empleada en el desarrollo del proyecto” Fuente: Elaboración propia.

Objetivo	Actividad	Como llevar a cabo	Entregable
1. Establecer el diseño de los módulos de prueba del sistema informático de admisión de Urgencias.	A Definición de módulos de prueba y mensajes de admisión de urgencias.	En base diagrama de flujo de datos definido en la guía de implementación de admisión de urgencias, se seleccionaran los actores y los mensajes HL7 a implementar. Se estructurará un pequeño diseño inicial, junto a la descripción del propósito de cada módulo. Además de describir el modelo de conectividad de la mensajería HL7 de admisión de Urgencias.	Diseño general del sistema informático hospitalario

	B	Determinar el Modelo Funcional de los sistemas de prueba.	En base al Modelo Funcional "HL7 EHR System Functional Model" del documento DSTU del 2004. Se definirán las funciones básicas de cada módulo de prueba a implementar del sistema informático de admisión de Urgencias.	Modelos funcionales de módulos de prueba a implementar del sistema informático de admisión de Urgencias.
	C	Diseño de modelo de codificación y decodificación de mensajes HL7 de admisión de Urgencias Chile	En base al formato estrictamente establecido en la guía de implementación de mensajería HL7 de Admisión de Urgencias Chile.	Modelo de codificación y decodificación de mensajes HL7 de admisión de Urgencias.
	D	Diseñar un modelo para la verificación de errores en mensajería HL7 de admisión de Urgencias.	En base al modelo de decodificación de mensajería y a la guía de implementación, se deben establecer los errores de estructura del mensaje a identificar. Definir técnicamente las capacidades o funciones del verificador.	Especificación técnica del verificador de errores de mensajería.
2. Implementar el sistema y su comunicación interoperable.	A	Implementar diseño general de los módulos de prueba.	Generar divisiones a la interfaz. Diseñar menús. Aplicar estilos a la interfaz.	Interfaz general de módulos de prueba.
	B	Implementar servidores de recepción y envío de datos para módulos de prueba de admisión de urgencia.	Aplicar programación cliente-servidor a cada módulo de prueba. Definir puertos de comunicación exclusivos para cada módulo. Verificar funcionamiento de las funciones cliente-servidor	Servidores de recepción y envío de datos para cada módulo de prueba del sistema informático de admisión de Urgencia Chile.
	C	Implementar función codificadora y decodificadora de mensajería HL7 de admisión de Urgencias Chile.	Elección del mismo lenguaje de programación del diseño general. Aplicar modelo de decodificación de mensajería definido en el punto 1.C	Código de implementación de codificador y decodificador para mensajería HL7 de admisión de Urgencias Chile.
	D	Implementar herramienta verificadora de errores de mensajería HL7.	Programación divisiones de la interfaz. Aplicar estilos a la interfaz. Implementar decodificador de mensajes.	Herramienta funcional verificadora de mensajería HL7

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

			Aplicación de funciones del verificador en base a lo estipulado en el punto 1.D. y a la incorporación del decodificador de mensajería declarado en el punto 1.C.	de admisión de Urgencias Chile.
	E	Implementación de modelos de bases de datos para cada módulo de prueba.	Selección de plataforma de aplicación de base de datos. Programar bases de datos para cada módulo.	Bases de datos implementadas para cada módulo de mensajería HL7 de admisión de Urgencias Chile.
	F	Implementación de módulos de prueba del sistema de admisión de Urgencias.	Programación de funciones básicas de los módulos de acuerdo al modelo definido en 1.B. Programación interfaces de monitor de mensajes. Aplicación de mensajes HL7 de admisión de Urgencias. Aplicación de funciones HL7 a cada servidor de prueba.	Módulos implementados
3. Probar funcionamiento y aplicar correcciones al prototipo del sistema informático de admisión de Urgencias.	A	Aplicación de marcha blanca	Detección de errores de funcionamiento. Corrección de errores.	Sistema corregido
	B	Incorporación de elementos de ayuda	Incorporar elementos visuales y de texto para facilitar el uso del sistema.	Implementación final de herramienta verificadora de mensajería HL7 y módulos de prueba del sistema de admisión de Urgencias Chile.
	C	Incorporación de documento de ayuda.	Redacción de documento de instalación y uso del prototipo.	Documento soporte del prototipo.

8. Resultados.

La presentación de los resultados y de la ejecución del proyecto estará estructurado de acuerdo al ítem entregable de cada etapa del proyecto definido en la metodología del capítulo anterior.

8.1 Diseño general del prototipo del sistema informático de admisión de Urgencias.

La primera etapa para el desarrollo del sistema fue seleccionar algunos actores del diagrama de flujo de datos definido en la guía de implementación de Admisión de urgencias, tal como se aprecia en la Figura 8 donde se visualiza el diagrama de flujo de datos acotados de parte del proceso clínico que se desea informatizar.



Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

31

Figura 8: “Diagrama de flujo de datos acotado de Admisión de Urgencias”, Fuente: (Urzúa, 2014)

Los módulos a desarrollar del sistema están sujetos a los siguientes actores seleccionados: “Pre-Atención de Urgencias”, “Registro Datos de Urgencias” y “Registro Datos Paciente”. El alcance de las funciones de cada módulo, que serán definidas más adelante, están sujetas a los tipos de mensajes a enviar según lo indicado por el modelo de conectividad (ver “Figura 9”) definido en la Guía de implementación de Admisión de Urgencias.

La Figura 9 detalla las conexiones entre actores del flujo de datos del sistema de admisión de urgencias, es decir, especifica el tipo de mensaje y el origen. El detalle de la estructura de cada mensaje se encuentra en la guía de implementación de Admisión de Urgencia. De la totalidad de los mensajes descritos se implementaran o codificaran los siguientes mensajes:

- ADT 01:** Notificación de la Admisión o visita.
- ADT 02:** Transferencia de un Paciente.
- ADT 04:** Registro de un Paciente.
- ADT 06:** Cambiar paciente externo a uno interno.

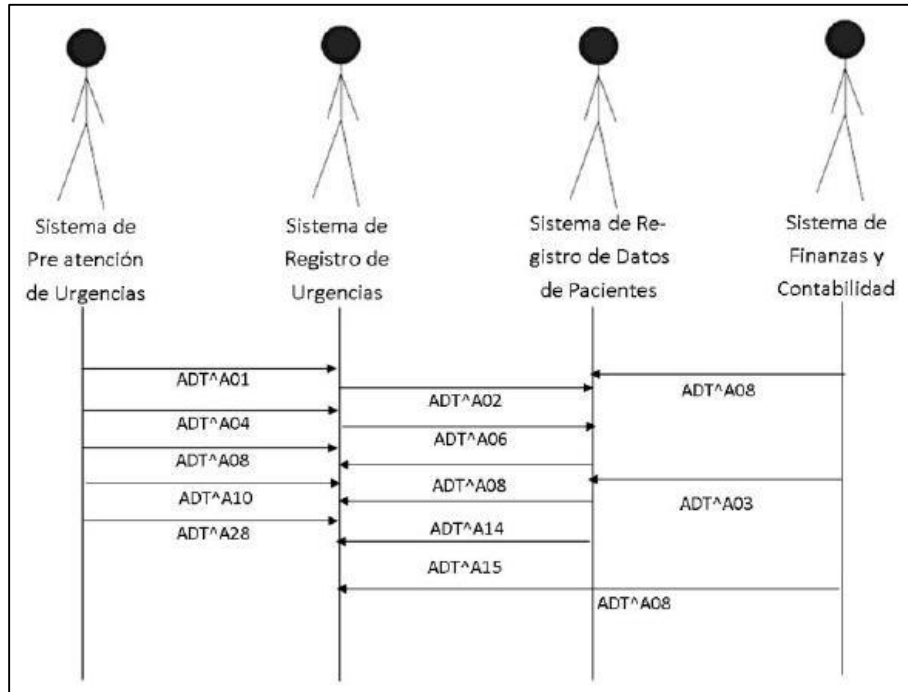


Figura 9: Modelo de conectividad de actores de Admisión de Urgencias. Fuente: (Urzúa, 2014).

La lectura o decodificación de los mensajes será para la totalidad de los mensajes de Admisión de Urgencias. Los módulos de prueba estarán diseñados por una interfaz y un servidor para mensajería HL7 de Admisión de Urgencias.

En base a lo anterior, el esquema o diseño general del sistema se visualiza en la “Figura 10”, donde se aprecian los tres módulos del sistema, la forma de conectividad, el tipo de mensaje y el verificador de errores de mensajería incluido dentro de los servidores y también como herramienta externa. Este modelo servirá como pauta para el diseño e implementación del prototipo. Además, por el tipo de aplicación de este prototipo, se requiere de una plataforma de implementación compatible con múltiples sistemas; conforme a esto, se escogió un diseño web por sus características de funcionamiento en los diversos sistemas operativos y dispositivos electrónicos presentes en el mercado.

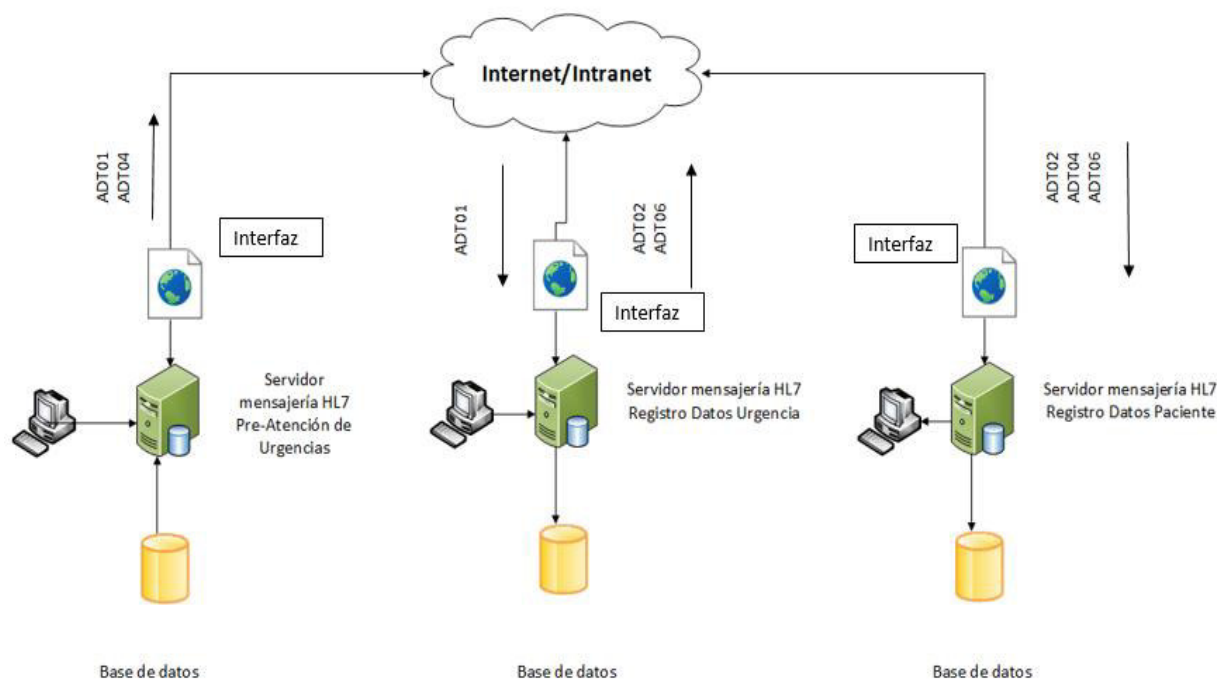


Figura 10: “Modelo General de los sistemas de prueba de Admisión de Urgencia”.
Fuente: Elaboración propia.

8.2 Modelo funcional de módulos de prueba.

El desarrollo del modelo funcional para cada módulo se llevó a cabo en base al documento “HL7 EHR System Functional Model DSTU” (Health Level Seven, 2004). El contenido de las funciones de este modelo se realizó en gran parte por lo descrito en la “Guía de implementación de Urgencias” (Urzúa, 2014) y el “Estándar de atención de urgencias” (Ministerio de Salud, 2013). El resultado del modelo con la descripción de las funciones que debe tener cada módulo se encuentra en el ANEXO 1 en mayor detalle. A continuación se presenta un pequeño resumen del contenido de dicho modelo en donde se describen las funciones y su respectiva sigla:

- a) Pre-Atención:
 - Ingreso datos pacientes (S3.1).
 - Ingreso datos acompañante (S3.2).
 - Visualización datos paciente (S3.3).
 - Pre-Admitir paciente (S3.4).
- b) Registro Datos de Urgencia.
 - Visualización datos de pacientes de urgencia (S3.1).
 - Traslado de pacientes (S3.2).
 - Admisión de un Paciente (S3.3).
- c) Registro Datos Pacientes.
 - Almacenamiento datos demográficos del paciente (S3.1).
 - Almacenamiento datos clínicos del paciente (S3.1).

a) Codificación de mensajería HL7 de Admisión de Urgencias Chile.

El procedimiento o modelo de codificación en la “Figura 11” describe el orden en que se deben construir los mensajes HL7 de Admisión de Urgencias. La diferencia en la elaboración de estos mensajes aplicados en Admisión de Urgencias Chile, respecto a lo referido al estándar HL7 internacional, radica en la selección de segmentos y del contenido por medio de una definición precisa de sus componentes en relación a la semántica nacional. Este modelo permitirá estructurar la implementación al momento de programar y armar cada mensaje definido en la Guía de Implementación de Urgencias Chile.

Figura 11: “Modelo de codificación de mensajería HL7” Fuente: Elaboración propia, basado en (Villagrasa), (Urzúa, 2014).

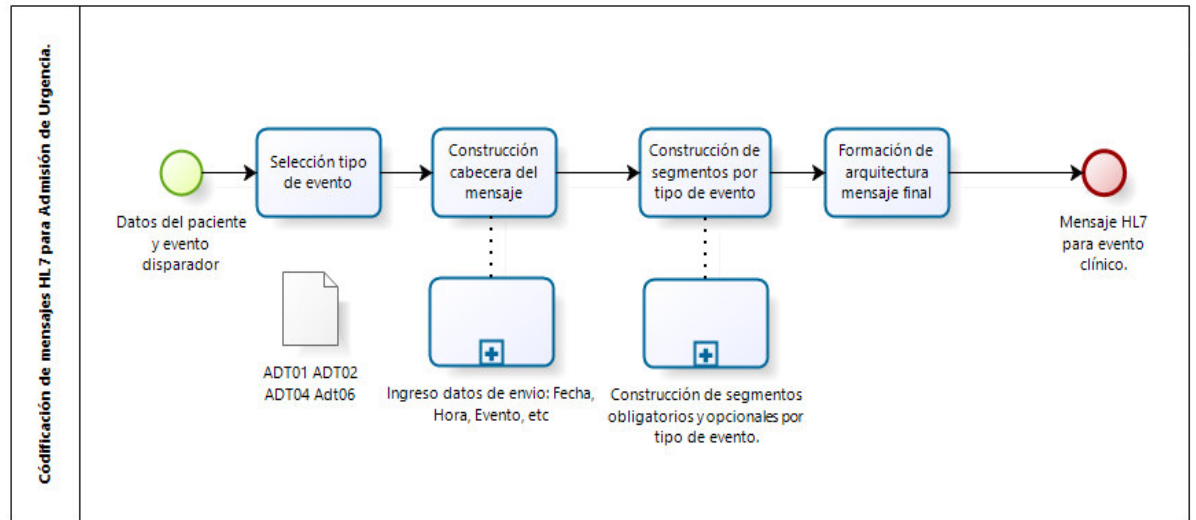
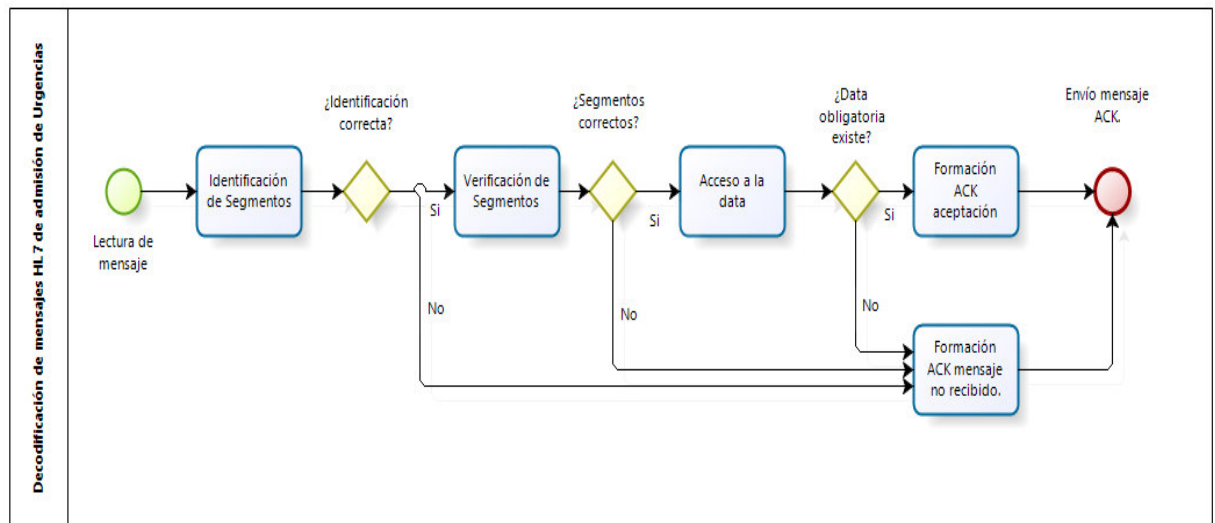


Figura 12: “Modelo de decodificación de mensajería HL7” Fuente: Elaboración propia, basado en (Villagrasa), (Urzúa, 2014).



b) Decodificación de mensajería HL7 de Admisión de Urgencias Chile.

El Modelo de la “Figura 12” describe el procedimiento para decodificar el mensaje HL7 y acceder a la información contenida en este. El mensaje HL7 a decodificar debe estar estructurado bajo lo establecido en la guía de implementación tanto para el tipo de mensaje como para los segmentos y campos contenidos. En caso de que no se cumpla con la semántica ni la estructura del mensaje, se debe generar un mensaje de respuesta informando aquello. La guía de implementación dispone de la codificación de errores para ser implementada, por lo que se debe detectar los errores y seleccionar el código de este al momento de construir el ACK.

8.4 Diseño del verificador de errores de mensajería HL7 de Admisión de Urgencia.

Tabla 3: “Especificación del verificador de errores de mensajería”. Fuente: Elaboración propia.

Zonificación	Nº	Especificación	Cumplimiento
Características generales	1	Compatible con mensajería HL7 2.5.1	Debe cumplir
	2	Diseñado para mensajería HL7 de Admisión de Urgencias	Debe cumplir
	3	Ingreso de mensajes HL7 en formato texto.	Debe cumplir
	4	ACK de respuesta.	Debe cumplir
Detección de errores	5	Errores de Formato de arquitectura de mensaje	Debe cumplir
	6	Segmentos obligatorio faltantes	Debe cumplir
	7	Evento ingresado, solo se aceptan los descritos en la guía de implementación	Debe cumplir
	8	Errores por no contener componentes obligatorios	Debe cumplir
	9	Generar código de error	Debe cumplir
Interfaz	10	Interfaz sencilla	Deseable
	11	Ingreso de mensaje HL7 por pantalla	Deseable
	12	Monitor con resultados de la verificación	Deseable
	13	Botones de operación	Deseable
	14	Textos en español.	Deseable

La “Tabla 3” detalla las características de la herramienta verificadora de la mensajería HL7 de Admisión de Urgencias. Si bien esta herramienta se implementará con una interfaz propia, el contenido funcional será implementado en los servidores de los módulos de pruebas del sistema de Admisión de Urgencias.

8.5 Implementación del diseño general del sistema.

El diseño general del sistema requería una plataforma de programación compatible con múltiples sistemas, es por ello que se escogió el diseño web como medio de implementación.

La programación se llevó a cabo en los lenguajes HTML, PHP, CSS6 y JAVASCRIPT. Las herramientas usadas en la implementación son las que se muestran en la “Tabla 4”.

Tabla 4: Herramientas Utilizadas. Fuente: Elaboración propia.

	Nombre Herramienta	Versión	WEB	Comentario
1	Notepad ++	6.6.9	http://notepad-plus-plus.org	Software de edición de diferentes lenguajes de programación
2	XAMPP	1.8.3	https://www.apachefriends.org	Plataforma de libre distribución que contiene servidor Apache (PHP 5.5.15), MYSQL, PERL, etc.

Con las herramientas descritas en la “Tabla 4”, se diseñó e implementó una interfaz general del sistema, para luego aplicarla en cada módulo de prueba como se muestra en las imágenes de la “Figura 13”.

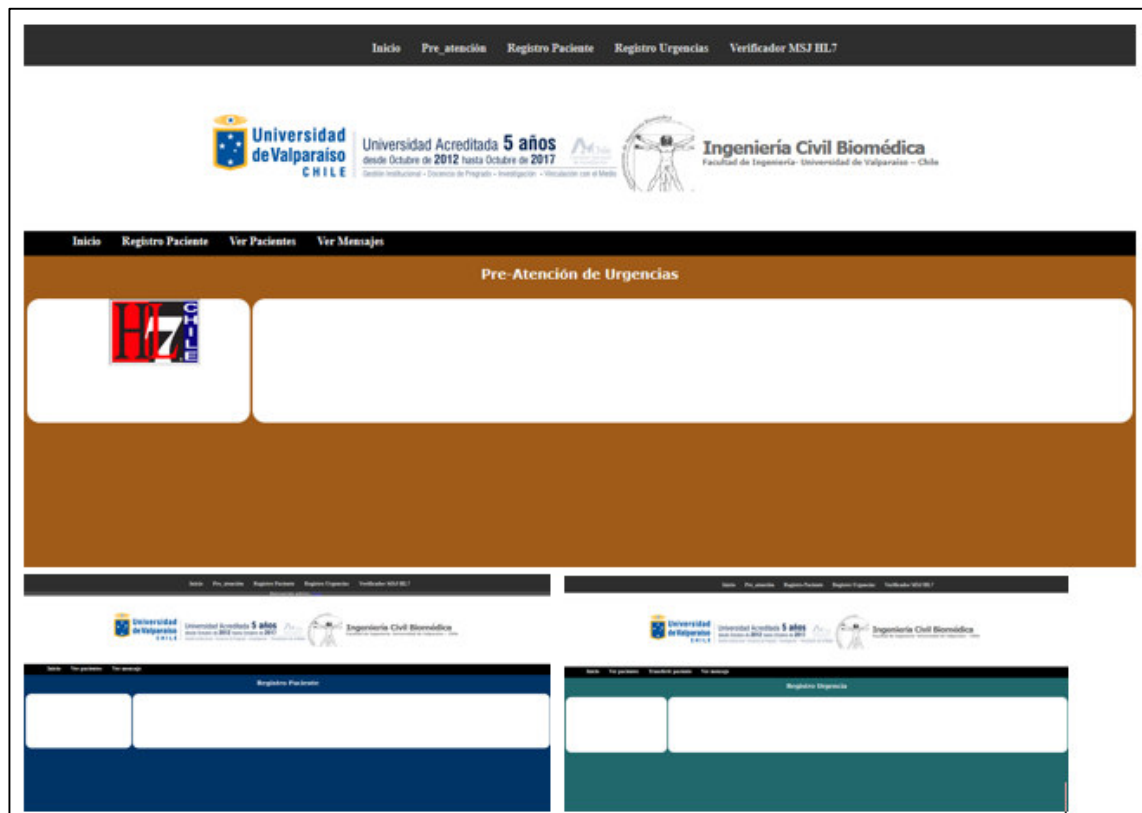


Figura 13: “Interfases generales de los módulos de prueba” Fuente: Elaboración propia.

Además, se diseñó las interfaces generales para los sistemas de acceso a los módulos de prueba como se aprecia en la “Figura 14”.

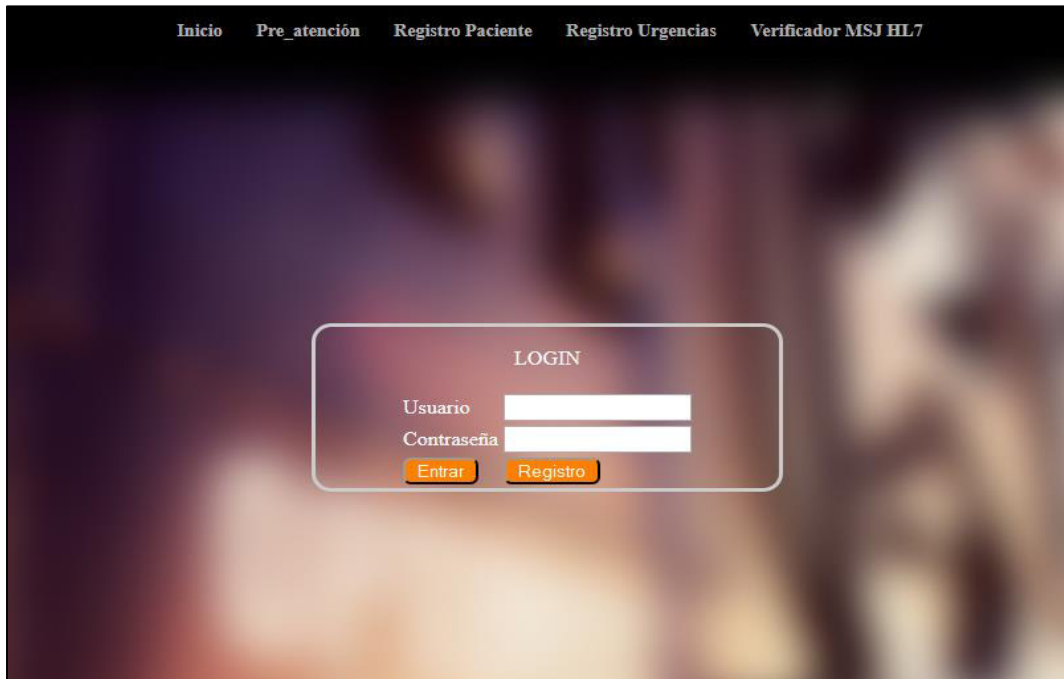


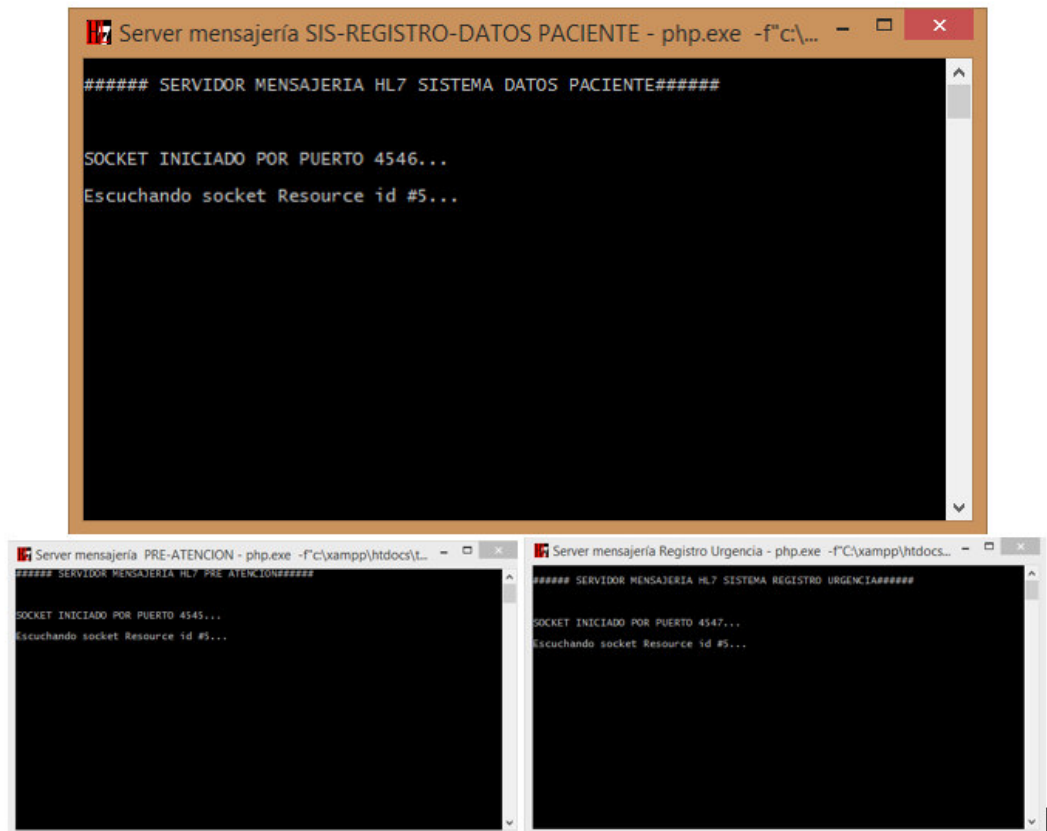
Figura 14: “Interfaz general de acceso usuario”. Fuente: Elaboración propia.

8.6 Implementación de servidores de recepción y envío de mensajes.

La implementación del cliente-servidor general se realizó por medio de la programación en PHP de un webservice o servidor web (ver detalle del código en el ANEXO 3). Por otro lado, a partir de la implementación general del servidor, se aplicó este modelo para cada módulo de prueba, en donde cada servidor fue capaz de correr bajo la consola de Windows como se muestra en la “Figura 15”.

A los servidores de mensajería HL7 para los diferentes módulos de prueba se les asignó un puerto exclusivo para cada uno; Puerto 4545, para el “Módulo de Pre-Atención”; Puerto 4546, para el “Módulo de Registro de datos paciente”; y el Puerto 4547, para el “Módulo de Registro de datos de Urgencia”. La IP de cada servidor puede ser modificada dentro del código de programación.

Figura 15: “Servidores de mensajería HL7 de módulos de prueba del Sistema de Admisión de Urgencias”.
Fuente: “Diseño Propio”.



8.7 Implementación de codificador y decodificador de mensajería HL7.

El codificador y decodificador de mensajes fueron implementados con el mismo lenguaje de programación de las interfaces (PHP 5.5.15). En la implementación se siguió el mismo modelo codificación y decodificación de mensajes HL7 definido anteriormente. El código de la implementación puede ser visto en mayor detalle en el ANEXO 4.

8.8 Implementación de la herramienta funcional verificadora de mensajería HL7.

Se implementó la herramienta verificadora de mensajería HL7 de admisión de Urgencias. Su programación se hizo en base al código del decodificador del ANEXO 4 y además se le agregó una interfaz para su fácil operación (ver Figura 16). La Interfaz está compuesta por dos ventanas: una para el ingreso del mensaje y otra donde se muestra el análisis de este. Además, hay dos botones para su operación; el primero para dar inicio a la verificación y el segundo para borrar el mensaje.

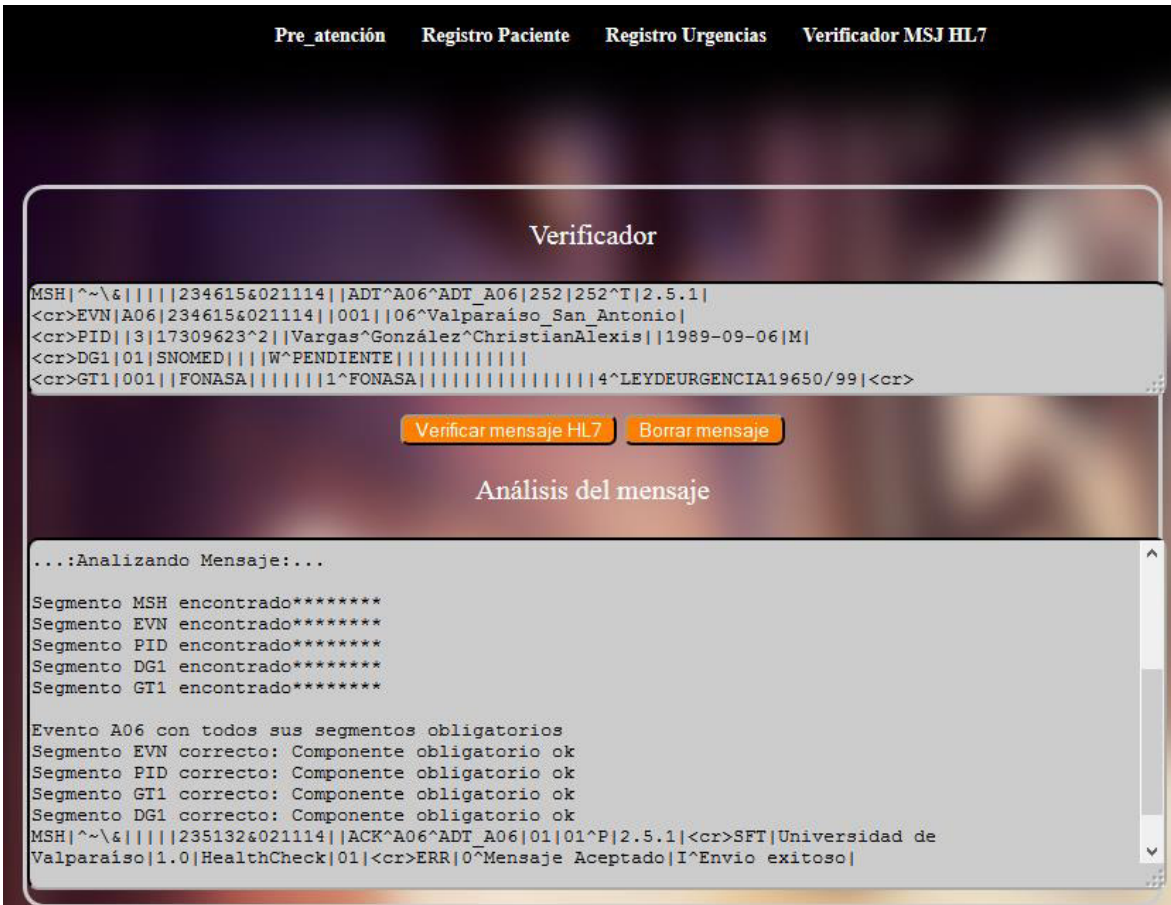


Figura 16: “Herramienta de verificación de mensajería HL7 Admisión Urgencias Chile”, Fuente: Elaboración Propia.

Por otro lado, se incorporó el reconocimiento de errores en la estructura y contenido del mensaje a analizar. Se verifica la existencia de los segmentos y componentes obligatorios según sea el tipo de mensaje analizado. En última instancia, se genera el mensaje ACK donde se muestra el código del error o aceptación según sea el caso.

8.9 Implementación de bases de datos de módulos de prueba.

Se implementaron las bases de datos de forma independiente para cada módulo de prueba. Los datos almacenados son compartidos solamente por el envío del mensaje HL7 codificado y posteriormente decodificado. El contenido se realizó en base a lo descrito en cada Modelo Funcional de los módulos de pruebas. El diseño de las bases de datos así como su implementación se realizó con el módulo del software XAMPP llamado “PHP ADMIN”. El detalle del diseño puede ser visto en el ANEXO 4.

8.10 Implementación de funciones a los módulos del prototipo del Sistema de Admisión de Urgencias.

Se implementaron las funciones a cada módulo de prueba, por medio del formato web de las interfaces resultantes. Los formularios de captación de información del paciente, así como el almacenamiento en cada base de datos propia de cada módulo, funcionan de forma independiente. Las funciones implementadas en cada módulo permiten como resultante generar un “Evento de Admisión de Urgencias” que desencadena en la determinación y formación de su respectivo “Mensaje HL7 de Admisión de Urgencias” a través de la incorporación de la función codificadora de mensajes implementada en un punto anterior.

La presentación, así como el cliente de envío del mensaje HL7 conformado, se desarrolló por medio de la implementación de una interfaz constituida por dos secciones; una, donde hay un monitor de visualización del envío y recepción del mensaje HL7 hacia el servidor correspondiente al evento generado; y otra, en donde se aprecia el mensaje con la capacidad de seleccionarlo y copiarlo. Además, se implementaron botones de operación para el reenvío del mensaje y finalización de la operación, ver “Figura 17”.

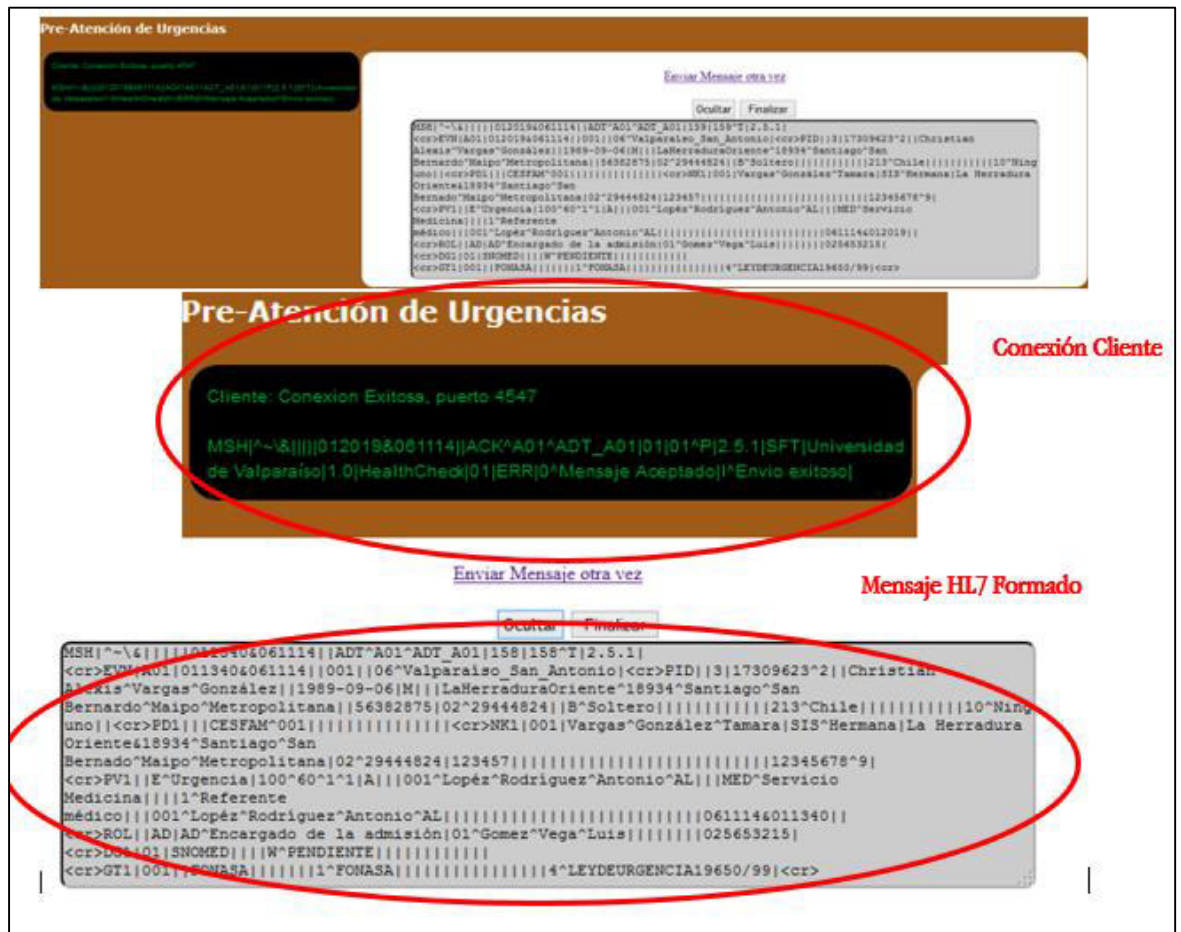


Figura 17: “Formación y envío de Mensaje HL7 a Servidor de prueba” Fuente: Elaboración propia.

Por otro lado, para la correcta recepción de los mensajes por los servidores de prueba se incorporó la función decodificadora de los mensajes junto con la aplicación de la incorporación y actualización de datos en las respectivas bases de datos asociadas a cada módulo. En la “Figura 18” se aprecia la recepción del mensaje descrito en la “Figura 15” por uno de los servidores de prueba.

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

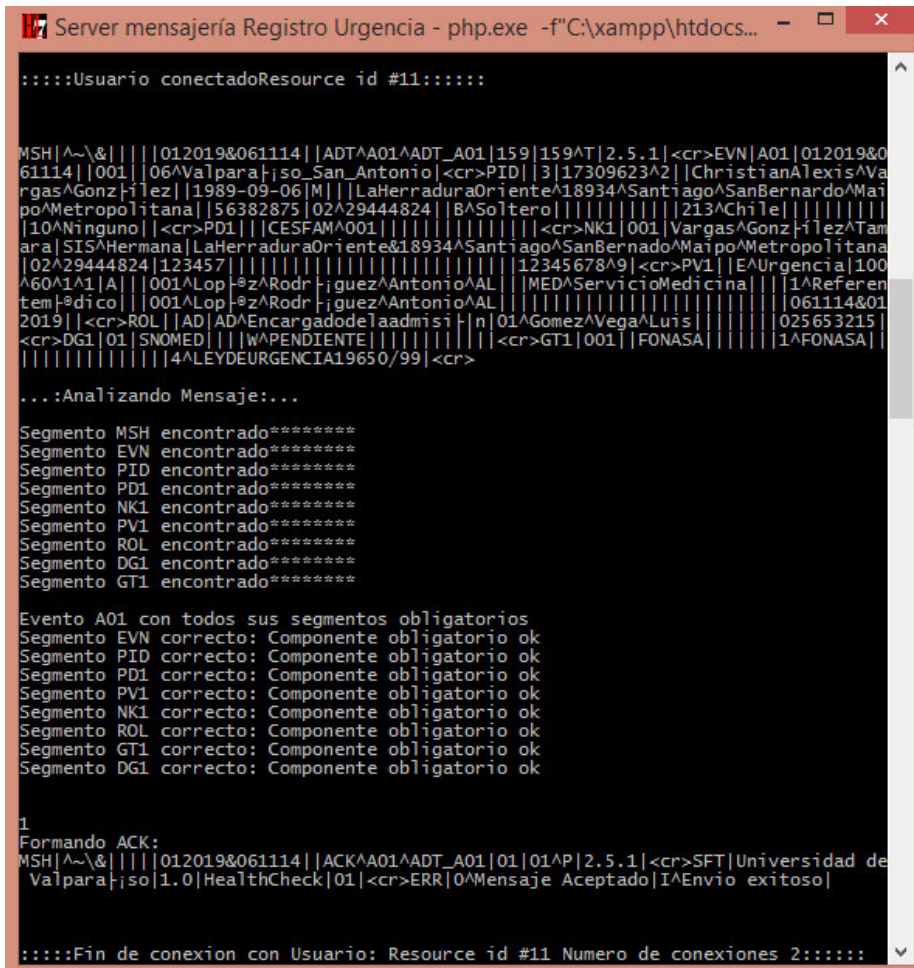


Figura 18: “Recepción de mensaje HL7 por Servidor de módulo de Pre-Atención de pacientes” Fuente: Elaboración Propia.

Por último, para registrar tanto la recepción como el envío de un mensaje desde un módulo, se implementaron las interfaces correspondientes para la visualización del almacenamiento de los respectivos mensajes, ver “Figura 19”.

ID Mensaje;	Mensaje HL7	Tipo Mensaje	Hora Envío	Fecha Envío
221	MSH ^~\& 105121&161014 ADT^A02^ADT_A02 221 221^T 2.5.1 <cr>EVN A02 105121&161014 001 06^Valparaíso_San_Antonio <cr>PID 3 17309623^2 Vargas^González^Christia	A02	10:51:21	2014-10-16

Figura 19: Historial de mensajes enviados y recibidos. Fuente: Elaboración propia.

8.11 Funcionamiento general del prototipo y correcciones.

Se probó el funcionamiento general del prototipo dando como resultado la compilación o codificación correcta de cada mensaje HL7 con el formato establecido en la Guía de Implementación. Además, el mensaje -al momento de crearse- es enviado al servidor de pruebas de destino, en donde se verifica la estructura semántica del mensaje, para luego generar un mensaje de respuesta ACK especificando el cumplimiento o no de la recepción de este. El mensaje de aceptación y de no aceptación es devuelto al módulo de origen del mensaje.

Los errores encontrados durante la verificación del funcionamiento tienen mayor frecuencia en relación a la presentación de los datos. En menor relevancia existían errores en las funciones de verificación y codificación de mensajes en base a la guía de implementación.

Por último, a cada módulo de prueba se le agregó información relacionada a las funciones de cada uno y menús de ayuda para la operación. También se conformó un documento de soporte a la instalación y uso del prototipo, en donde además se especifican algunos de talles de como agregar más mensajes o modificar elementos según corresponda para futuras incorporaciones y actualizaciones de la herramienta, ver ANEXO 5.

9. Discusiones.

El prototipo del sistema se estructura y define exclusivamente en lo especificado por la “Guía de implementación de Admisión de Urgencias Chile”, por lo tanto, será una herramienta válida para ser aplicada en la verificación de mensajería en diversos proveedores de software en salud, cuando la Guía esté normada y aprobada tanto por el Ministerio de Salud y el capítulo Chileno de HL7. De esta manera, este prototipo debe ir actualizando sus funciones de acuerdo a las correcciones o versiones de la Guía de implementación. Además, se debe incorporar la totalidad de mensajes actores de Admisión de Urgencias.

Por otro lado, Chile debe disponer de políticas que faciliten la normalización del uso de estándares de información por medio de la creación de Guías de Implementación. Además, se debe exigir el cumplimiento de estas guías por medio de la creación de entidades propias del estado, que dispongan de herramientas que validen la interoperabilidad sintáctica y semántica entre sistemas de información en salud. Sería deseable que el estado especificara en el convenio marco que los sistemas deben ser interoperables y deben adecuarse a las normativas de uso de guías de implementación diseñadas y especificadas para Chile.

10. Conclusiones.

La correcta adopción de estándares mediante el uso de Guías de Implementación permite en cierta medida lograr la Interoperabilidad ente dos o más sistemas de información en Salud. Los avances en Chile respecto a la implementación de Sistemas informáticos en Salud se deben fundamentalmente a la iniciativa del Proyecto SIDRA. Las políticas del proyecto sobre Interoperabilidad no hacen referencia al desarrollo de documentos de referencia técnica como las “Guías de Implementación” para abordar el correcto uso de los Estándares de información en Salud usados en Chile, por lo que existe una dificultad para demostrar Interoperabilidad entre los actuales sistemas implementados en Chile por medio de pruebas o validaciones que corroboren la adecuada utilización de los estándares.

Los resultados obtenidos en el capítulo anterior reflejan la aplicación de la metodología de desarrollo establecida para diseñar e implementar un prototipo de sistema informático con el propósito de aplicar la “Guía de implementación de Urgencias”. El prototipo permite generar cuatro mensajes HL7 que se transmiten entre los módulos de Pre-Atención- Registro Datos Urgencias y Registro de Datos Pacientes. Además, los servidores de cada módulo otorgan la posibilidad de validar la correcta estructura de los mensajes.

El prototipo informático diseñado e implementado permite verificar la estructuración de los mensajes HL7 de Admisión de Urgencias mediante la incorporación de servidores de prueba que permiten recibir mensajes, detectar errores, y generar mensajes de aceptación o rechazo. Es importante destacar que en el desarrollo de esta herramienta al ser un prototipo, solo se implementaron algunos actores de admisión de urgencia junto con las funcionalidades necesarias para dar solvencia a la generación de los cuatro (4) mensajes de Admisión de Urgencias y demostrar el uso de la herramienta. Es por ello, que este prototipo permite sustentar una base de desarrollos futuros tanto en la incorporación de nuevos mensajes y actores de Admisión de Urgencias, como la utilización de herramientas que permitan genera documentos clínicos o la utilización de servidores de terminologías de vocabulario.

En relación a la compatibilidad técnica en la instalación del prototipo, es importante mencionar que su diseño es en base a un lenguaje web, lo que permite su uso en un nivel multiplataforma. Solo se requiere fijar los puertos de comunicación y la dirección IP donde esté alojado. Por otro lado, este prototipo fue diseñado para su uso a nivel académico, con el fin de desarrollar Guías de Implementación.

Resumen de las contribuciones.

Las contribuciones en el desarrollo del presente proyecto son las siguientes:

- Se establece un prototipo de sistema informático de Admisión de Urgencias donde es posible enviar y recibir mensajes HL7 entre los módulos que lo conforman. Además, se verifica el contenido estructural de cada mensaje y se valida su recepción.
- Se implementó una herramienta de verificación de mensajería exclusiva para los mensajes HL7 de admisión de Urgencias definidos en la “Guía de implementación de Urgencias”.

- El prototipo de sistema informático de Admisión de Urgencias, permite establecer una base para incorporar elementos de otros estándares de información en salud, como es el caso de OpenEHR, SNOMED CT, etc.
- Se establece un prototipo inicial para la implementación de futuros desarrollos de Guías de implementación en Chile, junto con una herramienta inicial para la validación de estándares aplicados en Chile.

Alcance de las contribuciones.

El alcance de este proyecto se encuentra a nivel académico. Se deben establecer mejoras al prototipo basado en lo dictado en el proceso de validación final de la Guía de implementación de Admisión de Urgencias por parte del Capítulo Chileno de HL7. Posteriormente, será posible validar este prototipo o herramienta, para ser utilizada en la verificación de mensajería entre las plataformas de salud implementadas en Chile.

Por otro lado, el prototipo puede ser instalado en servidores o en computadores locales para ser usado en la formación de estudiantes por medio de experiencias en el uso de mensajería HL7.

Investigaciones futuras.

Las principales investigaciones futuras para este proyecto es la de consolidar el prototipo, incorporando la totalidad de actores y mensajes HL7 del proceso de Admisión de Urgencias. Además, se deben perfeccionar tanto los métodos de transmisión de datos como la incorporación de nuevas formas de codificación y decodificación de mensajes HL7 por medio del uso de XML. Otras de las áreas posibles a desarrollar es la de incorporar documentos clínicos (“Epicrisis de Urgencias”), y de un registro clínico electrónico (FHR de HL7 u OpenEHR). En esta perspectiva, también se encuentra la posibilidad de insertar terminologías y vocabularios médicos.

Conforme avance los desarrollos de normativas que describan técnicamente el uso adecuado de estándares en salud se podrá hacer uso de la base de diseño e implementación de este prototipo para replicar la validación de mensajería HL7 en otros procesos clínicos dentro de las instituciones sanitarias.

Referencias Bibliográficas

- Arevalo, O. L. (s.f.). *Estándares para la interoperabilidad en salud*. Recuperado el 4 de Abril de 2014, de Ministerio de Salud Colombia:
http://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/OT/4%20Interoperabilidad%20SEMANTICA_MSPS.pdf
- British Medical Informatics Society. (s.f.). *British Medical Informatics Society*. Recuperado el 2015 de Mayo de 2014, de <http://www.bmis.org>
- CENETEC. (Junio de 2007). *Modelo Funcional Expediente Electrónico Nacional*. Recuperado el 26 de Abril de 2014, de http://www.cenetec.salud.gob.mx/descargas/comite_e-Salud/ModeloFuncional_esalud.pdf
- Ecri. (Noviembre de 2013). *2014 Top 10 Hazards Executive Brief*. Recuperado el 2 de Noviembre de 2014, de Ecri:
https://www.ecri.org/Forms/Documents/2014_Top_10_Hazards_Executive_Brief.pdf
- Estándares Minsal. (13 de Mayo de 2014). *Salud-e*. Obtenido de <http://www.salud-e.cl/estandares/>
- Health Level Seven. (Julio de 2004). *HL7 EHR System Functional Model Draft Standard for Trial Use*. Recuperado el 26 de Abril de 2014, de http://www.hl7.org/documentcenter/public_temp_62989EB5-1C23-BA17-0CB29141FA5D02A6/wg/ehr/hl7_ehr-s_dstu.pdf
- Health Level Seven International. (2012). *GLOSSARY*. Recuperado el 10 de Octubre de 2014, de https://www.hl7.org/documentcenter/public_temp_E6839396-1C23-BA17-0C3A48D94A2436D8/calendarofevents/FirstTime/Glossary%20of%20terms.pdf
- Health Level Seven. (10 de Abril de 2014). *Health Level Seven*. Recuperado el 19 de Mayo de 2014, de <http://www.hl7.org/>
- Hoffmann, C. (2011). *ACTI Estándares Salud v1.0.6*.
- IHE Europe. (7 de Enero de 2014). *Connectathon IHE*. Recuperado el 2 de Noviembre de 2014, de IHE:
http://www.ihe-e.org/docweb/2014/WP_Connectathon_final.pdf
- IHE. (s.f.). *Integrating Healthcare the Enterprise*. Recuperado el 10 de Abril de 2014, de <http://www.ihe.net>
- López, A. P. (Julio de 2008). *Estándares*. Obtenido de CENETEC:
http://www.cenetec.salud.gob.mx/descargas/telemedicina/taller_directrices/estandares.pdf
- Marzo, J. V. (Mayo de 2012). *Sociedad Española de Informática de la Salud*. Recuperado el 14 de Mayo de 2014, de Seis: http://www.seis.es/documentos/foros/SEIS_EstadoArteInteroperabilidad%20JOSEP%20VILALTA.pdf
- MercadoPúblico. (8 de Agosto de 2008). *CM 06-2008 Software de salud y servicios asociados*. Recuperado el 21 de Mayo de 2014, de MercadoPúblico:
<http://www.mercadopublico.cl/Procurement/Modules/RFB/DetailsAcquisition.aspx?qs=4plkT+hbJpOCN2C6JSkrpQtKq30ZemAl1M48IVoMaUvwd9CYjvdpQOhB2vSy9Bjf>
- Ministerio de Salud. (2006). *El Libro Azul, Agenda digital del Ministerio de Salud Gobierno de Chile*. LOM Ediciones Ltda.
- Ministerio de Salud. (8 de Septiembre de 2011). *Exento n° 820 Aprueba Norma técnica sobre estándares en información en salud*. Recuperado el 10 de Mayo de 2014, de Deis: http://www.deis.cl/wp-content/uploads/2011/09/Decreto_Norma_TecnicaEstandares_de_Informacion_DEIS.pdf
- Ministerio de Salud. (2011). *Solicitud de información (RFI) para el nuevo convenio marco software de salud y servicios asociados*. Santiago: Ministerio de Salud. Recuperado el 2 de Junio de 2014
- Ministerio de Salud. (27 de Mayo de 2013). *Exento n° 467 Aprueba Norma Técnica N°149 sobre Estándares de Información de Salud- Estándar de Atención de Urgencias de los niveles de Alta, Mediana y Baja complejidad y de la Atención Primaria*. Recuperado el 20 de Mayo de 2014, de Deis: <http://www.deis.cl/wp-content/uploads/2013/07/EST%C3%81NDAR-DE-ATENCION-DE-URGENCIA.pdf>
- Ministerio de Salud. (24 de Abril de 2013). *Informe Público de Avance SIDRA*. Recuperado el 15 de Octubre de 2014, de <http://www.salud-e.cl/wp-content/uploads/2013/11/Informe-Pu%C3%81blico-SIDRA-2013-03.pdf>

- Ministerio Salud. (2008). *Ficha técnica componente SIDRA "Registro de atenciones de urgencia"*. Ministerio Salud.
- Ministerio Salud. (15 de Septiembre de 2009). *Ficha técnica componentes SIDRA. Alcance Funcional*. Coquimbo, Coquimbo, Chile: Ministerio Salud.
- OMS. (4 de 2012). *Foro de la OMS sobre la estandarización y la interoperabilidad de los datos sanitarios*. Recuperado el 16 de Mayo de 2014, de World Health Organization:
<http://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/INEC/OPS/OMS-estandareseinteroperabilidad%20%282%29.pdf>
- Pablo Pazos, S. I. (2011). *Estándares e interoperabilidad en salud electrónica: Requisitos para una gestión sanitaria efectiva y eficiente*. Santiago: Naciones Unidas.
- Pazos, P. (5 de Noviembre de 2009). *Super estándares en informática médica*. Recuperado el 20 de Septiembre de 2014, de <http://informatica-medica.blogspot.com/2009/12/super-estandares-en-informatica-medica.html>
- Pazos, P. (s.f.). *Informática Médica*. Recuperado el 5 de Abril de 2014, de [Informatica-medica.blogspot.com](http://informatica-medica.blogspot.com)
- Salud-e. (s.f.). *Estrategía Digital de Salud*. Recuperado el 25 de Mayo de 2014, de <http://www.salud-e.cl/proyectos/sidra/>
- Selene Indarte. (s.f.). *Capítulo XV Interoperabilidad*. Recuperado el 20 de Mayo de 2014, de http://www.seis.es/documentos/informes/secciones/adjunto1/15_Interoperabilidad.pdf
- Silva, E. A. (s.f.). *Ingeniería Biomédica*. Recuperado el 4 de Abril de 2014, de Ingeniería Biomédica:
<http://www.slideshare.net/cheo.torres/4-informtica-en-salud-estndares-e-interoperabilidad>
- Urzúa, J. P. (2014). *Desarrollo de una Guía de Implementación Chilena para eventos de Admisión de Urgencia por medio de mensajería HL7 2.5.1 como medio para optar al título de Ingeniero Civil Biomédico*. Trabajo de Título, Universidad de Valparaíso, Valparaíso.
- Villagrasa, J. (s.f.). *HL7 Detalles Versión 2.X*. Recuperado el 20 de Septiembre de 2014, de HL7 Spain:
http://www.hl7spain.org/documents/tutoriales_HL7/SemHL7_Detalles_V2.pdf

Glosario.

SIS (Sistemas de información en Salud): Software de apoyo a la gestión de procesos clínicos, administrativos y en el mantenimiento de tecnologías al interior de una institución sanitaria.

Urgencias: (Unidad de Emergencias): Dentro de una institución sanitaria, es aquella unidad que atiende a pacientes que presentan un cuadro clínico grave y que requieren de atención médica-quirúrgica inmediata.

HL7 (Health Level Seven): Estándar de información en salud para facilitar la estructuración y comunicación de información entre dos o más sistemas de información en salud al interior de una institución sanitaria.

Servidor web: Tecnología que permite intercambiar información entre diferentes dispositivos por medio del uso de protocolos de comunicación.

PHP (Personal Home Page): Es un lenguaje de programación orientado al diseño de sitios web dinámicos. Su operación se encuentra desde el lado del servidor y no del cliente.

HTML (HyperText Markup Language): Es un lenguaje de programación que conforma sitios web por medio de la estructuración ordenada y simplificada de hipertexto.

CSS6 (Cascading Style Sheets): Es un lenguaje de programación que permite definir la presentación de un documento en HTML con estilos.

Base de Datos: Almacenaje de datos de un mismo contexto almacenados sistemáticamente para su posterior uso.

MYSQL: Sistema de gestión de base de datos. Permite crear, editar y ver base de datos de manera simple.

**Desarrollo de un
prototipo informático
para la verificación de
mensajes HL7 2.5.1
Admisión de Urgencias**

Anexos.

Anexo 1. "Aplicación modelo funcional para módulos de prueba del sistema de admisión de urgencias".

SISTEMA DE PRE-ATENCIÓN DE URGENCIAS.

		ID	NOMBRE	DECLARACIÓN	DESCRIPCIÓN	REFERENCIA	OTRO
DIRECT CARE	D 1		Soporte Clínico.				
	D 2		Medición, Análisis, Investigación y Reportes				
	D 3		Administrative and Financial				
SUPPORTIVE	S3		Administración de Operaciones y Comunicación	Administrar información del paciente por medio del registro y operación de datos	Aplicación que permita registrar los datos demográficos del paciente y acceder a ejecutar la pre-admisión.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
	S3.1		Datos del Paciente	Registro de datos demográficos iniciales del paciente.	Función que permite ingresos por medio de una interfaz, los datos demográficos del paciente para indicar el proceso de atención. Registro inicial. -Nombre. - Primer Apellido. -Segundo Apellido. -Edad. -RUN. -Fecha de nacimiento. -Previsión. -Teléfono. -Dirección. -Comuna. -Sexo.	Estándar de atención de urgencias de los niveles de alta, mediana y baja complejidad de la atención primaria. Decreto N° 467 23 Mayo 2013. Decreto Exento N° 820/11 sobre Estándares de Información en Salud. Guía de implementación de Mensajería 2.5.1 de	

Tabla 5: Sistema de Pre-Atención de Urgencias. (Fuente: Elaboración propia basado en Health Level Seven, 2004).

			-Estado conyugal. -Nacionalidad. -Pueblo originario.	Admisión de Urgencias Chile.	
S3.2	Datos del Acompañante	Registro de datos demográficos del acompañante	<p>Función que permite ingresar por medio de una interfaz, los datos demográficos del acompañante para indicar el proceso de atención.</p> <p>-Nombre: -Primer Apellido. -Segundo Apellido. -Parentesco. -Calle. -Numero calle. -Ciudad. -Comuna. -Provincia. -Región. -Código área. -Teléfono fijo. -Móvil. -RUN.</p>	<p>Estándar de atención de urgencias de los niveles de alta, mediana y baja complejidad de la atención primaria. Decreto N° 467 23 Mayo 2013.</p> <p>Decreto Exento N° 820/11 sobre Estándares de Información en Salud.</p> <p>Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.</p>	
S3.3	Visualizar datos del paciente	Ver datos registrados del paciente.	Aplicación que permita ver los datos del paciente registrados por medio de este sistema informático.	No aplica	

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

	S3.4	Pre-Admisión de un Paciente	Generar la pre-admisión del paciente.	Función que permita generar la pre-admisión del paciente, en virtud de la información registrada del paciente con anticipación.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
INFORMATION INFRASTRUCTURE	ID	NOMBRE	DECLARACIÓN	DESCRIPCIÓN	REFERENCIA	OTRO
	I 1	Seguridad	Establecer seguridad en el acceso de los datos del sistema	Aplicación que permita filtrar el acceso a la información por medio de correctas autorizaciones validadas mediante usuario y contraseña.	HL7 EHR System Functional Model Draft Standard for Trial Use July, 2004	
	I 5	Interoperabilidad basada en estándares.	Permitir la Interoperabilidad del sistema.	Función que administre mensajería basada en estándares HL7 v2.5.1	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
	I 5.1	Codificación de mensajes	Codificar mensajería HL7 según guía de implementación de mensajería HL7 de admisión de Urgencias Chile.	Función que permita estructurar mensajes HL7 en función de lo normado en la guía de implementación. Se codificarán los mensajes. -A01 -A02 -A04 -A06	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
	I 5.2	Decodificación de mensajes	Decodificar mensajería HL7 según guía de implementación de mensajería HL7 de admisión de Urgencias Chile.	Función que permita decodificar mensajes HL7 o leer su contenido en función de lo normado en la guía de implementación. Además de verificar si hay errores y formar ACK.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	

	I 5.3	Envío y recepción de mensajes	Enviar mensajes HL7 mediante protocolo TPC IP.	Aplicación que permita enviar y recibir mensajes entre plataformas por medio de la definición de ip y puerto de comunicación. Además se debe visualizar el envío de mensajes y respuestas ACK.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
	I 5.4	Monitorización de mensajes	Monitorizar construcción de mensajes y recepción de estos	Aplicación que permita visualizar la correcta codificación, decodificación, envío y recepción de los mensajes HL7	No aplica	
	I 5.5	Almacenamiento y visualización de mensajes.	Almacenar datos de los mensajes	Función que permita almacenar los mensajes codificados enviados y los mensajes decodificados recibidos. Información del almacenamiento: -Mensaje HL7. -Tipo Mensaje. -Hora envío y recibido. -Fecha envío y recibido.	No aplica	

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

SISTEMA REGISTRO DE DATOS DE URGENCIA.

	ID	NOMBRE	DECLARACIÓN	DESCRIPCIÓN	REFERENCIA	OTRO
DIRECT CARE	D 1	Soporte Clínico.				
	D 2	Medición, Análisis, Investigación y Reportes				
	D 3	Administrative and Financial				
SU	ID	NOMBRE	DECLARACIÓN	DESCRIPCIÓN	REFERENCIA	OTRO

Tabla 6: Sistema de Registro de Datos de Urgencias. (Fuente: Elaboración propia basado en Health Level Seven, 2004).

	S3	Administración de Operaciones y Comunicación	Administrar información del paciente por medio del registro y operación de datos	Aplicación que permita registrar los datos del paciente y ejecutar traslados y admisión del paciente.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
	S3.1	Visualizar datos del paciente	Ver datos registrados del paciente.	Aplicación que permita ver los datos del paciente y de su acompañante registrados al momento de generar la pre-admisión desde el módulo de pre-atención.	<p>Estándar de atención de urgencias de los niveles de alta, mediana y baja complejidad de la atención primaria. Decreto N° 467 23 Mayo 2013.</p> <p>Decreto Exento N° 820/11 sobre Estándares de Información en Salud.</p> <p>Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.</p>	
	S3.2	Traslado de un Paciente	Generar el traslado de un paciente.	Función que permita generar la pre-admisión del paciente, en virtud de la información registrada del paciente con anticipación.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
	S3.3	Admisión de un Paciente	Generar la admisión del paciente a la institución	Función que permita generar la admisión del paciente a la institución de salud.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
INF	ID	NOMBRE	DECLARACIÓN	DESCRIPCIÓN	REFERENCIA	OTRO

I 1	Seguridad	Establecer seguridad en el acceso de los datos del sistema	Aplicación que permita filtrar el acceso a la información por medio de correctas autorizaciones validadas mediante usuario y contraseña.	HL7 EHR System Functional Model Draft Standard for Trial Use July, 2004	
I 5	Interoperabilidad basada en estándares.	Permitir la Interoperabilidad del sistema.	Función que administre mensajería basada en estándares HL7 v2.5.1	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
I 5.1	Codificación de mensajes	Codificar mensajería HL7 según guía de implementación de mensajería HL7 de admisión de Urgencias Chile.	Función que permita estructurar mensajes HL7 en función de lo normado en la guía de implementación. Se codificarán los mensajes. -A01 -A02 -A04 -A06	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
I 5.2	Decodificación de mensajes	Decodificar mensajería HL7 según guía de implementación de mensajería HL7 de admisión de Urgencias Chile.	Función que permita decodificar mensajes HL7 o leer su contenido en función de lo normado en la guía de implementación. Además de verificar si hay errores y formar ACK.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

	I 5.3	Envío y recepción de mensajes	Enviar mensajes HL7 mediante protocolo TPC IP.	Aplicación que permita enviar y recibir mensajes entre plataformas por medio de la definición de ip y puerto de comunicación. Además se debe visualizar el envío de mensajes y respuestas ACK.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
	I 5.4	Monitorización de mensajes	Monitorizar construcción de mensajes y recepción de estos	Aplicación que permita visualizar la correcta codificación, decodificación, envío y recepción de los mensajes HL7	No aplica	
	I 5.5	Almacenamiento y visualización de mensajes.	Almacenar datos de los mensajes	Función que permita almacenar los mensajes codificados enviados y los mensajes decodificados recibidos. Información del almacenamiento: -Mensaje HL7. -Tipo Mensaje. -Hora envío y recibido. -Fecha envío y recibido.	No aplica	

SISREMA DE REGISTRO DE DATOS PACIENTE.

	ID	NOMBRE	DECLARACIÓN	DESCRIPCIÓN	REFERENCIA	OTRO
DIRECT CARE	D 1	Soporte Clínico.				
	D 2	Medición, Análisis, Investigación y Reportes				
	D 3	Administrative and Financial				
SU	ID	NOMBRE	DECLARACIÓN	DESCRIPCIÓN	REFERENCIA	OTRO

Tabla 7: Sistema de Registro de Datos Paciente. (Fuente: Elaboración propia basado en Health Level Seven, 2004).

	S3	Administración de Operaciones y Comunicación	Administrar información del paciente por medio del registro y operación de datos	Aplicación que permita registrar los datos del paciente y ejecutar traslados y admisión del paciente.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
	S3.1	Almacenamiento datos del paciente	Almacenar datos del paciente	Aplicación que permita almacenar los datos del paciente desde información enviada desde otros módulos. Debe ser capaz de permitir la visualización de esta información por medio de una interfaz. Datos demográficos. Datos clínicos básicos.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
INFORMATION INFRASTRUCTURE	ID	NOMBRE	DECLARACIÓN	DESCRIPCIÓN	REFERENCIA	OTRO
	I 1	Seguridad	Establecer seguridad en el acceso de los datos del sistema	Aplicación que permita filtrar el acceso a la información por medio de correctas autorizaciones validadas mediante usuario y contraseña.	HL7 EHR System Functional Model Draft Standard for Trial Use July, 2004	
	I 5	Interoperabilidad basada en estándares.	Permitir la Interoperabilidad del sistema.	Función que administre mensajería basada en estándares HL7 v2.5.1	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

I 5.1	Codificación de mensajes	Codificar mensajería HL7 según guía de implementación de mensajería HL7 de admisión de Urgencias Chile.	Función que permita estructurar mensajes HL7 en función de lo normado en la guía de implementación. Se codificarán los mensajes. -A01 -A02 -A04 -A06	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
I 5.2	Decodificación de mensajes	Decodificar mensajería HL7 según guía de implementación de mensajería HL7 de admisión de Urgencias Chile.	Función que permita decodificar mensajes HL7 o leer su contenido en función de lo normado en la guía de implementación. Además de verificar si hay errores y formar ACK.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
I 5.3	Envío y recepción de mensajes	Enviar mensajes HL7 mediante protocolo TPC IP.	Aplicación que permita enviar y recibir mensajes entre plataformas por medio de la definición de ip y puerto de comunicación. Además se debe visualizar el envío de mensajes y respuestas ACK.	Guía de implementación de Mensajería 2.5.1 de Admisión de Urgencias Chile.	
I 5.4	Monitorización de mensajes	Monitorizar construcción de mensajes y recepción de estos	Aplicación que permita visualizar la correcta codificación, decodificación, envío y recepción de los mensajes HL7	No aplica	

	I 5.5	Almacenamiento y visualización de mensajes.	Almacenar datos de los mensajes	<p>Función que permita almacenar los mensajes codificados enviados y los mensajes decodificados recibidos.</p> <p>Información del almacenamiento:</p> <ul style="list-style-type: none"> -Mensaje HL7. -Tipo Mensaje. -Hora envío y recibido. -Fecha envío y recibido. 	No aplica	
--	----------	---	---------------------------------	--	-----------	--

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

Código de implementación del Servidor

```
<?php
//Referencia del código http://www.entrecodigos.net/2012/06/socket-en-php-cliente-servidor/
//include 'lector.php';
include 'error_mensaje.php';
echo "##### SERVIDOR MENSAJERIA HL7 SISTEMA REGISTRO
URGENCIA#####\r\n ";
$socket=socket_create(AF_INET,SOCK_STREAM,0);
$direccion=0;
include 'funcionest2.php';
$puerto=4547;
$con=0;
socket_bind($socket, $direccion,$puerto);
echo "SOCKET INICIADO POR PUERTO ".$puerto."...\n\n";
socket_listen($socket);
echo "Escuchando socket ".$socket."...\n";
$tamaño=2048;
while(1){
    $cliente=socket_accept($socket);
    echo "\n\n\n\n\n::::Usuario conectado".$cliente."::::\n\n\n\n";
    $con = $con + 1;
    $buffer=socket_read($cliente, $tamaño);
    $ACK=error_mensaje($buffer);
    $buffer=$ACK;
    socket_write($cliente, $buffer);
    socket_close($cliente);
    echo "\n\n\n\n\n::::Fin de conexion con Usuario: ".$cliente." Numero de conexiones
".$con."::::\n\n\n\n";
    $buffer="";
}
socket_close($socket);
?>
```

Código de implementación del Cliente:

```
<?php
//Referencia del código http://www.entrecodigos.net/2012/06/socket-en-php-cliente-servidor/
function enviar_mensaje_datos_paciente($mensaje){
$host="127.0.0.1";
$socket=socket_create(AF_INET,SOCK_STREAM,SOL_TCP);
$puerto=4546;
$conexion=socket_connect($socket,$host,$puerto);
```

```

$tamaño=2048;
if($conexion){
    echo "Cliente: Conexion Exitosa, puerto ".$puerto."<br/><br/>";
    $salida="";
    socket_write($socket,$mensaje);

    $salida=socket_read($socket,$tamaño);

    return $salida;

    // echo "Servidor: ".$salida."<br/>";
}
else{
echo "Conexión fallida, puerto: ".$puerto;
}

socket_close($socket);
}
?>

```

Anexo 3. “implementación codificación y decodificación de mensajería hl7 de admisión de urgencias chile”.

Codificador mensajería HL7 de Admisión de Urgencias Chile.

```

<?php
//Formación de cabecera MSH//
$fecha = timedate();
$ftMSH=$fecha[0].'&'.$fecha[1];
$tipo_mensaje='ADT ^ A01 ^ ADT_A01!;//Tipo mensaje depende del mensaje a enviar.
$ID_CONTROL='00001';
$ID_PROCESAMIENTO='01';
$MODO_PROCESAMIENTO='P';
if($ID_PROCESAMIENTO==""){ $IM=$MODO_PROCESAMIENTO;}
if($MODO_PROCESAMIENTO==""){ $IM=$ID_PROCESAMIENTO;}
if (($ID_PROCESAMIENTO!=") and
($MODO_PROCESAMIENTO!=")){ $IM=$ID_PROCESAMIENTO.' ^ '.$MODO_PROCESAMIENT
O;}
$ID_VERSION='2.5.1';

$MSH='MSH| ^ ~\&| | | | |'.$ftMSH.'| |'.$tipo_mensaje.'|'.$ID_CONTROL.'|'.$IM.'|'.$ID_VER
SION.'| <cr>';
////////////////////////////////////
//Formación EVN//
$tipo_evento='A01!;//definir segun evento

```

```
$fecha=timedate();
$ftEVN=$fecha[0].'&'.$fecha[1];
$responsable='001';//definir desde base de datos;
$lugar_evento='06^Valparaíso_San_Antonio';
$EVN='EVN|'.$tipo_evento.'|'.$ftEVN.'|'.$responsable.'|'.$lugar_evento.'|<cr>';
////////////////////////////////////
//Formación PID//
/*
$numero_ficha_c=;
$run=;
$nombre_paciente=;
$fecha_de_nacimiento=;
$sexo=;
$direccion=;
$telefono_fijo=;
$estado_conyugal=;
$nacionalidad=;
$pueblo_originario=;

$PID='PID|'.$numero_ficha_c.'|'.$run.'|'.$nombre_paciente.'|'.$fecha_de_nacimiento.'|'.
$sexo.'|'.$direccion.'|'.$telefono_movil.'|'.$telefono_fijo.'|'.$estado_conyugal.'|'.
|'.$nacionalidad.'|'.$pueblo_originario.'|<cr>';
*/
////////////////////////////////////
//Formación PD1//
/*
$consultorio=;
$minusvalia=;
$fecha_registro_inmunizacion=;

$PD1='PD1|'.$consultorio.'|'.$minusvalia.'|'.$minusvalia.'|<cr>';
*/
////////////////////////////////////
//Formación NK1//
/*
$ID_acompañante=;
$nombre_acompañante=;
$parentesco=;
$direccion_acompañante=;
$telefono_fijo_acompañante=;
$telefono_movil_acompañante=;
$Run_acompañante=;
```



```
$DG1='DG1|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|<cr>';
*/
$DG1='DG1|01|SNOMED||||W|||||||||<cr>';
////////////////////////////////////
//Formación GT1//
/*
    $Id_transaccion=;
    $nombre_cobertura=;
    $tipo_de_cobertura=;
    $ajustes_cobros=;
$GT1='GT1|'|.$Id_transaccion.'|'|.$nombre_cobertura.'|'|'|'|'|.$tipo_de_cobertura.'|'|'|
|'|'|'|'|'|'|'|'|.$ajustes_cobros.'|<cr>';
*/
////////////////////////////////////
//Formación SFT Segmentos del software.//
/*
$SFT_ACK=SFT|Universidad de Valparaíso|1.0|Health Check|01|;

*/
//EJEMPLO DE APLICACIÓN.
$ADT01=$MSH.$EVN.$PID.$PD1.$NK1.$PV1.$ROL.$DG1.$GT1;
?>
```

Decodificador mensajería HL7 de Admisión de Urgencias Chile.

```
<?php
function error_mensaje($mensaje){
include 'funcionest2.php';
//-----
// Preparar mensaje.
$mensaje=sacar_espacios($mensaje);
echo "\n".$mensaje."\n\n";
//-----
//Separar y vectorizar segmentos con sus respectivos componentes, SEGUN SALTO DE LINEA
"<cr>".
$segmento=explode('<cr>', $mensaje); //Separar y vectorizar segmentos con sus respectivos
componentes, SEGUN SALTO DE LINEA "<cr>".
//-----
// MOSTRAR SEGMENTOS SEPARADOS EN VECTORES
echo "...:Analizando Mensaje:... \n";
//echo "Segmentos del mensaje \n";
for($i=0;$i<count($segmento);$i++){
//echo $segmento[$i]."\n";
}
}
```

```

echo "\n";
//-----
//LECTURA GENERAL DE TODO EL MENSAJE
$MSH_flag=0;
$EVN_flag=0;
$PID_flag=0;
$PD1_flag=0;
$NK1_flag=0;
$PV1_flag=0;
$PV2_flag=0;
$ROL_flag=0;
$GT1_flag=0;
$DG1_flag=0;
$PDA_flag=0;
$ERR_flag=0;
$SFT_flag=0;
$MSA_flag=0;
for($i=0;$i<count($segmento);$i++){
    $componente=explode('|',$segmento[$i]);
    for($j=0;$j<count($componente);$j++){
        if(($componente[$j])=="MSH"){echo "Segmento MSH encontrado*****\n";
$MSH=$componente;$MSH_flag=1;}
        if(($componente[$j])=="EVN"){echo "Segmento EVN encontrado*****\n";
$EVN=$componente;$EVN_flag=1;}
        if(($componente[$j])=="PID"){echo "Segmento PID encontrado*****\n";
$PID=$componente;$PID_flag=1;}
        if(($componente[$j])=="PD1"){echo "Segmento PD1 encontrado*****\n";
$PD1=$componente;$PD1_flag=1;}
        if(($componente[$j])=="NK1"){echo "Segmento NK1 encontrado*****\n";
$NK1=$componente;$NK1_flag=1;}
        if(($componente[$j])=="PV1"){echo "Segmento PV1 encontrado*****\n";
$PV1=$componente;$PV1_flag=1;}
        if(($componente[$j])=="PV2"){echo "Segmento PV2 encontrado*****\n";
$PV2=$componente;$PV2_flag=1;}
        if(($componente[$j])=="ROL"){echo "Segmento ROL encontrado*****\n";
$ROL=$componente;$ROL_flag=1;}
        if(($componente[$j])=="GT1"){echo "Segmento GT1 encontrado*****\n";
$GT1=$componente;$GT1_flag=1;}
        if(($componente[$j])=="DG1"){echo "Segmento DG1 encontrado*****\n";
$DG1=$componente;$DG1_flag=1;}
        if(($componente[$j])=="PDA"){echo "Segmento PDA encontrado*****\n";
$PDA=$componente;$PDA_flag=1;}
        if(($componente[$j])=="ERR"){echo "Segmento ERR encontrado*****\n";
$ERR=$componente;$ERR_flag=1;}
    }
}

```

```
if (($componente[$i]) == "SFT"){echo "Segmento SFT encontrado*****\n";
$SFT=$componente;$SFT_flag=1;}
if (($componente[$i]) == "MSA"){echo "Segmento MSA encontrado*****\n";
$MSA=$componente;$MSA_flag=1;}
}
}
echo "\n";
//-----
//ANALISIS DE SEGEMENTO CABECERA MSH-----
-----
$flag_error=0;
$flag_ACK=0;
$flag_ADT=0;
$flag_ACK_CHECK=0;
//////////

//////////
if($MSH_flag==1){
// $flagMSH2=0;
// $flagMSH7=0;
// $flagMSH9=0;
// $flagMSH10=0;
// $flagMSH11=0;
// $flagMSH12=0;

$flag_errorACK=0;
$MSH1=$MSH[0];

if($MSH[1] == '^ ~\&'){
$MSH2=$MSH[1];
else {$flag_error=200;}
}
else {$flag_error=1; $flagACK=200;}
if($MSH[6]!=""){
$MSH7=explode('&',$MSH[6]);
}
else {$flag_error=1; $flagACK=200;}
if($MSH[8]!=""){
$MSH9=explode('^',$MSH[8]);
if($MSH9[0]=='ADT'){
if($MSH9[1]=='A01' or $MSH9[1]=='A02' or $MSH9[1]=='A03' or
$MSH9[1]=='A04' or $MSH9[1]=='A06' or $MSH9[1]=='A08'
or $MSH9[1]=='A09' or $MSH9[1]=='A10' or $MSH9[1]=='A14' or
$MSH9[1]=='A15' or $MSH9[1]=='A17' or $MSH9[1]=='A28')
```

```

    {
        if($MSH9[2] == 'ADT_A01' or $MSH9[2] == 'ADT_A02' or
$MSH9[2] == 'ADT_A03' or $MSH9[2] == 'ADT_A04' or $MSH9[2] == 'ADT_A06' or
$MSH9[2] == 'ADT_A05' or $MSH9[2] == 'ADT_A08'
        or $MSH9[2] == 'ADT_A09' or $MSH9[2] == 'ADT_A15' or
$MSH9[2] == 'ADT_A17')
        {
        }
        else { $flag_error=201; $flagACK=0; }
    }
    else { $flag_error=201; $flagACK=0;}
    $tipo_eventoACK=$MSH9[1];
    $tipo_evento2ACK=$MSH9[2];
    $flag_ACK=1;
    $flag_ADT=1;
}

```

```

if($MSH9[0] == 'ACK'){
    $flag_ACK_CHECK=1;
    echo "revisaremos el ACK y agregaremos a la DB\n";
}
if ($MSH9[0] != 'ACK' and $MSH9[0] != 'ADT'){ $flag_error=200;}
}
else { $flag_error=200; $flagACK=1;}
if($MSH[10] != ""){
    $MSH10=$MSH[10];
}
else { $flag_error=200; $flagACK=1;}
if($MSH[10] != ""){
    $MSH11=explode('^',$MSH[10]);
}
else { $flag_error=200; $flagACK=1;}
if($MSH[11] != ""){
    if($MSH[11] == '2.5.1'){
        $MSH12=$MSH[11];
    }
}
}
else { $flag_error=200; $flagACK=1;}
////////////////////////////////////

```

```

if ($flag_error == 0 and $flag_ADT == 1)
{
    switch ($MSH9[1]){
        case 'A01' :

```

```
if( $EVN_flag==1 and $PID_flag==1 and $PD1_flag==1 and $NK1_flag==1  
and $PV1_flag==1 and $ROL_flag==1 and  
$GT1_flag==1 and $DG1_flag==1)  
{
```

```
echo "Evento A01 con todos sus segmentos obligatorios\n";
```

```
//SEGMENTO EVN.....  
if($EVN[1]!=" and $EVN[2]!="){  
    echo "Segmento EVN correcto: Componente obligatorio ok  
\n";  
}  
else {$flag_error=200; echo "Segmento EVN incorrecto: Faltan Componente  
obligatorios \n";}  
//SEGMENTO PID.....  
if($PID[2]!=" and $PID[3]!=" and $PID[5]!="){  
    echo "Segmento PID correcto: Componente obligatorio ok  
\n";  
}  
else {$flag_error=200; echo "Segmento PID incorrecto: Faltan Componente  
obligatorios \n";}  
//SEGMENTO PD1.....  
if($PD1[3]!="){  
    echo "Segmento PD1 correcto: Componente obligatorio ok  
\n";  
}  
else {$flag_error=200; echo "Segmento PD1 incorrecto: Faltan Componente  
obligatorios \n";}  
//SEGMENTO PV1.....  
if($PV1[2]!="){  
    echo "Segmento PV1 correcto: Componente obligatorio ok  
\n";  
}  
else {$flag_error=200; echo "Segmento PV1 incorrecto: Faltan Componente  
obligatorios \n";}  
//SEGMENTO NK1.....  
if($NK1[1]!="){  
    echo "Segmento NK1 correcto: Componente obligatorio ok  
\n";  
}  
else {$flag_error=200; echo "Segmento NK1 incorrecto: Faltan Componente  
obligatorios \n";}  
//SEGMENTO ROL.....  
if($ROL[2]!=" and $ROL[3]!=" and $ROL[4]!="){
```

```

                                echo "Segmento ROL correcto: Componente obligatorio ok
\n";
                                }
                                else {$flag_error=200; echo "Segmento ROL incorrecto: Faltan Componente
obligatorios \n";}
                                // /SEGMENTO GT1.....
                                if($GT1[1]!=" and $GT1[3]!="){
                                echo "Segmento GT1 correcto: Componente obligatorio ok
\n";
                                }
                                else {$flag_error=200; echo "Segmento GT1 incorrecto: Faltan Componente
obligatorios \n";}
                                // /SEGMENTO DG1.....
                                if($DG1[1]!=" and $DG1[2]!=" and $DG1[6]!="){
                                echo "Segmento DG1 correcto: Componente obligatorio ok
\n";
                                }
                                else {$flag_error=200; echo "Segmento DG1 incorrecto: Faltan Componente
obligatorios \n";}
                                // // /SEGMENTO ERR.....
                                // if($ERR[3]!=" and $ERR[4]!="){
                                // echo "Segmento ERR correcto: Componente obligatorio
ok \n";
                                // }
                                // else {$flag_error=200; echo "Segmento ERR incorrecto: Faltan
Componente obligatorios \n";}
                                // // /SEGMENTO MSA.....
                                // if($MSA[3]!=" and $MSA[4]!="){
                                // echo "Segmento MSA correcto: Componente obligatorio
ok \n";
                                // }
                                // else {$flag_error=200; echo "Segmento MSA incorrecto: Faltan
Componente obligatorios \n";}
                                // // /SEGMENTO SFT.....
                                // if($SFT[1]!=" and $SFT[2]!=" and $SFT[3]!=" and $SFT[4]!="){
                                // echo "Segmento MSA correcto: Componente obligatorio
ok \n";}
                                // else {$flag_error=200; echo "Segmento SFT incorrecto: Faltan Componente
obligatorios \n";}
                                }
                                else {echo "Evento A01 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}
                                break;
                                case 'A02' :
                                if( $EVN_flag==1 and $PID_flag==1 and $PV1_flag==1 and $ROL_flag==1
and $DG1_flag==1)

```

```
{
  echo "Evento A02 con todos sus segmentos obligatorios\n";
  // /SEGMENTO EVN.....
  if($EVN[1]!=" and $EVN[2]!="){
    echo "Segmento EVN correcto: Componente obligatorio ok \n";
  }
  else {$flag_error=200; echo "Segmento EVN incorrecto: Faltan Componente
obligatorios \n";}
  // /SEGMENTO PID.....
  if($PID[2]!=" and $PID[3]!=" and $PID[5]!="){
    echo "Segmento PID correcto: Componente obligatorio ok
\n";
  }
  else {$flag_error=200; echo "Segmento PID incorrecto: Faltan Componente
obligatorios \n";}

  // /SEGMENTO PV1.....
  if($PV1[2]!="){
    echo "Segmento PV1 correcto: Componente obligatorio ok
\n";
  }
  else {$flag_error=200; echo "Segmento PV1 incorrecto: Faltan Componente
obligatorios \n";}

  // /SEGMENTO ROL.....
  if($ROL[2]!=" and $ROL[3]!=" and $ROL[4]!="){
    echo "Segmento ROL correcto: Componente obligatorio ok
\n";
  }
  else {$flag_error=200; echo "Segmento ROL incorrecto: Faltan Componente
obligatorios \n";}

  // /SEGMENTO DG1.....
  if($DG1[1]!=" and $DG1[2]!=" and $DG1[6]!="){
    echo "Segmento DG1 correcto: Componente obligatorio ok
\n";
  }
  else {$flag_error=200; echo "Segmento DG1 incorrecto: Faltan Componente
obligatorios \n";}

}
else {echo "Evento A02 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}
break;
case 'A03':
  if( $EVN_flag==1 and $PID_flag==1 and $ROL_flag==1
```

```

and $GT1_flag==1 and $PDA_flag==1)
{
  echo "Evento A03 con todos sus segmentos obligatorios\n";
  // /SEGMENTO EVN.....
  if($EVN[1]!=" and $EVN[2]!="){
    echo "Segmento EVN correcto: Componente obligatorio ok
\n";
  }
  else {$flag_error=200; echo "Segmento EVN incorrecto: Faltan Componente
obligatorios \n";}
  // /SEGMENTO PID.....
  if($PID[2]!=" and $PID[3]!=" and $PID[5]!="){
    echo "Segmento PID correcto: Componente obligatorio ok
\n";
  }
  else {$flag_error=200; echo "Segmento PID incorrecto: Faltan Componente
obligatorios \n";}
  // /SEGMENTO ROL.....
  if($ROL[2]!=" and $ROL[3]!=" and $ROL[4]!="){
    echo "Segmento ROL correcto: Componente obligatorio ok
\n";
  }
  else {$flag_error=200; echo "Segmento ROL incorrecto: Faltan Componente
obligatorios \n";}

  // /SEGMENTO GT1.....
  if($GT1[1]!=" and $GT1[3]!="){
    echo "Segmento GT1 correcto: Componente obligatorio ok
\n";
  }
  else {$flag_error=200; echo "Segmento GT1 incorrecto: Faltan Componente
obligatorios \n";}
  // /SEGMENTO DG1.....
  if($DG1[1]!=" and $DG1[2]!=" and $DG1[6]!="){
    echo "Segmento DG1 correcto: Componente obligatorio ok
\n";
  }
  else {$flag_error=200; echo "Segmento DG1 incorrecto: Faltan Componente
obligatorios \n";}
}
else {echo "Evento A03 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}
break;
case 'A04' :
  if( $EVN_flag==1 and $PID_flag==1 and $GT1_flag==1 and $PV1_flag==1)
  {

```

```
        echo "Evento A04 con todos sus segmentos obligatorios\n";
        //SEGMENTO EVN.....
        if($EVN[1]!=" and $EVN[2]!="){
            echo "Segmento EVN correcto: Componente obligatorio ok
\n";
        }
        else {$flag_error=200; echo "Segmento EVN incorrecto: Faltan Componente
obligatorios \n";}
        //SEGMENTO PID.....
        if($PID[2]!=" and $PID[3]!=" and $PID[5]!="){
            echo "Segmento PID correcto: Componente obligatorio ok
\n";
        }
        else {$flag_error=200; echo "Segmento PID incorrecto: Faltan Componente
obligatorios \n";}

        //SEGMENTO PV1.....
        if($PV1[2]!="){
            echo "Segmento PV1 correcto: Componente obligatorio ok
\n";
        }
        else {$flag_error=200; echo "Segmento PV1 incorrecto: Faltan Componente
obligatorios \n";}

        //SEGMENTO GT1.....
        if($GT1[1]!=" and $GT1[3]!="){
            echo "Segmento GT1 correcto: Componente obligatorio ok
\n";
        }
        else {$flag_error=200; echo "Segmento GT1 incorrecto: Faltan Componente
obligatorios \n";}

        //if($flag_error==0){ //Insertar datos en la DB
        //include ('insertar_A04.php');
        //}
    }
    else {echo "Evento A04 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}
break;
case 'A06':
    if( $EVN_flag==1 and $PID_flag==1 and $DG1_flag==1 and $GT1_flag==1)
    {
        echo "Evento A06 con todos sus segmentos obligatorios\n";
        //SEGMENTO EVN.....
        if($EVN[1]!=" and $EVN[2]!="){
```

```

                                echo "Segmento EVN correcto: Componente obligatorio ok
\n";
                                }
                                else {$flag_error=200; echo "Segmento EVN incorrecto: Faltan Componente
obligatorios \n";}
                                // /SEGMENTO PID.....
                                if($PID[2]!=" and $PID[3]!=" and $PID[5]!="){
                                echo "Segmento PID correcto: Componente obligatorio ok
\n";
                                }
                                else {$flag_error=200; echo "Segmento PID incorrecto: Faltan Componente
obligatorios \n";}

```

```

                                // /SEGMENTO GT1.....
                                if($GT1[1]!=" and $GT1[3]!="){
                                echo "Segmento GT1 correcto: Componente obligatorio ok
\n";
                                }
                                else {$flag_error=200; echo "Segmento GT1 incorrecto: Faltan Componente
obligatorios \n";}
                                // /SEGMENTO DG1.....
                                if($DG1[1]!=" and $DG1[2]!=" and $DG1[6]!="){
                                echo "Segmento DG1 correcto: Componente obligatorio ok
\n";
                                }
                                else {$flag_error=200; echo "Segmento DG1 incorrecto: Faltan Componente
obligatorios \n";}
                                }
                                else {echo "Evento A06 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}
                                break;
                                case 'A08' :
                                if( $EVN_flag==1 and $PID_flag==1 and $PD1_flag==1 and $NK1_flag==1
and $PV1_flag==1 and $ROL_flag==1)
                                {
                                echo "Evento A08 con todos sus segmentos obligatorios\n";
                                // /SEGMENTO EVN.....
                                if($EVN[1]!=" and $EVN[2]!="){
                                echo "Segmento EVN correcto: Componente obligatorio ok \n";
                                }
                                else {$flag_error=200; echo "Segmento EVN incorrecto: Faltan Componente
obligatorios \n";}
                                // /SEGMENTO PID.....
                                if($PID[2]!=" and $PID[3]!=" and $PID[5]!="){

```



```

else {$flag_error=200; echo "Segmento PID incorrecto: Faltan Componente
obligatorios \n";}

//SEGMENTO PV1.....
if($PV1[2]!="){
echo "Segmento PV1 correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento PV1 incorrecto: Faltan Componente
obligatorios \n";}
}
else {echo "Evento A09 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}
break;
case 'A10' :
if( $EVN_flag==1 and $PID_flag==1 and $PV1_flag==1)
{
echo "Evento A10 con todos sus segmentos obligatorios\n";
//SEGMENTO EVN.....
if($EVN[1]!=" and $EVN[2]!="){
echo "Segmento EVN correcto: Componente obligatorio ok \n";
}
else {$flag_error=200; echo "Segmento EVN incorrecto: Faltan Componente
obligatorios \n";}
//SEGMENTO PID.....
if($PID[2]!=" and $PID[3]!=" and $PID[5]!="){
echo "Segmento PID correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento PID incorrecto: Faltan Componente
obligatorios \n";}

//SEGMENTO PV1.....
if($PV1[2]!="){
echo "Segmento PV1 correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento PV1 incorrecto: Faltan Componente
obligatorios \n";}
}
else {echo "Evento A10 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}
break;
case 'A14' :
if( $EVN_flag==1 and $PID_flag==1 and $PV1_flag==1)
{
echo "Evento A14 con todos sus segmentos obligatorios\n";

```

```
// /SEGMENTO EVN.....
if($EVN[1]!=" and $EVN[2]!="){
    echo "Segmento EVN correcto: Componente obligatorio ok \n";
}
else {$flag_error=200; echo "Segmento EVN incorrecto: Faltan Componente
obligatorios \n";}

// /SEGMENTO PID.....
if($PID[2]!=" and $PID[3]!=" and $PID[5]!="){
    echo "Segmento PID correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento PID incorrecto: Faltan Componente
obligatorios \n";}

// /SEGMENTO PV1.....
if($PV1[2]!="){
    echo "Segmento PV1 correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento PV1 incorrecto: Faltan Componente
obligatorios \n";}
}
else {echo "Evento A14 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}
break;
case 'A15' :
if( $EVN_flag==1 and $PID_flag==1 and $NK1_flag==1 and $PV1_flag==1 and
$ROL_flag==1 and
$DG1_flag==1)
{
    echo "Evento A15 con todos sus segmentos obligatorios\n";
}
// /SEGMENTO EVN.....
if($EVN[1]!=" and $EVN[2]!="){
    echo "Segmento EVN correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento EVN incorrecto: Faltan Componente
obligatorios \n";}

// /SEGMENTO PID.....
if($PID[2]!=" and $PID[3]!=" and $PID[5]!="){
    echo "Segmento PID correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento PID incorrecto: Faltan Componente
obligatorios \n";}
```

```

//SEGMENTO PV1.....
if($PV1[2]!="){
    echo "Segmento PV1 correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento PV1 incorrecto: Faltan Componente
obligatorios \n";}
// /SEGMENTO NK1.....
if($NK1[1]!="){
    echo "Segmento NK1 correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento NK1 incorrecto: Faltan Componente
obligatorios \n";}
// /SEGMENTO ROL.....
if($ROL[2]!=" and $ROL[3]!=" and $ROL[4]!="){
    echo "Segmento ROL correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento ROL incorrecto: Faltan Componente
obligatorios \n";}

// /SEGMENTO DG1.....
if($DG1[1]!=" and $DG1[2]!=" and $DG1[6]!="){
    echo "Segmento DG1 correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento DG1 incorrecto: Faltan Componente
obligatorios \n";}
}
else {echo "Evento A15 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}
break;
case 'A17' :
    if( $EVN_flag==1 and $PID_flag==1 and $PD1_flag==1 and $NK1_flag==1
and $PV1_flag==1 and $ROL_flag==1 and
$GT1_flag==1 and $DG1_flag==1)
    {
        echo "Evento A17 con todos sus segmentos obligatorios\n";
// /SEGMENTO EVN.....
if($EVN[1]!=" and $EVN[2]!="){
    echo "Segmento EVN correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento EVN incorrecto: Faltan Componente
obligatorios \n";}
}

```

```
//SEGMENTO PID.....
if($PID[2]!=" and $PID[3]!=" and $PID[5]!="){
    echo "Segmento PID correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento PID incorrecto: Faltan Componente
obligatorios \n";}
//SEGMENTO PD1.....
if($PD1[3]!="){
    echo "Segmento PD1 correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento PD1 incorrecto: Faltan Componente
obligatorios \n";}
//SEGMENTO PV1.....
if($PV1[2]!="){
    echo "Segmento PV1 correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento PV1 incorrecto: Faltan Componente
obligatorios \n";}
//SEGMENTO NK1.....
if($NK1[1]!="){
    echo "Segmento NK1 correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento NK1 incorrecto: Faltan Componente
obligatorios \n";}
//SEGMENTO ROL.....
if($ROL[2]!=" and $ROL[3]!=" and $ROL[4]!="){
    echo "Segmento ROL correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento ROL incorrecto: Faltan Componente
obligatorios \n";}
//SEGMENTO GT1.....
if($GT1[1]!=" and $GT1[3]!="){
    echo "Segmento GT1 correcto: Componente obligatorio ok
\n";
}
else {$flag_error=200; echo "Segmento GT1 incorrecto: Faltan Componente
obligatorios \n";}
//SEGMENTO DG1.....
if($DG1[1]!=" and $DG1[2]!=" and $DG1[6]!="){
```

```

                                echo "Segmento DG1 correcto: Componente obligatorio ok
\n";
                                }
                                else {$flag_error=200; echo "Segmento DG1 incorrecto: Faltan Componente
obligatorios \n";}
                                }
                                else {echo "Evento A17 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}
                                break;
                                case 'A28' ://Por implementar...
                                if( $EVN_flag==1 and $PID_flag==1 and $PV1_flag==1 and $PR1_flag==1)
//FALTA PR1.
                                {
                                echo "Evento A28 con todos sus segmentos obligatorios\n";

                                }
                                else {echo "Evento A28 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}

                                break;
                                }
                                }

if ($flag_error==0 and $flag_ACK_CHECK==1){
    if( $SFT_flag==1 and $MSA_flag==1 and $ERR_flag==1)
    {
        echo "ACK con todos sus segmentos obligatorios\n";
        //SEGMENTO ERR.....
        if($ERR[3]!=" and $ERR[4]!="){
            echo "Segmento ERR correcto: Componente obligatorio ok
\n";
            }
        //SEGMENTO MSA.....
        if($MSA[3]!=" and $MSA[4]!="){
            echo "Segmento MSA correcto: Componente obligatorio ok
\n";
            }
        //SEGMENTO SFT.....
        if($SFT[1]!=" and $SFT[2]!=" and $SFT[3]!=" and $SFT[4]!="){
            echo "Segmento MSA correcto: Componente obligatorio ok
\n";
            }
        }
        }
        else {echo "ACK con segmentos obligatorios faltantes\n\n"; $flag_error=1;}
    }
}

```

```
    }

    else $flag_error=200;
if ($flag_error!=0){
    echo "Error en la estructura del mensaje\n";
}

if($flag_ADT==1){
//Formar ACK
//Cabecera.
$fecha=timedate();
$ft=$fecha[0].'&'.$fecha[1];
$MSH_ACK="MSH|^~\&|||||".$ft."||".ACK^".$tipo_eventoACK."^".$tipo_evento2AC
K."|01|01^P|2.5.1|<cr>";
//SFT Segmentos del software.
$SFT_ACK='SFT|Universidad de Valparaíso|1.0|HealthCheck|01|<cr>';
//ERR Segmento notificación de error.
if($flag_error==0){ $codigo_error='0 ^ Mensaje Aceptado!;}
if($flag_error==100){ $codigo_error='100 ^ Error en la secuencia de segmentos!;}
if($flag_error==101){ $codigo_error='101 ^ Archivo requerido perdido!;}
if($flag_error==102){ $codigo_error='102 ^ Error de tipo de data!;}
if($flag_error==103){ $codigo_error='103 ^ Tabla no encontrada!;}
if($flag_error==200){ $codigo_error='200 ^ Tipo de mensaje no valido!;}
if($flag_error==201){ $codigo_error='201 ^ Codigo de evento no valido!;}
if($flag_error==202){ $codigo_error='202 ^ ID de procesamiento no valida!;}
if($flag_error==203){ $codigo_error='203 ^ Version de Id no valida!;}
if($flag_error==204){ $codigo_error='204 ^ Identificador no valido!;}
if($flag_error==205){ $codigo_error='205 ^ Identificador duplicado!;}
if($flag_error==206){ $codigo_error='206 ^ Aplicación de registro bloqueada!;}
if($flag_error==207){ $codigo_error='207 ^ Error interno en la aplicación!;}

if($flag_error==0){$Gravedad='I ^ Envio exitoso!;}
if($flag_error!=0){$Gravedad='E ^ Envio fallido!;}
$ERRMSH='ERR|'.$codigo_error.'|'.$Gravedad.'|';
//MSA Segmento notificación de error. pendiente!!

$ACKMSH=$MSH_ACK.$SFT_ACK.$ERRMSH;
echo "\nFormando ACK: \n".$ACKMSH;
return $ACKMSH;
}
}?:>
```

Anexo 4. “diseño de bases de datos para módulos de prueba del sistema de admisión de urgencias”.

Bases de datos de Pre-atención de Urgencias.

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 misión de Urgencias

79

Figura 20: Base de datos de Pre-Atención de Urgencias.

pre_atencion_registro
Numero ficha clinica : int(250)
RUN : varchar(250)
Digito verificador : varchar(1)
Nombres : varchar(45)
Primer Apellido : varchar(45)
Segundo Apellido : varchar(45)
Fecha nacimiento : date
Sexo : varchar(1)
Nombre Calle : varchar(45)
Numero Calle : varchar(45)
Ciudad : varchar(45)
Comuna : varchar(45)
Provincia : varchar(45)
Region : varchar(45)
numero telefono movil : varchar(250)
codigo area : varchar(3)
numero telefono fijo : varchar(250)
identificador conyugal : varchar(10)
texto conyugal : varchar(250)
identificador nacionalidad : varchar(10)
texto nacionalidad : varchar(250)
valor pueblo : varchar(10)
texto pueblo : varchar(250)
Nombre acompañante : varchar(25)
Primer Apellido acompañante : varchar(25)
Segundo Apellido acompañante : varchar(25)
Parentesco : varchar(25)
Calle acompañante : varchar(25)
numero calle acompañante : varchar(25)
ciudad acompañante : varchar(25)
comuna acompañante : varchar(25)
provincia acompañante : varchar(25)
region acompañante : varchar(25)
codigo area acompañante : varchar(25)
telefono fijo acompan : varchar(25)
movil acompañante : varchar(25)
RUN acompañante : varchar(25)
digito verificador acompañante : varchar(25)
Consultorio Paciente : varchar(250)
Minusvalía : varchar(2)
Fecha de Registro de inmunizacion : date
Tipo de paciente : varchar(250)
Unidad enfermería : varchar(20)
Habitacion : varchar(20)
Cama : varchar(20)
Edificio : varchar(20)
Tipo de admision : varchar(2)
Medico Asignado : varchar(250)
Servicio Asignado : varchar(250)
Origen de la Admision : varchar(250)
Medico durante la admision : varchar(250)
Tipo de alta : varchar(25)
servicio alta : varchar(47)
fecha alta : date
Hora y fecha de admision : date
Hora y fecha del alta : date

pre_atencion_mensajes enviado
id Mensaje : int(11)
Mensaje : varchar(5000)
Tipo mensaje : varchar(4)
hora_mensaje : time
fecha_mensaje : date

pre_atencion_login
User : varchar(20)
Pass : varchar(20)
idEncargado : int(11)
Nombre : varchar(20)
Primer Apellido : varchar(20)
Segundo Apellido : varchar(20)
codigo de area : varchar(4)
numero de telefono fijo : varchar(20)

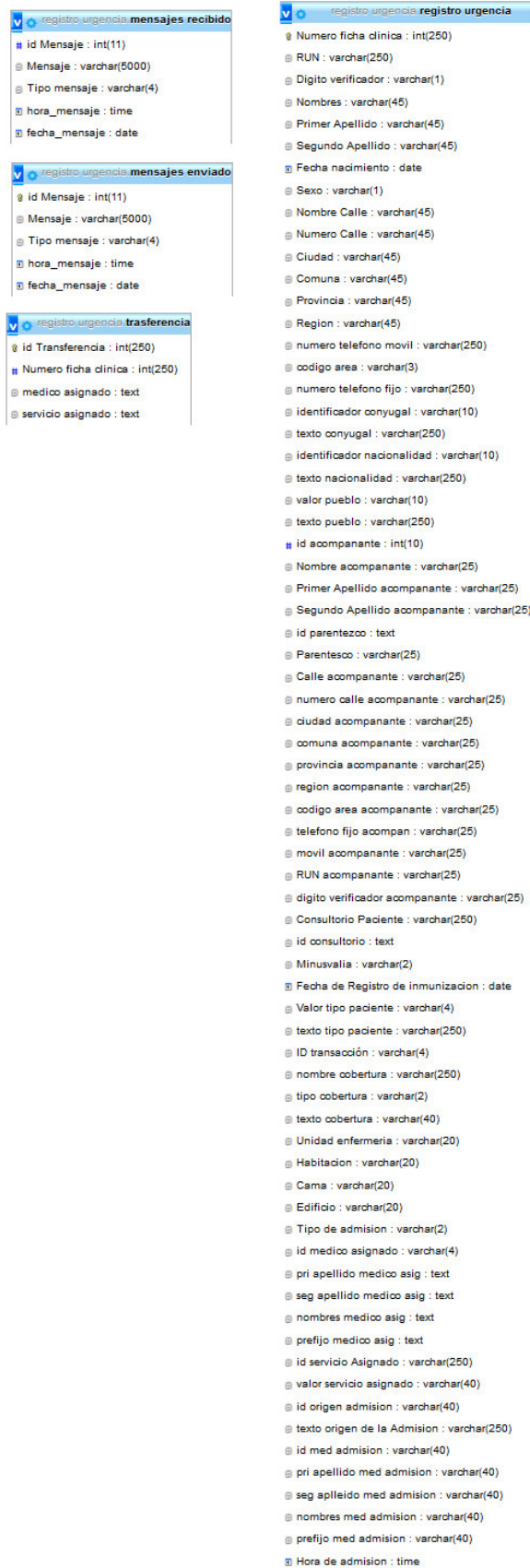
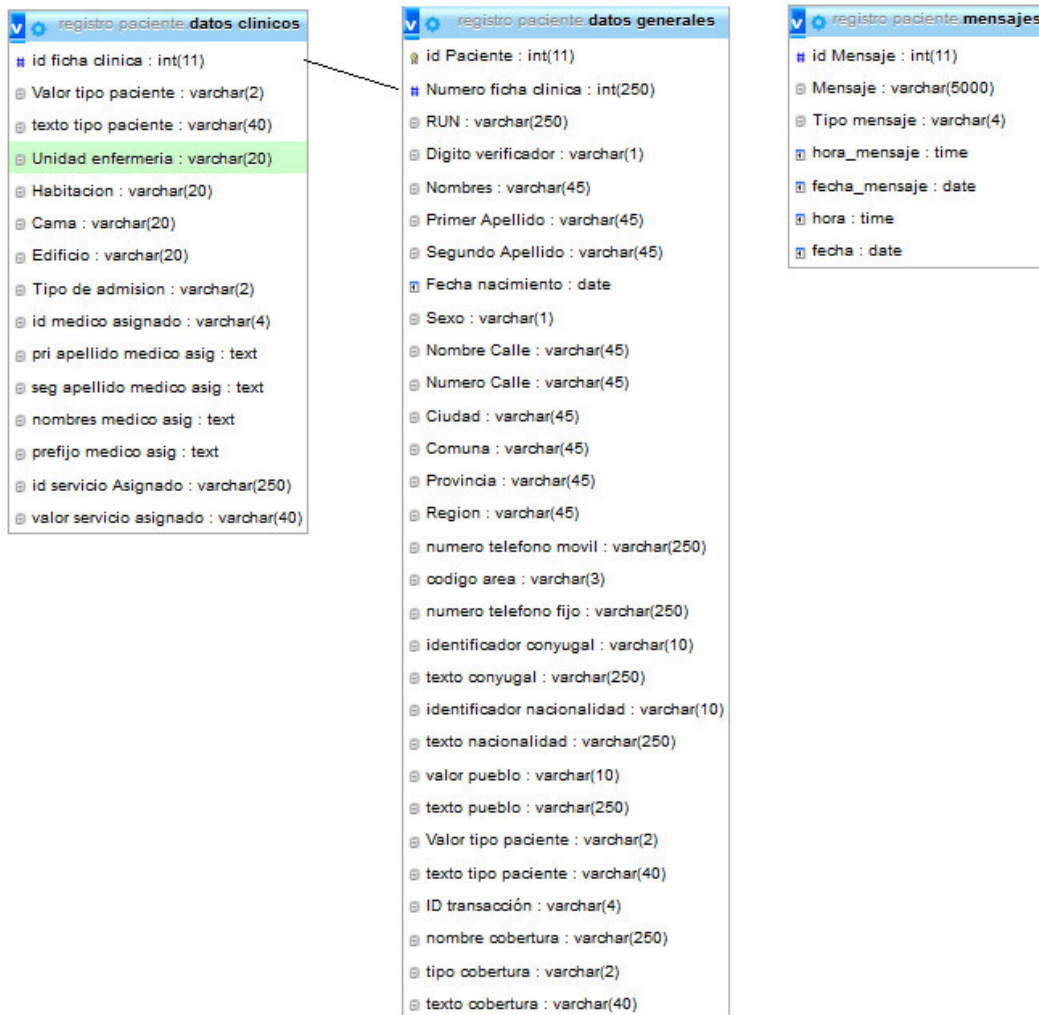


Figura 21: Bases de datos de Registro de Datos de Urgencia.

Bases de datos de Registro de datos de Paciente.

Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

Figura 22: Base de datos de Pre-Atención de Urgencias



Anexo 5. Guía de instalación “prototipo informático para la verificación de mensajes hl7 2.5.1 de la guía de implementación chilena de admisión de urgencias”.

Introducción.

La presente guía tiene como propósito ayudar al usuario en la instalación y uso del prototipo informático para la verificación de mensajería de Admisión de Urgencias.

Descripción del prototipo.

Prototipo informático elaborado por tres módulos de la totalidad de actores del proceso de Admisión de Urgencias.

Módulos y funciones.

- Pre-Atención:
 - Ingreso datos pacientes.
 - Ingreso datos acompañante.
 - Visualización datos paciente.
 - Pre-Admitir paciente.

- Registro datos Urgencia.
 - Visualización datos de pacientes de urgencia.
 - Traslado de pacientes.
 - Admisión de un Paciente.

- Registro datos pacientes.
 - Almacenamiento datos demográficos del paciente.
 - Almacenamiento datos clínicos del paciente.

Mensajes implementados.

- e) **ADT 01:** Notificación de la Admisión o visita.
- f) **ADT 02:** Transferencia de un Paciente.
- g) **ADT 04:** Registro de un Paciente.
- h) **ADT 06:** Cambiar paciente externo a uno interno.

Herramientas necesarias.

Antes de abordar la instalación el prototipo es requisito descargar e instalar las siguientes herramientas.

- 1) **Xampp v 1.8.3:** Plataforma gratuita que integra servidor PHP Apache, base de datos Mysql, etc. Es compatible con los sistemas operativos de “Windows”, “Linux” y “OS x”.
Link de descarga: <https://www.apachefriends.org/es/index.html>. Nota: Instalar por defecto en la raíz del disco duro. Ej: C: //xampp.

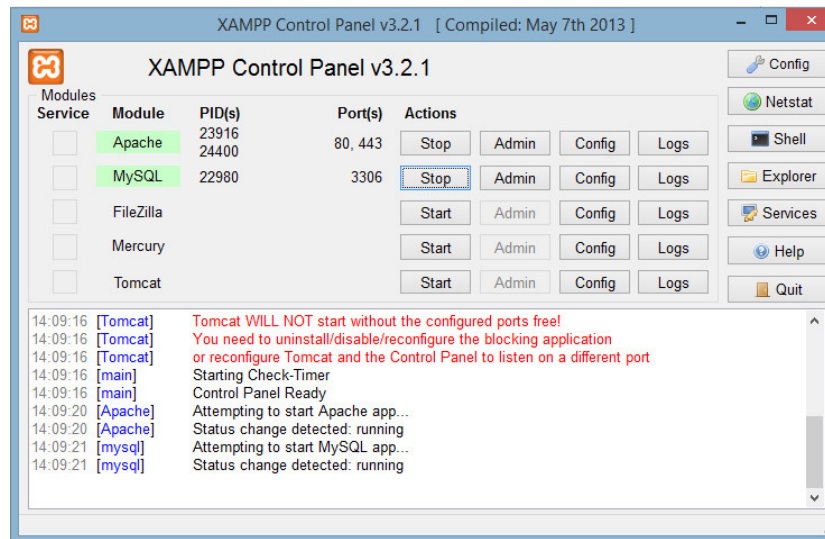


Figura 23: Panel de control XAMPP; Fuente: Elaboración propia.

- 2) Notepad++ v 6.6.9: Software para la edición de múltiples lenguajes de programación. Link de descarga: <http://notepad-plus-plus.org/download/v6.6.9.html>.

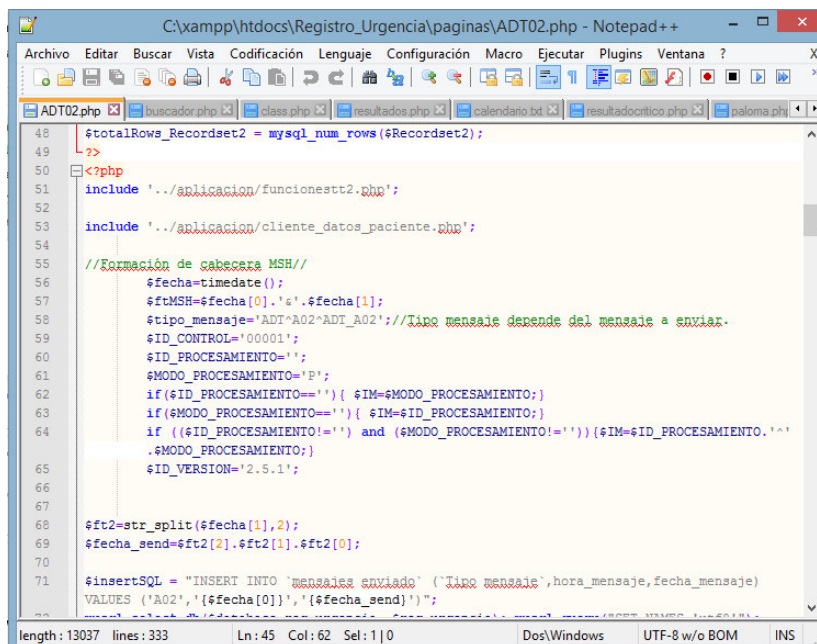


Figura 24: Panel Notepad++; Fuente: Elaboración propia.

Instalación del prototipo

1) Instalación de aplicaciones web.

En la carpeta de instalación se encuentran los archivos del prototipo.

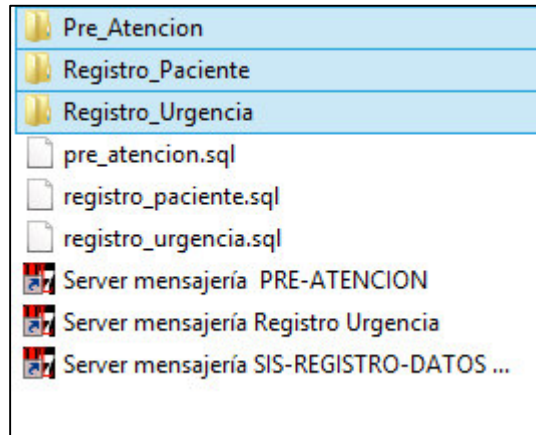


Figura 25: Aplicación web en carpeta prototipo.: Fuente: Elaboración propia.

Seleccionarlos todas las carpetas, copiarlas y pegarlas en c:\xampp\htdocs.

2) Instalación de servidores de mensajería.

En Windows 8 y 7:

- i. Ingresar a “Panel de control/sistema y seguridad/sistema” y hacer click en “Configuración avanzada del sistema.

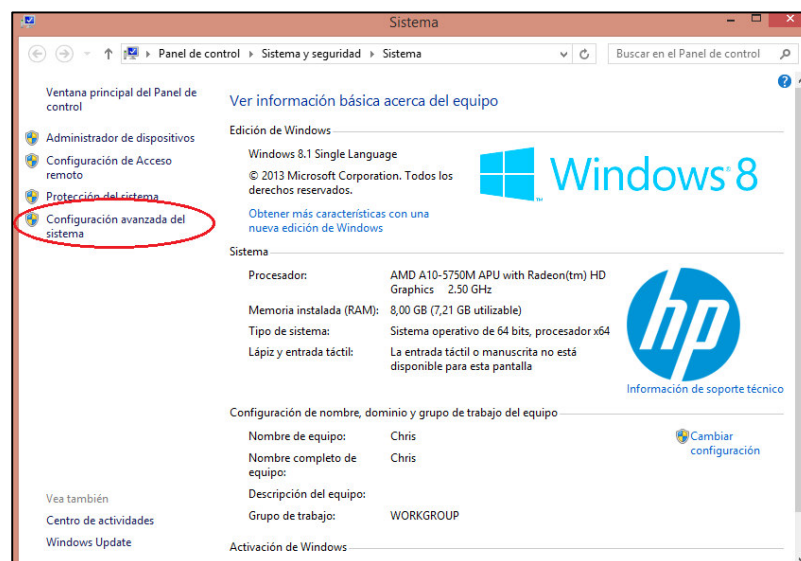
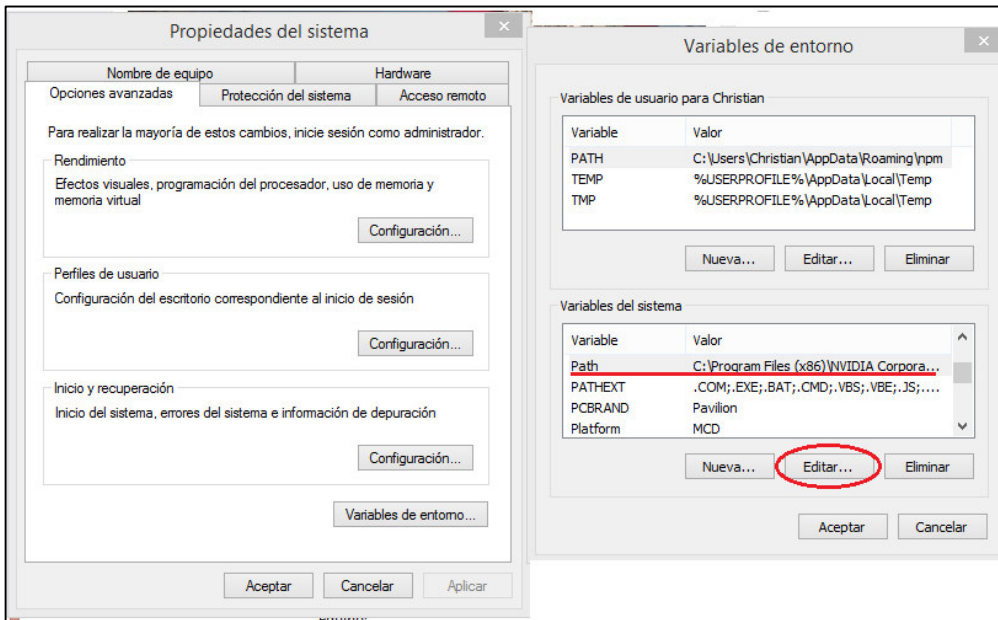


Figura 26: Sistema Windows: Fuente: Elaboración propia.

- ii. En configuración avanzada del sistema /opciones avanzadas hacer click en variables de entorno, seleccionar la variable “Path” y hacer click en editar.



Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

Figura 27: Variables de entorno Windows. Fuente: Elaboración propia.

Aparecerá una ventana de variables del Sistema en donde se debe escribir lo siguiente: “C:\xampp\php;”

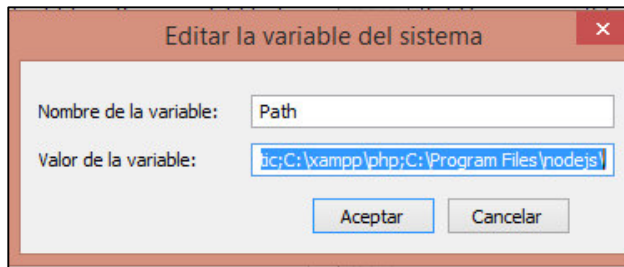


Figura 28: Editar Variables de entorno Windows. Fuente: Elaboración propia.

- iii. Para verificar si está todo correcto, abrir un bloc de notas y escribir:

```
<?php
echo "Esta todo en orden";
?>
```

Guardar en mis documentos con el nombre “prueba.php” y abrir la línea de comandos de Windows “cmd”. Ir a la ruta donde se guardó el archivo y escribir “php prueba.php”, aparecerá el mensaje “Está todo en orden”. En caso de no ser así repasar los puntos anteriores de la guía.

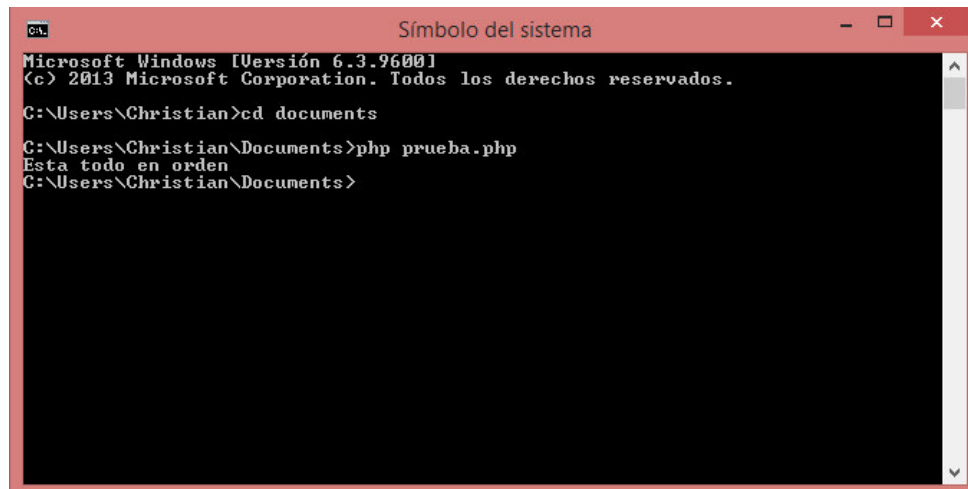


Figura 29: PHP CMD.: Fuente:
Elaboración propia.

- iv. En los archivos del prototipo se encuentran los servidores, copiar y pegar en el escritorio.
Nota: "En caso de instalar XAMPP en una ruta diferente a la especificada en esta Guía modificar el acceso directo".

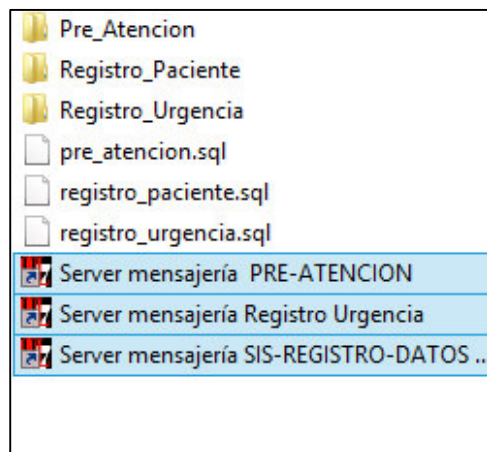
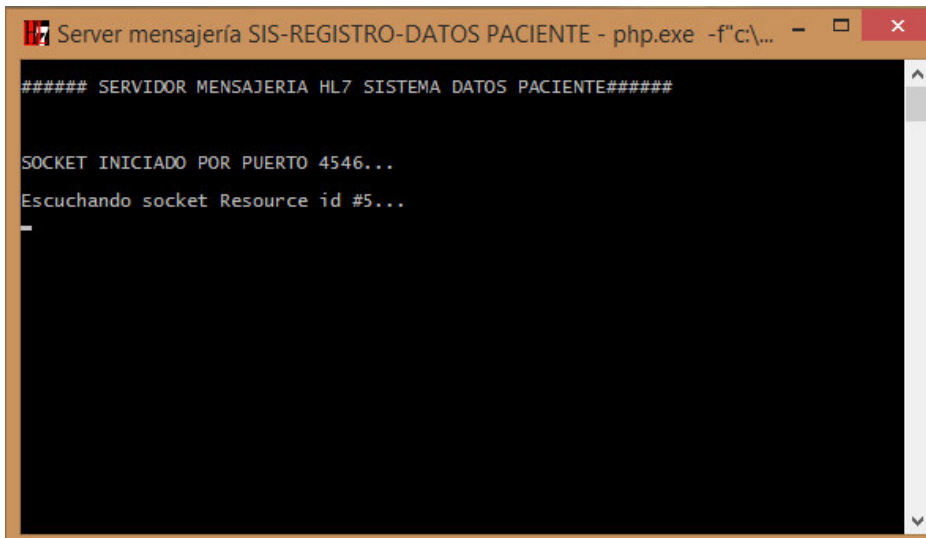


Figura 30: Servidores en
carpeta prototipo.: Fuente:
Elaboración propia.

- v. Hacer click en uno de los archivos para verificar funcionamiento.



Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

87

Figura 31: Servidores mensajería.: Fuente: Elaboración propia.

- 3) Instalación de Bases de Datos.

- a) En el panel de control de XAMPP hacer click en ADMIN del módulo MYSQL.

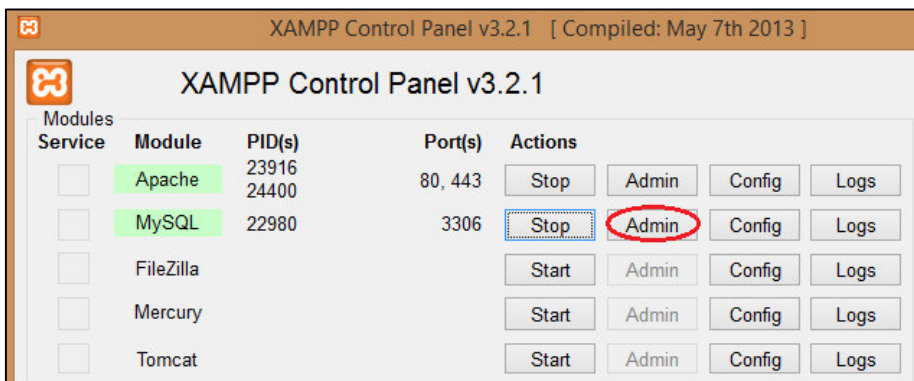


Figura 32: Panel control XAMPP.: Fuente: Elaboración propia.

- b) En PHP ADMIN crear una nueva base de datos con el nombre de uno de los archivos de base de datos presente en las carpetas de los módulos del prototipo. Repetir el proceso para las dos bases de datos restantes.

Figura 33: PHP ADMIN crear DB.: Fuente: Elaboración propia.

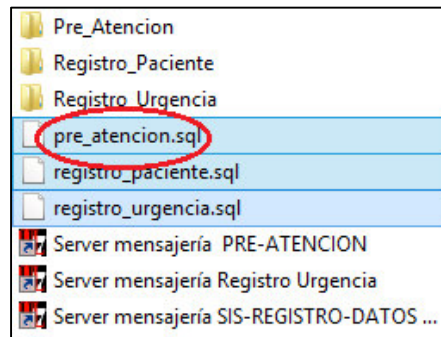
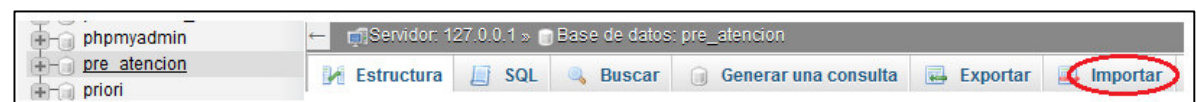


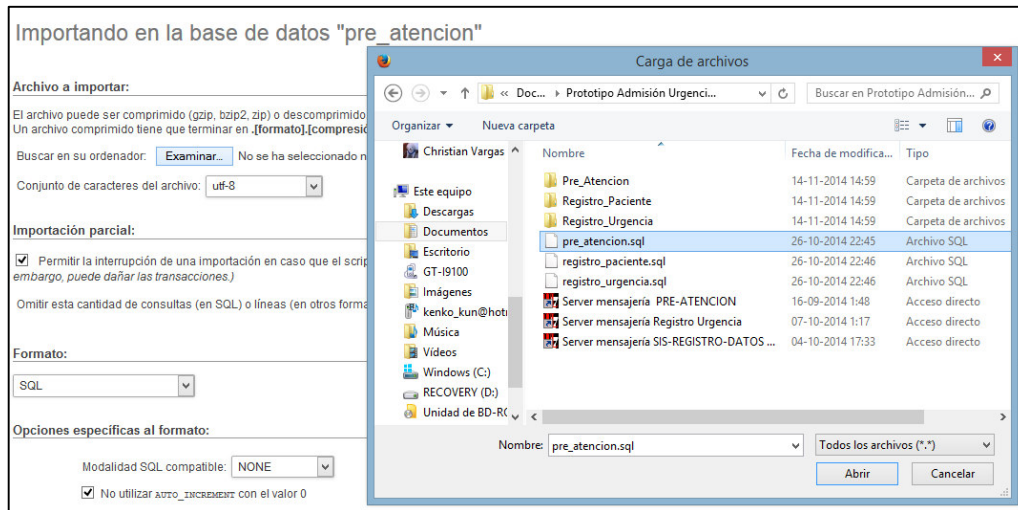
Figura 34: Bases de datos en carpeta prototipo.: Fuente: Elaboración propia.



- c) Click en una de las base de datos creadas. Seleccionar importar y buscar el archivo de la base de datos correspondiente. Repetir el proceso para el resto de bases de datos. Nota: Usuario: root y Password: vacío.

Figura 35: Importar base de datos.: Fuente: Elaboración propia.





Desarrollo de un prototipo informático para la verificación de mensajes HL7 2.5.1 Admisión de Urgencias

Figura 36: Selección base de datos.: Fuente: Elaboración propia.

Iniciar funcionamiento prototipo.

1) Arrancar servidor Apache y Mysql desde panel de control de Xampp.

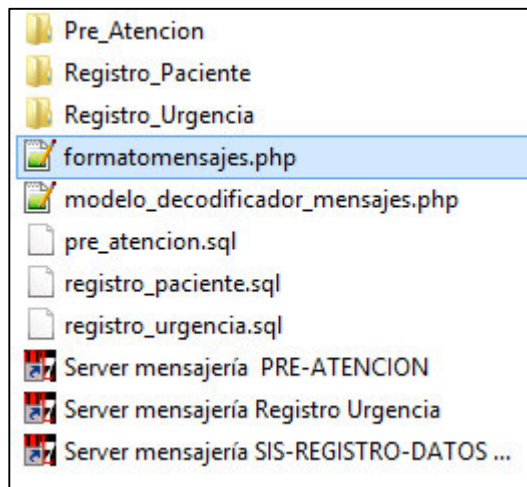
2) En un navegador web escribir:

- Registro paciente: http://localhost/registro_paciente/index.php.
- Pre- Atención: http://localhost/Pre_Atencion/index.php.
- Registro Urgencia: http://localhost/Registro_Urgencia/index.php.

Agregar nuevos mensajes.

Abrir el archivo “Formatomensajes.php” (Ver Figura 37), en él se encuentran escritos todos los segmentos utilizados en los mensajes de Admisión de Urgencias. Para implementar un mensaje en particular, primero se deben identificar los segmentos lo componen según lo especificado en la “Guía de implementación”. Luego, se deben llenar los campos del formato de aquellos segmentos necesarios con la correspondiente data, ya sea sacada directamente desde una base de datos o por algún otro medio.

Figura 37: Archivo
formatomensajes.php.:
Fuente: Elaboración propia.



Finalmente, se deben tomar todos los segmentos y unirlos para formar el mensaje tal como se aprecia en la “Figura 38”.

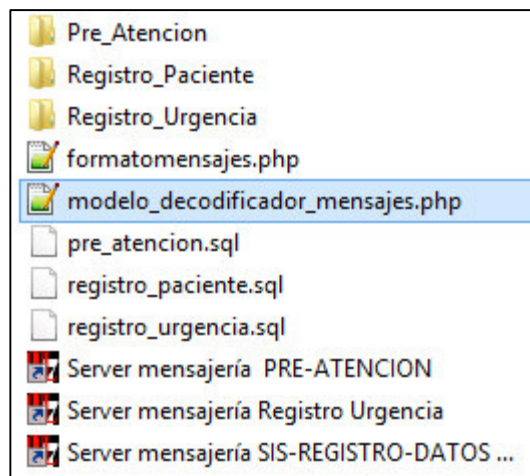
Figura 38: Ejemplo de
formación de mensaje.:
Fuente: Elaboración propia.

```
// Ejemplo $ADT01=$MSH.$EVN.$PID.$PD1.$NK1.$PV1.$ROL.$DG1.$GT1;
```

Lectura de nuevos mensajes.

Para leer nuevos mensajes se debe tomar como base el archivo “modelo decodificador mensajes” que se muestra en la “Figura 39”. En este archivo se encuentra el algoritmo de lectura de la estructura general y la lectura particular definida según el tipo de evento o tipo de mensaje.

Figura 39: Archivo modelo
decodificador.: Fuente:
Elaboración propia.



Finalmente para utilizar los datos, se debe crear un programa de lectura aparte y debe ser llamado desde el segmento de código que se aprecia en la “Figura 40”.

```
case 'A02' :
  if( $EVN_flag==1 and $PID_flag==1 and $FV1_flag==1 and $ROL_flag==1
    and $DG1_flag==1)
  {
    echo "Evento A02 con todos sus segmentos obligatorios\n";
    // /SEGMENTO EVN.....
    if($EVN[1]!='' and $EVN[2]!=''){
      echo "Segmento EVN correcto: Componente obligatorio ok \n";
    }
    else {$flag_error=200; echo "Segmento EVN incorrecto: Faltan Componente obligatorios \n";}
    // /SEGMENTO PID.....
    if($PID[2]!='' and $PID[3]!='' and $PID[5]!=''){
      echo "Segmento PID correcto: Componente obligatorio ok \n";
    }
    else {$flag_error=200; echo "Segmento PID incorrecto: Faltan Componente obligatorios \n";}
    // /SEGMENTO FV1.....
    if($FV1[2]!=''){
      echo "Segmento FV1 correcto: Componente obligatorio ok \n";
    }
    else {$flag_error=200; echo "Segmento FV1 incorrecto: Faltan Componente obligatorios \n";}
    // /SEGMENTO ROL.....
    if($ROL[2]!='' and $ROL[3]!='' and $ROL[4]!=''){
      echo "Segmento ROL correcto: Componente obligatorio ok \n";
    }
    else {$flag_error=200; echo "Segmento ROL incorrecto: Faltan Componente obligatorios \n";}
    // /SEGMENTO DG1.....
    if($DG1[1]!='' and $DG1[2]!='' and $DG1[6]!=''){
      echo "Segmento DG1 correcto: Componente obligatorio ok \n";
    }
    else {$flag_error=200; echo "Segmento DG1 incorrecto: Faltan Componente obligatorios \n";}
    if($flag_error==0) { //insertar datos en la DB
      include ('insertar_A02.php');
    }
  }
  else {echo "Evento A02 con segmentos obligatorios faltantes\n\n"; $flag_error=100;}
break;
```

Figura 40: Ejemplo llamado de programa de lectura particular.: Fuente: Elaboración propia.

Para tener acceso a la lectura de los campos de los segmentos, dentro del archivo se encuentra una sección donde están todos los posibles campos a leer. Si se desea usar alguno, desde el nuevo programa para leer mensajes se debe copiar y pegar para ser utilizados, ver ejemplo de la “Figura 41”

Figura 41: Ejemplo código lectura particular mensaje.:
Fuente: Elaboración propia.

```
//Insertar A02
$PID2=$PID[2]; //
// $PID3=explode('^',$PID[3]);
// $PID5=explode('^',$PID[5]);
// $PID7=$PID[7];
// $PID8=$PID[8];
// $PID11=explode('^',$PID[11]);
// $PID13=$PID[13];
// $PID14=explode('^',$PID[14]);
// $PID16=explode('^',$PID[16]);
// $PID28=explode('^',$PID[28]);
// $PID39=explode('^',$PID[39]);

//$FV12=explode('^',$FV1[2]);
// $FV13=$FV1[3];
// $FV14=$FV1[4];
$FV17=explode('^',$FV1[7]);
$FV110=explode('^',$FV1[10]);
// $FV114=$FV1[14];
// $FV117=$FV1[17];
// $FV136=$FV1[36];
// $FV137=$FV1[37];
// $FV144=$FV1[44];
// $FV145=$FV1[45];

// $GT11=$GT1[1];
// $GT13=$GT1[3];
// $GT110=explode('^',$GT1[10]);
// $GT126=explode('^',$GT1[26]);

//Datos clínicos
$insertSQL = "UPDATE `datos clínicos` SET `id medico asignado`='$FV17[0]', `dx apellido medico asig`='$FV17[1]', `sex apellido medico asig`='$FV17[2]', `nombre medico asig`='$FV17[3]', `apellido medico asig`='$FV17[4]', `id servicio asignado`='$FV110[0]', `valor servicio asignado`='$FV110[1]'
WHERE `id ficha clinica`='$PID2';"
```