



Instituto Politécnico Nacional

Centro de Investigación en Computación

*Aprendizaje de protocolos ASTM para equipos
clínicos*

TESIS

Que para obtener el grado de

Maestro en Ciencias de la Computación

Presenta

Ing. Juan Rivero Mercado

Director de Tesis

Dr. Adolfo Guzmán Arenas



México D.F., Noviembre de 2011

Resumen

Los equipos de laboratorio procesan muestras clínicas y obtienen sus resultados. Cada equipo tiene ciertas características de comunicación, y son definidas en sus respectivos manuales técnicos. Si se desea enviar los resultados a un software de laboratorio clínico es necesario programar su protocolo de comunicación (un conjunto de órdenes correctas para que el equipo efectúe los análisis que se le indican), el protocolo se encuentra bajo cierto estándar llamado ASTM E1394-97. Cabe mencionar que el nombre de “protocolo de comunicación” se usa normalmente para los equipos clínicos, con el significado de “el formato correcto (subconjunto del estándar ASTM E1394-97) que hay que darle a ese equipo para que efectúe los análisis que se le indican”. Cada equipo específico requiere de un “protocolo de comunicación” ad-hoc. La programación de este protocolo se hace actualmente en forma manual, por un programador que tiene que estar familiarizado con el estándar, leer un voluminoso manual y hacer innumerables pruebas (en un equipo que ya está en producción en forma manual). El tiempo que se toma en desarrollar este protocolo (para un equipo que acaba de llegar al laboratorio) puede oscilar entre dos y cuatro semanas.

Muchos factores influyen ampliamente en el tiempo de desarrollo para lograr la comunicación entre un equipo clínico y un software de laboratorio, como pueden ser; falta de manuales técnicos, poca experiencia en el área, tiempo extremadamente corto en el uso del equipo para pruebas (poca disponibilidad del equipo clínico para pruebas), el tipo lenguaje de programación usado, modelo y versión del equipo, etc.

Todos estos problemas se traducen en tiempo, dinero y recursos perdidos para la empresa que se encarga de realizar la comunicación entre el equipo clínico y el software, es decir, que realiza el “protocolo de comunicación”.

El presente trabajo de investigación tiene como objetivo desarrollar un sistema capaz de reducir tiempo y dinero a las empresas dedicadas este rubro. Específicamente, dado un equipo de laboratorio que carece de protocolo, el sistema aquí presentado construirá el protocolo de manera automática, eliminando así la necesidad de su confección manual.

La solución ofrecida en esta investigación es un motor de aprendizaje de protocolos de comunicación, el aprendizaje se realiza a base de las respuestas recibidas por el equipo clínico, adaptando y reconstruyendo el esquema de comunicación. El sistema de aprendizaje tiene una base de datos con rasgos que describen partes de la comunicación ASTM E1394-97, estos rasgos son mutados de acuerdo a la respuesta y la etapa del flujo de trabajo del equipo clínico.

Durante las pruebas realizadas se obtuvieron dos tablas de medidas de éxito; la primera indica el porcentaje de éxito del aprendizaje, la segunda muestra el porcentaje de éxito con respecto al tiempo de desarrollo de una interfaz clínica.

Los resultados indican que el éxito y el tiempo de desarrollo del protocolo de comunicación es por mucho superior al tiempo que le lleva a un desarrollador realizar la misma tarea.

Las aportaciones del trabajo de investigación van desde aplicar una herramienta rápida, un sistema de aprendizaje único para este campo, reducción considerable de tiempo y dinero, hasta en algunos casos la eliminación de la programación manual y del entendimiento y asimilación del manual técnico para el desarrollo.

Palabras clave: ASTM, E1394-97, Interfaz, equipo clínico, protocolo, comunicación, laboratorio clínico, aprendizaje, interfaces, software de laboratorio, manual técnico.

Abstract

The laboratory equipment processed clinical samples and obtain their results. Each equipment has certain communication features, which are defined in their respective technical manuals. If one wants to send the results to clinical laboratory software, it is necessary to program their communication protocol (a set of correct orders that the equipment needs in order to perform the analysis), the protocol is under a standard called ASTM E1394-97. It is worth mentioning that the name of "communication protocol" is commonly used for clinical equipment, meaning "the correct format (subset of ASTM E1394-97) that we should give to that computer to perform the analysis that is required." Each equipment requires a specific, ad-hoc "communication protocol". The programming of this protocol is currently done manually, by a programmer who needs to be familiar with the standard, reads a lengthy manual and makes countless tests (on a clinical equipment that is already in production on a manual setup). The time taken to develop this protocol (for an equipment that just arrived at the laboratory) can range from two to four weeks.

Many factors contribute greatly to the development time to achieve communication between a clinical equipment and the laboratory software, such as missing ~~of~~ technical manuals, little experience in the area, extremely short time available for the use of the clinical laboratory for, the type of programming language used, the model, and version of the equipment, etc.

All these problems result in time, money and losses for the company that undertakes the communication between the clinical equipment and the computer software.

The present research aims to develop a system capable of reducing time and money to companies active in this area. Specifically, lacking a protocol for a given laboratory equipment, the system presented here will build the protocol automatically, eliminating the need for manual preparation.

The solution offered in this research is a learning engine that produces communication protocols; learning takes place based on the responses received by the clinic team, adapting and rebuilding the communication scheme. The learning system has a database of features that describe parts of the communication ASTM E1394-97, the individual parts are mutated according to the response and the stage of the workflow of the clinical equipment.

During the tests we obtained two tables of measures of success: the first indicates the percentage of successful learning, the second shows the percentage of success with respect to speed.

The results indicate that the success and development time of the communication protocol is far greater than the time it takes for a developer to perform the same task.

The contributions of research ranging from implementing a fast, unique learning system for this field, considerable reduction of time and money, even

in some cases eliminating manual programming and understanding and assimilation of the technical manual for the development.

Keyowrds: ASTM, E1394-97, Interface, clinical team, protocol, communication, clinical laboratory, learning, interfaces, laboratory software, technical manual.

Agradecimientos

Le doy las gracias a mis padres que cada uno con su forma especial y única me ha formado en todos estos años, mi madre que con su amor y apoyo me ha ayudado a salir adelante en la adversidad, a mi padre que con su ejemplo de trabajo duro y éxito me ha enseñado que debo pelear siempre por más, le agradezco infinitamente a mi hermano, sin su ayuda de ninguna manera estuviera vivo y cosechando mis éxitos que sin lugar a dudas son de él. Gracias a toda mi familia que ha estado conmigo siempre, todos y cada uno de ellos enseñándome algo importante de la vida.

Y también un agradecimiento a mis amigos y amigas, siempre me han apoyado cuando he perdido el camino, es incontable nombrarlos a todos pero saben que pienso en ustedes cuando consigo superar mis obstáculos.

Agradezco al IPN y CONACYT, por el apoyo económico, mediante el cual ha sido posible la realización de mis estudios de posgrado.

También extiendo mi agradecimiento a todos mis compañeros de estudio y trabajo por su apoyo y los gratos momentos de convivencia.

Finalmente agradezco al Centro de Investigación en Computación (CIC), por los recursos, espacio y años de formación que me ha brindado para lograr mis estudios de posgrado.

Glosario

- **Algoritmo:** Conjunto de reglas bien definidas para la solución de un problema en un número finito de pasos.
- **Análisis clínico:** Forma parte del proceso de atención a la salud que se apoya en el estudio de distintas muestras biológicas mediante su análisis en laboratorio y que brinda un resultado objetivo que puede ser tanto cuantitativo (un número, como en el caso de la cifra de glucosa) o cualitativo (positivo o negativo).
- **ANSI:** *American National Standards Institute.*
- **ASCII:** *American Standard Code for Information Interchange.*
- **ASTM:** *American Section of the International Association for Testing Materials.*
- **Base de datos:** Conjunto de datos organizados que siguen criterios muy sofisticados y hacen referencia al mismo tema. Sistema de almacén de información valiosa con características similares.
- **Cable RS-232:** Cable para el envío de datos, de una interfaz que designa una norma para el intercambio de una serie de datos binarios.
- **Checksum:** Medida muy simple para proteger la integridad de los datos, verificando que no haya datos corruptos.
- **Equipo clínico:** Aparato automático para el análisis y evaluación de muestras clínicas.
- **Estructura jerárquica:** Conjunto de sentencias o estructuras bien definidas ordenadas por rango de importancia.
- **Expresión regular:** También llamado patrón es una expresión que describe un conjunto de cadenas sin enumerar sus elementos.
- **HL7 Health Level Seven:** Conjunto de estándares para el intercambio electrónico de información clínica.
- **Interfaz:** Conexión e interacción entre hardware, software y usuario para el intercambio de información.
- **Laboratorio de análisis clínico:** Lugar donde se realiza análisis clínicos que contribuyen al estudios, prevención, diagnóstico y tratamiento de los problemas de salud de los pacientes.

- **Léxico:** Relativo al conjunto de vocablos perteneciente a un grupo específico de hablantes o que comprenden determinado campo semántico.
- **LIMS Laboratory Information Management System:** Software para la administración de un laboratorio de análisis clínico. Permite recoger, almacenar, administrar, calcular y gestionar datos.
- **Muestra clínica:** Este proceso conocido como toma de muestras, abarca la flebotomía, proceso por el cual se extrae una muestra de sangre; la obtención de otro tipo de muestras, como orina y heces; y la extracción de otros líquidos corporales, como líquido cefalorraquídeo o líquido articular.
- **Paquete:** Es un segmento de información que contiene propiedades que son necesarias en la comunicación total, está conformado por secuencias de caracteres especiales.
- **Periférico:** Se denomina periférico tanto a las unidades o dispositivos a través de los cuales la computadora se comunica con el mundo exterior, como los sistemas que almacena o archivan información, sirviendo de memoria auxiliar de la memoria principal.
- **Protocolo:** es la relación que se reconoce en la comunicación o la transferencia de información. En informática, un protocolo es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red.
- **Rasgo:** Propiedad o peculiaridad distintiva de un elemento.
- **Semántica:** Se refiere a los aspectos del significado, sentido o interpretación del significado de un determinado elemento.
- **Simulador de equipos clínicos:** Software que permite la reproducción de la forma de trabajo de un equipo clínico.
- **Sintaxis:** Estudio de la estructura del lenguaje y conjunto de reglas combinatorias de las palabras para formar unidades inteligibles mayores.
- **Sistema de aprendizaje:** Es una rama de la inteligencia Artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender o realizar una clasificación determinada.
- **XML Extensible Markup Language:** Metalenguaje extensible de etiquetas desarrollador por el *World Wide Web Consortium*.

Índice

Glosario	IX
CAPÍTULO 1. INTRODUCCIÓN.....	1
1.1 Objetivo general.....	3
1.2 Objetivos específicos	3
1.3 Problemática	3
1.4 Justificación	4
1.5 Aportaciones	5
1.6 Delimitación del trabajo de investigación.....	5
1.6.1 Lo que abarca el trabajo de investigación	5
1.6.2 Lo que no abarca el trabajo de investigación.....	5
1.7 Descripción general del proyecto	7
1.7.1 Base de datos	7
1.7.2 Seleccionador de rasgos.....	8
1.7.3 Protocolo Alfa	8
1.7.4 Sistema de reglas.....	8
1.7.5 Analizador léxico y sintáctico	8
1.7.6 Equipo/Simulador	8
1.7.7 Traductor ASTM-XML y XML-ASTM	9
1.7.8 Clasificador	9
1.8 Estructura de la tesis	9
CAPÍTULO 2. ESTADO DEL ARTE	11
2.1 Simuladores ASTM E1394-97	11
2.1.1 <i>Simulador - ASTM HS</i>	12
2.1.2 <i>Simulador - ASTM Connector</i>	13
2.2 Especialistas en el desarrollo de interfaces clínicas.....	14
2.3 HL7.....	15
2.4 Software para el manejo del estándar HL7	17
2.4.1 7Edit.....	17
2.4.2 Iguana HL7	18
2.4.3 HL7Spy.....	18
2.5 Reflexiones.....	19
CAPÍTULO 3. MARCO TEÓRICO.....	20
3.1 Analizador léxico	20
3.2 Analizador sintáctico.....	21
3.3 Lenguajes formales	22
3.4 Expresiones regulares.....	25
3.5 Forma Backus-Naur y Backus-Naur extendida	27
3.6 Extensible Markup Language (XML)	27
3.7 Especificaciones del estándar para la transferencia de información entre instrumentos clínicos y sistemas de cómputo	29
3.7.1 Definiciones de términos específicos para el estándar ASTM.....	29
3.7.2 Significado y uso	30
3.7.3 Requisitos de información en los ensayos clínicos	31
3.7.4 Contenido del mensaje	34
3.8 Estructura de la comunicación.....	57
3.9 Estructura de un paquete.....	58
3.10 Descripción de términos específicos para ASTM	61
3.10.1 Capa de aplicación	61
3.10.2 Capa de presentación	67
3.10.3 Capa de enlace de datos	68
3.10.4 Capa física.....	78
CAPÍTULO 4. CONSTRUCCIÓN DEL PROYECTO	79
4.1 Método propuesto	79
4.1.1 Pseudocódigo.....	80
4.2 Modelo de desarrollo de software	80

4.2.1	Modelo lineal secuencial.....	80
4.3	Desarrollo y especificaciones	81
4.4	Base de datos.....	81
4.4.1	Estructura jerárquica	82
4.4.2	Organización del metalenguaje.....	83
4.5	Seleccionador de rasgos	83
4.5.1	Etapa de conglomerado.....	84
4.5.2	Etapa de selección	85
4.5.3	Etapa de generación de objeto	86
4.6	Traductor XML-ASTM.....	86
4.6.1	Objeto: Protocolo Alfa	87
4.6.2	Objeto: Estructura XML.....	87
4.7	Sistema de reglas	87
4.7.1	Sistema de mutación.....	87
4.7.2	Analizador	88
4.8	Sistema de aprendizaje.....	88
4.8.1	Motor de comunicación.....	89
4.8.2	Motor de aprendizaje	89
4.8.3	Aprendizaje.....	90
4.9	Software	93
4.9.1	Bloque de Datos	93
4.9.2	Selección de reglas	93
4.9.3	Selección de rasgos.....	94
4.9.4	Protocolo Alfa	94
4.9.5	Análisis léxico y traductor.....	95
4.9.6	Terminal.....	95
	CAPÍTULO 5. PRUEBAS Y RESULTADOS.....	96
5.1	Metodología experimental	96
5.1.1	Entorno simulador	97
5.1.2	Entorno de laboratorio de análisis clínico	99
5.2	Resultados.....	99
5.2.1	Interpretación de los resultados	99
5.2.2	Resultados del entorno de simulación.....	101
5.2.3	Resultados del entorno de laboratorio.....	103
5.2.4	Medidas de éxito.....	105
5.3	Conclusiones sobre los resultados.....	106
5.3.1	Objetivos específicos y resultados.....	107
	CAPÍTULO 6. CONCLUSIONES Y TRABAJO A FUTURO	108
6.1	Sobre el sistema de aprendizaje	108
6.2	Sobre el estándar ASTM y sus reglas	109
6.3	Limitaciones	109
6.4	Trabajo a futuro	109
6.4.1	Limitaciones	109
	REFERENCIAS	111
	ANEXO A. EJEMPLO DE CÓDIGO FUENTE	113
	ANEXO B. ORGANIZACIÓN DEL ESTÁNDAR ASTM E1394-97	116

Índice de ilustraciones

Ilustración 0-A Flujo de trabajo de un laboratorio clínico	2
Ilustración 0-B Delimitación del proyecto	6
Ilustración 2-A Interfaz gráfica simulador ASTM Host Simulator	12
Ilustración 2-B Interfaz gráfica simulador ASTM Connector	13
Ilustración 2-C Software 7Edit	17
Ilustración 2-D Software Iguala HL7	18
Ilustración 2-E Software HL7Spy	18
Ilustración 2-F Software HL7Spy	19
Ilustración 3-A Árbol gramatical	21
Ilustración 3-C Ejemplo de estructura XML	28
Ilustración 4-A Modelo lineal secuencial	81
Ilustración 4-B Sistema de archivos	82
Ilustración 4-C Árbol de derivaciones	85
Ilustración 4-E Software - Base de datos	93
Ilustración 4-F Software - Selección de reglas	93
Ilustración 4-G Software - Selección de rasgos	94
Ilustración 4-H Software - Protocolo Alfa	94
Ilustración 4-I Software - Análisis	95
Ilustración 4-J Software - Terminal	95
Ilustración 5-A Lista de equipos de la herramienta simulador	98
Ilustración 5-B Resultados simulación Elecsys 2010	101
Ilustración 5-C Resultados simulación Urisys	102
Ilustración 5-D Resultados Elecsys 2010	103
Ilustración 5-E Resultados Urisys	104
Ilustración 5-F Porcentaje de éxito	105
Ilustración 5-G Medida de tiempo	106

Índice de Tablas

Tabla 3-A Estructura lógica para el almacenamiento de información.....	33
Tabla 3-B Estructura lógica de un mensaje	33
Tabla 3-C Tabla de requerimientos de retransmisión	34
Tabla 3-D Delimitadores de escape.....	36
Tabla 4-A Esquema XML	83
Tabla 4-B Ejemplo de rasgos.....	83
Tabla 4-C Ejemplo protocolo A	84
Tabla 4-D Ejemplo protocolo B.....	84
Tabla 4-E Estructura generalizada.....	84
Tabla 4-F Estructura conglomerada	85
Tabla 4-G Estructura organizacional XML	86
Tabla 4-H Traducción XML-ASTM	86

CAPÍTULO 1.

INTRODUCCIÓN

Refiriéndonos al sector salud y en especial a un laboratorio de análisis clínico, cada laboratorio cuenta con aparatos electrónicos modernos que le ayudan al proceso de análisis clínicos, estos sistemas electrónicos son llamados instrumentos clínicos o equipos clínicos, su función es procesar muestras clínicas (sangre, orina, heces, saliva, etc.) y obtener las mediciones de dichas muestras, que son utilizadas para generar un diagnóstico médico.

El flujo general del trabajo de un laboratorio se muestra en la Ilustración 0-A.

Ahora bien, un laboratorio de análisis clínico también cuenta con un programa de cómputo de administración, el cual es utilizado para organizar citas; programar pruebas de análisis clínicos, imprimir resultados, generar estadísticas, etc. El software se encuentra instalado y dirigido desde un equipo de cómputo común y corriente, es por esto que la conexión para el envío y recepción de datos se realiza de manera física entre el equipo clínico y el sistema de cómputo. Esta conexión, además de ser física, requiere de un software que le transmita al equipo clínico las instrucciones correctas (en un formato definido) para que lleve a cabo las pruebas que se le solicitan. A este software se le llama en el ambiente de laboratorios clínicos “protocolo de comunicación”, siendo que en realidad es un conjunto de instrucciones de pruebas clínicas a llevar a cabo.

Ya que los equipos clínicos y los equipos de cómputo no utilizan el mismo lenguaje para el intercambio de información, la *ASTM American Society for Testing and Materials* por sus siglas en inglés, definió la norma E1394-97[1] para dicha tarea.

Por lo tanto, aquel laboratorio que desee compartir información entre el equipo clínico y el software de administración (implementado en sistema de cómputo), requiere contratar los servicios de un especialista que programe un software de intercambio de datos. Este programa informático es llamado interfaz de laboratorio de análisis clínico y se encarga de recibir, enviar, decodificar e interpretar la información recibida. El manual técnico del equipo clínico proporciona las reglas y demás detalles de comunicación, a las que el programador se tiene que apegar forzosamente para desarrollar la interfaz. Esta interfaz es lo que en esta tesis se llama “protocolo de comunicación” entre el equipo y el software particular del laboratorio, y es el objetivo de esta tesis producirlo de forma automática (en vez de que un programador lo produzca).

Cada empresa que se dedica a desarrollar interfaces de laboratorio clínico tiene predilección por un tipo lenguaje de programación, por lo tanto, las interfaces pueden ser desarrolladas en *Java*, *C#*, *Visual Basic*, *C*, *Ruby* entre muchos otros lenguajes de programación. Por tal motivo el fabricante no proporciona ningún controlador de dispositivo para el envío de datos. El desarrollo de interfaces en la mayoría de los casos es excesivo y poco eficiente.

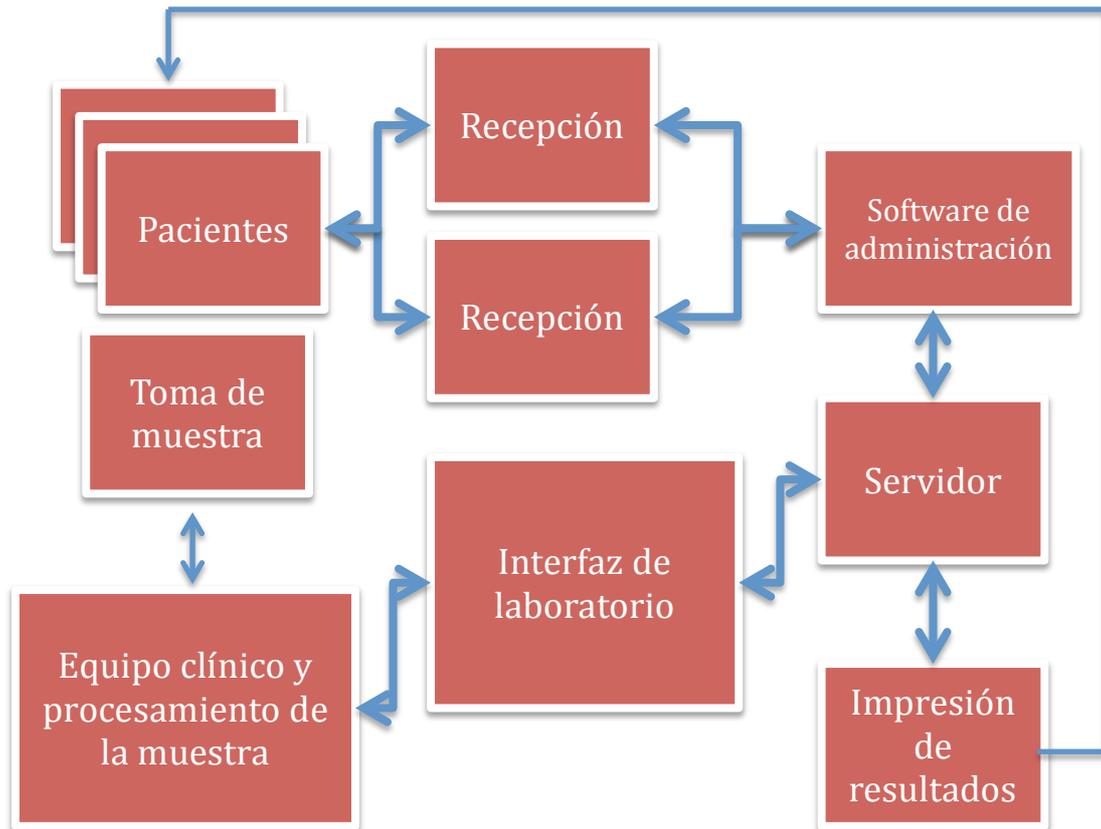


Ilustración 0-A Flujo de trabajo de un laboratorio clínico

1.1 Objetivo general

Aportar una herramienta capaz de generar una comunicación exitosa y valida de forma automática, para el intercambio de información entre el equipo clínico y el sistema de cómputo, con mínima intervención del usuario.

1.2 Objetivos específicos

- Proponer una herramienta única de software que reduzca el tiempo de investigación y desarrollo de una interfaz de laboratorio de análisis clínico.
- Generar un motor de aprendizaje de protocolos de comunicación apegado al estándar ASTM E1394-97 para el intercambio de datos de manera automática.
- Crear una comunicación válida exitosa, independientemente de la versión, modelo, marca y/o fabricante de equipos clínicos.
- Eliminar en la medida de lo posible la necesidad de manuales técnicos para desarrollar la comunicación de los equipos clínicos.
- Servir como plataforma para impulsar el desarrollo de nuevas investigaciones en el campo de los estándares de comunicación clínica.

1.3 Problemática

El desarrollo de interfaces para el intercambio de datos entre el instrumento clínico y el sistema de cómputo, puede llevar un tiempo excesivamente largo en ser completada, esto se debe a los siguientes factores:

- Cada uno de los fabricantes de instrumentos clínicos implementa su propia versión del protocolo de comunicación ASTM utilizando la norma E1394-97, por esa razón, cada equipo fabricado tiene un cierto nivel de afinidad al estándar, aunque no en todos los casos es 100% compatible.
- El manual técnico se encuentra inaccesible o en algunos casos tiene un costo excesivo.
- La información del manual es por lo general excesiva, demasiado técnica y en algunas ocasiones compleja de comprender.
- La versión del protocolo de comunicación que el fabricante haya implementado, podrá cambiar con respecto al tiempo, versión, marca y/o modelo del equipo clínico.

- El protocolo de comunicación del equipo puede ser rápidamente discontinuado por el fabricante, y dando por terminado su ciclo de vida sin dar más material o soporte técnico.
- Poca o nula experiencia por parte del programador.

Aunado a todo estos problemas, existe una gran pérdida de tiempo en realizar las pruebas pertinentes en el desarrollo de la interfaz clínica, ya que es necesario que el desarrollador esté físicamente en el laboratorio y usando el equipo clínico, por lo tanto, utiliza tiempo crítico del equipo clínico que es necesario para pruebas urgentes.

1.4 Justificación

El valor teórico proporcionado es implementar el primer modelo de aprendizaje de protocolos de comunicación ASTM bajo la norma E1394-97, por lo que no existe en la actualidad ningún modelo, técnica o herramienta para el desarrollo de interfaces clínicas de forma automática, como el presente trabajo.

El aspecto práctico de este trabajo de tesis, es proporcionar la primera herramienta de apoyo que se aplica directamente al desarrollo de interfaces clínicas, por lo tanto, soluciona las problemáticas mencionadas en 1.3, y por consiguiente aumenta la velocidad y el rendimiento del desarrollo, así como también, reduce las erogaciones de las empresas. Además, la misma técnica puede ser aplicada a diferentes formatos de comunicación e incluso a diferentes normas que aparezcan en un futuro.

Desde punto de vista metodológico, el programa informático proporcionado por el resultado de esta investigación será útil a aquellas investigaciones y trabajos que versen sobre crear un método o técnica para el rápido y eficiente desarrollo de interfaces clínicas, cuya utilidad puede expandirse en diferentes protocolos de comunicación o modelos de comunicación.

La conclusión del trabajo de investigación nos expone la utilidad de esta herramienta y el beneficio a corto plazo en la reducción de tiempo de desarrollo independiente de la marca, modelo, versión y/o fabricante del equipo clínico.

Esta herramienta es altamente requerida en líneas de producción especializadas en el desarrollo de interfaces, en la mayoría de licitaciones manejadas por los laboratorios de análisis clínicos, se requiere un tiempo de desarrollo efectivo y rápido para evitar el entorpecimiento de la tarea vital del laboratorio.

1.5 Aportaciones

Las aportaciones del presente trabajo se describen de manera concisa a continuación:

- Minimizar el tiempo de desarrollo y pérdida de recursos económicos para las empresas enfocadas en el rubro.
- Servir de plataforma para la creación de nuevas herramientas para el desarrollo de interfaces de laboratorio con el modelo aquí propuesto.
- Reducir el tiempo de uso de un equipo clínico para el desarrollo de una interfaz clínica.

1.6 Delimitación del trabajo de investigación

La Ilustración 0-B describe el flujo general de un laboratorio clínico, los bloques con un estilo diferente al grupo general son las áreas donde se concentrara el presente trabajo de investigación.

Por otro lado, la delimitación del trabajo de tesis se divide en dos partes principales; en la primera se describe lo que abarca la investigación, en la segunda parte se define claramente lo que no cubre.

1.6.1 Lo que abarca el trabajo de investigación

- Trabajar y alterar la transmisión de datos entre el equipo clínico y un sistema de cómputo.
- Crear un software con interfaz gráfica sencilla, con la única finalidad de realizar las pruebas pertinentes y ejecutar el sistema de aprendizaje para lograr una comunicación exitosa.

1.6.2 Lo que no abarca el trabajo de investigación

- Modificar directa o indirectamente la información que contenga el equipo clínico.
- Entregar una lista de instrucciones que el equipo clínico maneja e identificar la versión del protocolo de comunicación.
- Proporcionar inmediatamente una interfaz gráfica al desarrollador de interfaces para el uso y administración del sistema en las líneas de producción.
- Realizar de forma total o parcial el trabajo del equipo clínico.
- Interferir y realizar el trabajo del software de administración de un laboratorio de análisis clínico, como puede ser; generar citas al paciente, administrar la información de pacientes y pruebas de laboratorio, reporte de resultados, estadísticas, etc.

- Alterar de cualquier forma la información que se encuentra contenida en la base de datos del laboratorio de análisis clínico.
- Almacenar la información de pacientes y resultados obtenidos de la comunicación.
- Perturbar el sistema operativo del instrumento clínico, así como, cambiar de cualquier forma la estructura interna de la comunicación del equipo clínico.
- Interferir con el flujo de trabajo de un laboratorio de análisis clínico.

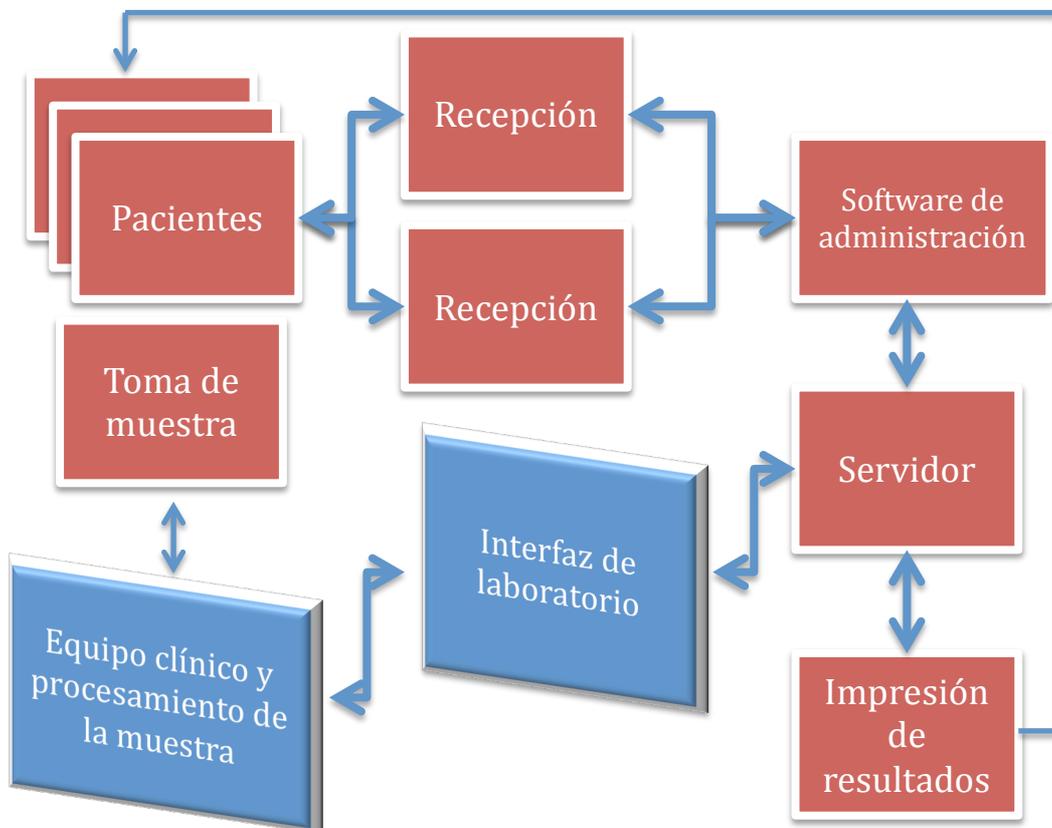


Ilustración 0-B Delimitación del proyecto

La línea entre los dos bloques azules muestra el protocolo de comunicación que aquí se desarrollará. Los cuadros rojos no se involucran en este desarrollo.

1.7 Descripción general del proyecto

El actual trabajo de investigación busca crear un sistema de aprendizaje de protocolos de comunicación del estándar ASTM E1394-97 de forma automática, para el intercambio de información entre el equipo clínico y el sistema de cómputo con mínima intervención del usuario.

El proyecto se ilustra de manera general en la Ilustración 0-D, la descripción de cada uno de los bloques del proyecto viene a continuación.

1.7.1 Base de datos

Uno de los puntos centrales del sistema de aprendizaje es la base de datos, en ella se almacenan y describen las versiones de los protocolos, así como también sus rasgos más descriptivos. El Universo completo del estándar ASTM E1394-97 no maneja orden de la transmisión, modificaciones de comunicación, rasgos y en general las características complejas de la base de datos. Si el aprendizaje ha fallado, el sistema recurrirá al estándar para utilizar aquellos rasgos generales para tratar de entablar la comunicación exitosamente. Por lo tanto, es necesario tener un sistema de rasgos característicos de los protocolos (base de datos) y el conjunto universo del estándar ASTM como sistema de respaldo.

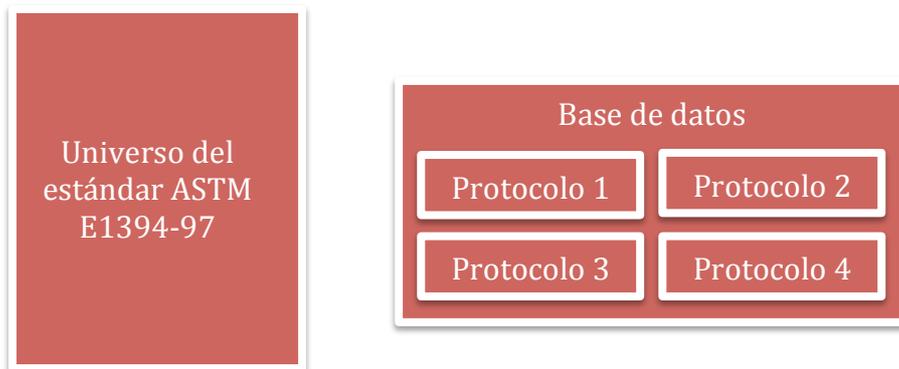


Ilustración 0-C Comparación Universo ASTM y base de datos

El bloque definido como: “Universo del estándar ASTM E1394-97”, contiene de manera completa toda la estructura de comunicación definida en **[1]** y organizada en una estructura jerárquica, véase Anexo A.

El bloque definido como: “Base de datos”, contiene todos aquellos protocolos de comunicación que han sido aprendidos exitosamente y sirven para el intercambio de información, cada uno de ellos cuenta con rasgos, organización de transferencia y modificaciones variadas, aunque algunos de esos protocolos contenidos en la base de datos comparten características similares, no las comparten del todo con el primer bloque definido y demás protocolos de la base de datos.

Los dos bloques vistos anteriormente, están estructurados internamente con la organización jerárquica XML *Extensible Markup Language* por sus siglas en inglés[2] refiérase a la sección 3.6 del marco teórico para mayor información.

1.7.2 Seleccionador de rasgos

En particular, el bloque de selección de rasgos se encarga de seleccionar los rasgos más importantes y representativos que no han sido utilizados en el protocolo Alfa, véase Protocolo Alfa. Además se encargará de realizar las modificaciones pertinentes dentro de la búsqueda dirigida.

1.7.3 Protocolo Alfa

El protocolo Alfa es el bloque encargado de armar un protocolo válido a base del estándar ASTM E1394-97. Este protocolo aún se encuentra en fase de desarrollo y todavía no es útil para realizar el aprendizaje con los equipos clínicos. Por lo tanto, el protocolo tiene que pasar para algunos pasos más para ser factible en las pruebas. El software gradualmente transforma este protocolo Alfa en uno final, que funciona correctamente para el equipo clínico en cuestión.

1.7.4 Sistema de reglas

Este bloque cuenta con dos funciones principales, la primera se encarga de analizar de manera semántica el protocolo Alfa generado por el bloque anterior. Por otra parte, cuando las pruebas de aprendizaje hayan finalizado sin éxito, el presente bloque podrá tomar aquellos rasgos (segmentos de comunicación) del conjunto universo ASTM para realizar un último intento de comunicación exitosa. Este bloque se comunica de manera indirecta con el traductor y generador de protocolo Alfa.

1.7.5 Analizador léxico y sintáctico

Un bloque encargado de analizar de manera sintáctica y léxica el protocolo que será transmitido al equipo para verificar que se encuentra apegado a las reglas del estándar base. La salida generada y analizada por el bloque es nombrado protocolo nativo y se encuentra listo para ser probado. Por otro lado, el presente bloque tendrá la función de enviar el protocolo nativo al equipo clínico o simulador para realizar las respectivas pruebas.

1.7.6 Equipo/Simulador

Es la parte principal del proyecto, pues es el encargado de verificar que la comunicación se realiza de manera correcta y entendible para el equipo. Este mismo bloque se encargará de publicar el error en alguna de las fases de enlace y transmisión.

1.7.7 Traductor ASTM-XML y XML-ASTM

El traductor tiene la única tarea de convertir una estructura organizacional a otra, esto es, hará una traducción literaria del formato XML a la estructura ASTM y de manera inversa si así se requiere.

1.7.8 Clasificador

El bloque clasificador es el encargado de clasificar y almacenar el protocolo que fue aprendido de manera exitosa, la clasificación es a base de sus rasgos principales para su posterior utilización.

1.8 Estructura de la tesis

Este documento se encuentra estructurado en seis capítulos. El primero versa sobre el contexto general del trabajo que abarca la presente introducción, en la cual se presenta el origen de esta tesis, el objetivo general, objetivos específicos, problemática, justificación del trabajo, delimitación y finalmente, el proyecto general del aprendizaje de protocolos del cual forma parte.

En el segundo capítulo se expone el estado del arte de la presente investigación, en donde se da un repaso de las herramientas de software que tienen relación con un laboratorio clínico pero no solventan la lentitud del desarrollo de una interfaz.

El marco teórico correspondiente al capítulo tres, exponer la metodología utilizada para este trabajo de investigación.

En el cuarto capítulo se expone a detalle la construcción de la presente tesis.

El capítulo quinto muestra las pruebas realizadas con el sistema de aprendizaje, así como, alcances y resultados obtenidos.

Finalmente, en el capítulo seis se reflexiona sobre las conclusiones, de trabajos futuros sobre el mismo tema.

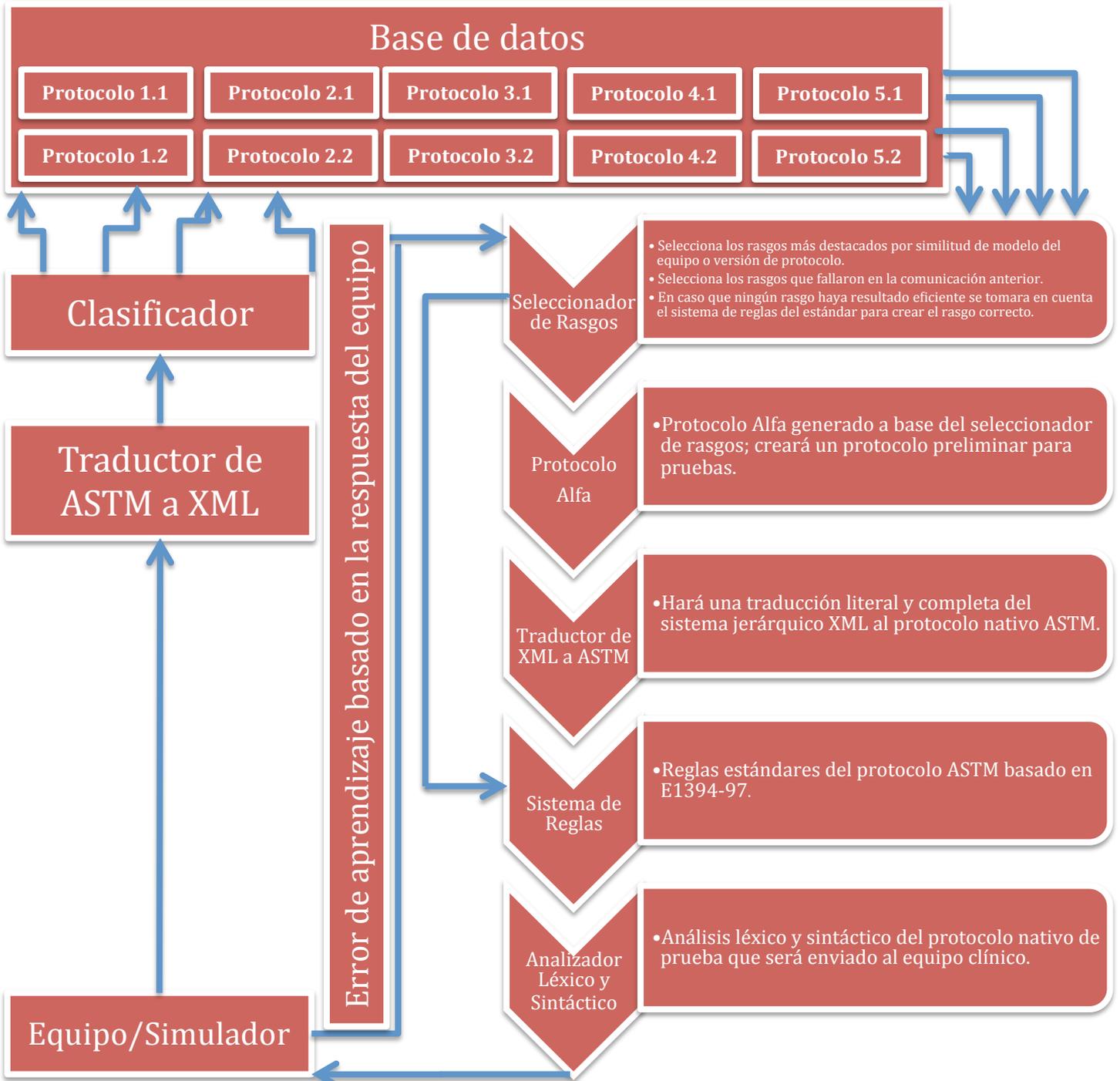


Ilustración 0-D Arquitectura del proyecto de tesis

CAPÍTULO 2.

ESTADO DEL

ARTE

Actualmente, no existe ninguna herramienta, técnica o modelo capaz de realizar (construir) de manera automática y exitosa una comunicación, que sirva para el intercambio de información entre el equipo clínico y el sistema de cómputo. Es decir, no existe (hasta antes del presente trabajo) una herramienta que genere automáticamente los protocolos de comunicación. Se usan programadores y técnicos informáticos para este fin.

Sin embargo, existe un tipo de software especial que es útil a todos los desarrolladores de interfaces clínicas, es una herramienta llamada simulador y como su nombre lo indica, es capaz simular de manera muy detallada la forma de comunicación de un equipo clínico real.

Los simuladores son provechosos para realizar todo tipo de pruebas de comunicación de un equipo clínico específico. Sin embargo tiene un lado bastante inconveniente y problemático, son herramientas difíciles de conseguir y sobre todo no cuentan con una amplia gama de equipos clínicos para ser simulados. Puesto que, pocos fabricantes de equipos clínicos desarrollan simuladores y aquellos que los desarrollan trabajan únicamente con sus propios modelos.

2.1 Simuladores ASTM E1394-97

En el mercado actual existen muy pocas herramientas de simulación, muchas de ellas son propietarias y de uso exclusivo de los fabricantes, sin embargo, se presentan dos herramientas que son de fácil acceso y autorizadas para ser usadas por el desarrollador habitual.

Algunas de las principales características de las herramientas de simulación son:

- Comunicación a bajo nivel.
- Rastro de registros.
- Editor de datos.
- Simulación de errores.
- Edición y modificación de registros.

2.1.1 Simulador - ASTM HS

ASTM Host Simulator Ilustración 0-A es el sistema de simulación por elección para equipos de la marca Hitachi, fue desarrollado por la empresa *Roche Diagnostics*[3] con el fin de proporcionar a los desarrolladores una herramienta que sirva de pruebas de comunicación para sus equipos clínicos. La última versión disponible es la 6.3.

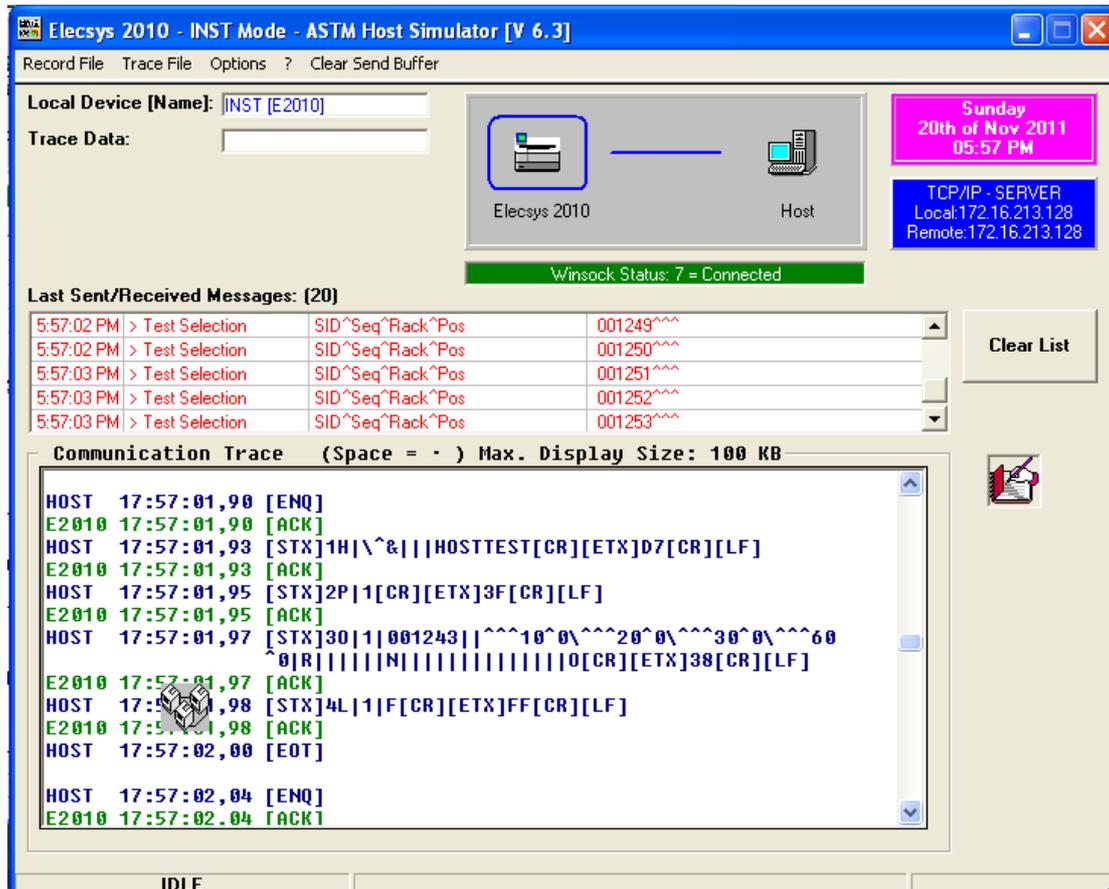


Ilustración 2-A Interfaz gráfica simulador ASTM Host Simulator

Sus principales ventajas son:

- Amplia gama de equipos a simular.
- Sistema sencillo y amigable.
- Fácil seguimiento de errores.
- Gran variedad de características y configuraciones.

Sus principales desventajas son:

- Poca estabilidad.
- Lenguaje de programación obsoleto.
- No existen nuevas versiones.

2.1.2 Simulador - ASTM Connector

Esta herramienta fue desarrollada por *Henodics Management AG*[3] Ilustración 2-B Al igual que las demás herramientas de simulación solo se enfoca al estándar ASTM.

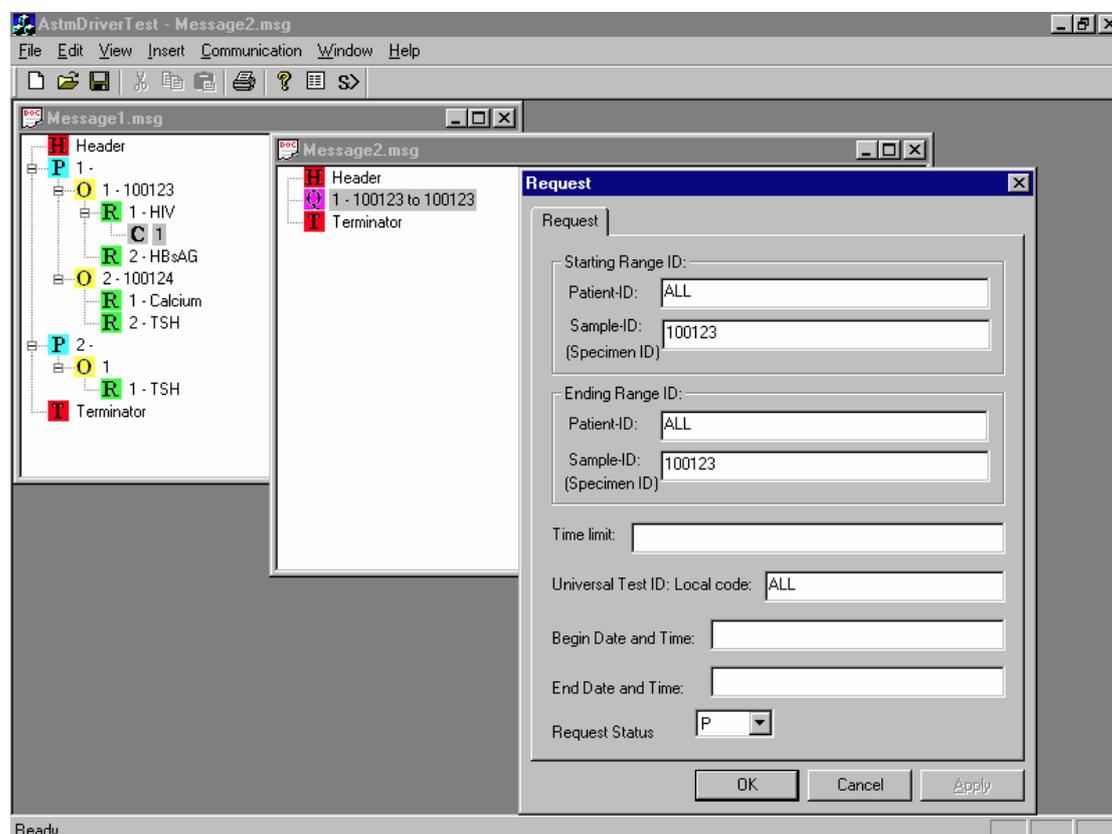


Ilustración 2-B Interfaz gráfica simulador ASTM Connector

Sus principales ventajas son:

- Simple e intuitiva.
- Verificación de datos corruptos.
- Configuración para diferentes tipos de puertos de comunicación.

Sus principales desventajas son:

- Pocas características de configuración.
- No cuenta con manejo de errores.
- Versión obsoleta y descontinuada.

Son las únicas herramientas disponibles al público en general al momento de escribir la presente tesis de investigación.

2.2 Especialistas en el desarrollo de interfaces clínicas

A nivel nacional existen diferentes empresas dedicadas al desarrollo de interfaces clínicas, algunas de ellas son:

- Sofilab

Empresa dedicada a la venta de reactivo clínico para laboratorios en general, con amplia experiencia en el manejo de laboratorio de análisis clínico, generando software para laboratorio e interfaces para equipos clínicos.

- Sigelab

Empresa dedicada a la venta de software para laboratorio de análisis clínico y venta de reactivo clínico.

Adicionalmente Sigelab soporta el desarrollo de interfaces de comunicación con los equipos analizadores.

- Cubasoft

Cubasoft es una empresa ecuatoriana dedicada al desarrollo de software de gestión y negocios verticales, software especializado desarrollo a medida.

Cubasoft es una empresa que ha estado relacionada totalmente con el área de salud, estando dispuesta a dar asesoría y ayudar a la solución de la gestión de salud.

- Interlab

Dedicado a soluciones empresariales, soluciones modernas y acordes a la tendencia de la consulta de resultados por internet, dedicada especialmente a la creación de interfaces de laboratorio, cuentan con más de 76 interfaces desarrolladas de diferentes marcas y equipos.

- StarLims

StarLims es una herramienta configurable para la gestión de complejos flujos de trabajo; mantiene el cumplimiento de las normas regulatorias; impone las funciones de acuerdo con las normas establecidas; ofrece un rápido formato y entrega de resultados clínicos de acuerdo con el método de preferencia del cliente; almacena, rastrea y distribuye inventarios de especímenes biológicos; facilita la producción de kits y pruebas de distribución, reportes y la gestión de consultas para especímenes de ensayos clínicos

- eLabo

El sistema eLabo es un sistema de Información para laboratorios clínicos que fue diseñado para facilitar la labor de los operadores, la administración

eficiente y el reporte de los resultados de las pruebas realizadas en un laboratorio de análisis clínico moderno. eLabo tiene la capacidad de enlazar sus propios productos del mercado con interfaces clínicas de otras compañías.

Las empresas tienen un método en común para el desarrollo de interfaces, la cual puede ser catalogado como “esperar, desarrollar y probar” ya que, cuando un nuevo cliente solicita el desarrollo de un nuevo protocolo en particular, es necesario que el encargado de desarrollar el proyecto espere las especificaciones del cliente, además, debe aguardar para recolectar las especificaciones técnicas del protocolo específico y solo si esto es posible, se procede a la fase desarrollo. Todo esto requiere considerable tiempo que dependerá de la experiencia del propio desarrollador, posteriormente deberá poner en marcha la fase de pruebas contando con un tiempo limitado, que es directamente proporcional al tiempo que el equipo clínico se encuentre sin carga de trabajo. (Como estos equipos son costosos se utilizan aunque sea en forma manual para análisis de muestras, mientras se desarrolla el protocolo de comunicación).

Existen algunos intentos para simplificar el desarrollo de interfaces clínicas para equipos en particular. En la sección 2.3 se definen un intento de facilitar todo lo anteriormente mencionado.

2.3 HL7

HL7 *Health Level Seven* por sus siglas en inglés, es un conjunto de estándares para el intercambio electrónico de información clínica de diferentes formas y medios de comunicación.

Los estándares HL7 son desarrollados por la organización *HL7 International*[5].

Existe una idea errónea acerca de la organización HL7, consistente en que desarrolla software. En realidad, HL7 desarrolla especificaciones que posteriormente son usadas por los implementadores para solucionar problemas de integración entre sistemas de información heterogéneos.

A partir del 17 de diciembre de 2003, la ANSI ha aprobado una serie de estándares contenidos en la especificación de la Versión 3 HL7. Estos estándares utilizan una notación formal de modelado (UML) y un metalenguaje extensible de marcado con etiquetas (XML).

La organización HL7 hoy en día es una de las organizaciones más importantes de informática médica a nivel internacional.

Una lista de los comités técnicos y grupos de intereses especiales, así como sus misiones, alcances y coordinadores está disponible en el sitio Web de HL7.

La creciente necesidad de generar sistemas de información integrados regionalmente (ciudades, regiones, países) hizo necesario el desarrollo de un espectro más amplio de estándares que faciliten la interoperabilidad.

Por esta razón, en la actualidad la organización HL7 cuenta con un proceso para definir una serie de herramientas de interoperabilidad (mensajes, documentos electrónicos, reglas, modelos de referencia), esto ha dado origen a varios estándares que facilitan los procesos de intercambio de información de salud.

Debido a ello, hoy en día, se habla de estándares HL7.

Algunos de estos estándares son:

- **Mensajería HL7 versión 2:** Estándar de mensajería para el intercambio electrónico de datos de salud.
- **Mensajería HL7 versión 3:** Estándar de mensajería para el intercambio electrónico de datos de salud basada en el RIM (*Reference Information Model*).
- **CDA HL7:** (*Clinical Document Architecture*) Estándar de arquitectura de documentos clínicos electrónicos.
- **SPL HL7:** (*Structured Product Labeling*) Estándar electrónico de etiquetado de medicamentos.
- **HL7 Medical Records:** Estándar de administración de Registros Médicos.
- **GELLO:** Estándar para la expresión de reglas de soporte de decisiones clínicas.
- **Arden Syntax:** Es estándar sintáctico (*if then*) para compartir reglas de conocimiento clínico.
- **CCOW:** Es un estándar para *Framework* para compartir contexto entre aplicaciones.

2.4 Software para el manejo del estándar HL7

2.4.1 7Edit

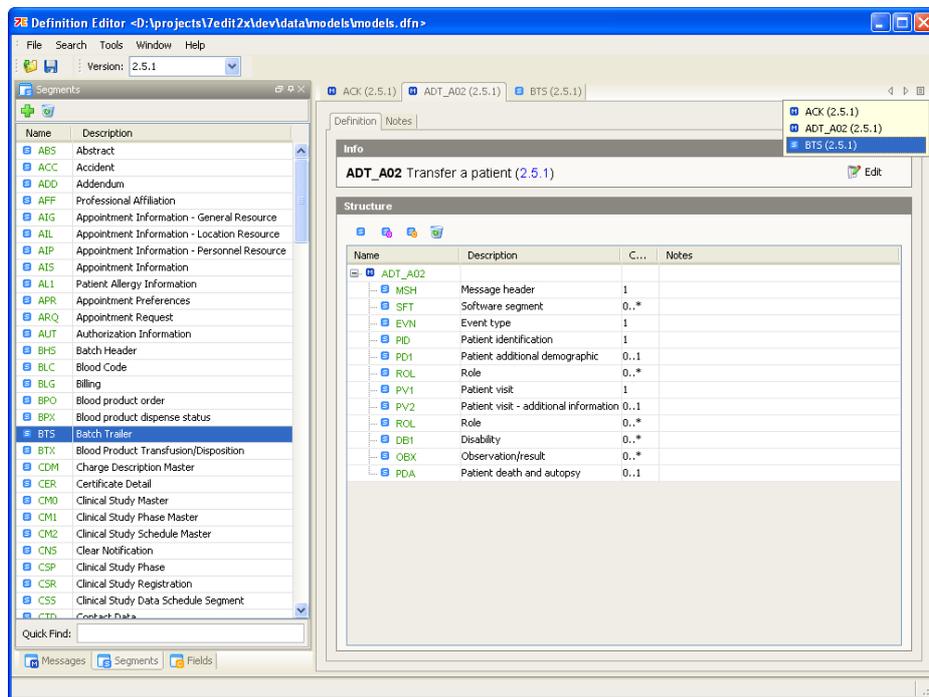


Ilustración 2-C Software 7Edit

7Edit es una herramienta de productividad para navegar, edición, validación de mensajes de HL7 e intercambio de datos con las aplicaciones HL7. Algunas de las características clave de 7Edit son:

1. Ver mensajes de HL7;
2. Editar mensajes de HL7;
3. Validar mensajes de HL7;
4. Enviar y recibir mensajes de HL7;
5. Mensajes de HL7 de exportación a Excel y XML;
6. Personalizar las definiciones y tablas.

2.4.2 Iguana HL7

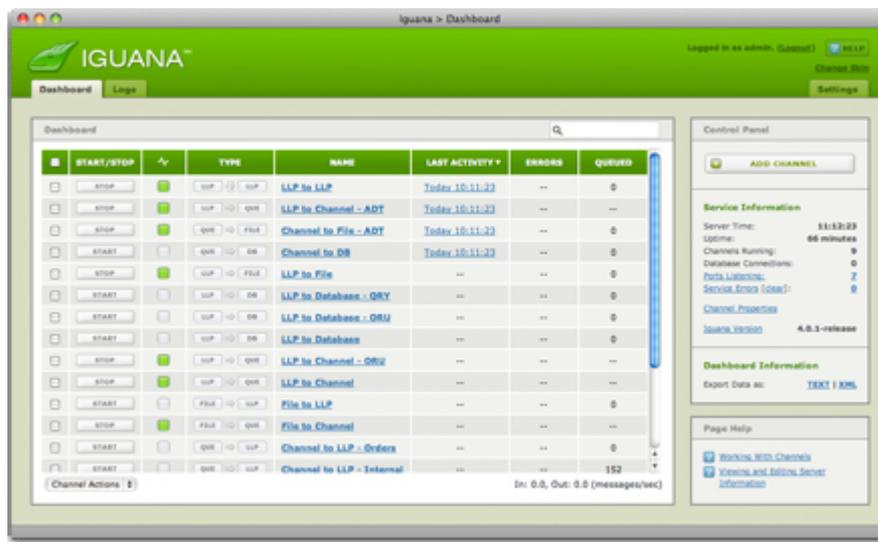


Ilustración 2-D Software Iguana HL7

Una solución completa a los complejos problemas de integración de la salud. Diseñado específicamente para la industria de la salud, Iguana permite a la transferencia transparente de la información electrónica sin necesidad de reemplazar los sistemas existentes.

1. Elimina la necesidad de una programación personalizada;
2. Interfaz de control, fácil de usar y ambiente Web;
3. Todas las plataformas y base de datos soportadas;
4. Enviar multitud de mensajes HL7 por segundo.

2.4.3 HL7Spy

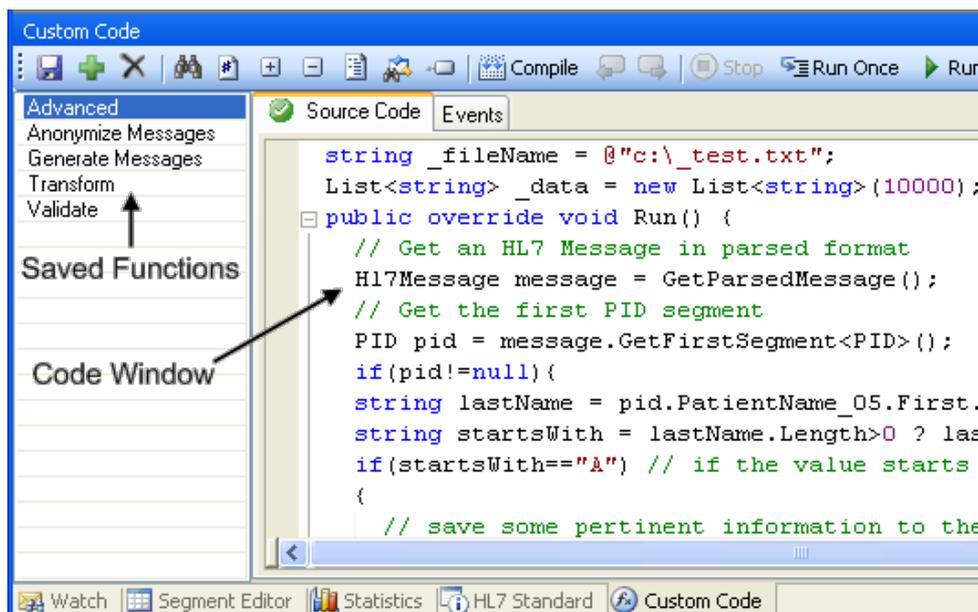


Ilustración 2-E Software HL7Spy

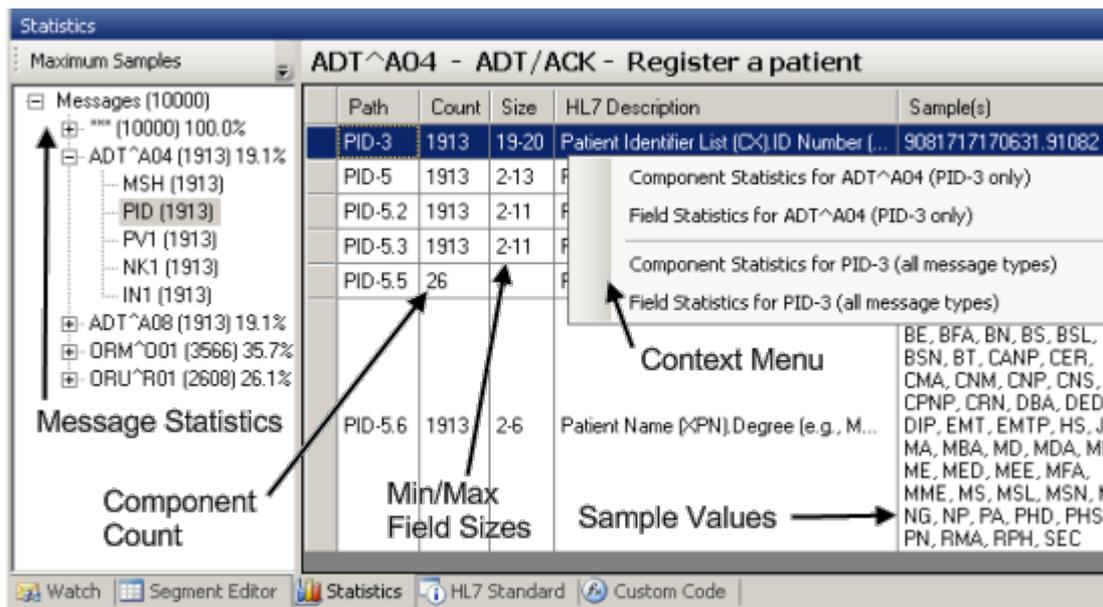


Ilustración 2-F Software HL7Spy

Uno de los aspectos más complejos de configurar y mantener, aplicaciones de salud es la interpretación de la semántica de los datos que se intercambian. HL7Spy ha sido diseñado para ayudar a HL7 analistas y programadores con rapidez interpretar, y caracterizar muy grande de mensajes HL7 arroyos. Rápidamente se lee en 1000 de los mensajes y proporciona estadísticas en el mensaje, el campo, los componentes y en los niveles sub-componente proporciona una gran comprensión de la naturaleza del intercambio de datos.

2.5 Reflexiones

Como se mencionó, en la actualidad no existe ningún software especializado únicamente a la simplificación de comunicación de equipo clínico y un sistema de cómputo.

Tanto como las empresas como los desarrolladores independientes siguen usando el antiguo modelo de crear el protocolo de manera manual y probarlo, corregir errores y repetir pruebas, ya que es lo único que se tiene hasta el momento.

Ahora bien, este tipo de metodología de “esperar, desarrollar y probar” se está volviendo rápidamente obsoleta, ya que con las tendencias actuales, existe la necesidad de estar constantemente actualizando el software, debido a los nuevos problemas que surgen con cada nueva versión del equipo.

CAPÍTULO 3.

MARCO

TEÓRICO

3.1 Analizador léxico

Un analizador léxico[7] consiste en examinar en un orden lexicográfico (ordenación de caracteres) un flujo de caracteres. El rastreador -como también suele llamarse- realiza una recolección de secuencia de caracteres en unidades significativas denominadas *tokens*, estos *tokens* son un conjunto de caracteres con un significado específico y en su totalidad forma una palabra en lenguaje natural. Así, de ésta forma podemos imaginar a un analizador léxico como un sistema de deletreo.

Por lo tanto se encarga de procesar carácter a carácter el programa fuente y buscar una secuencia o formato (expresión regular) que le corresponda y que ésta exista en el lenguaje que se desea, si es así se dice que cambia de estado, es decir, se mueve al siguiente carácter y verifica si se sigue cumpliendo el patrón que se analiza, así sucesivamente hasta que termina en el final del archivo con el último carácter del archivo fuente. Al analizador léxico se le conoce también como autómata finito determinista.

En esta etapa se genera al mismo tiempo un conjunto de tablas que almacenan las variables de interés de nuestro lenguaje (tabla de símbolos), que manipula el analizador sintáctico en la siguiente etapa. Otra función que interesa del analizador léxico es la capacidad de saltar posibles comentarios, espacios, saltos de línea y cualquier otro símbolo que puede causar un error en las etapas siguientes o que no se desee incluir en el lenguaje.

Para mostrar un ejemplo haremos uso de una sintaxis de un lenguaje muy conocido como es C [6]. Por lo que tenemos la siguiente línea de código:

```
Array [index] = 2 * 3;
```

El código anterior nos muestra una simple instrucción clásica de dicho lenguaje, sin embargo, para motivos didácticos nos será útil para ver como está funcionando el rastreador. Podemos notar a simple vista que la instrucción consta de tan solo 17 caracteres, los cuales son agrupados en 8 *tokens* como se muestra a continuación:

1. Array – Identificador
2. [- Corchete izquierdo, que simboliza la apertura de un elemento
3. Index – Identificador
4.] Corchete derecho, que simboliza el cierre de un elemento
5. = Asignación
6. 2 Número
7. * Operación aritmética
8. 3 Número

3.2 Analizador sintáctico

Una vez completada la fase de análisis léxico, es común proceder a una siguiente etapa denominada análisis sintáctico, ésta fase se encarga de recolectar información que fue creada en *tokens* por el analizador léxico. El análisis sintáctico determina los elementos estructurales de la fuente de datos y sus relaciones, los resultados de este análisis sintáctico por lo general se representan como un árbol de análisis gramática o árbol sintáctico [7].

Como ejemplo continuaremos con la línea de código en C anterior, ya que representa un elemento estructural denominado expresión [6] la cual es una expresión de asignación compuesta de una expresión con subíndice a la izquierda y subíndice a la derecha. Es representada como una estructurada de árbol gramatical como se muestra en la Ilustración 3-A.

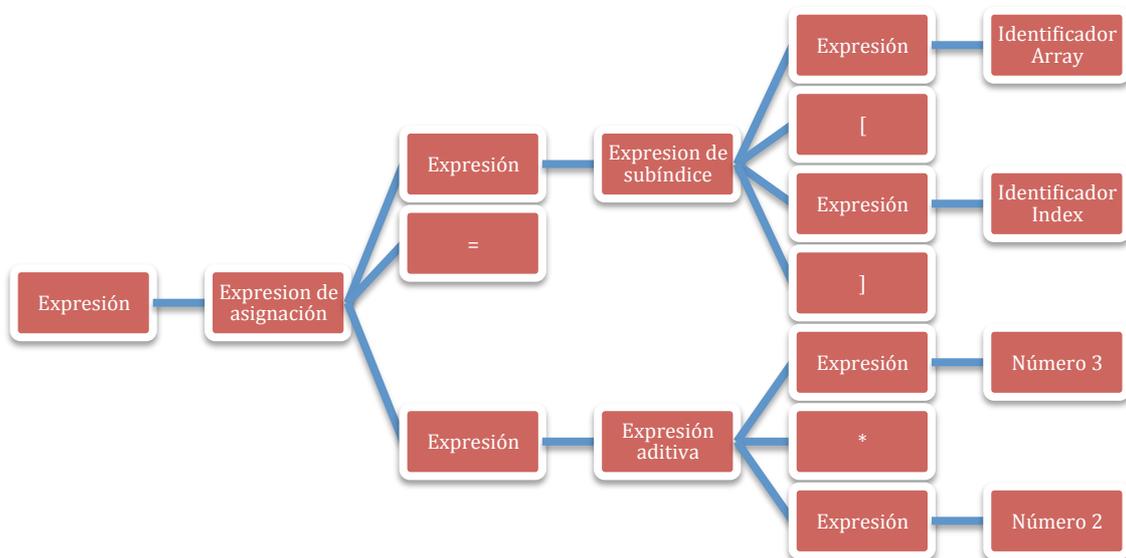


Ilustración 3-A Árbol gramatical

La etapa de análisis léxico y sintáctico (llamado “parser” en inglés) es la responsable de corroborar la estructura del protocolo de comunicación generado por el motor de datos, este análisis es basado según las normas y características del estándar E-1394-97 del estándar internacional ASTM. Dichas características son empleadas en todos los equipos clínicos que

desean manejar el protocolo el estándar para la comunicación entre equipo clínico y el equipo de cómputo.

El desarrollo de este *parser* [8] es para un dominio en específico y una tarea única, la cual es validar un estándar determinado, tendrá como única entrada aquel esqueleto de protocolo formado por una capa anterior, se encargará de revisar y validar el protocolo dando como resultado un único valor si la validación fue correcta, o un conjunto de datos con información concreta sobre el error generado. La información recogida por la salida del parser será necesaria para continuar con el proceso de aprendizaje o para la corrección de errores de una capa o varias capas anteriores y superiores.

En la presente tesis no se realizó el trabajo de la creación del analizador léxico y sintáctico, sin embargo, fue utilizada de una manera amplia la herramienta *Irony - .Net Language implementation Kit* [9] bajo la licencia de MIT [10]. La cual da la libertad de usarla, modificarla o distribuirla bajo los términos mencionados en dicha licencia.

3.3 Lenguajes formales

Las especificaciones de lenguajes formales se especifican mediante gramáticas, para entender cuáles son sus alcances y limitantes es útil entender su naturaleza [11]. Los lenguajes existen desde los primeros avances de la humanidad, ellos permiten obtener un significado de los objetos y fenómenos que se suscitan al diario en la vida, como por ejemplo, la oración “Los niños corren hacia la casa” indica la dirección, acción y los objetos de la acción, es por ello que se entiende, en este caso, lo que “los niños” hacen; por otra parte las matemáticas modelan muchos otros fenómenos que con otra notación o lenguaje utilizando símbolos como ‘+’, ‘-’, ‘{’, ‘}’ entre otros, también denotan acciones, fenómenos, etc. como los conjuntos, de manera matemática se puede definir todos los números pares utilizando la notación “ $pares = \{x | x = 2m, m < 0 \text{ y } m \in Z^+\}$ ” lo que en nuestro lenguaje se traduce como los pares son todos aquellos números formados por 2 por m con m mayor que cero y m siendo un entero positivo, si se quiere denotar en nuestro lenguaje cuales son los números pares se puede enumerar todos los pares los cuales como se sabe son infinitos, de la siguiente manera “pares: 2, 4, 6,…” y se entiende que “…” significa que se extiende hasta el infinito.

Como se puede percatar, los dos lenguajes ejemplificados utilizan algunos símbolos que son similares y otros que no, por ello, se entiende que son lenguajes diferentes, pero existen varios otros lenguajes que son subconjuntos de otros, es decir, que existen lenguajes que contienen a muchos otros, o a su unión. Aún así el estudio de lenguajes es extenso, sin embargo, existen lenguajes que no pueden ser anotados de una forma matemática completa, ya que pueden producir oraciones muy complejas, en otros casos, ambiguas que al carecen de sentido o este no se puede comprender.

Este problema es tan grande como la limitación de la computación misma, existen problemas que no pueden resolverse con computadoras o la solución

tardaría muchos siglos e incluso miles de años en poder ser resueltos mediante computadoras (problemas no computables y problemas NP). Pese a estas limitantes, varios matemáticos han intentado acotar los fenómenos que son modelables matemáticamente para posteriormente ser resueltos con computadoras, y han llegado a taxonomías bastante claras que desde su origen ha dado paso a lenguajes que permiten denotar: acciones, procesos, procedimientos y gran cantidad de elementos que hacen útiles a las computadoras. Los lenguajes de programación datan desde el origen de la computación, es la forma en que nosotros, los humanos, interactuamos con las computadoras y a su vez, ellas, devuelven cálculos. Este lenguaje es conocido como código de máquina y, aún en estos días se usa, cabe recalcar que todo aquello que se introduce a una computadora y es capaz de devolver un resultado, es llamado “computable”, al ser computable, existe un procedimiento secuencial que es capaz de terminar en un punto que devuelve un resultado, ésta aseveración fue demostrada por el padre de la computación Alan Turing en los años 40's.

Conforme el hombre se fue acostumbrando a las computadoras empezó a desarrollar nuevas formas que facilitarían la interacción con ellas, de ahí surgieron los lenguajes mnemotécnicos los cuales tienen instrucciones como saltar, limpiar, cargar, entre otras, que después son traducidas a código máquina y ejecutadas. No había pasado mucho tiempo cuando se empezaron a proponer otros lenguajes más complejos en su estructura, haciéndola similar al idioma inglés. De ahí en adelante existe una proliferación de lenguajes que no se restringen a dar secuencias de instrucciones, sino que modelan los problemas de otra forma unos en base a objetos, atributos y métodos (programación orientada a objetos), otros a reglas (programación lógica), a funciones (programación funcional), etc., haciendo su notación bastante variada y en muchos casos compleja en relación al código máquina inicial.

Ahora bien, todos los lenguajes computables como se explico al inicio de ésta sección cuentan con una gramática asociada que los genera. La gramática incluye 4 aspectos:

1. Los símbolos permitidos en el lenguaje.
2. Los nombres de las reglas para generar los símbolos y el lenguaje mismo.
3. Las reglas.
4. El símbolo (variable) inicial.

Para dejar claro cada uno de estos aspectos se tomará como ejemplo la el lenguaje de la aritmética (subconjunto de las matemáticas). Al ser así, se tiene para cada punto lo siguiente:

1. En la aritmética son válidos todos los números y cuatro símbolos extra, denotados '+', '-', '*', y '/' y los paréntesis '(', ')'. Su unión forma los símbolos permitidos en el lenguaje, matemáticamente se denota a este conjunto: $A_p = Ru\{(,)\}u\{+, -, *, /\}$ Llamando conjunto A_p .

2. En aritmética se cuenta con las siguientes operaciones: suma, resta, multiplicación y división, las primeras dos tienen precedencia menor a las tercera y cuarta, así también con expresiones y números, así pues, los nombres generadores del lenguaje son:
 - a. Operaciones con menor precedencia.
 - b. Operaciones con mayor precedencia.
 - c. Expresión.
 - d. Números reales.

Al conjunto de nombres de reglas se le llamará Ar .

3. Ahora bien, las reglas P permiten generar sentencias válidas en nuestro lenguaje, indican la estructura de nuestras sentencias.

La estructura tiene la siguiente Notación:

Nombre_de_regla: subreglas derivadas.

Para nuestro ejemplo, las reglas para formar nuestro lenguaje son:

Operaciones_menor_precedencia: suma o resta o
Operaciones_mayor_precedencia

Operaciones_mayor_precedencia: multiplicación o división o expresiones

Expresiones: números o (expresiones)

Números: reales o decimales

4. La regla que inicia todo el proceso de generar sentencias válidas es Expresiones, cualquier sentencia válida en la aritmética será una expresión. A ésta regla se le conoce como S .

Así entonces conjuntando los cuatro elementos de nuestro lenguaje es posible generalizar la definición de gramática.

Sea una gramática G formada por: $G = \{A_p, Ar, P, S\}$

Donde:

- G es el nombre de la gramática
- A_p el alfabeto de palabras
- Ar el alfabeto de reglas
- P el conjunto de reglas del lenguaje
- S la regla inicial

3.4 Expresiones regulares

A cualquier palabra que pueda ser formada por una combinación de símbolos y para esto no se requiere una estructura que abarque reglas para ser generada, se conocerá como expresión regular [12,9]. Para ello existe un lenguaje matemático simple que, utilizando símbolos '*', '+', '(', ')', '|', '-' denotan lo siguiente:

Símbolo '-' se usa para denotar un conjunto inmerso entre el elemento de la izquierda y el de la derecha, actualmente, solo es útil para denotar del cero al nueve y de la a "a" la "z" en minúsculas o mayúsculas.

- 0-9 Permite crear palabras del cero al nueve decimal.
- a-z Crea palabras con cualquier letra del abecedario.

Símbolo '|' Se utiliza para denotar la O lógica.

Ejemplo:

Formar siempre una palabra con a o con v

Expresión regular

Ejemplo:

O: a | v;

Símbolos '()' Se usan juntos para agrupar símbolos que se requiere aparezcan en cierto orden.

Ejemplo

Formar siempre una palabra que tenga 'ab'

La expresión regular que genera estas palabras es:

Tener_ab: (a-z)(ab)(a-z)

En este caso Tener_ab tiene 3 grupos de los cuales el primero y tercero son el mismo (a-z) y (ab), genera palabras como aaba, babc, vabd, etc.

Símbolo '*' Se usa para denotar cero o más apariciones del grupo de símbolos inmediato a la izquierda.

Ejemplo

Cualquier nombre en español puede ser formado por una combinación de letras.

La expresión regular para cualquier nombre es:

Nombre: (a-z |A-Z)*

Nombre puede generar palabras como aaaa, abce, ahjdf, e inclusive el nombre _ (vacío).

Símbolo '+' Denota al menos una aparición del grupo de símbolos inmediato a la izquierda.

Ejemplo

Cualquier número decimal

Expresión regular

Decimal: (0-9)+

Decimal genera todos los números enteros, como 0, 99, 91234, 12421, etc.

Las expresiones regulares se representan a través de máquinas de estados, cada vez que se encuentra un símbolo se procede a cambiar en un estado que acepte el símbolo que se está analizando o a mover la máquina hacia el siguiente símbolo a analizar.

Ejemplo

Reconocer la palabra **aei**

La máquina de estados o autómeta que reconoce la palabra es:

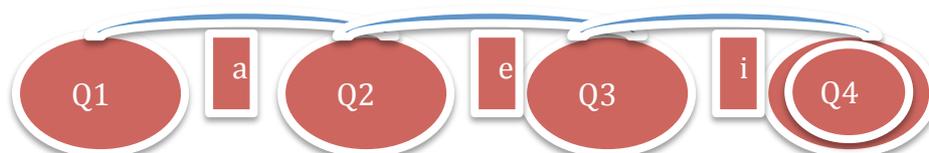


Ilustración 3-B Máquina de expresiones regulares

Cada vez que se procesa un símbolo del alfabeto, se cambia de estado, como se puede observar, cuando se encuentra una a se cambia del estado Q1 a Q2 y así sucesivamente, cabe señalar que varios autómatas pueden reconocer una misma expresión regular, por lo que el autómeta anterior es solo un ejemplo para reconocer esa palabra. A los autómatas también se les conoce como autómatas finitos, el término finito se debe a que en cierto punto la máquina debe dejar de procesar y terminar. Existen dos variantes de los autómatas finitos, los deterministas y los no-deterministas, la diferencia entre uno y otro es la capacidad de tomar diferentes decisiones para procesar un símbolo (caso no-determinista), y solo poder tomar una decisión (caso determinista). Está demostrado que estos modelos son análogos.

3.5 Forma Backus-Naur y Backus-Naur extendida

La forma Backus-Naur o BNF, fue creada por John Backus y Peter Naur para describir la sintaxis de un lenguaje dado. Apareció por primera vez en el lenguaje ALGOL60 [13,12]. Es frecuentemente utilizado para describir las reglas de un lenguaje utilizando la siguiente Notación:

'::=' es "se define como"
'< nombre>' "nombre de la regla"
'|' "O lógica"

Un ejemplo para formar cualquier número es:

```
<número-decimal> ::= <entero-sin-signo> | <fracción-decimal> |  
<fracción-decimal> ::= <entero-sin-signo> <fracción-decimal>  
<entero-sin-signo> ::= <dígito> | <dígito> <entero-sin-signo>  
<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
```

La regla <dígito> también puede ser mostrada como 0-9 en vez de colocar todos los números, en el caso de la versión original si es necesario hacerlo, la versión extendida contiene el símbolo '-'.

Prácticamente este es lenguaje más utilizado para definir gramáticas en todo el mundo, ya sea por su facilidad o claridad en su sintaxis, lo cierto es que hasta hoy no se ha propuesto una notación más fácil o aceptada.

3.6 Extensible Markup Language (XML)

XML[2] fue desarrollado en los años setenta por la empresa estadounidense IBM y está formado bajo el auspicio del *World Wide Web Consortium* (W3C) en el año de 1996, y fue precedido por Jon Bosak.

Cada documento XML tiene estructuras tanto lógica como física. Físicamente, el documento está compuesto de unidades llamadas entidades. Una entidad podría referirse a otras entidades que causa su inclusión en los documentos. Un documento inicia en una raíz o documento de entidad. Lógicamente, el documento está compuesto de declaraciones, elementos, comentarios, referencias a caracteres e instrucciones de procesamiento, todos ellos marcados explícitamente en el documento.

Los objetivos principales son:

1. XML será directamente utilizable en internet.
2. XML deberá soportar una amplia variedad de aplicaciones.
3. XML deberá ser compatible con SGML.
4. Deberá ser fácil escribir programa que procesen documentos XML.
5. El número de características opcionales en XML es que se mantenga en lo mínimo absoluto, idealmente cero.
6. Los documentos XML deberán ser legibles por humanos y razonablemente claros.

7. El diseño de XML deberá ser preparado rápidamente
8. El diseño de XML deberá ser formal y conciso
9. Los documentos XML deberán ser fáciles de crear
10. La consistencia de las marcas en XML es de mínima importancia

Algunas de las ventajas del metalenguaje XML son:

1. Extensible, Después de haber sido diseñado y puesto en producción, es posible extenderlo con la adición de nuevas etiquetas sin modificar la esencia de la estructura.
2. El procesador de documentos XML es un componente estándar, una vez diseñado no es necesaria una actualización para nuevas versiones.
3. Los documentos XML creado por diferentes empresas no afecta en el entendimiento de la estructura por lo que es posible procesarla de una manera rápida y eficiente.
4. La estandarización entre documentos crear un nivel de intercambio entre diferentes plataformas de comunicación.
5. Los datos almacenados en este tipo de estructuras tiene un significado concreto y asociado un contexto, con lo que existe flexibilidad para estructurar documentos.

A continuación, se muestra un ejemplo de estructura de un documento desarrollado en XML:

```
<?xml version="1.0" standalone="yes"?>
<document>
<bookcategory>
  Non Java
<book>
  <title>Algorithms in C++</title>
  <author>Robert Sedgewick</author>
  <publisher>Addison-Wesley</publisher>
  <isbn>0-201-51059-6</isbn>
  <pages>658</pages>
  <price>46.25</price>
  <recommendation>Buy It</recommendation>
  <blurb>
    This is the best book I know about basic algorithms
    and data structures. I refer to it frequently.
  </blurb>
</book>
</bookcategory>
</document>
```

Ilustración 3-C Ejemplo de estructura XML

3.7 Especificaciones del estándar para la transferencia de información entre instrumentos clínicos y sistemas de cómputo

Descrito en la norma ASTM para el intercambio de información, el documento referido como E1394-97[1], define el estándar como:

“Norma que se aplica a la transmisión digital de dos vías de peticiones remotas para el intercambio de resultados entre instrumentos clínicos y sistemas informáticos. Su objetivo es documentar los convenios comunes necesarios para el intercambio de los resultados clínicos y datos de los pacientes entre los diferentes instrumentos.”

Esta norma también especifica el contenido del mensaje para transferir información entre un equipo clínico y un sistema de cómputo. Permite a cualquiera de los dos sistemas establecer un vínculo lógico para la comunicación de texto para enviar resultados, peticiones o información demográfica en un formulario normalizado e interpretable.

3.7.1 Definiciones de términos específicos para el estándar ASTM

3.7.1.1 Batería

Un grupo de pruebas ordenadas conjuntamente, por ejemplo, una batería de admisión. El término batería es usado en el protocolo sinónimamente con el término perfil o panel. Los elementos de pruebas dentro de una batería pueden ser característicos de un sistema fisiológico único o múltiples sistemas fisiológicos diferentes. La batería es una simple convención que un usuario puede ordenar múltiples pruebas especificando un solo nombre.

3.7.1.2 Campo componente

Un único elemento de datos o elementos de datos que expresan un agregado fino extensión de elementos de datos que le preceden. Esto es, partes de un campo o campos de entrada repetidos, por ejemplo, el nombre del paciente es registrado como apellido paterno, nombre e iniciales, cada uno de ellos se separa por un componente delimitador. Los componentes no pueden tener campos repetidos.

3.7.1.3 Descarga

Datos transmitidos de un sistema de cómputo a un instrumento clínico.

3.7.1.4 Campo

Un atributo específico de un registro que puede contener agregados de elementos de datos más refinados del atributo básico.

3.7.1.5 Mensaje

Un cuerpo textual de información consistente de un registro de cabecera (H) a través de un registro de terminación de mensajes (L).

3.7.1.6 Registro

Un agregado de campos describiendo un aspecto de un mensaje completo

3.7.1.7 Campo de repetición

Un elemento de datos únicos, que expresa una duplicación de la definición del campo que se está repitiendo. Se utiliza para la demográfica, las solicitudes, pedidos y similares, donde cada elemento de un campo de repetición debe ser tratado con la misma prioridad o esperar campos de repetición asociados.

3.7.1.8 Prueba

Una determinación de un único análisis o combinación de valores de otras determinaciones u observaciones que constituyen una medida de un atributo único de sistema.

3.7.1.9 Subidas

Datos transmitidos de un instrumento clínico a un sistema de cómputo.

3.7.2 Significado y uso

3.7.2.1 Información general

Estas especificaciones proporcionan transmisión de dos vías permitiendo flujos de datos en cualquier dirección. Se prevé el envío de información demográfica, información de pruebas o de instrumentos clínicos. La especificación tiene suficiente flexibilidad para permitir la adición de campos a tipos de registros existentes o la creación de un nuevo tipo de registro para acomodar nuevas pruebas y reportar metodologías.

Las especificaciones caen en los límites del estándar **[13]** y **[1]**, ambos estándares usa convención posicional para definir la estructura de mensajes que intercambian información sobre pruebas clínicas, solicitudes y resultados. El conjunto de convenciones que especifica un conjunto de jerarquías de registros en el que los registros de mayor rango dentro de la jerarquía contienen información que es común para todos los registros de menor rango en la jerarquía, y por lo tanto evita la redundancia en la conexión de los datos en conjunto. La posición convencional es simple y directa para implementar, requiriendo solo una secuencia de cadenas de cada campo variable que tiene longitud delimitada que están específicamente posicionadas.

El estándar E-1238, no es apropiado para ser usado en un instrumento clínico para enlazar un sistema de cómputo. Estas convenciones con respecto a los tipos de registros y la organización de los datos dentro de los

registros se han adherido a la medida de lo posible para garantizar que los elementos comunes de datos definidos por allí y usados dentro de los instrumentos que son especificados en la mayor medida posible.

3.7.3 Requisitos de información en los ensayos clínicos

Los mensajes podrán contener uno o más peticiones/resultados para uno o más pacientes. Las pruebas pueden ser solicitadas como grupos de muchas pruebas individuales. Estos grupos son referenciados como baterías.

Los mensajes consistentes en una jerarquía de registros de varios tipos, como son:

Registros de nivel cero contienen información perteneciente a la identificación del remitente y la finalización de la transmisión.

Registros en el nivel uno contienen información sobre pacientes individuales.

Registros de nivel dos contienen información sobre las órdenes de pruebas y especímenes.

Registros de nivel tres contienen información sobre el resultado de las pruebas.

Registros de comentarios pueden ser insertados en cualquier nivel de la jerarquía. Un registro de comentario está inmediatamente precedido a un paciente, orden, resultado, o información de manufactura. Por lo tanto, si un registro de comentario que está seguido de un registro de paciente (Nivel uno), entonces el registro de comentario deberá ser tratado como un registro de nivel dos. Un registro de comentario podría no seguir el mensaje de registro de término.

La información de registro de manufactura puede ser insertada en cualquier nivel dentro de la jerarquía. Un registro de manufactura está inmediatamente precedido a un paciente, orden, resultado, o comentario. Por lo tanto, si un registro de manufactura que está seguido de un registro de paciente (Nivel uno), entonces el registro de comentario deberá ser tratado como un registro de nivel dos. Un registro de comentario podría no seguir el mensaje de registro de término.

Tipos de registros adicionales son, registro de solicitud de información y registro de término. El registro de solicitud de información proporciona solicitud de datos demográficos o resultados de pruebas para o del equipo clínico para pacientes especificados, especímenes, pruebas, fechas entre otros. El registro de término deberá ser el último registro del mensaje.

El elemento más pequeño de información en cualquier registro es el campo, conteniendo un solo elemento de información como fecha, nombre, o resultado numérico.

El registro de orden de prueba contiene información sobre orden de una única prueba, batería de pruebas, o series de pruebas o baterías.

Una secuencia de registro de pacientes, registro de órdenes, o registro de resultados en el nivel uno es terminado por la aparición de un registro del mismo tipo o de un nivel superior. Esto es, una secuencia de resultados para una batería de pruebas es terminada por la siguiente orden de pruebas, paciente, o información de manufactura o registro de término.

Un registro de orden nunca podrá aparecer sin un registro de pacientes precedente y un registro de resultado nunca podrá aparecer sin un registro de orden precedente.

Cuando una orden es transmitida, debe ser precedida por un registro de pacientes. Cuando un resultado es transmitido, este deberá ser precedido por un registro de orden y un registro de paciente para mantener la jerarquía prescrita.

Los requerimientos para el almacenamiento de información, es el orden para determinar requerimientos de búfer, tanto como transmisor y receptor deberán usar reglas comunes para el almacenamiento de datos transmitidos con el fin de asegurar un correcto registro de errores y recuperación de estos. Dado que el contenido de datos está estructurado de una manera jerárquica, cualquier cambio decreciente en el nivel jerárquico deberá activar el almacenamiento de todos los datos transmitidos con anterioridad a dicho cambio de nivel. Esta regla puede ser considerada como mínima.

Los requerimientos lógicos para la recuperación de errores de transmisión, son aquellos dados cuando la línea de transmisión falla, determinado en un nivel de protocolo de transmisión, esto requiere un mecanismo para reiniciar el mensaje incompleto. Si una falla de transmisión ocurre, la transmisión deberá de reiniciar en el último registro lógico. El siguiente esquema muestra las líneas de transmisión fallidas y sus respectivas acciones de almacenamiento para las recuperaciones de errores:

Número de línea	Tipo de registro	Nivel	Acción
A	Cabecera	(Nivel 0)+0	
B	Paciente 1	(Nivel 1)+1	
C	Orden 1	(Nivel 2)+1	
D	Resultado 1	(Nivel 3)+1	
E	Orden 2	(Nivel 2)-1	Salva A-
D			
F	Orden 3	(Nivel 2)+0	
G	Paciente 2	(Nivel 1)-1	Salva E-
F			
H	Orden 1	(Nivel 2)+1	
I	Comentario 1	(Nivel 3)+1	
J	Resultado 1	(Nivel 3)+0	
K	Comentario 1	(Nivel 4)+1	

L		Resultado 2	(Nivel 3)-1	Salva G-
K				
M		Orden 2	(Nivel 2)-1	Salva L
N		Paciente 3	(Nivel 1)-1	Salva M
O		Orden 1	(Nivel 2)+1	
P		Resultado 1	(Nivel 3)+1	
Q		Mensaje de término	(Nivel 0)-3	Salva N-
P				

Tabla 3-A Estructura lógica para el almacenamiento de información

(Nivel 0)	Cabecera			
(Nivel 1)	Información de manufactura			
(Nivel 1)	Paciente 1			
(Nivel 2)	Comentario			
(Nivel 2)	Orden 1			
(Nivel 3)	Comentario 1			
(Nivel 3)	Resultado 1			
(Nivel 3)	Resultado 2			
(Nivel 4)	Comentario 1			
(Nivel 4)	Comentario 2			
		.		
		.		
		Resultado n		
(Nivel 2)	Orden 2			
		Resultado 1		
		Resultado 2		
		.		
		.		
		.		
		Resultado n		
	.			
	.			
	.			
(Nivel 2)	Orden n			
(Nivel 3)		Resultado 1		
		.		
		.		
		.		
(Nivel 1)	Paciente 2 (Toda la estructura se repite)			
		.		
		.		
		.		
(Nivel 1)	Paciente n			
(Nivel 0)	Mensaje de término			

Tabla 3-B Estructura lógica de un mensaje

La siguiente tabla muestra el nivel de error que puede ocurrir y los datos necesarios que deben ser retransmitidos al dispositivo, la tabla se basa en la estructura lógica del mensaje en la Tabla 3-C:

Falla de la línea ocurre en	
A	A
B	A, B
C	A, B, C
D	A, B, C, D
E	A, B, C, D, E
F	A, B, E, F
G	A, B, E, F, G
H	A, G, H
I	A, G, H, I
J	A, G, H, I, J
K	A, G, H, I, J, K
L	A, G, H, I, J, K, L
M	A, G, H, L, M
N	A, G, M, N
O	A, N, O
P	A, N, O, P
Q	A, N, O, P, Q

Tabla 3-C Tabla de requerimientos de retransmisión

3.7.4 Contenido del mensaje

3.7.4.1 Código de Caracteres

Todos los datos deberán ser representados como *eight-bit*, *single-byte*, los valores codificados de caracteres gráficos son definidos en [14]. Los valores *eight-bit*, dentro del rango de 0 a 127 de [15] corresponden al conjunto de caracteres del estándar ASCII. Los valores de 0 a 31 no están permitidos con la excepción de 7 (timbre), 9 (Tabulador horizontal), 11 (tabulador vertical), y 13 (*carrier*) donde 13 es reservado como un registro de término. Valores del 32 al 126 y del 128 al 254 son permitidos. Los valores 127 y 255 tampoco son permitidos.

Caracteres permitidos: 7, 9, 11, 12, 13, 32-126, 128-254.

Caracteres no permitidos: 0-6, 8, 10, 14-31, 127, 255.

Dentro de los campos de datos textuales, solo el carácter Latin-1, los caracteres 32-126 y los caracteres no definidos 128-254 son permitidos como caracteres usables (Excluyendo aquellos usados como delimitadores en una transmisión particular). Además, todos los caracteres usados como delimitadores en una transmisión son excluidos del rango permitido. El trasmisor es responsable de analizar todos los campos de texto para garantizar que el texto no contiene delimitadores. Al menos que el protocolo indique lo contrario el contenido de los datos debe ser sensitivo a mayúsculas y minúsculas.

3.7.4.2 Longitud Máxima del campo

El protocolo define que todos los campos son de longitud variable. No se asigna almacenamiento (excepto por el delimitador) por un campo nulo. Cuando, por ejemplo, 10 caracteres de datos son ingresados dentro de un campo, solo 10 caracteres deberán ser usados. El protocolo no define una longitud máxima para cualquier campo o registro y se basa en la capacidad del buffer del receptor, y la capa lógica de los medios de transporte facilita analizar información de longitudes variables, para propósitos de transmisión y procesamiento. Esto es responsabilidad los proveedores de los instrumentos clínicos y sistemas de cómputo para acordar cualquier campo arbitrario o registro que pueda ser necesario y deberá ser impuesto.

3.7.4.3 Longitud máxima de registro

El protocolo define que no se encuentra impuesta en ningún momento.

3.7.4.4 Delimitadores

Caracteres alfanuméricos no deberán utilizarse como delimitadores, ya que es probable que aparezcan en el contenido del campo. Sin embargo, caracteres alfanuméricos tienen usos especiales como la siguiente lista:

Identificadores de tipos de registros: H, P, O, R, C, Q, S, L, M

Punto decimal: Latin-1(46)

Coma: Latin-1(44)

Prioridad de códigos: S, P, R, C

Código de resultados: L, H, <, >, N, U, D, B, W

Estado del resultado: C, P, F, X, I, O

Para el propósito de proporcionar ejemplos, los siguientes delimitadores son usados en el protocolo:

Delimitador de campo = barra vertical (|) Latin-1 (124).

Delimitador de repetición = diagonal invertida (\) Latin-1 (96).

Delimitador de componente = acento circunflejo (^) Latin-1 (94).

Delimitador de escape = ampersand (&) Latin-1 (38).

3.7.4.4.1 Delimitador de registro

Retorno de carro (13) deberá ser el delimitador para el final de cualquiera de los tipos de registros definidos.

3.7.4.4.2 Delimitador de campo

Un único carácter permitido como se definió en **3.7.4.1**, excluyendo Latin-1 (13), deberá separar los campos adyacentes. El campo delimitador es variable y definido dentro de la cabecera del mensaje. El mismo delimitador deberá ser usado en todos los registros siguiendo una cabecera y precediendo un mensaje de término de registro.

3.7.4.4.3 Delimitador de repetición

Un único carácter permitido como se definió en 4.7.4.1, excluyendo Latin-1 (13), y el valor para el campo delimitador definido en 4.7.4.4.4. El delimitador de repetición debe ser definido dentro de la cabecera del mensaje y usado para separar números variables de descriptores para los campos que contengan partes de los miembros iguales de la misma serie.

3.7.4.4.4 Delimitador de componente

El delimitador de componente es usado para separar elementos de datos o campos de una jerarquía, por ejemplo, direcciones, ciudades, códigos postales, calles, etc.

3.7.4.4.5 Delimitador de escape

El delimitador de escape es usado dentro de campos de texto para significar casos de operaciones especiales. Las aplicaciones del delimitador de escape son opciones y pueden ser usados o ignorados. Sin embargo todas las aplicaciones son requeridas para aceptar el delimitador de escape y utilizarlo para analizar correctamente los campos dentro del registro.

3.7.4.4.6 Uso del delimitador de escape

Del delimitador de escape podría ser usado para señalar ciertas características especiales de porciones de un campo de texto (por ejemplo, delimitadores incrustados, líneas de alimentaciones, retornos de carro). Una secuencia de escape consiste del carácter de delimitador de escape por un solo identificador de escape, como se muestra en la Tabla 3-D Delimitadores de escape, seguido por cero o más caracteres de datos seguido por otras ocurrencias de carácter de delimitador de escape.

&H&	Comenzar a resaltar el texto
&N&	Texto normal (fin de resaltar texto)
&F&	Incrustar carácter delimitador de campo
&S&	Incrustar carácter delimitador de componente
&R&	Incrustar carácter delimitador de repetición
&E&	Incrustar carácter delimitador de escape
&Xh&	Dato hexadecimal

Tabla 3-D Delimitadores de escape

3.7.4.4.7 Especificaciones de los delimitadores

Los delimitadores actuales serán empleados en una transmisión dada deberán ser especificados dentro de la cabecera del mensaje. Es responsabilidad del emisor evitar la inclusión de cualquier carácter delimitador dentro del contenido del campo. La computadora receptora determinara que caracteres usar para leer las especificaciones de la cabecera que recibe.

3.7.4.4.8 Delimitadores para valores nulos

Los campos deberán estar identificados por su posición, obtenido por el conteo de campos delimitadores desde la parte frontal del edificio. Este procedimiento de identificación de la posición sensible exige que cuando el contenido del campo es nulo, el delimitador de campo correspondiente deberá ser incluido en el registro para asegurar que el *i*-ésimo campo puede ser encontrado contando (*i*-1) delimitadores. Los delimitadores no se incluyen por detrás de campos nulos, esto es, si el décimo campo fue el último campo que contiene los datos, el registro podría terminar después del décimo campo, y por lo tanto contiene solo nueve delimitadores.

3.7.4.4.9 Campos sin importancia en el sistema de recepción

Registros transmitidos pueden incluir más campos que son requeridos por un sistema receptor. Cuando se procesa un mensaje, el sistema receptor podrá ignorar cualquier campo que no requiera. Los campos deben ser siempre transmitidos, sin embargo, dentro del orden posicional especificado.

3.7.4.4.10 Campos con valores nulos

Un sistema podrá transmitir un valor nulo para un campo porque:

- No conoce el valor
- Conoce el valor pero es irrelevante para el sistema receptor
- El valor no tiene cambios desde la última transmisión, o cualquier transmisión anterior

Porque el emisor puede usar valores nulos para indicar que no hubo cambios, un valor nulo no sobrescribe datos existentes dentro del sistema receptor. En circunstancias raras, por ejemplo, si un sistema erróneamente envía la fecha de nacimiento de un paciente cuando la fecha de nacimiento está actualmente desconocida, el sistema receptor deberá reemplazar este valor existente por un campo con valor nulo.

Un campo que contiene sólo un par de comillas dobles deberá ser tratado como una instrucción para el receptor que los contenidos existentes pertenecientes a esa definición de campo deberán ser borrados.

3.7.4.5 Registros de datos de uso general

Los datos deberán ser intercambiados dentro de los diferentes tipos de registros. Cada registro es introducido por el un campo identificador que identifica el tipo de registro, y terminado por un retorno de carro. Los siguientes tipos de registros son definidos:

3.7.4.5.1 Registro de cabecera (H)

Este registro deberá contener información sobre el emisor y el receptor, esto es, deberá identificar los instrumentos, el sistema de cómputo y los registros que se están intercambiando. También define los campos de caracteres de delimitadores, de repetición y de componentes.

3.7.4.5.2 Registro de identificación de paciente (P)

Este tipo de registro contiene información sobre un paciente de manera individual.

3.7.4.5.3 Registro de orden de prueba (O)

Cuando es enviado del sistema de cómputo al instrumento, este registro deberá representar una orden de prueba y podrá ser seguida por uno o más registros de resultados que podrán contener información pertinente para la prueba sea ordenada.

Cuando es enviada del instrumento clínico al sistema de cómputo, este deberá proporcionar información sobre la solicitud de muestra/prueba y podrá ser seguida por registros de resultados (al menos un registro por cada prueba dentro de las baterías ordenadas).

3.7.4.5.4 Registro de resultado (R)

Cada registro de resultado deberá contener solamente un valor por cada prueba.

3.7.4.5.5 Registro de comentario (C)

Los registros de comentarios deberán aplicarse en cualquier otro registro excepto en el registro de término. Pueden ser mensaje de pie enviado hacia o desde el instrumento, sin relación con un paciente en particular o procedimiento de prueba.

3.7.4.5.6 Registro de término (L)

Es un registro especial en el manejo de comunicación, tiene como única finalidad que indique a las entidades involucradas en el intercambio de datos el fin de la transmisión.

3.7.4.5.7 Registro de solicitud de información (Q)

Este registro deberá ser usado para solicitar información para nuevas pruebas, para pruebas anteriores ordenadas, y posibilidad para pruebas anteriores reportadas. Un simple registro de solicitud de información puede ser solicitar información demográfica, o resultado para una prueba individual, múltiples pruebas, o todas las pruebas de una fecha en particular, series de fechas, o rangos de fechas y para información de pacientes individuales, en grupo, especímenes o grupos de especímenes.

3.7.4.5.8 Registro científico (S)

Este registro deberá ser usado para intercambiar resultados entre sitios clínicos para propósitos de ensayos de aptitud o métodos de desarrollo.

3.7.4.5.9 Registro de información de manufactura (M)

Este registro, que es similar al registro comentario, puede ser usado para enviar complejas estructuras donde el uso de los tipos de registros existentes no sería apropiado. Los campos dentro de este tipo de registro son definidos por el fabricante.

3.7.4.6 Tipos de campos comunes

3.7.4.6.1 Identificador de pruebas universales

Este campo se define como un campo de cuatro partes de las disposiciones para definir más la prueba de identificación a través del uso de los campos componentes. El campo identificador de prueba es usado para identificar una prueba o nombre de la batería. Las cuatro partes que son definidas a continuación son identificadoras de pruebas universales, el nombre de la prueba, el identificador del tipo de prueba y el fabricante define el código de prueba. Todas las partes de identificadores de prueba deberán ser separadas por un delimitador de componente y situadas de manera dependiente. A modo de ejemplo, la información adicional que se puede incluir en este tipo de campo es instrumento de identificación, id organismo, un número de copa, número de ubicación, número de la bandeja, el número de código de barras entre otros. Es responsabilidad del fabricante del instrumento definir el contenido de los datos del campo de la identificación de la prueba. Cuando el identificador de prueba se utiliza en el registro del resultado, debe haber suficiente información en el campo de identificación de prueba para determinar la relación del resultado la relación del resultado de prueba a la prueba, batería, o baterías ordenadas.

3.7.4.6.2 Identificador de pruebas universales (Parte 1)

Este es el primer componente del campo de identificación de la prueba. Este campo está actualmente sin uso pero es reservado para la aplicación del código de prueba universal, debe estar disponible un sistema para usar en un tiempo futuro según el estándar.

3.7.4.6.3 Identificador de pruebas universales (Parte 2)

Este deberá ser la prueba o nombre de la batería asociada con el código de identificador de prueba universal como se mencionó en **3.7.4.6.1**

3.7.4.6.4 Identificador de pruebas universales (Parte 3)

En este caso donde múltiples nacionalidades o códigos internacionales existen, este campo podrá ser usado para determinar que esquema de

código es empleado en el identificador de prueba y nombre del identificador de prueba.

3.7.4.6.5 Código local o de fabricante (Parte 4)

Este es el código definido por el fabricante. Este código podrá ser un número, caracteres, o diseñador de pruebas múltiples basados en los fabricantes.

3.7.4.6.6 Fechas y hora

En todos los casos, las fechas deberán ser registradas con el formato YYYYMMDD, donde Y es año, M mes y D día, el formato es requerido por [16], por ejemplo, Diciembre 1, 1989 deberá ser representado como 19891201. Cuando la hora es transmitida, deberá ser representado por HHMMSS, donde H es hora, M minuto y S segundo, la información deberá ser ligada a la fecha especificado por [17]. Fechas y horas juntas deberán ser especificados como 14 caracteres formando una sola cadena: YYYYMMDDHHMMSS.

3.7.4.6.6.1 Zona horaria

La zona horaria puede ser opcionalmente añadida a la fecha/hora en el formato +HHMM o -HHMM. La zona horario predeterminada es la del emisor.

3.7.4.6.7 Números telefónicos

Los números telefónicos deberán ser registrados como texto libre, que puede contener extensiones como código de área, código de país, número personal, horario para llamar, etc.

3.7.4.6.7.1 Múltiples números telefónicos

Cuando múltiples número telefónicos aplican, estos pueden ser incluidos en un campo y separados cada uno por un delimitador de repetición. La primera entrada es considerada como número principal.

3.7.4.6.8 Ajuste de medidas y unidades

Cuando un campo contiene observaciones específicas, por ejemplo, peso del paciente, altura del paciente, o colección de volumen urinario, las unidades predeterminadas de medidas para las observaciones son especificadas en la definición de campo. Cuando la observación es medida en las unidades predeterminadas, las unidades no necesitan ser enviadas. Si la medida es registrada en unidades diferentes de las predeterminadas, por ejemplo, si el peso es medido en libras en lugar de kilogramos, entonces la unidad de medida debe ser transmitida. En este caso las unidades son transmitidas en el mismo campo que la medida. Las unidades de seguimiento de medida están separadas por un delimitador de componente, por ejemplo, 100^lb. Las unidades deberán ser expresadas en abreviaciones estándar ISO de acuerdo con [18].

3.7.4.6.9 Direcciones

Una dirección ocupa un único campo dentro del registro. La dirección puede ser compuesta de cinco componentes (calle, ciudad, estado, código postal, y país) separado por el componente delimitador, así de este modo el receptor pueda separar la dirección en los campos necesarios, por ejemplo, Lisboa 45^Colonia Juárez^Distrito Federal^56743^Mexico. Los componentes de este campo son dependientes de la posición.

3.7.4.6.10 Proveedor e identificador de usuario

Los identificadores de médicos y otros cuidadores pueden ser transmitidos como números de código interno, como nombres, o ambos, de común acuerdo entre el emisor y receptor. Cuando ambos el nombre y el número de identificación son enviados, el número deberá llegar primer y será separado del nombre por un campo delimitador. Cada componente del nombre también es separado por un campo delimitador. El orden de los componentes del nombre deberá ser:

1. Apellido.
2. Nombre.
3. Segundo nombre.
4. Sufijo.

Por ejemplo, Si el Dr. José G. Pérez tiene como identificador 401-0, entonces su número y nombre deberá ser transmitido 401-0^José^G^Perez^Dr. Cuando es necesario más de un identificador podrá ser enviado dentro un campo. Múltiples identificadores en un campo son separados por el delimitador de repetición.

3.7.4.6.11 Registro número de secuencia

Este es un campo requerido usado en los tipos de registro que podría ocurrir múltiples veces dentro de un solo mensaje. El número usado define la i-énésima ocurrencia de tipo de registro asociado en un nivel de jerarquía en particular y es reiniciado a uno cada vez que un registro de una mayor significación jerárquica (número menor) es transmitido o si el mismo registro es usado en un diferente nivel de jerarquía (por ejemplo, registros de comentarios).

3.7.4.7 Registro de cabecera

El registro de cabecera deberá contener identificadores del emisor y el receptor. El mensaje de cabecera es un registro de nivel cero y deberá ser seguido en el mismo punto por un mensaje de registro de término antes de terminar la sesión o transmitir otro registro de cabecera. Este tipo de registro deberá siempre ser el primer registro en ser transmitido.

3.7.4.7.1 Identificador de tipo de registro

El carácter H identifica el registro como un mensaje de registro de encabezado.

3.7.4.7.2 Definición del delimitador

Los cinco caracteres latinos que son inmediatamente seguidos después del carácter H (El identificador de cabecera) se definen los delimitadores que se utilizan en los registros posteriores del mensaje. El segundo carácter en el registro de cabecera es el campo delimitador, el tercero es el delimitador de repetición, el cuarto es el delimitador de componente, y por último es el carácter de escape. Un campo delimitador seguido de estos caracteres para sepáralos de los campos subsecuentes. Otra forma de ver esto es que el primer campo contenga H y el segundo campo contenga los delimitadores repetidor, componente y escape. Usando los delimitadores de ejemplo, los primeros seis caracteres en el registro cabecera será lo siguiente: **H|\^&|**.

3.7.4.7.3 Identificador de mensaje de control

Este es un número único u otro identificador que identifica de forma única la transmisión para el uso en sistemas de redes que han definido protocolos de reconocimiento de que están fuera del alcance de la especificación.

3.7.4.7.4 Contraseña de acceso

Este es una contraseña de nivel de seguridad/acceso como un mutuo acuerdo entre el emisor y el receptor. Si el chequeo de seguridad falla la transmisión será abortada en el emisor será notificado de una violación de acceso.

3.7.4.7.5 Nombre del emisor o identificador

El propósito de este campo es definir el fabricante o el instrumento clínico. Usando delimitadores de componentes y/o repetidores este campo podrá reflejar la revisión del software o firmware, múltiples instrumentos pueden estar disponibles en una línea.

3.7.4.7.6 Dirección del emisor

Este valor de texto deberá contener la dirección del emisor como se definió en 3.7.4.6.9.

3.7.4.7.7 Campo reservado

Este campo está actualmente sin uso pero reservado para futuro uso.

3.7.4.7.8 Número telefónico del emisor

Este campo identifica un número telefónico para comunicación por voz con el emisor como se especifica en 3.7.4.6.7.

3.7.4.7.9 Características del emisor

Este campo contiene cualquier característica del emisor tal como, paridad, suma de comprobación, protocolos opcionales, etc. Necesarios para establecer un vínculo de comunicación con el remitente.

3.7.4.7.10 Identificador del receptor

Este valor de texto incluye el nombre u otro identificador del receptor. Su propósito es verificar que la transmisión es necesaria para el receptor.

3.7.4.7.11 Comentario o instrucciones especiales

Este campo de texto debe contener cualquier comentario o instrucción especial relacionando los registros subsecuentes para ser transmitidos

3.7.4.7.12 Identificador de proceso

Indica como este mensaje será procesado:

- P – Producción: Tratar el mensaje como un aviso activo para ser completado de acuerdo al tratamiento estándar.
- T – Entrenamiento: Mensaje iniciado por un entrenador y no deberá tener un efecto en el sistema.
- D – Depuración: Mensaje iniciado para el propósito de un programa de depuración.
- Q – Control de calidad: Mensaje iniciado para el propósito de transmitir controles de calidad o datos regulatorios.

3.7.4.7.13 Número de versión

Este valor identifica la versión de nivel de la especificación.

3.7.4.7.14 Fecha y hora del mensaje

Este campo contiene la fecha y hora que el mensaje fue generado usando el formato especificado en 3.7.4.6.6.

3.7.4.8 Registro de información de pacientes

Cada línea del registro de pacientes debe iniciar con un tipo de registro y termina con un retorno de carro.

3.7.4.8.1 Tipo de registro

El carácter P identifica el registro como un registro de paciente.

3.7.4.8.2 Número de secuencia

Para el primer paciente transmitido, o debe ser ingresado, para el segundo, 2, hasta el último.

3.7.4.8.3 Práctica de identificación asignado al paciente

Este identificador debe ser un identificador único asignado y usado por la práctica para identificar al paciente y regresar sus resultados de las pruebas.

3.7.4.8.4 Identificador de paciente asignado al laboratorio

Este identificador debe ser un número de procesamiento único asignado al paciente por el laboratorio.

3.7.4.8.5 Identificación del paciente no. 3

Este campo debe ser opcional, usado para identificadores adicionales, universales o de fabricantes (Tal como número de seguro social), como acuerdo entre el receptor y emisor.

3.7.4.8.6 Nombre del paciente

El nombre del paciente deberá ser presentado en el siguiente formato: apellido, nombre, segundo nombre o inicial, sufijo, y título, y cada uno de esos componentes debe ser separado por un delimitador de componente como se describe en 3.7.4.4.

3.7.4.8.7 Nombre de soltera de la madre

Este campo es opcional y podrá ser requerido para distinguir entre pacientes con la misma fecha de nacimiento y apellido cuando los archivos de registro son muy largos. Este nombre debe ser presentado como el nombre de soltera de la madre, por ejemplo, Martínez.

3.7.4.8.8 Fecha de nacimiento

La fecha de nacimiento debe ser presentada en el formato estándar especificado en 3.7.4.6.6.

3.7.4.8.9 Sexo del paciente

Este campo deberá ser representado por M (Masculino), F (Femenino), U (Desconocido).

3.7.4.8.10 Paciente origen de raza étnica

Los siguientes ejemplos pueden ser usados:

- W – Blanco.
- B – Negro.
- O – Asiático/isla del pacifico.
- NA – Nativo americano/nativo de Alaska.
- H – Hispánico.

3.7.4.8.11 Dirección del paciente

Este valor de texto debe registrar la dirección del paciente como se define en 3.7.4.7.6.

3.7.4.8.12 Campo reservado

Este campo es reservado para futuras expansiones.

3.7.4.8.13 Número telefónico del paciente

El número telefónico del paciente está en el formato definido en 3.7.4.6.7.

3.7.4.8.14 Identificador de atención medica

Este campo debe identificar los cuidados físicos para el paciente como nombre, o código, como un acuerdo entre el emisor y receptor. Identificadores nombres, ambos deberán ser separados por el delimitador de componentes como se especificó en 3.7.1.7.

Múltiples nombres (por ejemplo, órdenes, atención física, análisis físico, etc.) debe ser separado por el delimitador de repetición.

3.7.4.8.15 Campo especial 1

Un campo de texto opcional para uso del vendedor (cada laboratorio usa este campo indistintamente).

3.7.4.8.16 Campo especial 2

Un campo de texto opcional para uso del vendedor.

3.7.4.8.17 Altura del paciente

Un campo numérico opcional que contiene la altura del paciente. Las unidades por defecto son centímetros. Si la medida en términos de otra unidad, la unidad deberá ser transmitida como se especifica en 3.7.1.4.

3.7.4.8.18 Peso del paciente

Un campo numérico opcional que contiene el peso del paciente. Por defecto las unidades son en kilogramos. Si la medida en términos de otra unidad, por ejemplo, libras, el nombre de la unidad debe también ser transmitida como se especifica en 3.7.1.4. La información de altura y pesos no es actualmente requerida por todos los laboratorios.

3.7.4.8.19 Diagnóstico del paciente conocido o presunto diagnóstico

Este valor deberá ser ingresado como en el código ICD-9 o texto libre. Si múltiples diagnósticos son registrados, ellos deben ser separados por delimitadores de repetición.

3.7.4.8.20 Medicamentos activos del paciente

Los medicamentos activos o aquellos sospechosos situaciones de sobredosis. El nombre genérico debe ser usado. Este campo es de uso en la interpretación de los resultados clínicos.

3.7.4.8.21 Dieta del paciente

Este campo de texto libre es opcional y deberá ser usado para indicar que condiciones afectan los resultados de las pruebas.

3.7.4.8.22 Campo de práctica no. 1

Es un campo de texto libre para usarlo para prácticas, el texto de transmisión opcional será devuelto con los resultados.

3.7.4.8.23 Campo de práctica no. 2

Similar al campo 3.7.4.8.22.

3.7.4.8.24 Fecha de admisión y de alta

Estos valores deben ser representados como se especifica en 3.7.4.6.6. La fecha de alta seguida de la fecha de admisión es separada por un delimitador de componente.

3.7.4.8.25 Estado de admisión

Este valor deberá ser representado por la siguiente lista mínima o por un acuerdo de extensión entra el emisor y el receptor, la lista es la siguiente:

- OP – Ambulatorio.
- PA – Pre admitido.
- IP – Hospitalización.
- ER – Sala de emergencia.

3.7.4.8.26 Lugar

Este valor de texto debe reflejar el lugar de la clínica general, unida de enfermería, sala, cama, o ambos del paciente en términos de acuerdo por el emisor y receptor.

3.7.4.8.27 Código de diagnóstico natural o alternativo y clasificador.

Identifica la clase de códigos o clasificadores que son transmitidos.

3.7.4.8.28 Código de diagnóstico alternativo y clasificador

Si múltiples códigos son incluidos, estos deberán ser separados por un delimitador de repetición.

Códigos individuales pueden ser seguidos por un descriptor optativo de prueba y deberá ser separado por delimitadores de componentes.

3.7.4.8.29 Religión del paciente

Cuando es necesario, este valor debe incluir la religión del paciente. Códigos o nombres pueden ser enviados como acuerdo entre el emisor y receptor. Nombres completos de religiones pueden también ser enviados como sea requerido.

3.7.4.8.30 Estado marital

Cuando es requerido, este valor debe indicar el estado marital del paciente como es mostrado a continuación:

- M – Casado.
- S – Soltero.
- D – Divorciado.
- W – Viudo.
- A – Separado.

3.7.4.8.31 Estado de aislamiento

Códigos de aislamiento indican las precauciones que se deben aplicar para proteger el paciente o personal frente a la infección.

Múltiples precauciones pueden ser listadas separadas por un delimitador de repetición. Texto completo sobre las precauciones también puede ser enviado.

3.7.4.8.32 Idioma

El valor de este campo indica el idioma primario del paciente. Este puede ser necesitado cuando el paciente habla con fluidez en el idioma local.

3.7.4.8.33 Servicio hospitalario

Este valor indica el actual servicio hospitalario asignado al paciente. Tanto como códigos y texto completo pueden ser enviados cuando es separado por el delimitador de componente.

3.7.4.8.34 Institución hospitalaria

Este valor indica la institución hospitalaria actualmente asignada al paciente. Tanto como códigos y texto completo pueden ser enviados cuando es separado por el delimitador de componente como se definió en 3.7.4.4.

3.7.4.8.35 Categoría de dosis

Este valor indica el grupo de dosis del paciente. Subcomponentes de este campo puede ser usado para definir subgrupos de dosis.

3.7.4.9 Registro de órdenes de prueba

El registro de orden de prueba define los atributos de una solicitud en particular para los servicios de un instrumento clínico. Y contiene toda la información de los especímenes. Un registro de orden deberá ser generado por el sistema de cómputo para solicitar una prueba, batería, o conjunto de pruebas dadas. La información contenida en un registro de la orden por lo general se aplicara en una sola muestra. Sin embargo, no hay necesariamente una relación uno a uno entre la muestra y la prueba ordenada. Diferentes tipos de baterías de prueba por lo general se ordenaran en los registros de orden diferente, incluso se puede realizar en una sola muestra.

3.7.4.9.1 Múltiples órdenes

Más de una prueba o batería de pruebas puede ser ordenada en un solo registro de orden, usando delimitadores de repetición entre las pruebas individuales ordenadas en ese registro. Sin embargo, en algunos casos, todos los atributos almacenados dentro del registro de orden deberán ser los mismos para todas las pruebas ordenadas dentro del registro. Esto es, si uno desea ordenar una prueba como una prueba inmediata y otra como una prueba de rutina, dos registros de órdenes de prueba deberán ser requeridos. En el caso que la batería de pruebas requiera más que una muestra, como ocurre con la depuración de creatinina, la información sobre cada una de las muestras de prueba se puede incluir en un solo registro de orden, para identificar múltiples muestras se usa el delimitador de repetición dentro el campo de identificación de muestra.

3.7.4.9.2 Tipo de registro

El carácter asignado al registro de orden debe ser O.

3.7.4.9.3 Número de secuencia

Este campo debe ser representado como se describe en 3.7.4.7.13.

3.7.4.9.4 Identificador de muestra

Este campo de texto debe representar un identificador único para la muestra asignada por el sistema de cómputo y retornada por el instrumento. Si la muestra tiene múltiples componentes más la identificación de cultivos derivados de ella, estos componentes identificadores serán seguidos por el identificador de la muestra y será separada por delimitadores de componentes.

3.7.4.9.5 Muestra de identificación de instrumentos

Este campo de texto debe representar un identificador único asignado por el instrumento, si difiere del identificador del sistema de cómputo, y regresara con resultados para usar como referencia a cualquier resultado.

3.7.4.9.6 Identificador de prueba universal

Este campo debe usar el identificador de prueba universal como se describe en 3.7.4.6.1.

3.7.4.9.7 Prioridad

Los códigos de prioridad de las pruebas son los siguientes:

- S – Inmediata.
- A – Tan pronto como sea posible.
- R – Rutina.
- C – Devolución de llamada.
- P – Preoperatoria.

Si más que un código de prioridad aplica, estos deberán ser separados por el delimitador de repetición.

3.7.4.9.8 Fecha y hora de solicitud/orden

El contenido de este campo debe ser representado como se especifica en 3.7.4.6.6.

3.7.4.9.9 Fecha y hora de recolección de muestra

Este campo debe representar el tiempo actual cuando la muestra fue recolectada u obtenida.

3.7.4.9.10 Tiempo final de recolección

Este campo debe contener la fecha y hora final de una recolección de muestra., como puede ser recolección de orina de 24-hrs, este valor debe ser especificado como se describe en 3.7.4.6.6.

3.7.4.9.11 Recolección de volumen

Este valor debe representar el volumen total de la muestra tal como orina, u otras recolecciones a granel. Por defecto la unidad de medida es en milímetros. Cuando las unidades son explícitamente representadas estas deberán ser separadas del valor numérico por el delimitador de componente.

3.7.4.9.12 Código de acción

Este campo debe indicar la acción que será tomada con respecto al espécimen que acompaña o precede la muestra. Los siguientes códigos deben ser usados:

- C – Cancelar solicitud.
- A – Añadir pruebas a la solicitud.
- N – Nuevas solicitudes acompañando nuevas muestras.
- P – Muestra pendiente.

- L – Reservado.
- X – Muestra o prueba en proceso.
- Q – Tratar a la muestra como un control de calidad.

3.7.4.9.13 Código de peligro

Representa cualquier prueba o un código indicando algún riesgo especial asociado con la muestra, por ejemplo, paciente con hepatitis o VIH.

3.7.4.9.14 Información clínica relevante

Cualquier información adicional sobre la muestra deberá ser proporcionada aquí.

3.7.4.9.15 Fecha y hora de la recepción de la muestra

Este campo es opcional y debe contener la actual fecha y hora registrado en el laboratorio.

3.7.4.9.16 Descriptor de la muestra

Este campo podrá contener dos elementos separados, tipo y origen de la muestra.

3.7.4.9.17 Tipo de muestra

El tipo de cómo es la muestra, por ejemplo, sangre, orina, cabello etc.

3.7.4.9.18 Origen de la muestra

Este es siempre el segundo componente del descriptor de la muestra y es usado especialmente para determinar el origen de la muestra de un cuerpo humano, por ejemplo, (brazo izquierdo, brazo derecho).

3.7.4.9.19 Solicitud de médico

Este campo debe contener el nombre de la solicitud de médico.

3.7.4.9.20 Número telefónico del médico

Este campo debe contener un número telefónico del médico solicitante. Usa el formato descrito en 3.7.4.6.7.

3.7.4.9.21 Campo de usuario no. 1

Texto enviado por el solicitante, deberá ser regresado por el remitente junto con la respuesta.

3.7.4.9.22 Campo de usuario no. 2

Similar al punto 3.7.4.9.22.

3.7.4.9.23 Campo de laboratorio no. 1

Es un campo opcional para ser usado por cualquier laboratorio

3.7.4.9.24 Campo de laboratorio no. 2

Similar al punto 3.7.4.9.23.

3.7.4.9.25 Fecha y Hora del reporte de resultados o última modificación

Este campo es usado para indicar la fecha y hora de los resultados de la orden que se compone en un reporte, o en este mensaje. Las fechas y horas deben ser realizadas con formato 3.7.4.6.7.

3.7.4.9.26 Instrumento de carga en el sistema informático

Este campo contiene el cargo de la factura o de referencia contable.

3.7.4.9.27 Sección de instrumentos de identificación

Este identificador podrá denotar la sección del instrumento donde la prueba ha sido realizada. En el caso donde múltiples instrumentos están en una sola línea o las pruebas han sido movidas de un instrumento a otro, este campo mostrara que instrumento realizó la prueba.

3.7.4.9.28 Tipos de reportes

Los siguientes códigos deben ser usados:

- O – Registro de orden.
- C – Corrección de resultados previos.
- P – Resultados preliminares.
- F – Resultados finales.
- X – Orden sin realizar o cancelada.
- I – Instrumento pendiente.
- Y – No hay orden o registro para la prueba.
- Z – Ningún registro para este paciente.
- Q – Responde la consulta.

3.7.4.9.29 Campo reservado

Este campo se encuentra reservado para futuras expansiones.

3.7.4.9.30 Ubicación de la recolección de muestras

Este campo identifica el lugar de recolección de la muestra si es diferente de la ubicación del paciente.

3.7.4.9.31 Bandera de infección nosocomial

Este campo es usado para propósitos de reportes epidemiológicos y es mostrado si el organismo identificado es el resultado de una infección nosocomial.

3.7.4.9.32 Muestra de servicio

En el caso donde un servicio individual puede aplicar a la muestra recolectada, y el servicio es diferente del servicio registrado para el paciente, este campo puede ser usado para definir el servicio específico responsable de la recolección.

3.7.4.9.33 Muestra de institución

En algunos casos donde la muestra puede que haya sido recolectada en una institución, o la institución es diferente de la institución registrada del paciente, este campo puede ser usado para registrar la institución donde se recolectó la muestra.

3.7.4.10 Registro de resultados

El registro de resultado debe incluir los siguientes campos:

3.7.4.10.1 Tipo de registro

Definido como el carácter P.

3.7.4.10.2 Número de secuencia

Debe ser asignado como está descrito en 3.7.1.4.

3.7.4.10.3 Identificador de prueba universal

Este campo debe ser usado el identificador de prueba universal definido como 3.7.4.6.1.

Dato o valor de medida. Ya sea numérico, texto, o valor de código, los datos deben ser registrados en texto ASCII. Si los datos contienen elementos calificadores deberán ser separados por el delimitador de componente.

3.7.4.10.4 Unidades

La abreviación de las unidades para resultados numéricos deberá aparecer aquí.

3.7.4.10.5 Rangos de referencia

Este campo debe contener referencias de los rangos usados por las unidades, representando un límite inferior o superior, se describe como texto libre.

3.7.4.10.6 Bandera de resultados anormales

Este campo debe indicar el estado de la normalidad del resultado. Los caracteres para representar los resultados anormales son los siguientes:

- N – Normal
- A – Anormal

3.7.4.10.7 Naturaleza de la prueba anormal

Este tipo de prueba normal realizado debe usar la siguiente representación:

- A – Denota que la población.
- S – Población basada en raza.
- R – Una población basada en raza.
- N – Implica que un rango genérico normal ha sido aplicado a todas las muestras.

Una combinación es posible, por ejemplo, si las pruebas en una población está basada en edad, sexo y raza, debe denotarse como: (A\S\R) y deberá ser transmitido.

3.7.4.10.8 Estado de los resultados

Los siguientes códigos deben ser usados:

- C – Corrección de resultados previamente transmitidos.
- P – Resultados preliminares.
- F – Resultados finales.
- X – Orden no pudo ser realizada.
- I – En equipo, resultados pendientes.
- S – Resultados parciales.
- M – Implica múltiples resultados.
- R – El resultado ha sido previamente transmitido.
- N – El registro de resultado contiene la información necesaria para iniciar una nueva orden.

Fecha de cambio en los valores o unidades de los instrumentos.

3.7.4.10.9 Identificación del operador

El primer componente identifica el operador del instrumento que realizó la prueba. El segundo componente identifica el verificador de la prueba.

3.7.4.10.10 Fecha y hora del inicio de la prueba

Indica la fecha y hora del instrumento que inicio los resultados reportados. Indicados con el formador descrito en 3.7.4.6.6.

3.7.4.10.11 Fecha y hora de la prueba completada

Indica la fecha y hora del instrumento que completó los resultados reportados. Descritos con el formato 3.7.4.6.6.

3.7.4.10.12 Identificación del instrumento

Identifica el instrumento o sección de instrumento que realizó la medida en particular.

3.7.4.11 Registro de comentarios

El registro de comentarios puede ser insertado en cualquier lugar excepto después del registro de término del mensaje. Cada registro de comentario debe aplicar al primer registro que no sea un comentario que le precede.

El registro incluye los siguientes campos:

3.7.4.11.1 Tipo de registro

El tipo de registro debe ser denotado por C.

3.7.4.11.2 Número de secuencia

Como se definió en 3.7.1.4.

3.7.4.11.3 Origen del comentario

Punto de origen del comentario:

- P – Práctica.
- L – Sistema informático.
- I – Sistema de instrumento clínico.

3.7.4.11.4 Texto del comentario

Cuando los códigos/mnemónicos de comentarios son usados, el código deber ser enviado primero, seguido -si se desea- por el texto del comentario y separado por un delimitador de componentes descrito en 3.7.4.4.3.

3.7.4.11.5 Tipo de comentario

Los siguientes códigos pueden ser usados para calificar el registro de comentario:

- G – Comentario genérico o texto libre.
- T – Comentario de nombre de la prueba.
- P – Comentario de prueba positiva.
- N – Comentario de prueba negativa.
- I – Comentario de bandera del instrumento.

3.7.4.12 Registro de solicitud de información

El registro de solicitud de información es usado por cualquier instrumento clínico o sistema de cómputo para solicitar información de forma remota a un sistema recíproco.

3.7.4.12.1 Tipo de registro

Codificado como Q.

3.7.4.12.2 Número de secuencia

Como fue descrito en 3.7.1.4.

3.7.4.12.3 Rango inicial de un número de identificación

Este campo podría contener tres o más componentes para definir un rango de criterio de selección para pacientes/muestras/fabricantes. El primer componente es el identificador del paciente asignado por el sistema informático. El segundo componente, de igual manera asignado por el sistema informático, es el identificador de la muestra. Por último el tercer componente es el identificador del fabricante y es usado para solicitar información de sub-resultados. Estos componentes de dependientes de la posición y los componentes son separados por el delimitador de repetición.

3.7.4.12.4 Rango final de un número de identificación

Similar ha como se describe en 3.7.4.12.3.

3.7.4.12.5 Identificador de prueba universal

Este identificador debe seguir el formato definido en 3.7.4.6.1.

3.7.4.12.6 Límite de tiempo de la solicitud

Específica si los límites de fecha y hora especificadas en 3.7.4.6.1 se refieren a la fecha de la muestra recolectada u ordenada:

- R – Indica que la fecha es de resultado.
- S – Indica que la fecha de recolección.
- Vacío – Asume que la fecha es de resultado.

3.7.4.12.7 Inicio de la fecha y hora de solicitud de resultados

Este campo debe representar cualquier inicio de fecha y hora para cuando los resultados han sido solicitados. Este campo podrá contener múltiples fechas individuales separadas por un delimitador de repetición. Cada fecha y hora se especifica en 3.7.4.6.6.

3.7.4.12.8 Fin de la fecha y hora de solicitud de resultados

Especifica la fecha y hora final de la solicitud de resultados, similar a 3.7.4.6.6.

3.7.4.12.9 Nombre del médico solicitante

Este campo identifica el médico que solicitó los resultados. La identificación es definida con el formato descrito en 3.7.1.4.

3.7.4.12.10 Número telefónico del médico solicitante

Campo descrito en 3.7.4.6.6.

3.7.4.12.11 Campo de usuario no. 1

Campo definido para el usuario.

3.7.4.12.12 Campo de usuario no. 2

Campo definido para el usuario.

3.7.4.12.13 Códigos del estado de la solicitud de información

Los siguientes códigos deben ser usados:

- C – Corrección de resultados previamente transmitidos.
- P – Resultado preliminares.
- F – Resultados finales.
- X – Resultados que no se realizaron o solicitud cancelada.
- I – Solicitud de resultados pendientes.
- S – Solicitud parcial/resultados sin finalizar.
- M – Múltiples resultados.
- R – Este resultado ha sido previamente transmitido.
- A – Abortado/Cancelado.
- N – Nueva solicitud o solo resultados editados.
- O – Solicitud de órdenes de prueba y datos demográficos solamente.
- D – Solicitud de datos demográficos solamente.

3.7.4.13 *Registro de mensaje terminador*

Este es el último registro en el mensaje. Un registro de cabecera podrá ser transmitido después de este registro significando el inicio de un segundo mensaje.

3.7.4.13.1 Tipo de registro

Codificado como L.

3.7.4.13.2 Número de secuencia

Para este tipo de registro el valor siempre deberá ser 1.

3.7.4.13.3 Código terminador

Proporciona una explicación del código del final de la sesión

- Nil/N – Terminación normal.
- T – Abortado por el emisor.
- R – Solicitud abortada por el receptor.
- E – Error de sistema desconocido.
- Q – Error en la última solicitud de información.
- I – No existe información disponible de la última consulta.
- F –Última solicitud de información procesada.

F, I, Q terminará una solicitud y permitirá procesar un nuevo registro de solicitud.

3.8 Estructura de la comunicación

Diagrama de la comunicación básica mostrado en la Ilustración 3-D

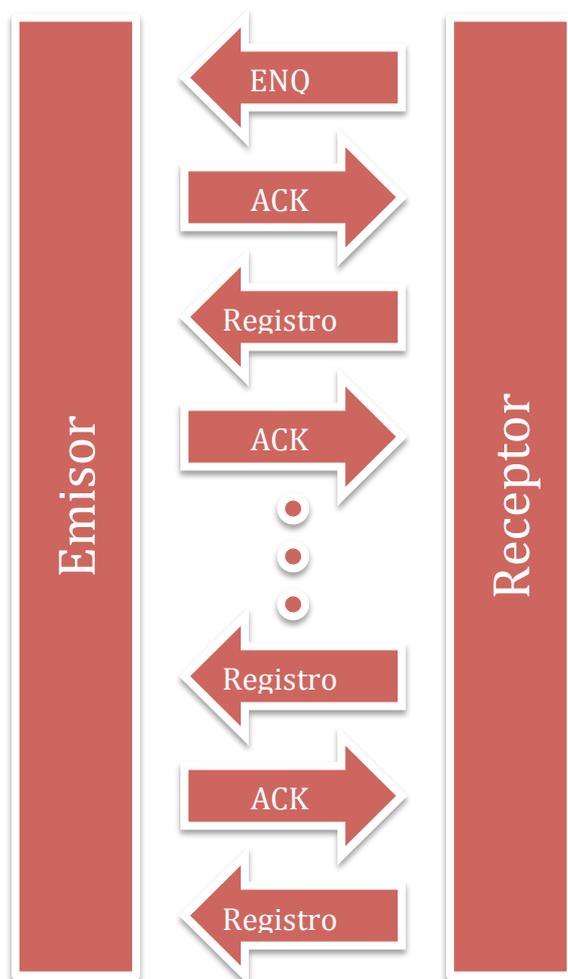


Ilustración 3-D Comunicación básica

3.9 Estructura de un paquete

La estructura de un paquete se muestra en la siguiente ilustración:

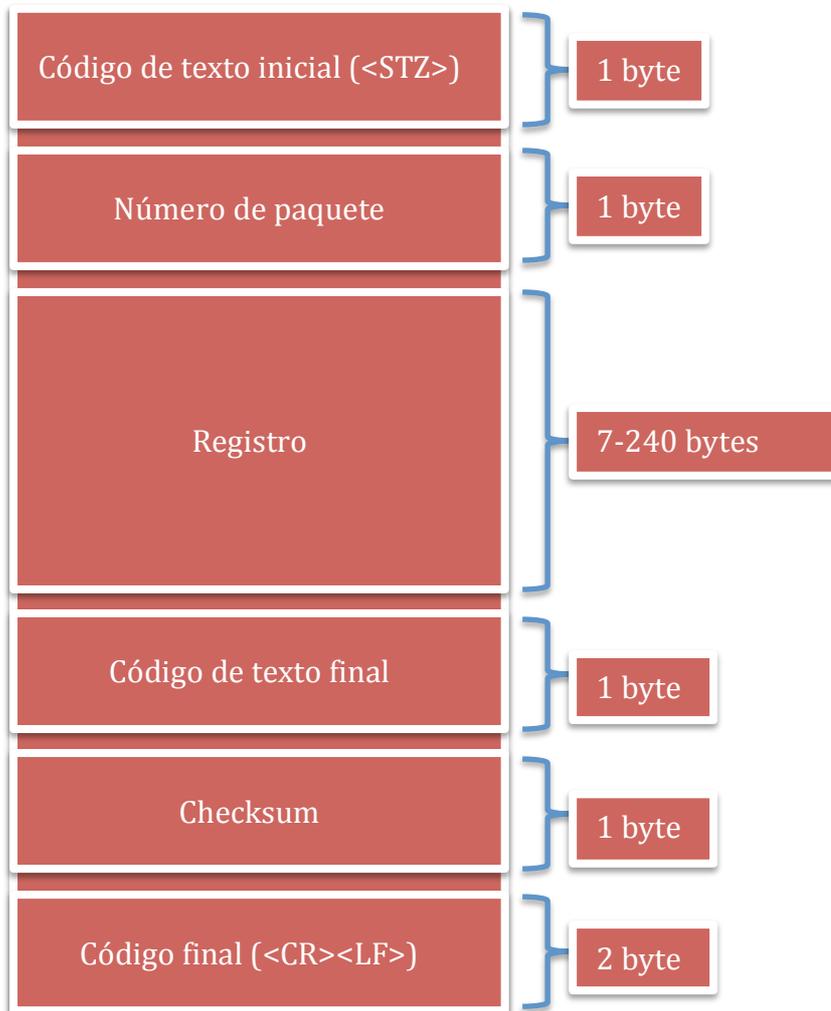


Ilustración 3-E Estructura de un paquete

El protocolo ASTM E1394-97 use cuatro capas jerárquicas para el procesar la comunicación de datos mostrado en la Ilustración 3-G Capas de comunicación, y son:

1. *Capa de aplicación*: Virtualmente intercambia mensajes.
2. *Capa de presentación*: Virtualmente intercambia registros. Un mensaje consiste de múltiples registros.
3. *Capa enlace de datos*: Virtualmente intercambia *frames* (marcos), Un registro consiste de uno o más *frames*.

4. *Capa física*: Físicamente intercambia códigos ASCII a través de la interfaz serial. Un *frame* consiste de máximo de 240 registros de letras (bytes) y 7 bytes de delimitadores y verificadores.

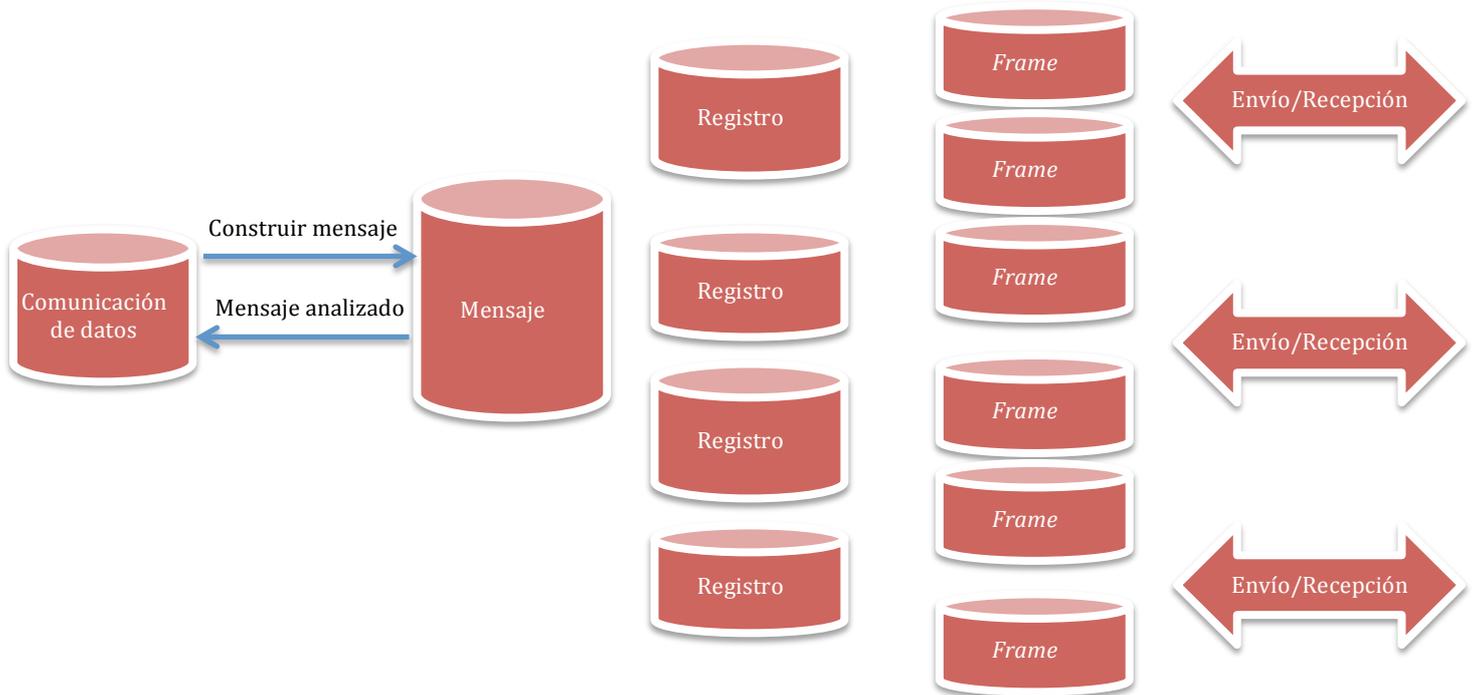


Ilustración 3-F Paquetes de comunicación

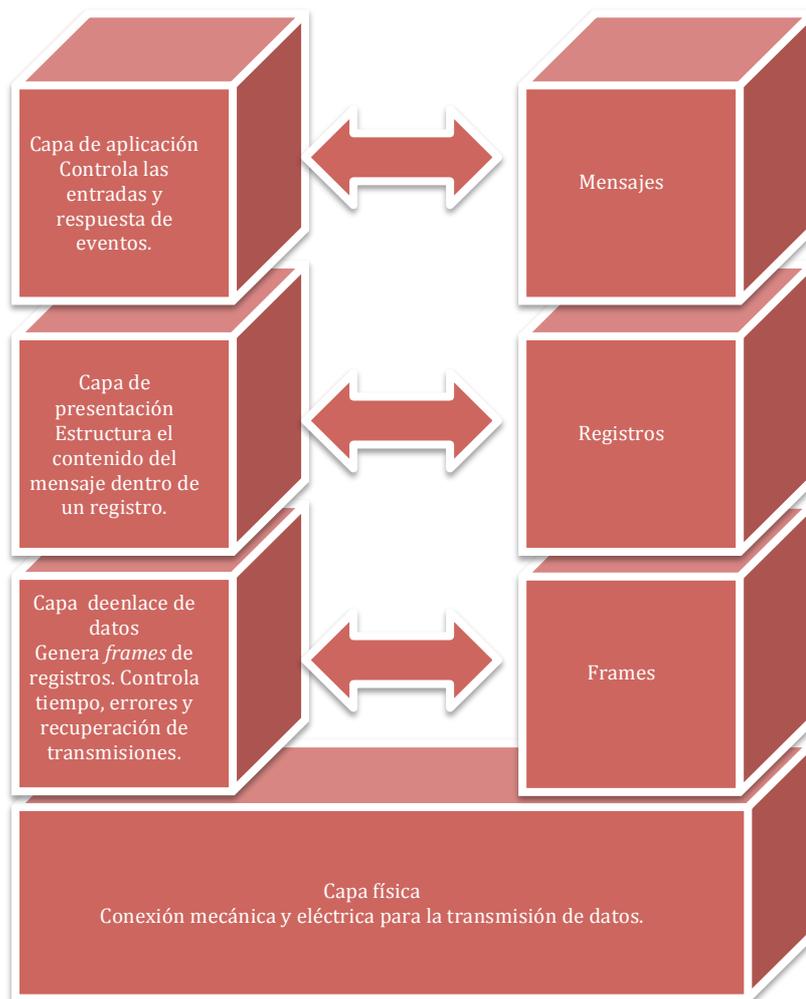


Ilustración 3-G Capas de comunicación

La definición del protocolo E1394-97 se encuentra dividida en protocolo de alto y bajo nivel. El protocolo de bajo nivel define los métodos de la comunicación de datos por *frames*. El protocolo de alto nivel define la estructura del mensaje, proceso de reenvío, estructura del registro y tipos de registro así como sus usos.

- **ASTM E1381-94 *Low Level Protocol*:**
Especificación para el protocolo de bajo nivel para transmitir mensajes entre equipo clínico y sistema de cómputo.
- **ASTM E1394-91 *High Level Protocol*:**
Especificación del estándar para transferir información entre equipo clínico y sistema de cómputo.

3.10 Descripción de términos específicos para ASTM

Para clarificar algunos términos este capítulo define algunas palabras específicas de acuerdo al estándar ASTM E1394-97.

- Mensaje.- Un cuerpo textual de información. Ejemplo,: El resultado de todas las ordenes para un grupo de pacientes y datos relacionados.
- Batería.- Un grupo de ordenes juntas, por ejemplo, una batería de admisión. El término *batería* es usado en el presente trabajo de investigación como sinónimo de perfil o panel. Los elementos de pruebas dentro de una batería podrían ser característicos de un solo sistema fisiológico, por ejemplo, pruebas de funciones vitales, o por el contrario, podrían ser de un múltiple conjunto de sistemas fisiológicos. La batería es una simple convención para que un usuario pueda ordenar múltiples pruebas especificando un solo nombre.
- Prueba.- Una determinación de un solo análisis o combinación de valores de otras determinaciones u observaciones.
- Registro.- Un agregado o campo describiendo un aspecto del mensaje completo. Ejemplo: El registro de información del paciente en un “mensaje de medida de datos” contiene información relacionada al paciente.
- Campo.- Un atributo específico de un registro que podría contener agregados de elementos de datos referenciando atributos básicos.
- Campo repetidor.- Un solo elemento de datos que expresa una duplicación del campo. Usado para demografía, solicitudes, ordenes entre otros.
- Campo componente.- Un solo elemento de dato que expresa un agregado de información básica.

3.10.1 Capa de aplicación

En la capa de aplicación entre el instrumento (es decir, el analizador) y el anfitrión (sea el sistema de cómputo) el intercambio de mensajes. El nivel alto del protocolo define la estructura de estos mensajes, pero no puede definir los eventos de activación de los mensajes o las reacciones a los mensajes. Estas especificaciones son parte de la aplicación individual del protocolo, por lo general dependen de las características del instrumento (y anfitrión).

3.10.1.1 Tipos de eventos

- Emisor (equipo clínico) – Receptor (equipo de cómputo).

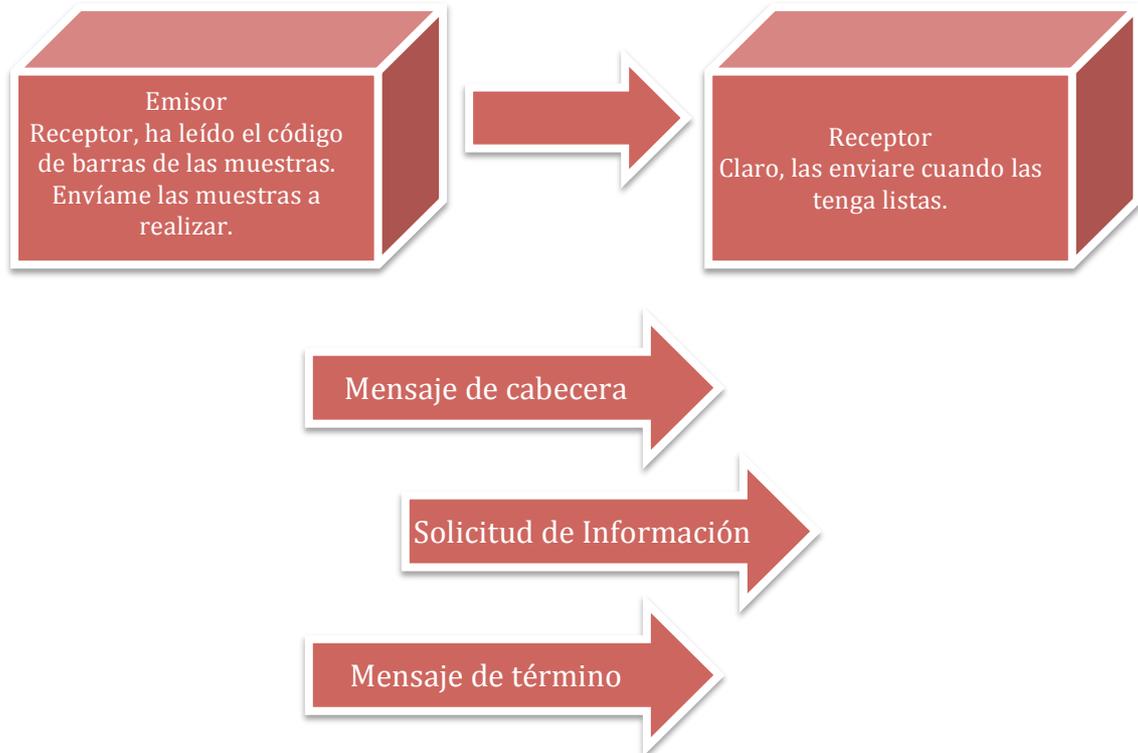


Ilustración 3-H Evento 1 Equipo clínico - Equipo de cómputo

Ejemplo: El instrumento ha comprobado que la muestra de los códigos de barras y las necesidades de la lista de trabajo (lista de las pruebas) para llevar a cabo con estas muestras (selección de la prueba).

Cuando el papel de emisor lo toma el equipo clínico, este automáticamente lee los tubos con códigos de barras de las muestras clínicas que serán procesadas, una vez cumplida esta fase, el emisor solicitara al emisor una lista de trabajo para poder procesar las muestras clínicas dicha lista de trabajo contiene los datos del paciente así como su respectivas pruebas a procesar por el equipo clínico.

- Emisor (equipo de cómputo) – Receptor (equipo clínico)

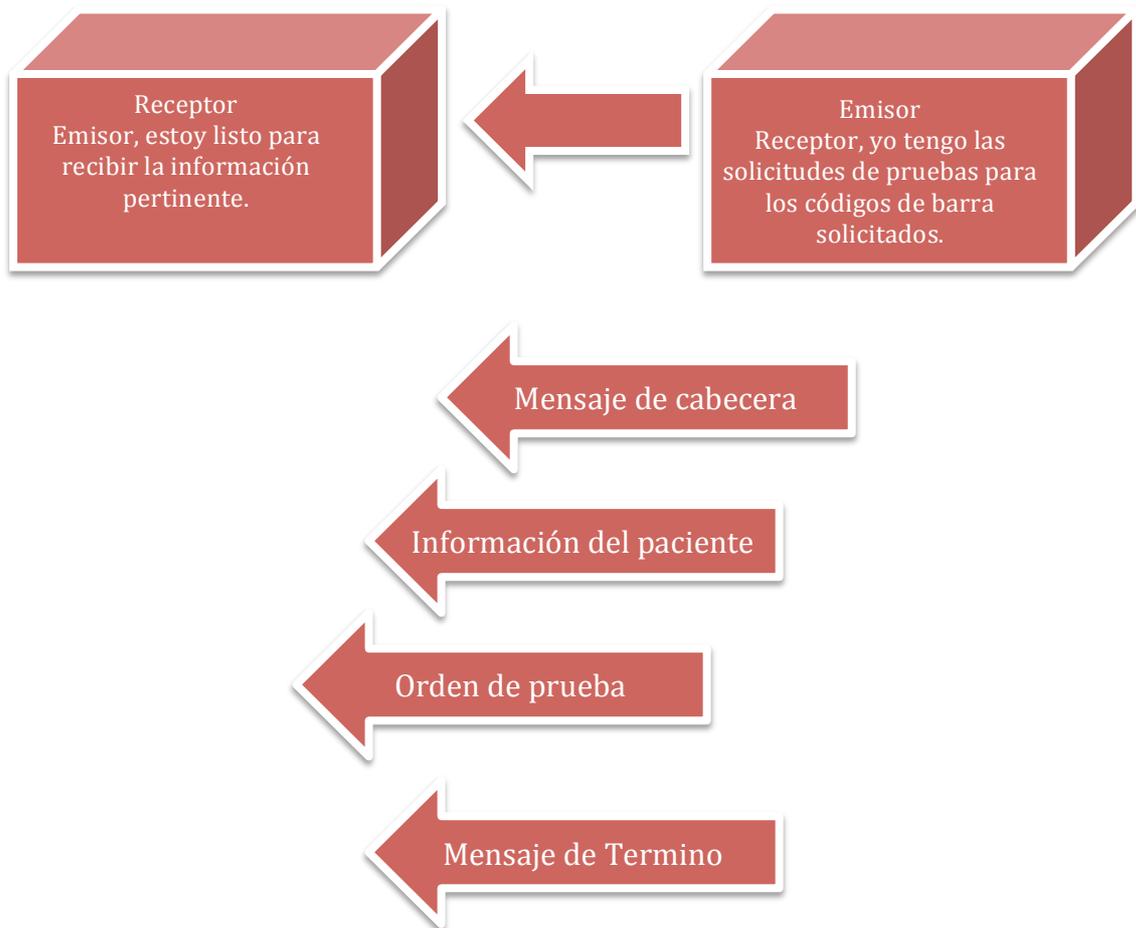


Ilustración 3-1 Evento 2 Equipo de cómputo - Equipo clínico

Ejemplo: El anfitrión recibe la solicitud para la selección de prueba. Reacciona mediante el envío de los datos mediante un mensaje de pedido de prueba desde el anfitrión.

Este tipo de evento, el rol del equipo ahora se enfoca en receptor, por lo tanto el emisor que es el equipo clínico envía toda la información pertinente de las pruebas a realizar por el receptor. El evento se lanzará independientemente de que en el evento uno se hayan leído las muestras clínicas.

- Emisor (equipo de cómputo) – Receptor (equipo clínico)

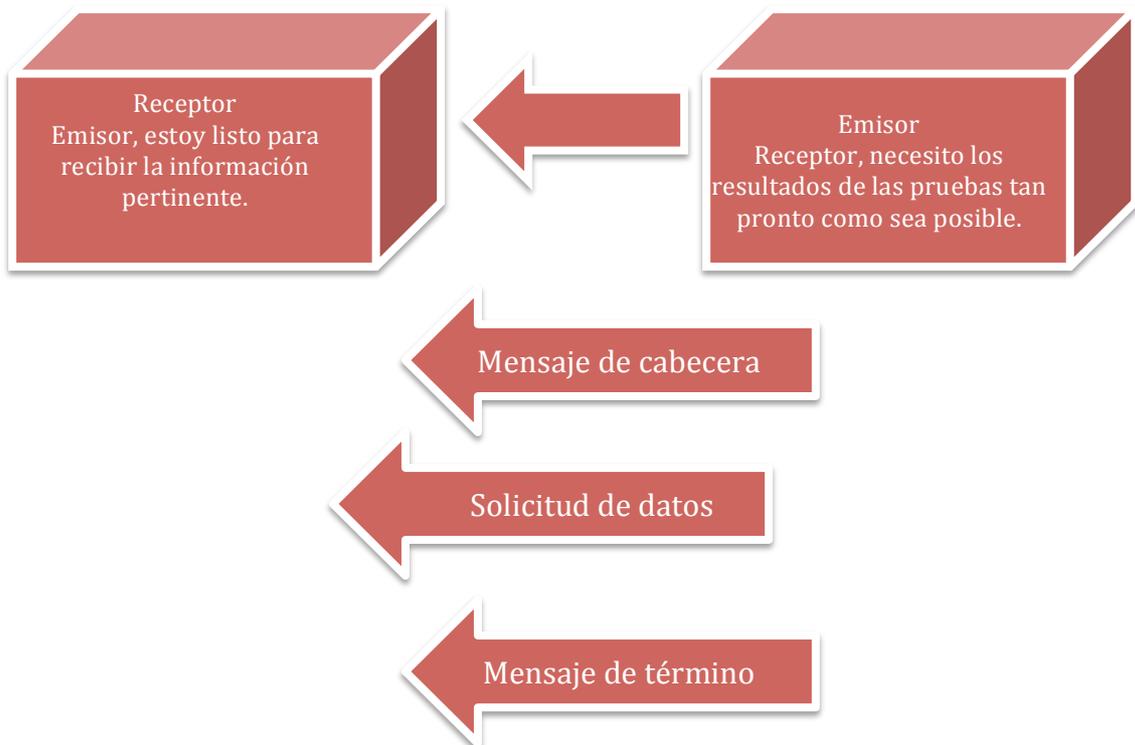


Ilustración 3-J Evento 3 Equipo clínico - Equipo de cómputo

Ejemplo: El anfitrión necesita los resultados de una muestra específica. Se informa a los instrumentos de ésta necesidad mediante el uso de un mensaje de solicitud de información a los instrumentos. Comentario: Por lo general, con la opción "Carga automática" a todos los resultados de una muestra será enviada a el anfitrión tan pronto como esté listo.

El receptor está en espera de alguna solicitud del emisor, en este nivel el emisor solicita los resultados procesados por el emisor.

- Emisor (equipo clínico) – Receptor (equipo de cómputo)

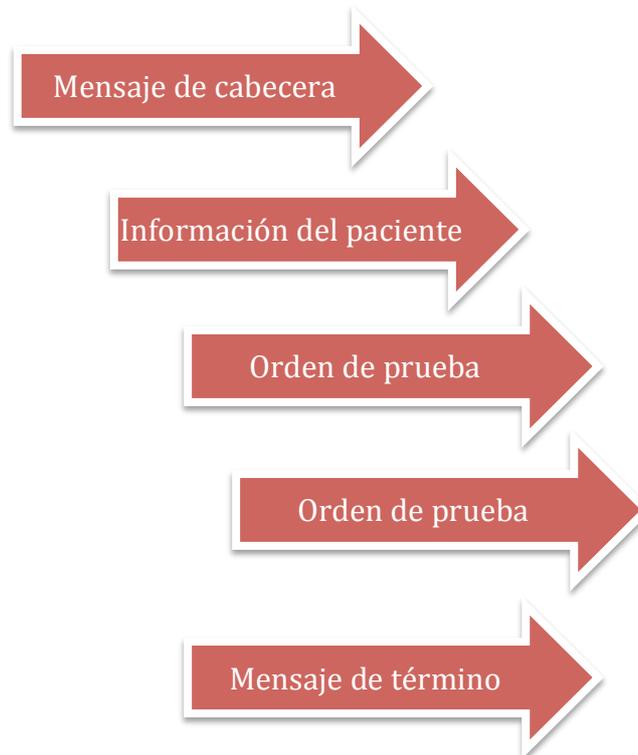
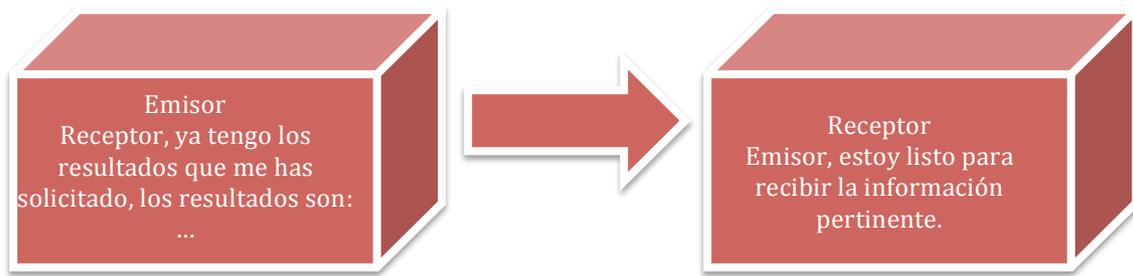


Ilustración 3-K Evento 4 Equipo clínico - Equipo de cómputo

Ejemplo: el instrumento recibe la investigación de resultados de la prueba. Reacciona mediante el envío de los datos mediante un mensaje de resultado del anfitrión.

El equipo clínico enviará todos los resultados solicitados por el equipo de cómputo para que posteriormente sean entregados a sus respectivos pacientes.

3.10.1.1.1 **Eventos relacionados al procesamiento de datos**

El instrumento solicita un mensaje del anfitrión (instrumento está enviando)

- Después de la petición, el instrumento espera la respuesta del anfitrión por un determinado tiempo.
- Cuando el tiempo ha caducado y otro mensaje tiene que ser enviado al instrumento, comenzará a enviar el nuevo mensaje al anfitrión.
- Cuando el instrumento falla en el envío de una petición al anfitrión el instrumento deja de reenviar la solicitud.

Instrumento responde a las peticiones del anfitrión (instrumento está enviando).

- Después de haber recibido la solicitud, el instrumento responde a la petición en un primer momento de prioridad.
- Cuando el instrumento falla en el envío de una petición al anfitrión el instrumento deja de reenviar la solicitud.

Instrumento ofrece al anfitrión con la información (instrumento está enviando).

- Cuando la transferencia de un mensaje de carga falla el instrumento reenvía el mensaje.
- Volver a enviar el mensaje es permitido dos veces. Cuando el reenvío del mensaje falla dos veces el envío de este mensaje de carga se detiene.

Anfitrión solicita información al instrumento (instrumento está recibiendo).

- Después de haber recibido una solicitud del anfitrión el instrumento responde a la petición en primera prioridad.
- Cuando el instrumento no envía el mensaje de respuesta que intenta de nuevo.
- Al volver a enviar la respuesta falla dos veces, la respuesta a la solicitud se detiene.

3.10.1.2 **Detección de errores y recuperación**

Los errores de comunicación se detectan en las diferentes capas. Las capas de tratar de corregir estos errores en su nivel. Si un error no puede corregirse mediante los procedimientos de recuperación de errores de una capa específica, se informa que el error a la capa inmediatamente superior. Por fin, la capa de aplicación tiene que tratar de una corrección de la condición de error detectado o tiene que generar una alarma.

3.10.2 Capa de presentación

El mensaje típico contiene un conjunto de información. Por ejemplo el mensaje de respuesta a una solicitud de resultados de la prueba consta de cuatro niveles (0-3) de la información:

- Los datos relativos a los pacientes que los resultados de las pruebas se transmiten.
- Los datos relativos a las órdenes (baterías de pruebas) los resultados del examen perteneciente.
- Los datos relativos a los resultados de las pruebas de cada batería de pruebas para cada paciente.
- Comentario de datos con información perteneciente a cada resultado.

Como puede se puede apreciar, hay una estructura jerárquica en este conjunto de información. Varios comentarios pueden pertenecer a un solo resultado. Los resultados de varias pruebas puede pertenecer a una orden (perfil solicitado la prueba). Varios pedidos pueden pertenecer a un paciente. Varios pacientes pueden pertenecer a un mensaje transmitido. En lugar de repetir los datos del paciente y los datos para que cada resultado de la prueba, los datos del pedido y el paciente puede ser utilizado como titulares en un esquema jerárquico como el de este documento. Esto reduce la transmisión de datos redundantes y refleja el típico modelo de base de datos relacional se utiliza para almacenar la información del mensaje.

ASTM E1394-97 define los mensajes que consiste en una jerarquía de los registros de varios tipos.

Los registros en el nivel cero contienen información referente a la identificación del remitente y la finalización de la transmisión. Pueden ser vistos como una especie de elaboración de la información relacionada con el mensaje.

Los registros en el nivel uno de la jerarquía contienen información sobre los pacientes individuales.

Los registros en el nivel dos contienen información sobre las solicitudes de pedido de prueba y las muestras.

Los registros en el nivel tres contienen información sobre resultados de la prueba.

Los registros de comentarios se pueden insertar en cualquier nivel en la jerarquía. Un registro de observaciones se refiere siempre al paciente inmediatamente anterior, el orden, el resultado, científicos o de registro de datos del fabricante.

Por lo tanto, si un registro de comentarios iban a seguir una historia clínica del paciente (nivel uno), entonces ese registro comentario podría ser tratado como un segundo nivel récord. Un registro comentario no puede seguir el registro de mensaje de terminación.

Los registros de información de los fabricantes puede ser insertado en cualquier nivel en la jerarquía (con la excepción de nivel 0). Este tipo de registro se refiere siempre al paciente inmediatamente anterior, resultado de la orden, científica o grabar un comentario. Por lo tanto, si un registro de información de los fabricantes iban a seguir una historia clínica del paciente (nivel uno), el registro sería tratado como un nivel de dos registros. Este registro no puede seguir al grabar el mensaje de terminación.

Otros tipos de registro son el registro de solicitud de información y el registro de terminación. El registro de solicitud de información proporciona a la solicitud de la demografía o resultados de pruebas o de los instrumentos clínicos para los pacientes especificada, muestras, pruebas, fechas, etc.

3.10.3 Capa de enlace de datos

Como se describió anteriormente la capa de enlace de datos tiene que realizar los siguientes servicios:

- Enlace y la liberación de la conexión: Establece que el sistema envía y que recibe los sistemas de información.
- Cortar los mensajes en pequeños cuadros de texto, delimitar *frames*: prevé el reconocimiento de los *frames*.
- Sincronizar la transferencia de imágenes y el control secuencial: Mantiene el orden secuencial de la información a través de la conexión.
- La detección de errores: la transmisión de los sentidos y los errores de formato.
- Recuperación de errores: Los intentos de recuperarse de los errores detectados por volver a transmitir los marcos defectuosos o devolver el enlace a un estado neutral, de los errores de otro modo irre recuperable.

El protocolo ASTM E1394-91 de bajo nivel es un simple protocolo de parada y espera. Por ejemplo la información siempre fluye en una dirección a la vez. Respuestas ocurren después de que la información se envía, nunca al mismo tiempo. En contradicción con otros protocolos de comunicación no existe una relación entre amo y esclavo. Ambos instrumentos así como el anfitrión puede iniciar la comunicación.

Para establecer que el sistema envía y que recibe el sistema de información y de garantizar las acciones de emisor y receptor están bien coordinados, hay tres fases distintas en la transferencia de la información:

- Fase de establecimiento (conexión de enlace).
- Fase de transferencia.
- Fase de terminación (libera el enlace).

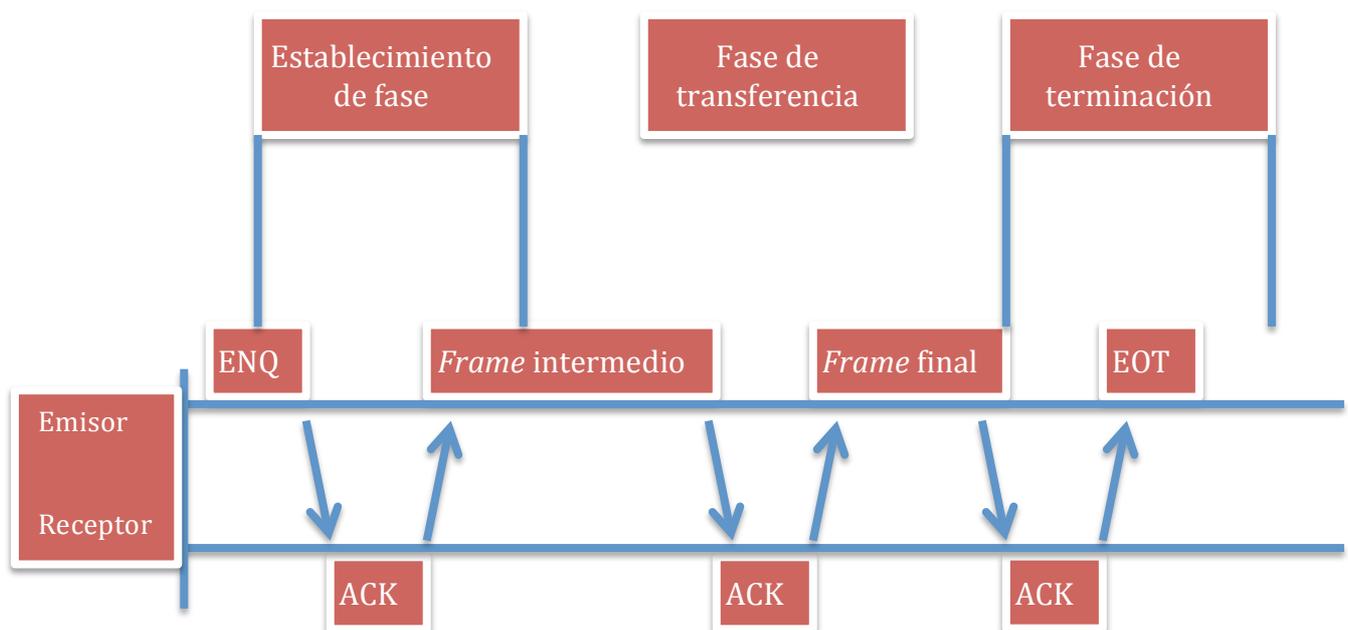


Ilustración 3-L Fases de enlace de datos

3.10.3.1 Fase de establecimiento (conexión de enlace)

En la capa de enlace de datos para ambos interlocutores pueden estar en uno de los tres estados siguientes:

Inactivo: A la espera de convertirse en un receptor

Remitente: Inicia la fase de establecimiento, envía la información de registro en los marcos y termina con la fase de terminación.

- Receptor: Recibe y envía tramas códigos de reconocimiento.

Al mismo tiempo, sólo un dispositivo puede ser emisor y un solo dispositivo puede ser receptor. Ambos dispositivos pueden estar en estado de reposo, al mismo tiempo. Si la capa de presentación solicita a la capa de enlace de datos para la transferencia de un registro, el equipo debe cambiar de estado de reposo al estado remitente.

Para asegurarse de que sólo un único dispositivo trata de convertirse en emisor al mismo tiempo y si el interlocutor está escuchando (es decir, en el estado receptor), el dispositivo con la intención de convertirse en emisor tiene que iniciar la fase de establecimiento.

Por lo tanto, envía el código ASCII 5 [ENQ] y espera a que el socio para enviar el código 6 [ACK] para indicar que ha cambiado desde el ralentí hasta el estado receptor. La recepción de [ACK] se completa la fase de establecimiento y se inicia la fase de transferencia. En estado de reposo cualquier otro carácter que recibió [ENQ] será ignorado. Recibir un [ENQ] debe ser respondida con [ACK] y se pone el receptor en el estado receptor. Este sería el procedimiento normal de la fase de establecimiento. En caso de error hay tres formas en que el receptor puede responder a una [ENQ]:

- El receptor envía un carácter que no sea [ACK].

Normalmente esto es un código ASCII 21 (15hex) [NAK] cuando el receptor está ocupado. El emisor espera una cantidad de tiempo y luego trata de establecer con otro [ENQ]. Este ciclo se repite hasta que [ACK], [NAK], o cualquier otro carácter que se recibe.

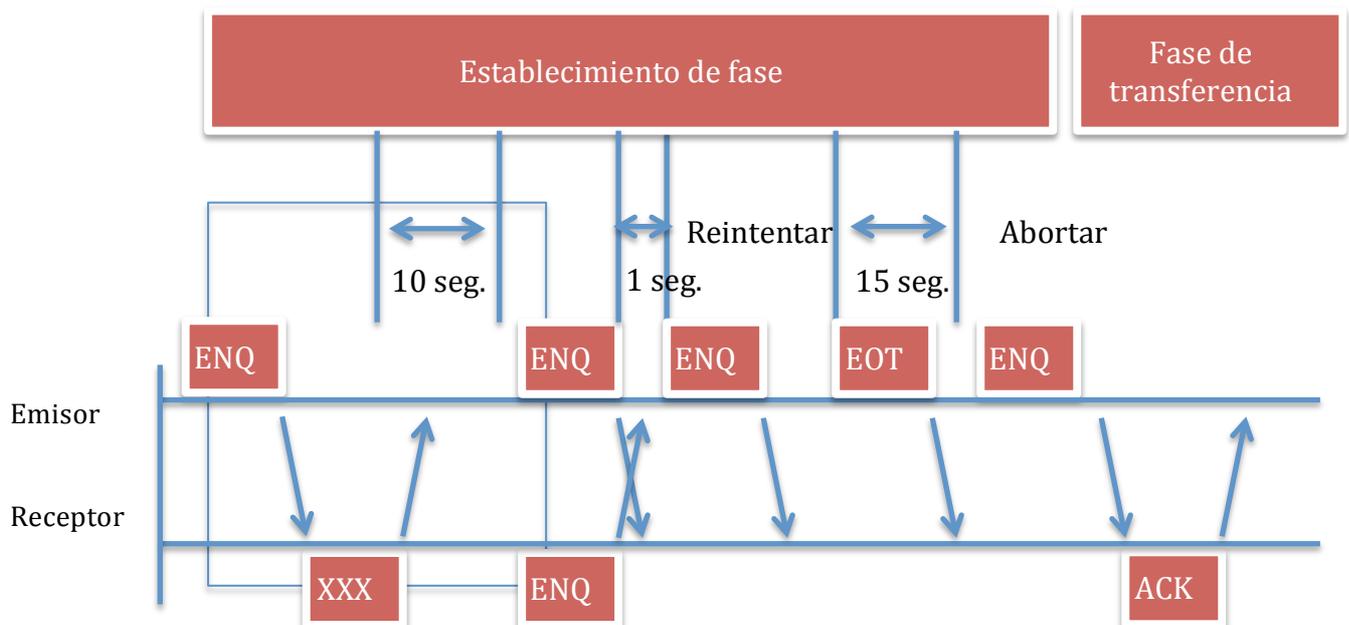


Ilustración 3-M Respuesta a cualquier caracter

- El receptor envía un [ENQ].

La situación se denomina "contención" ya que ambos equipos tratan de convertirse en emisores. El sistema de instrumento, por definición, la prioridad más alta para transmitir la información cuando se produce contienda. Por lo tanto el sistema de acogida tiene que dejar de enviar [ENQ], pero tiene que responder con [ACK] o [NAK] cuando detecta contención. Por otro lado los equipos en general esperan al menos 1 segundo y luego vuelve otra [ENQ]. Este ciclo se repite hasta que [ACK], [NAK], o cualquier otro carácter que se recibe.

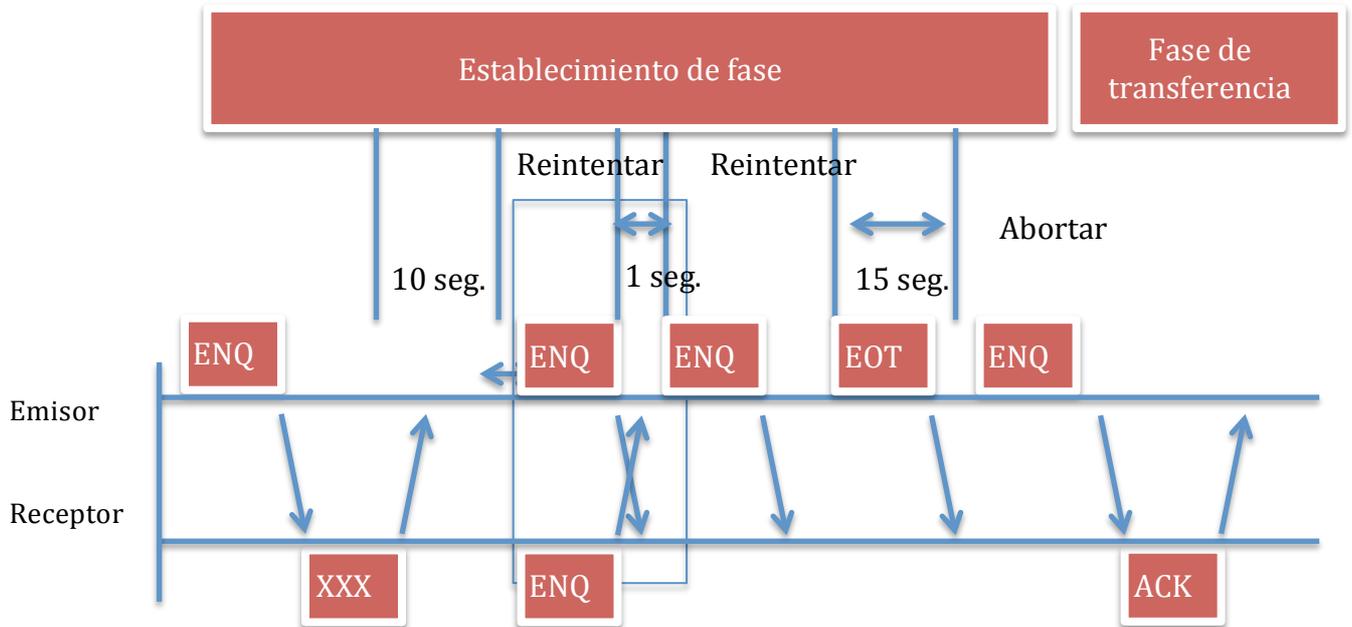


Ilustración 3-N Respuesta ENQ

- El receptor no responde.

El receptor espera 15 segundos, se inicia la fase terminal mediante el envío de un [EOT] (código ASCII 4), y muestra un mensaje de error.

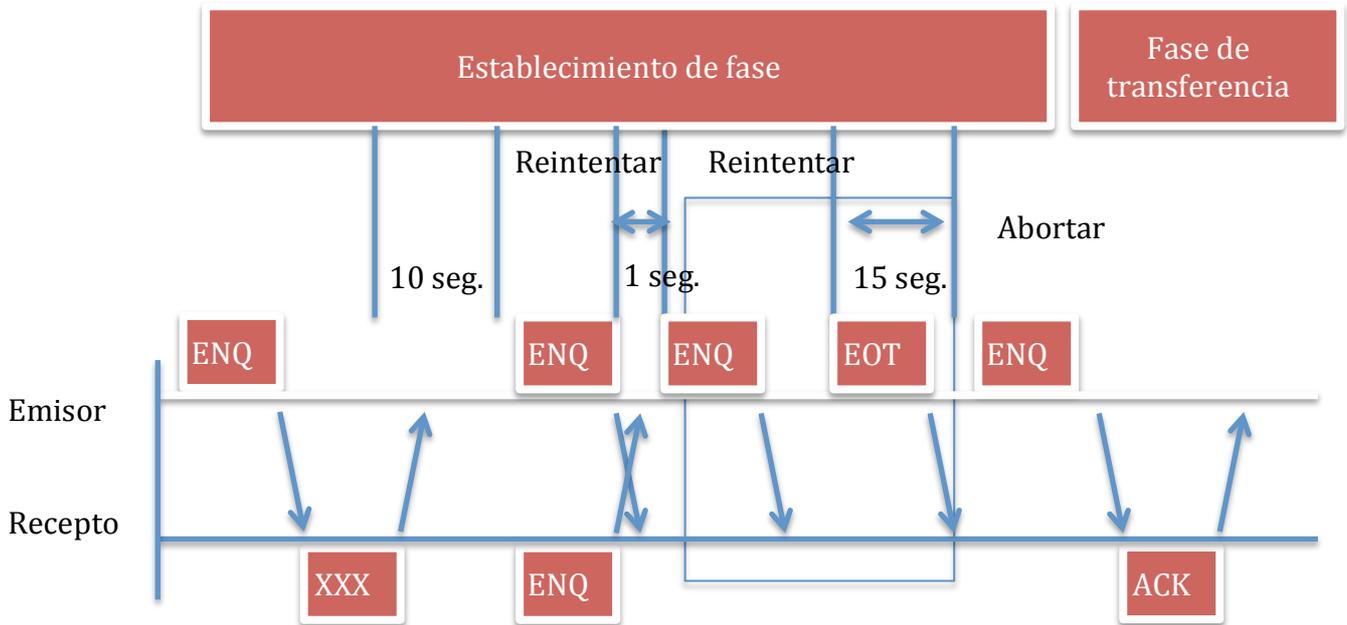


Ilustración 3-0 Tiempo de espera agotado

- El receptor envía un [ACK].

El remitente cambia en la fase de transferencia y se inicia el traslado de expediente. Alternativamente, si el equipo está en modo de prueba, se inicia la fase terminal mediante el envío de un [EOT] (código ASCII 4).

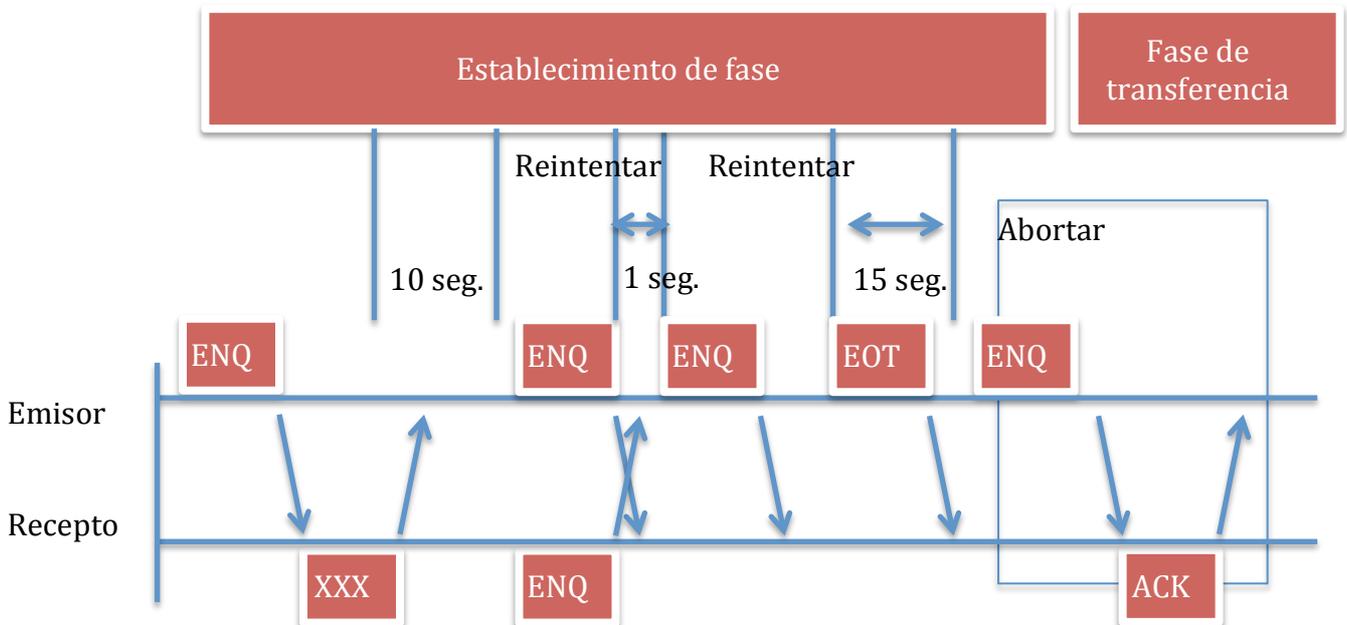


Ilustración 3-P Fase completada exitosamente

La siguiente figura muestra las cuatro posibles respuestas a una [ENQ] en una presentación de diagrama de flujo. Punto de entrada "A" es provocada por la capa de presentación, cuando se solicita el servicio de transmisión de la capa enlace de datos. El punto "B" es el punto de partida para un nuevo juicio en caso de errores. El punto "C" marca la transición a la fase de transferencia.

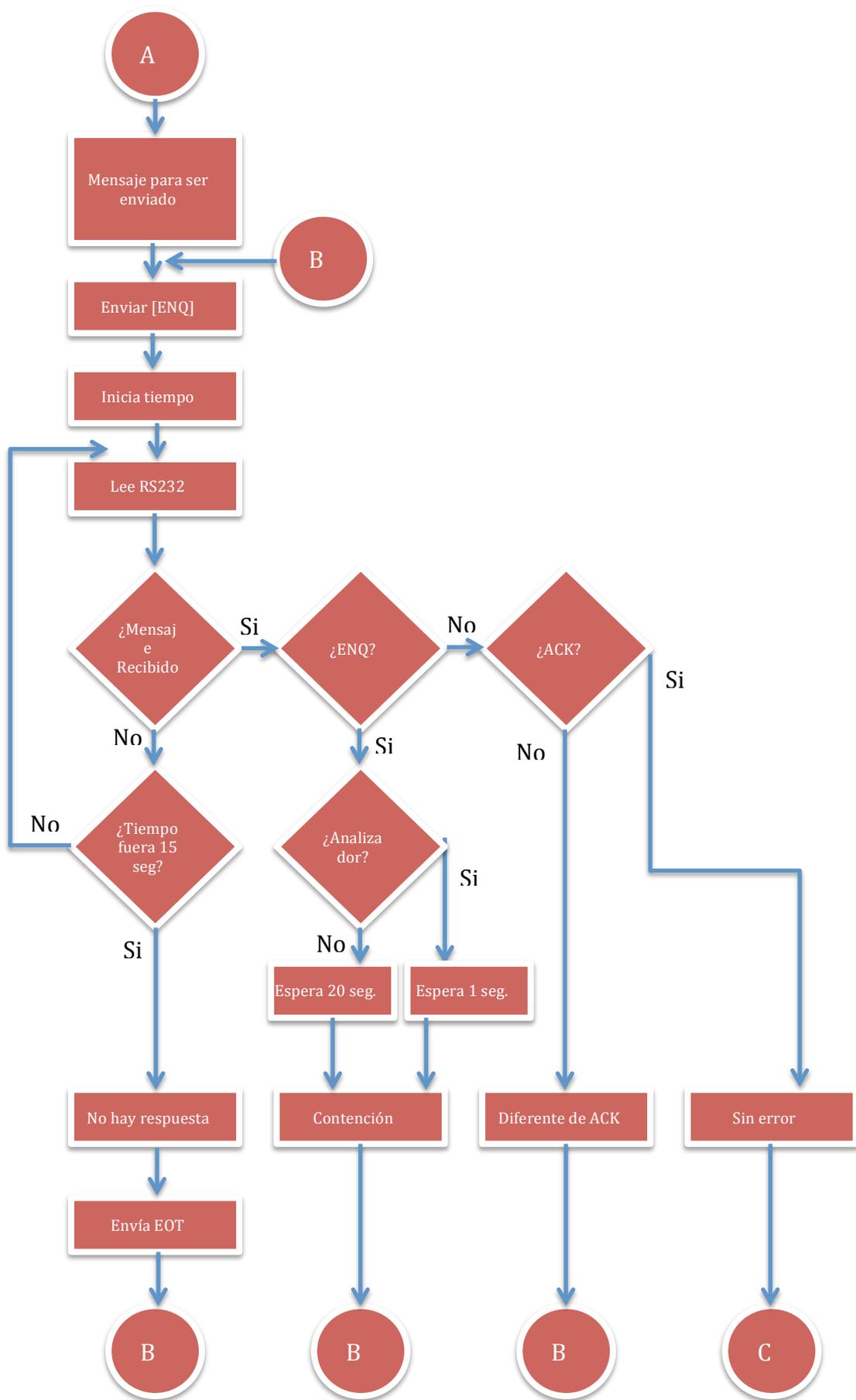


Ilustración 3-Q Fase de establecimiento

3.10.3.2 Fase de transferencia

La capa de presentación construye e interpreta los registros que contienen la información de los mensajes. Estos registros se intercambian con la capa de enlace de datos, que se espera que transmiten durante la fase de transferencia. Ya que los registros pueden tener un byte ampliado, una tarea de la capa de enlace de datos es cortar los registros en no más de 240 bytes. Estas porciones son empaquetadas en los *frames*. Otros siete bytes de delimitadores, suma de comprobación y se añaden identificadores como *frame* de la cabecera del mensaje. Así, la longitud de la trama puede variar entre 8 y 247 bytes.

Hay dos tipos de *frames*: *frames* intermedios y finales. Mensajes (es decir, registros) de más de 240 bytes se cortan en una o más porciones de 240 bytes para ponerlo en un *frame* intermedio. Los bytes restantes en el último cuadro se colocan al final del marco. Por ejemplo, hay dos *frames* intermedios y un *frame* final de los registros con una longitud de 481 hasta 720 bytes.

Después de cada *frame* enviado, el remitente de la transmisión se interrumpe hasta que se recibe una respuesta o un tiempo de espera termine. Normalmente, el receptor envía un [ACK] o [EOT] en la recepción de la señal de que el último *frame* fue recibido con éxito y que está preparado para recibir el siguiente *frame*.

[EOT] tiene el significado de [ACK], pero, además, pide al remitente la parada de transmisión tan pronto como sea posible (interrupción del receptor).

Especificaciones ASTM E1394-97 permite del remitente el tratamiento de una interrupción como un receptor normal [ACK], es decir, es capaz de implementar un receptor especial en una rutina de interrupción. El receptor envía un [NAK] de respuesta a la señal que el último *frame* no se ha recibido correctamente y que está preparado para una retransmisión del último *frame*. Así que hay tres formas en que un huésped puede responder a la transmisión de la trama analizador:

- El anfitrión envía un [ACK] o [EOT].

El equipo envía el siguiente registro. Ya que el equipo tiene más datos para transferir, prosigue a transmitir los registros (a partir de la fase de establecimiento).

- El anfitrión envía cualquier otro carácter, pero [ACK] o [EOT] (por ejemplo, [NAK]).

El equipo repite la transmisión del registro. Este ciclo se repite hasta que un número de seis reintentos.

En este punto el equipo inicia la fase terminal mediante el envío de un [EOT] y muestra un mensaje de error.

- El anfitrión no responde.

El equipo espera 15 segundos, se inicia la fase terminal mediante el envío de un [EOT] y muestra un mensaje de error.

Hay reacciones diferentes del respuesta del anfitrión a una transmisión de *frame* en el equipo. Cuando el equipo está en modo de receptor y espera para recibir un *frame* del anfitrión, hay tres posibles escenarios:

- El anfitrión envía un *frame* de caracteres

Si un *frame* completo es recibido se verifica para el número de *frame* de la derecha y una suma de comprobación correcta. Frames válidos son contestadas con un [ACK] por el equipo. Los frames inválidos son respondidos con [NAK] y el rechazo de la trama no válida.

- El anfitrión no completa el envío de *frames*

Cuando ninguna recepción de *frame* se ha completado ni un [EOT] se recibió en 30 segundos después de equipo respondió con la último [ACK] o [NAK], entonces el equipo descarta el mensaje incompleto, los cambios al estado de reposo y se cambian para estar en el estado neutral.

- El anfitrión envía un [EOT]

Esto restablece el equipo al estado de reposo. Frames sólo se consideran completamente recibido para ser válido.

La siguiente figura muestra las tres posibles respuestas en una transmisión de la trama. El punto de entrada "C" se desencadena por la superación de la fase de establecimiento. También es punto de reentrada para posteriores transmisiones de éxito. El punto "D" es el punto de partida para un nuevo juicio en caso de una respuesta diferente a [ACK]. El punto "E", marca la transición a la fase terminal.

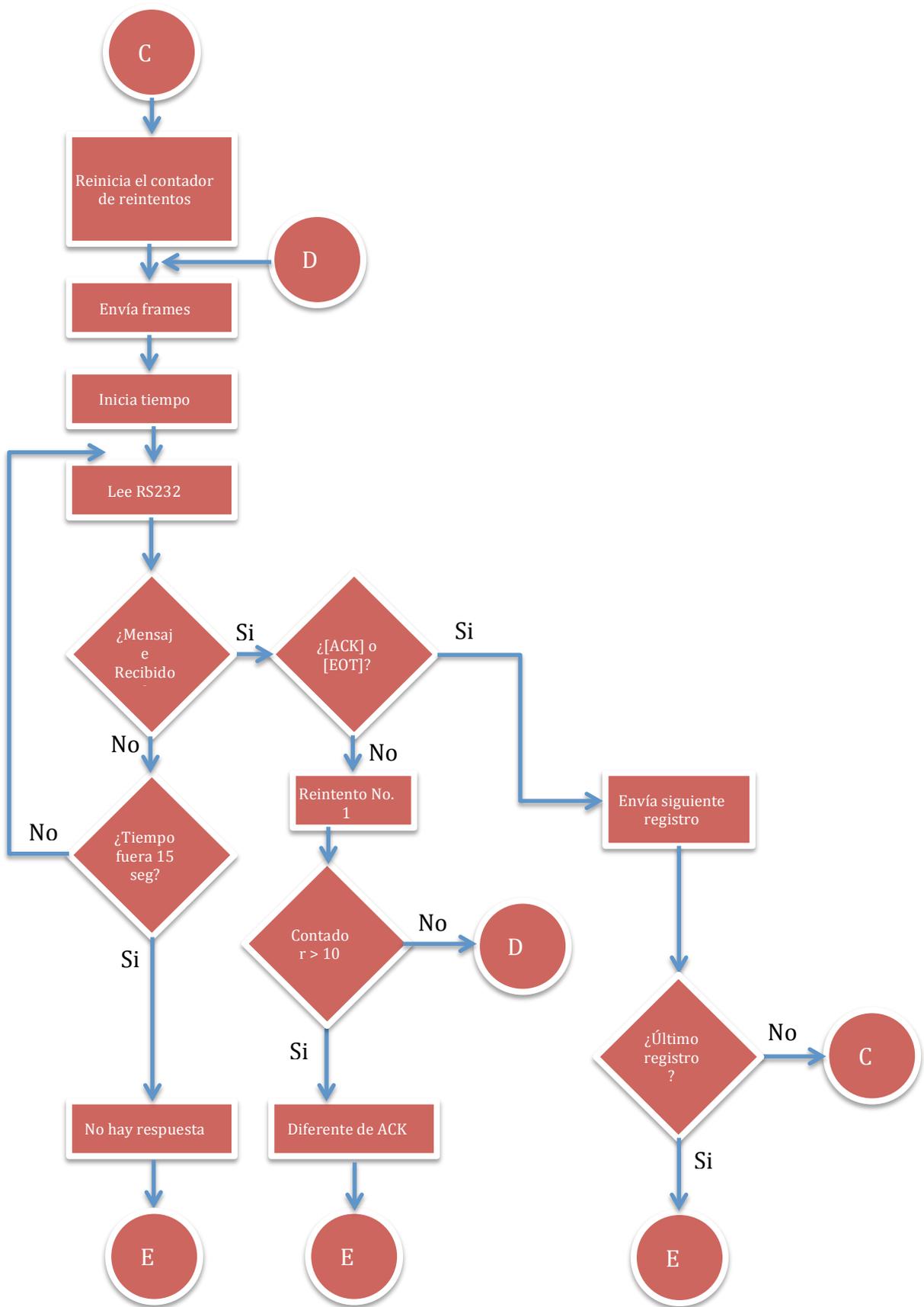


Ilustración 3-R Frase de transferencia

3.10.3.3 Fase de transferencia

La fase de terminación es la transición de ambos interlocutores al estado de reposo. Sólo puede ser iniciado por el remitente, simplemente enviando un código [EOT]. No hay respuesta del receptor de ese mensaje. Cada vez que un receptor detecta un [EOT] tiene que cambiar de estado de reposo, con respecto a la línea para estar en el estado neutral.

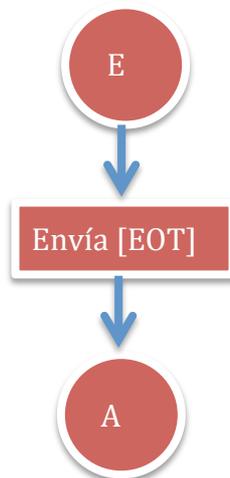


Ilustración 3-S Fase de transferencia

3.10.3.3.1 Formato de *frame*

Como se menciono anteriormente, hay dos tipos de *frames*:

- *Frame* intermedio

[STX]	NF	Texto	ETB	CH	CL	[CR]	[LF]
-------	----	-------	-----	----	----	------	------

- El *frame* final

[STX]	NF	Texto	ETX	CH	CL	[CR]	[LF]
-------	----	-------	-----	----	----	------	------

[STX]

Es 2 en código ASCII, indicando el inicio de la transmisión.

NF (Número de *frame*)

Es un dígito el cual va de 0 a 7. Los *frames* son numerados cíclicamente para habilitar el receptor y verificar la secuencia de *frames* válidos.

[ETB]

Número 23 en código ASCII, indicando el fin del texto de bloque de un *frame* intermedio.

[ETX] Es el 3 en código ASCII, indicando el final del bloque de texto del *frame* final.

CH, CL

Representa los bits más y menos significativos. Creando así el verificados de la transmisión que consta de dos dígitos hexadecimales. La suma de control es el módulo 8 de la suma de los valores ASCII de los caracteres del *frame* a partir desde "NF" y hasta [ETX], respectivamente [ETB].

Texto

Es el dato contenido en un *frame* con un máximo de 240 caracteres. Los registros son subdivididos dentro de *frames* intermedio con 240 máximos cada uno. Solo el último *frame* deberá de ser indicado por el [ETB].

3.10.4 Capa física

La conexión física entre el analizador y un sistema de computadora(anfitrión) se establece por medio de las interfaces de serie y un cable RS-232[19].

Para el cable y los conectores se usa un tipo especial de 0-modem de 9 pines hembra sub-D en el lado de acogida y de 9 pines macho sub-D en el lado del equipo clínico.

CAPÍTULO 4. CONSTRUCCIÓN DEL PROYECTO

4.1 Método propuesto

El método propuesto se describe de manera general en los siguientes pasos:

Uno.- Se seleccionan diferentes rasgos representativos de la base de datos.

Dos.- Se crea el árbol de derivaciones definido en 4.5.2.

Tres.- Se recorre el árbol y se genera el protocolo Alfa definido en 1.7.3.

Cuatro.- Se realiza una traducción literal del protocolo Alfa al formato ASTM.

Cinco.- Si es la primera vez que se realiza el aprendizaje el paso es omitido, de lo contrario, se cambian aquellos rasgos que no funcionaron con los rasgos del árbol de derivaciones o las mutaciones realizadas.

Seis.- Se realiza un análisis léxico del protocolo, si el análisis es correcto se procede al siguiente paso, de lo contrario, marca error.

Siete.- Se envía el protocolo al equipo clínico o simulador y se espera repuesta.

Ocho.- Si la información es aceptada por el equipo clínico o simulador, se continua con la siguiente rama del árbol (rasgo). De lo contrario, se realiza la mutación de los rasgos definido en 4.7.1.

Nueve.- Si es el fin de la transferencia y la información fue aceptada de manera exitosa se clasifica en la base de datos. De otra forma, se envía error.

4.1.1 Pseudocódigo

```
Mientras(Existen rasgos y no es fin de transmisión)
  Tomar rasgo y construir el árbol de derivaciones.
  If(existen nodos en el árbol y no se ha terminado la comunicación)
    Se recorre el árbol
    Se construye protocolo Alfa
    If Not(Es la primera pasada)
      Mutación
    If(Hay error en el análisis léxico)
      Enviar Error.
    Else
      Se envía el protocolo
    If(La parte del protocolo es aceptada)
      Continúa con el siguiente rasgo (rama)
    Else
      Mutación
  Else
    Enviar error.
```

4.2 Modelo de desarrollo de software

Existen una gran variedad de modelos en ingeniería de software, sin embargo, el siguiente modelo presentado es el que más apegado al proyecto de investigación.

4.2.1 Modelo lineal secuencial

El modelo lineal secuencial, también conocido como modelo de cascada, se basa en el enfoque sistemático y secuencial del desarrollo del software que comienza en un nivel de sistema y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

La ingeniería y el análisis del sistema comprende los requisitos que se recogen en el nivel de análisis y diseño. La ingeniería de información abarca los requisitos que se recogen en el nivel de empresa estratégico y en el nivel del área de negocio.

La Ilustración 4-A muestra la visión del modelo lineal secuencial.

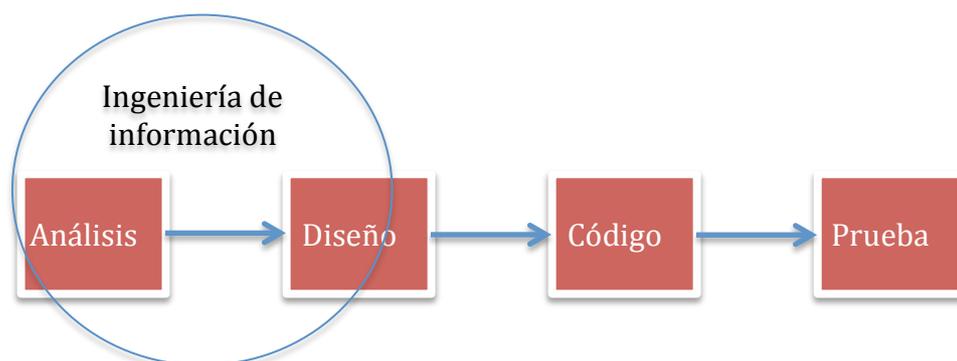


Ilustración 4-A Modelo lineal secuencial

Sin embargo, según [20] la ingeniería de software es una aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora, y la documentación asociada requerida para diseñar, desarrollar, operar, y mantener.

Ahora bien, el trabajo de investigación no tiene por objetivo inmediato o a corto plazo, hacer una interfaz gráfica para la administración, uso y manejo del motor de aprendizaje hacia el programador habitual.

El software gráfico utilizado y desarrollado en la presente tesis, tiene como único objetivo realizar pruebas del proyecto y obtener los resultados correspondientes.

Es así que, no cumple con las etapas de requerimientos de cliente/usuario, diseño gráfico, operación, mantenimiento de cualquier modelo de ingeniería de software.

Por lo tanto, el modelo lineal secuencial no es completamente adecuado y los demás modelo son innecesarios para el objetivo de la investigación.

4.3 Desarrollo y especificaciones

La construcción del proyecto del presente trabajo de investigación fue desarrollado en varias capas, cada una de ellas tiene una función específica en el flujo de trabajo del aprendizaje, la Ilustración 0-D Arquitectura del proyecto de tesis muestra las principales capas y una descripción de su arquitectura, así como funcionamiento principal.

4.4 Base de datos

La base de datos maneja dos tipos de protocolos de comunicación para los equipos clínicos, en primera instancia la capa contiene los protocolos ASTM E1394-97 base para el aprendizaje básico de la comunicación, esto es, al inicio de la prueba de aprendizaje se cuenta con algunos protocolos ya especificados y válidos para ciertos instrumentos. Por lo tanto, el aprendizaje partirá de reglas, estructuras, sintaxis y semántica correcta para realizar la comparación y pruebas respectivas.

Esta misma capa es retro alimentada por la capa de clasificación, esto es, la capa de datos se incrementa constantemente con nuevos protocolos aprendidos de las capas inferiores, de igual manera que el tipo de protocolo anterior están validados de manera, sintáctica, semántica y estructural para un protocolo único.

Al protocolo almacenado en esta capa también es llamada estructura jerárquica.

De forma general, de manera interna esta capa realiza ciertas acciones y procesos sobre los protocolos de comunicación de la siguiente manera:

1. Almacena.
2. Organiza.
3. Elimina y controla duplicados.
4. Reduce el espacio requerido.
5. Proporciona información para las capas superiores.

El sistema de archivos de la capa base de datos, está formada de una manera básica para su fácil manejo, la Ilustración 4-B Sistema de archivos muestra dicho sistema.

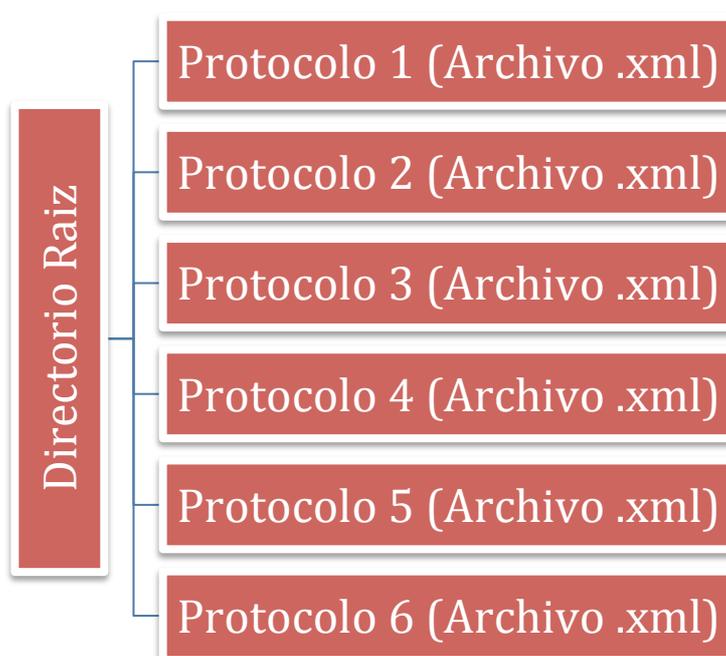


Ilustración 4-B Sistema de archivos

4.4.1 Estructura jerárquica

Los protocolos de comunicación están clasificados y ordenados con una estructura jerárquica en un metalenguaje llamado XML (*Extensible Markup Language*)[2] por sus siglas en inglés, cada estructura contiene un protocolo de comunicación único por equipo, versión, modelo o estándar determinado, la configuración del protocolo de comunicación está almacenada en un archivo de texto plano con la extensión y formato XML.

4.4.2 Organización del metalenguaje

Se ha definido una estructura generalizada para todos los protocolos de comunicación, la organización deriva directamente del estándar ASTM E1394-97, la estructura completa del estándar se puede ver en Anexo B Organización del estándar ASTM E139497, sin embargo, un ejemplo más abstracto de la organización jerárquica del protocolo de comunicación se encuentra definida en la siguiente tabla:

```
<?xml version="1.0" encoding="UTF-8"?>
<E1394-97>
  <NombreRegistro>
    <Campo1>
      <Elemento1>Tipo de dato</Elemento1>
      <Elemento2>Tipo de dato</Elemento2>
      .
      .
      <ElementoN>Tipo de dato</ElementoN>
    </Campo1>
  </NombreRegistro>
</E1394-97>
```

Tabla 4-A Esquema XML

4.5 Seleccionador de rasgos

Una de las principales capas del sistema de aprendizaje es la capa de selección de rasgos, como se vio en 1.7.2 la estructura y la validación del protocolo de comunicación están organizadas jerárquicamente, y separadas por etiquetas, cada etiqueta contiene intrínsecamente un fragmento de la forma de comunicación, por ejemplo:

```
<HeaderRecord>
  <RecordTypeID>H</RecordTypeID>
  <DelimiterDefinition>\^&amp;</DelimiterDefinition>
  <MessageControllID>1</MessageControllID>
  .
  .
  .
</HeaderRecord>
```

Tabla 4-B Ejemplo de rasgos

En el ejemplo anterior podemos notar que se muestra un pequeño ejemplo válido del registro de cabecera y sus respectivos nombres con parámetros *-recordemos que la comunicación consta de diferentes y variados registros con parámetros-*, por lo tanto, el conjunto del ejemplo anterior también puede denominarse rasgo.

En el presente trabajo de investigación, se define rasgo como el conjunto de reglas, nombres, parámetros e información representativa de comunicación del estándar, separados y organizados por etiquetas representativas.

La capa de selección de rasgos está dividida en tres etapas las cuáles son:

1. Etapa de conglomerado.
2. Etapa de selección.
3. Etapa de generación de objeto.

Estas dos etapas trabajan sobre el sistema de archivos de la base de datos mostrado en la Ilustración 4-B.

4.5.1 Etapa de conglomerado

La etapa de conglomerado es la encargada de revisar, administrar y validar cada una de las estructuras jerárquicas de los protocolos de comunicación, cada estructura está definida por rasgos ya sean únicos o idénticos, por lo tanto, esta capa se encargará de obtener una “vista panorámica” generalizada de todos los protocolos de comunicación almacenados y organizados en la capa *base de datos*, la vista panorámica no contendrá ningún rasgo repetido, inválido sintácticamente o semánticamente.

Se muestra a continuación el desarrollo de la etapa de conglomerado con un ejemplo simple.

Supongamos que la base de datos consta de solo dos protocolos de comunicación con su correspondiente estructura jerárquica, por ejemplo:

```
<HeaderRecord>
  <RecordTypeID>H</RecordTypeID>
  <DelimiterDefinition>\^&amp;</DelimiterDefinition>
</HeaderRecord>
```

Tabla 4-C Ejemplo protocolo A

```
<HeaderRecord>
  <RecordTypeID>H</RecordTypeID>
  <MessageControllID>1</MessageControllID>
</HeaderRecord>
```

Tabla 4-D Ejemplo protocolo B

Como podemos notar, ambos protocolos contienen solo un rasgo idéntico el cual es *RecordTypeID*, el *protocolo B* no cuenta con el rasgo *DelimiterDefinition* del *protocolo A* y a su vez, el *protocolo A* no contiene el rasgo *MessageControllID* del *protocolo B*, por lo tanto la etapa de conglomerado generara una única estructura.

```
<HeaderRecord>
  <RecordTypeID>H</RecordTypeID>
  <DelimiterDefinition>\^&amp;</DelimiterDefinition>
  <MessageControllID>1</MessageControllID>
</HeaderRecord>
```

Tabla 4-E Estructura generalizada

La estructura anterior muestra como la etapa de conglomerado genera una estructura jerárquica generalizada que elimina rasgos idénticos y une aquellos rasgos diferentes.

4.5.2 Etapa de selección

Una vez que la etapa de conglomerado haya concluido satisfactoriamente y además la estructura generalizada de los protocolos de comunicación esté creada, la etapa de selección de rasgos definirá las propiedades básicas de aprendizaje del proyecto de investigación, la etapa de conglomerado genera una amplia variedad de rasgos, los que serán tomados por la etapa de selección de rasgos para usarse en las siguientes etapas.

Se muestra a continuación el desarrollo de la etapa de conglomerado con un ejemplo simple.

```
<HeaderRecord>
  <RecordTypeID>H</RecordTypeID>
  <DelimiterDefinition>\^&</DelimiterDefinition>
  <MessageControlID>1</MessageControlID>
</HeaderRecord>
```

Tabla 4-F Estructura conglomerada

Dada la anterior estructura generalizada de la Tabla 4-F, la selección de rasgos se tomará de la siguiente manera:

- HeaderRecord
 - RecordTypeID
 - DelimiterDefinition
 - MessageControlID

Por lo tanto, el seleccionador de rasgos simplemente tomará el rasgo raíz y sus rasgos derivados, creando así un árbol de derivaciones que será usado en las siguientes capas, usando el mismo ejemplo anterior, el árbol de derivaciones se muestra en la Ilustración 4-C Árbol de derivaciones.

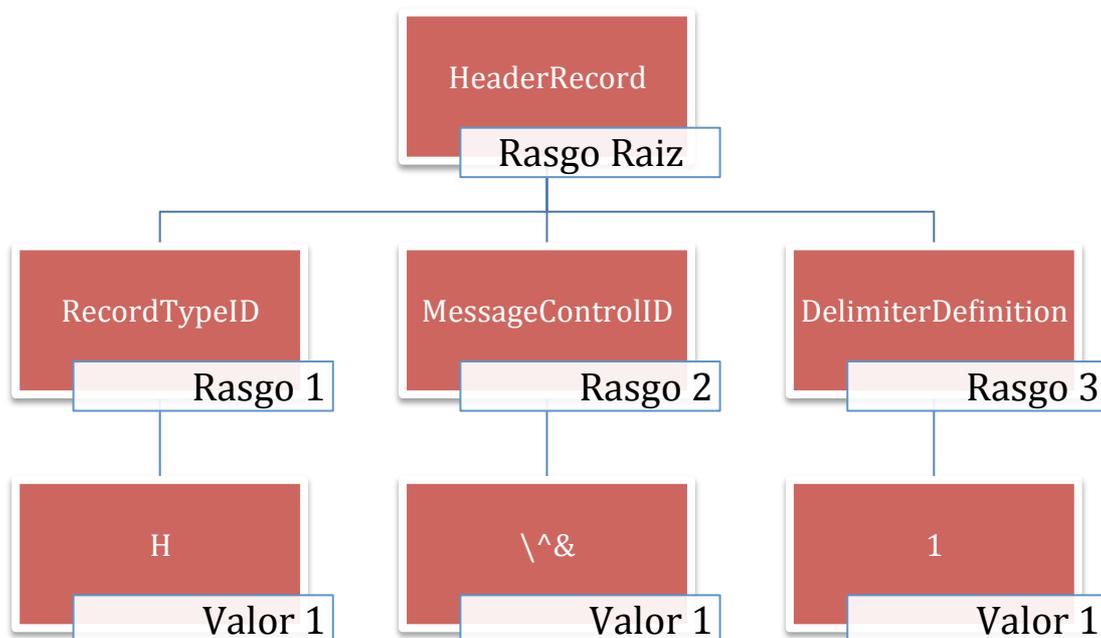


Ilustración 4-C Árbol de derivaciones

4.5.3 Etapa de generación de objeto

La última etapa de la capa de *seleccionador de rasgos* es denominada etapa de generación de objeto, el único fin de la capa final es crear un archivo físico XML a base del árbol de derivaciones de la etapa anterior, el archivo generado será utilizado en las siguientes capas.

4.6 Traductor XML-ASTM

La capa de traducción es la encargada de hacer la conversión literal y completa de un sistema estructural a otro, esta capa tiene un sentido completamente bidireccional, ya que la traducción puede realizarse en ambos sentidos.

Sea la Tabla 4-G con la estructura organizacional XML:

```
<HeaderRecord>
  <RecordTypeID>H</RecordTypeID>
  <DelimiterDefinition>\^&&</DelimiterDefinition>
  <MessageControlID>1</MessageControlID>
</HeaderRecord>
```

Tabla 4-G Estructura organizacional XML

La traducción de la tabla anterior generada por esta capa es la siguiente:

```
[ENQ]
[ACK]
[STX]1H|\^&||ELECTEST[CR][ETX]D7[CR][LF]
[ACK]
[STX]2P|1[CR][ETX]3F[CR][LF]
[ACK]
[STX]3O|1|01||^60^0R||||N|||||||O
[CR][ETX]3D[CR][LF]
[EOT]
```

Tabla 4-H Traducción XML-ASTM

La mayoría de los caracteres generados por la capa de traducción son de tipo no imprimibles. Algunos de estos caracteres son:

- [AKC]
- [CR]
- [EOF]
- [STR]
- [AKC]
- [LF]

Aunque estas secuencias de caracteres no son comprendidas de manera simple estos son entendidos y requeridos durante el flujo de comunicación de los equipos de laboratorio de análisis clínico.

4.6.1 Objeto: Protocolo Alfa

Si la salida de la traducción es de tipo ASTM el objeto generado es un protocolo de comunicación básica, este protocolo es llamado protocolo Alfa definido en 1.7.3.

Dicho protocolo Alfa tiene la función de servir como base para el inicio del sistema de aprendizaje, creando un protocolo casi listo para iniciar las pruebas con el equipo.

Sus características son:

- Sintácticamente correcto.
- Semánticamente inválido.
- Completamente estandarizado.

4.6.2 Objeto: Estructura XML

Si la salida del traductor es de tipo XML, genera un archivo con este formato para prepararlo para la primera capa (*base de datos*). Por lo tanto, el objeto generado no tendrá ninguna otra modificación ni utilidad viable o extra para la capa.

4.7 Sistema de reglas

El protocolo aún no cumple con todos los requerimientos para el proceso de aprendizaje, por lo tanto, la capa de *sistema de reglas* se encargará de validar y administrar que el protocolo Alfa cumpla de forma completa con la semántica establecida por el estándar.

Otra función primordial es ordenar el flujo de comunicación entre el software de aprendizaje y el equipo clínico, ya que, aunque cumplan con el estándar algunos instrumentos manejan diferente orden en el flujo de comunicación.

4.7.1 Sistema de mutación

El proceso de mutación se basa principalmente en recorrer todo la organización del estándar ASTM (véase Anexo B) donde aquellos nodos padres, contienen diferentes tipos de nodos hijos. Por ejemplo:

```
<Padre1>
  <item1>STRING</item1>
  <item2>STRING^STRING</item2>
  <item3>STRING\STRING</item3>
</Padre1>
<Padre2>
  <item1>STRING</item1>
  <item2>STRING^STRING^STRING</item2>
  <item2>STRING\STRING\STRING</item2>
</Padre2>
```

Hace una lista de posibles cambios y los va utilizando conforme son necesarios.
Por ejemplo:

```
<item1>STRING</item1>  
<item2>STRING^STRING^STRING</item2>  
<item2>STRING\STRING\STRING</item2>
```

Por otro lado, el sistema de mutación hará los siguientes cambios básicos en los rasgos del protocolo:

- Usara diferente tipo de dato para cada contenido del rasgo que haya fallado.
- Usara diferentes tipos de delimitadores (descrito en 3.7.4.4) para cada dato contenido en el rasgo.

4.7.2 Analizador

La capa analizador es utilizada en la fase de retroalimentación del sistema de aprendizaje. La primera instancia el protocolo Alfa generado no pasa directamente por la capa, ya que al inicio el protocolo Alfa es válido.

La función primordial de esta capa es realizar un análisis sintáctico de cada corrección realizada por el sistema de aprendizaje.

Esta herramienta de parseo (análisis) tiene una gramática libre de contexto donde cada regla de producción es de la forma:

$$V \rightarrow w$$

Donde V es un símbolo no terminal y w es una cadena de terminales y/o no terminales. El término *libre de contexto* se refiere al hecho de que el no terminal V puede siempre ser sustituido por w sin tener en cuenta el contexto en el que ocurra. La notación utilizada para expresar gramáticas libres de contexto en la estructura de comunicación es la forma Backus-Naur.

La capa fue desarrollada con la herramienta *Irony .Net*[9] por lo tanto, en el Anexo A da un pequeño ejemplo del uso de la herramienta y su relación con la gramática libre de contexto.

4.8 Sistema de aprendizaje

El sistema de aprendizaje está dividido en los siguientes módulos:

1. Motor comunicación del protocolo.
2. Motor de aprendizaje.
3. Sistema de reglas.
4. Interfaz de usuario.

Cada módulo tiene una función específica. Los módulos tienen la característica de ser completamente independientes, ayudando a que las

modificaciones para nuevas características sean mucho más fáciles de realizar.

4.8.1 Motor de comunicación

El sistema de aprendizaje utiliza el motor de comunicación para crear el enlace de datos entre el equipo clínico y el equipo de cómputo donde se encuentre instalado el software.

Todo el proceso y administración de los paquetes de comunicación del equipo o simulador es manejado por este motor, además de crear las secuencias de escape especiales definidas en 3.7.4 para la transmisión de manera correcta que son requeridas por el estándar de comunicación.

Todos los caracteres de transmisión especiales enviados o recibidos por el equipo clínico son capturados por el motor de comunicación y son enviados al sistema de aprendizaje para que procese las correcciones pertinentes. Estos, dependiendo de la respuesta del equipo, el sistema de aprendizaje ajustará los rasgos del protocolo de comunicación.

4.8.2 Motor de aprendizaje

El sistema de aprendizaje radica principalmente en los tipos de respuesta de los equipos clínicos, existen tres diferentes respuestas en un instrumento, estas se engloban principalmente en:

1. Transmisión válida.
2. Transmisión con errores.
3. Transmisión inválida.

4.8.2.1 Transmisión válida

El software captura este tipo de respuesta con un carácter de transmisión especial llamado ACK, cuando el sistema de aprendizaje ha recibido ACK, no se realiza ninguna modificación adicional de las ya realizadas y se procesa con el flujo de datos de manera continua.

4.8.2.2 Transmisión con errores

La bandera de transmisión *ERROR TRANSMISSION* indica al sistema de aprendizaje que la transmisión de los paquetes de información contiene errores de envío, y el último paquete de datos debe ser retransmitido inmediatamente.

En este punto ninguna modificación es requerida por el sistema de aprendizaje y se procesa con la transmisión de datos.

4.8.2.3 Transmisión inválida

Existen banderas de transmisión que indican que el protocolo de comunicación es inválido.

1. Checksum error.
2. NAK.
3. Sequence error.
4. Data type unknow.

Cualquiera de ellas le indica al sistema de aprendizaje que el protocolo no es válido y necesita ser reiniciada la transmisión.

El sistema procederá a realizar los cambios necesarios para volver a iniciar la comunicación de datos con el nuevo protocolo generado.

4.8.3 Aprendizaje

Todos aquellos protocolos de comunicación candidatos que pueden llegar a ser válidos en el flujo de comunicación de datos se encuentran almacenados en la memoria del equipo de cómputo.

El sistema de aprendizaje cuenta con una aglomeración de protocolos candidatos, cualquiera de ellos puede llegar a ser susceptible para ser seleccionado como protocolo candidato para ser enviado. El número de candidatos va a variar y va depender de cuantos protocolo totales existan en la base de datos y la diferencia de rasgos que exista entre ellos.

Como se definió en el punto 3.7.3 el envío de datos se realiza por niveles de comunicación, si existe un error en alguno de esos niveles, el sistema de aprendizaje tomará el siguiente rasgo de los protocolos candidatos y lo sustituirá con otro rasgo candidato, hasta hacer que encuentre el correcto.

La Ilustración 4-D Bloques principales del proyecto muestra de manera general el trabajo del sistema de aprendizaje.

Sin embargo el intercambio del rasgo incorrecto por el rasgo candidato pasa por varias fases antes de ser sustituido, el sistema de aprendizaje tomará el siguiente rasgo de la memoria del equipo, las fases son definidas al inicio de este capítulo.

Una vez que el rasgo candidato ha sido analizado se procede a realizar la sustitución, y se envía al equipo, si el equipo vuelve a enviar error el procedimiento se repetirá con cada uno de los rasgos candidatos de los protocolos aglomerados en la base de datos.

Si todos los rasgos candidatos han sido probados infructuosamente entonces el sistema de aprendizaje generara protocolos candidatos a base de E-1394-97 del estándar internacional ASTM. El estándar deberá funcionar en la

mayoría de los casos siempre y cuando el equipo clínico cumpla con las normas internacionales ASTM.

Si tanto los rasgos candidatos y los rasgos generados por el estándar fallan, el sistema de aprendizaje se detendrá y le dará oportunidad al usuario de modificar las reglas de comunicación de forma manual.

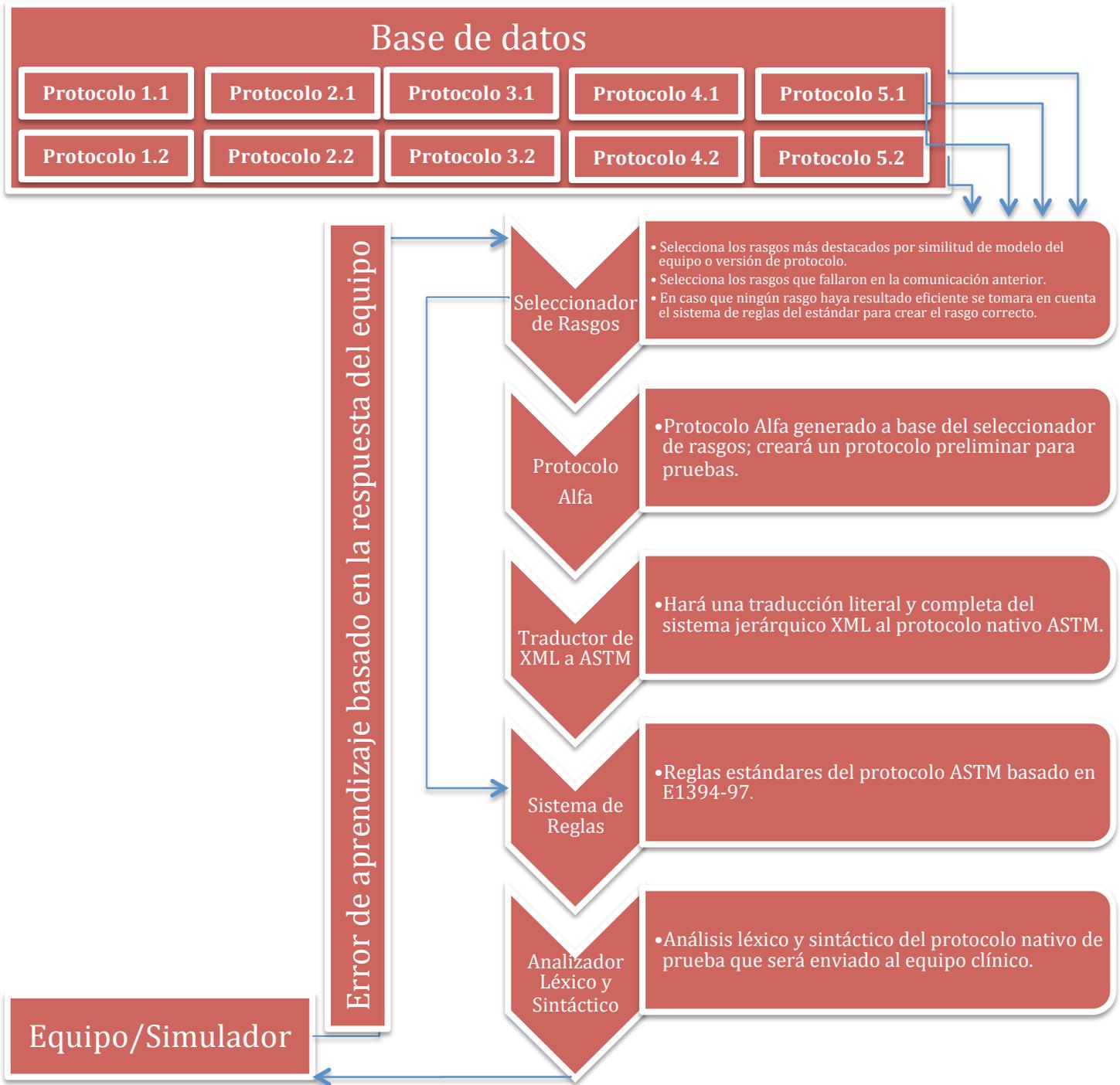


Ilustración 4-D Bloques principales del proyecto

4.9 Software

La interfaz de software así como su uso está dividido en bloques, cada uno de ellos estrechamente ligado con el anterior y el sucesor.

4.9.1 Bloque de Datos

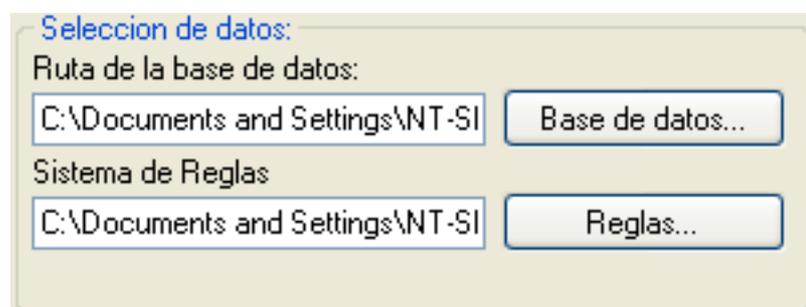


Ilustración 4-E Software - Base de datos

Como se puede apreciar, en la presente parte se selecciona la ruta de la base de datos, la cual fue definida en la Ilustración 4-B Sistema de archivos también se debe seleccionar el sistema de reglas estándar para poder realizar el análisis sintáctico y semántico, así como la generación de nuevos rasgos en caso que los rasgos candidatos fallen.

4.9.2 Selección de reglas

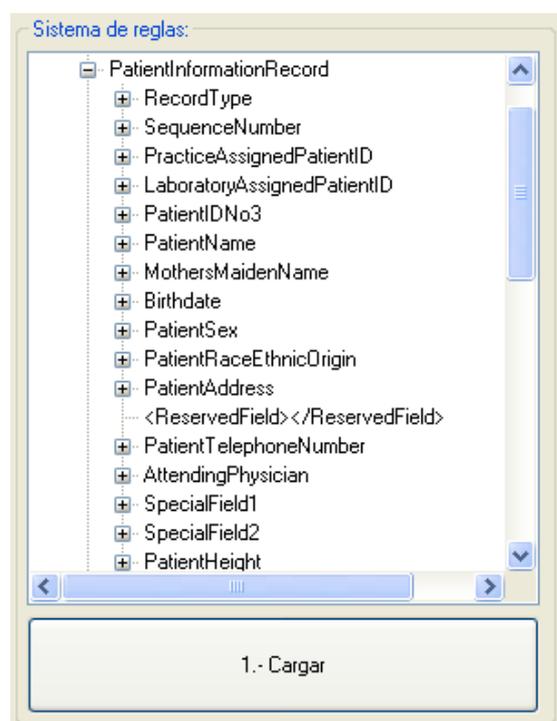


Ilustración 4-F Software - Selección de reglas

El sistema generará el árbol jerárquico de las reglas del estándar ASMT con todos los campos posibles y válidos requeridos para la comunicación de los equipos clínicos.

4.9.3 Selección de rasgos

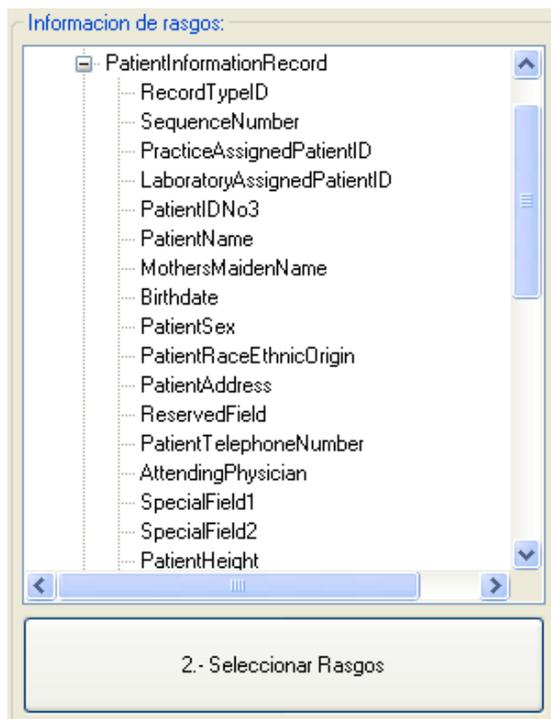


Ilustración 4-G Software - Selección de rasgos

4.9.4 Protocolo Alfa

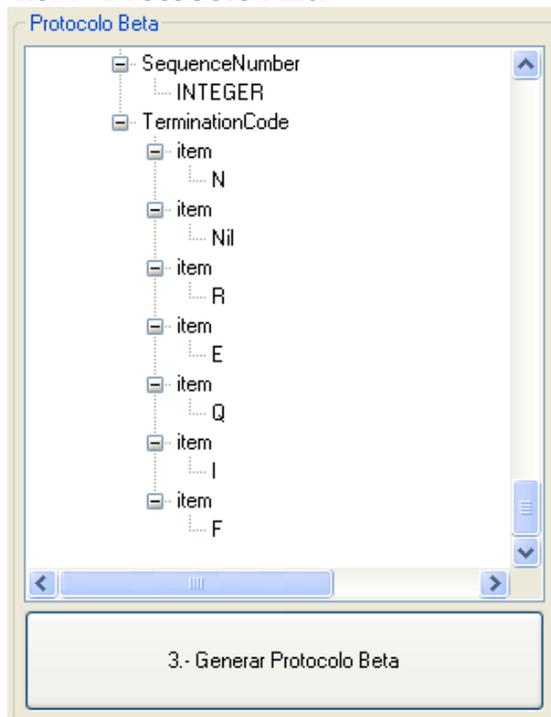


Ilustración 4-H Software - Protocolo Alfa

El módulo genera el protocolo Alfa inicial de toda la aglomeración de todos los protocolos de comunicación incluidos en la base de datos.

4.9.5 Análisis léxico y traductor

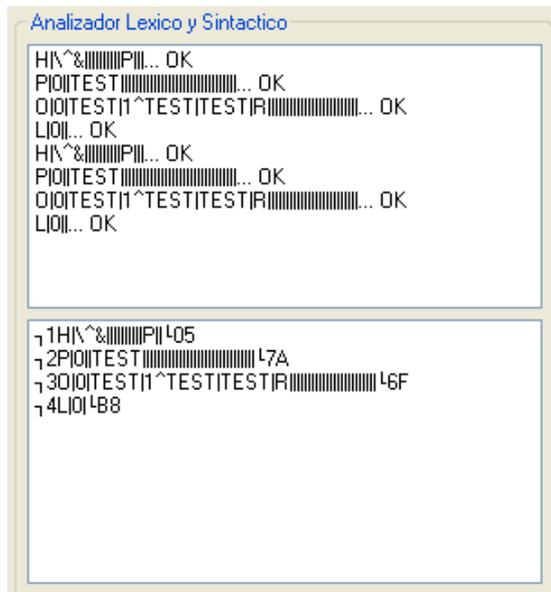


Ilustración 4-I Software - Análisis

En análisis léxico y sintáctico del protocolo Alfa, así como su traducción literal del esquema jerárquico XML al estándar ASTM E1394-97.

4.9.6 Terminal

La terminal es un módulo especial que sirve para dar un seguimiento de la transmisión de datos del software al equipo o simulador

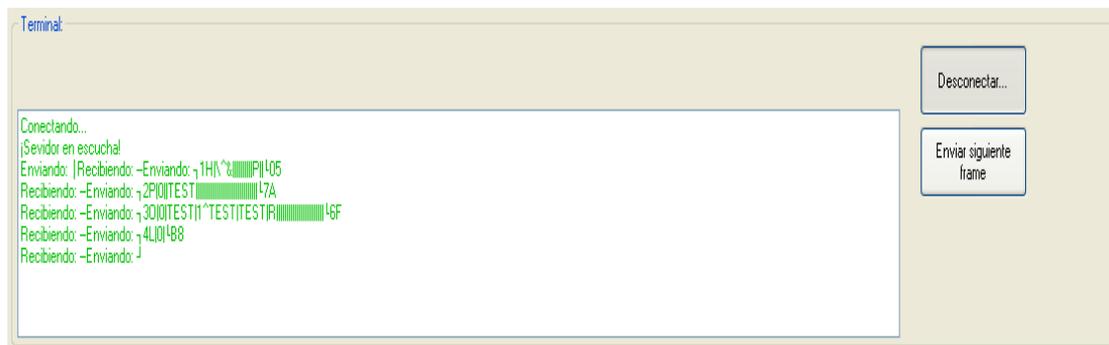


Ilustración 4-J Software - Terminal

CAPÍTULO 5.

PRUEBAS Y

RESULTADOS

Una vez concluido el proceso de cada una de las capas y cuando el sistema de aprendizaje finalizó satisfactoriamente, se procede a realizar diversas pruebas del software para evaluar el desempeño y velocidad de aprendizaje.

5.1 Metodología experimental

Los procedimientos para realizar las pruebas se engloban de la siguiente manera:

1. Configuración del equipo en modo de recepción de datos.
2. Conexión física entre el equipo clínico y el sistema de cómputo.
3. Limpieza de archivos temporales.
4. Inicio del sistema de aprendizaje.
5. Configuración de la herramienta de aprendizaje para el envío de datos.

Para el paso número uno, el equipo clínico contiene en su sistema operativo un panel de control, donde se configura el modo de trabajo, principalmente todos los equipos cuentan con tres distintos modos de comunicación; modo de envío de datos, modo de recepción de datos, modo de envío y recepción de datos.

En el segundo paso, la conexión física es a través de un cable *RS-232*[19], sin embargo, fue necesario utilizar un cable de tipo *USB RS-232 Adapter*, ya que en su mayoría los equipos actuales no cuentan con el puerto serial adecuado.

En el tercer paso pueden llegar a existir archivos o información temporal almacenada en el equipo clínico, se debe hacer una limpieza total para evitar confusiones de lo que fue o no transmitido exitosamente.

El quinto paso nos indica la necesidad de limpiar todos aquellos archivos temporales que hayan sido generados por la comunicación, con el fin de evitar confusiones en los resultados, la limpieza se realiza tanto del lado del equipo clínico como del sistema de cómputo.

El cuarto y quinto paso, consta de realizar los pasos definidos en 4.9 y enviar la información que el sistema haya generado.

Las pruebas para la fase de simulación fueron realizadas con la herramienta definida en 2.1.1 - *ASTM Host Simulator*. Las principales razones son las siguientes:

- Amplio conjunto de características y configuraciones.
- Manejo y seguimiento de errores.
- Lista de equipos necesarios para la simulación.
- No requiere una conexión física.
- Fácil disponibilidad.

Por otra parte, se simularon dos equipos de laboratorio de análisis clínico, el primero llamado Elecsys 2010 y el segundo nombrado Urisys. La selección de estos equipos de laboratorio tiene dos principales razones; la primera es que los equipos están disponibles de manera física en entorno de trabajo de laboratorio, la otra razón cae en que la herramienta de simulación utilizada cuenta con esos dos equipos para trabajar.

El equipo de cómputo número uno sirve para ejecutar el sistema de aprendizaje, así como manejar y corregir los paquetes de información que reciba.

Las características de ambos equipo son:

- Sistema Operativo Windows XP SP3.
- 2 GB de RAM.
- Procesador Intel Core i3.
- Puerto serial - *USB RS-232 Adapter*

Todas las pruebas de cada uno de los procesos de aprendizaje fueron realizados en dos entornos de trabajo diferentes. El primero fue realizado de manera simulada, por el contrario, el segundo entorno de trabajo, los procesos fueron corridos de manera real en los equipos de laboratorio.

5.1.1 Entorno simulador

El desarrollo de las pruebas y resultados fueron realizados de primera instancia en un ambiente controlado, este ambiente es propicio para realizar pruebas con diferentes tipos de errores, poder controlarlos y crear una depuración más precisa.

El entorno de simulación tiene como meta crear comunicaciones de equipos clínicos similares a los equipos reales que trabajan en un laboratorio de análisis clínico, creando paquetes de comunicación válidos y con características especiales.

Existen pocas herramientas de simulación para equipos clínicos, muchas de las cuales son proporcionadas solo en la venta de nuevos equipos clínicos de laboratorio, sin embargo, la herramienta utilizada para la simulación y realización de pruebas proporciona una lista amplia de una gran diversidad de equipos con protocolo clínicos muy específicos, esto nos proporciona un gran margen pruebas y resultados, la lista de los equipos se muestra en la Ilustración 5-A Lista de equipos de la herramienta simulador.

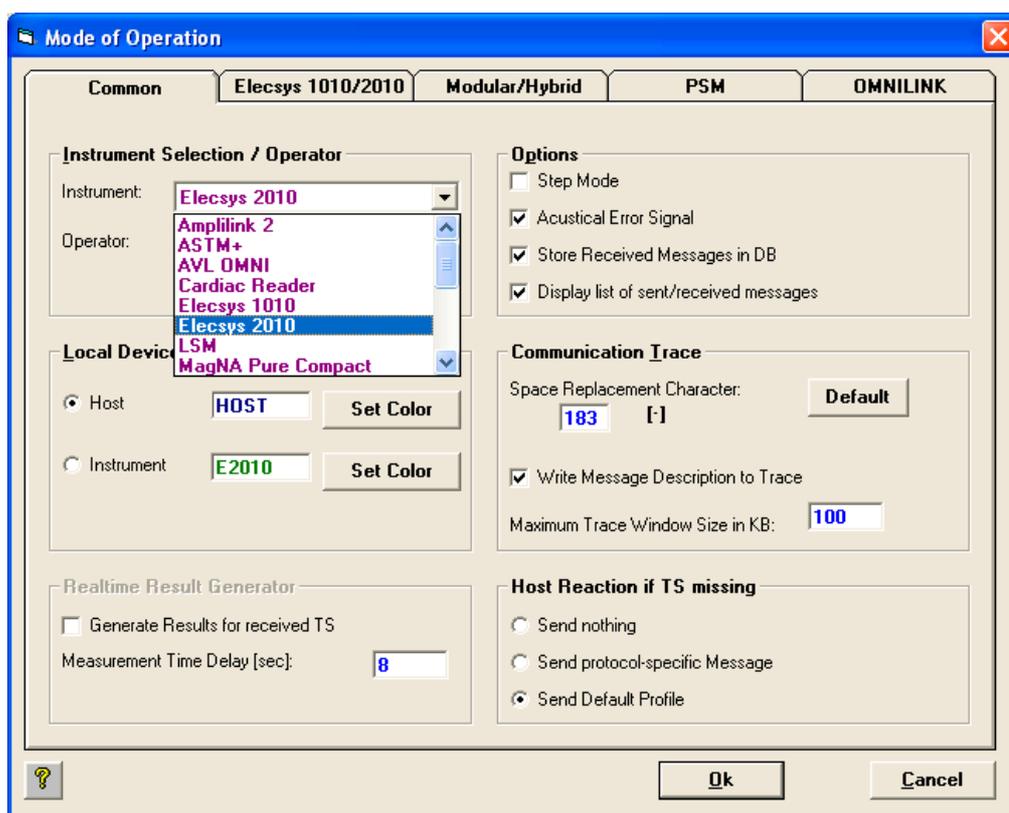


Ilustración 5-A Lista de equipos de la herramienta simulador

En general las herramientas de simulación proporcionan las siguientes características:

- Fáciles de configurar.
- Proporcionan diferentes equipos clínicos.
- Simulación por pasos.
- Manejo de errores.
- Respuestas personalizadas.

Esta herramienta de simulación cuenta con dos modos de comunicación, por un lado le permite comportarse como un verdadero equipo clínico y sus principales características, y por el otro, puede utilizarse como equipo de cómputo para recibir tramas de comunicación y crear un análisis del protocolo de comunicación.

5.1.2 Entorno de laboratorio de análisis clínico

A diferencia del entorno simulado, las pruebas y resultados son realizados con equipos reales con información de pruebas y pacientes auténticos.

Todo el proceso es igual al entorno de trabajo anterior, sin embargo, la herramienta de simulación es sustituida por un equipo de laboratorio de análisis clínico.

5.2 Resultados

Se realizaron un total de 68 pruebas de comunicación para dos equipos clínicos diferentes, las pruebas fueron realizadas tanto de manera simulada como de manera física con el equipo clínico, cada prueba contiene con cada una de sus etapas de comunicación definidas en el estándar ASTM.

Cada una de los ensayos consiste en lo siguiente:

1. Preparación de las entidades involucradas.
2. Configuración de la conexión de red.

5.2.1 Interpretación de los resultados

En la siguiente sección se muestran los resultados obtenidos de las pruebas realizadas, estos resultados son acumulados, organizados y mostrados en forma de estadística.

Cada resultado es catalogado por tipo de mensaje y etapas de transmisión, en general la transmisión es catalogada en tres etapas:

1. Etapa inicial
Comúnmente inicia con el registro cabecera definido en 3.7.4.5.1.
2. Etapas intermedias
Son las etapas donde se encuentran la mayoría de los registros ASTM E1394-97.
3. Etapa final
La etapa final va caracterizada por el registro de término definido en 3.7.4.5.6.

Existen también otros cuatro conceptos utilizados en la presentación de resultados.

Transmisión válida.- Indica al sistema de aprendizaje que el intercambio de información fue realiza de manera exitosa, esto es, el equipo clínico entendió de manera clara la información que le fue enviada y por lo tanto fue almacenada en sus sistema de datos para su posterior uso.

Error de comunicación.- Nos indica que el equipo clínico recibió de manera corrupta e incompleta la información que fue enviada por el sistema de aprendizaje. Los errores de comunicación se deben a diversos factores como:

- El algoritmo de verificación de datos tiene errores de diseño.
- La calidad del cable convertidor es de baja calidad.

Sin embargo, los errores de comunicación son de menor importancia y no afectan en ningún sentido el sistema de aprendizaje y no afectan enormemente los resultados obtenidos, cuando un error de comunicación es encontrado, simplemente se retransmite la información. No indica que el sistema aprendió o no.

Transmisión inválida.- El concepto indica que el equipo clínico no logró interpretar los datos recibidos de manera precisa, se debe a que el protocolo de comunicación generado por el sistema de aprendizaje no fue válido en alguno de sus niveles. Y el equipo rechazó la comunicación y requiere retransmisión completa.

Error de secuencia.- Señala que existe un error al enumerar los paquetes de transmisión, cada paquete transmitido tiene un identificador de secuencia, cuando el identificador esperado por el equipo no concuerda con el recibido se genera el error de secuencia y se retransmite la información. El error de secuencia no afecta radicalmente los resultados.

5.2.2 Resultados del entorno de simulación

5.2.2.1 Elecsys 2010

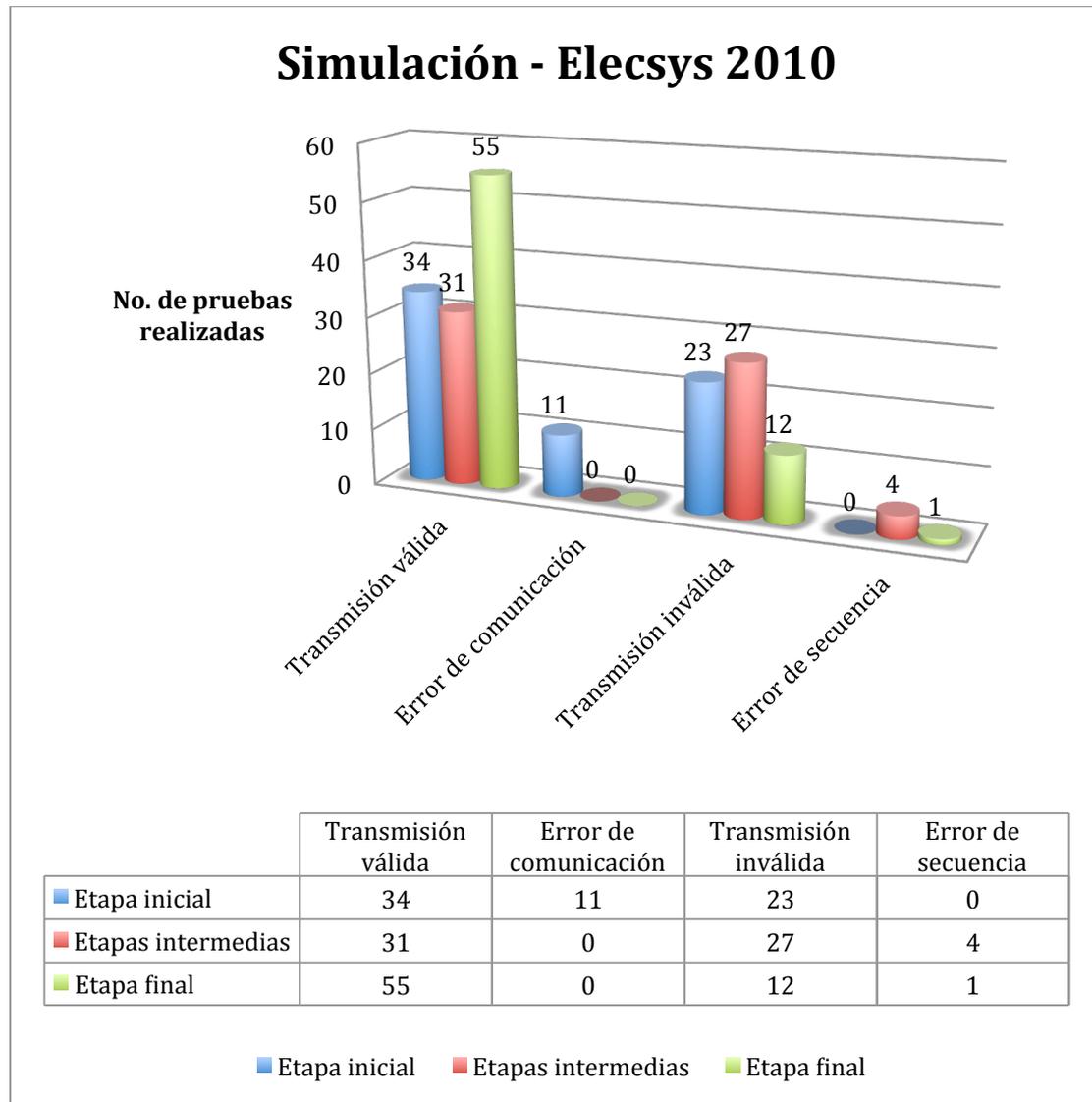


Ilustración 5-B Resultados simulación Elecsys 2010

El resultado de la primera prueba nos muestra que existe una mayor cantidad de comunicación válida y aprendida por el sistema de aprendizaje.

Los errores de comunicación en la etapa inicial son pocos y pueden ser atribuidos por el algoritmo de verificación de datos.

Las transmisiones inválidas se concentran principalmente en la etapa inicial e intermedia, es debido a que la mayor concentración de aprendizaje se encuentra en esas etapas.

Los errores de secuencia son mínimos y se concentra principalmente en la capa intermedia donde existe mayor cambio de rasgos.

5.2.2.2 Urisys

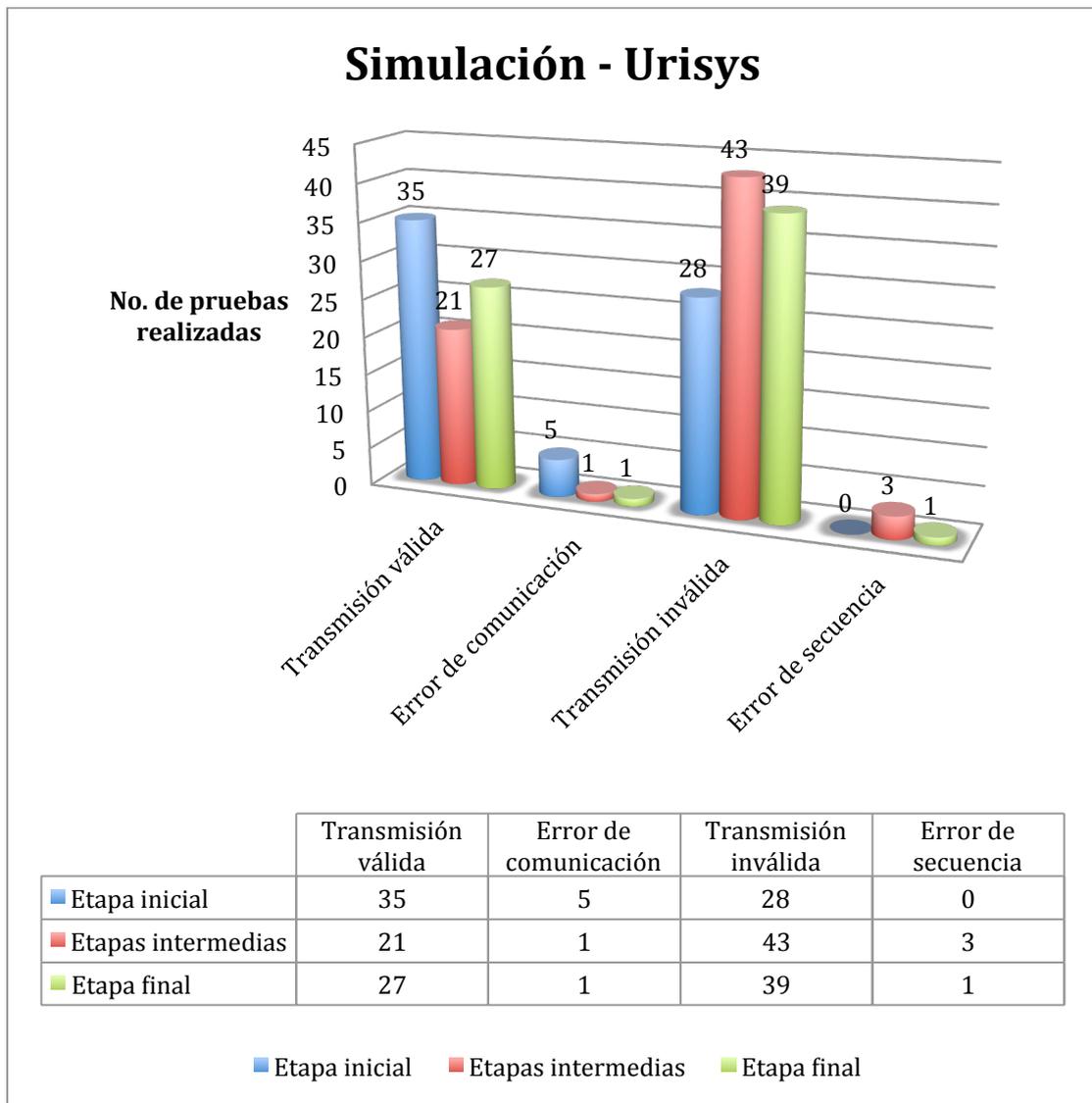


Ilustración 5-C Resultados simulación Urisys

Notamos claramente que las transmisiones válidas son ligeramente menores que las transmisiones inválidas, existe un mayor número de fallos en el aprendizaje de este equipo en particular.

El caso de errores de comunicación y demasía pocos, y no influyen de manera drástica en los resultados totales.

5.2.3 Resultados del entorno de laboratorio

5.2.3.1 Elecsys 2010

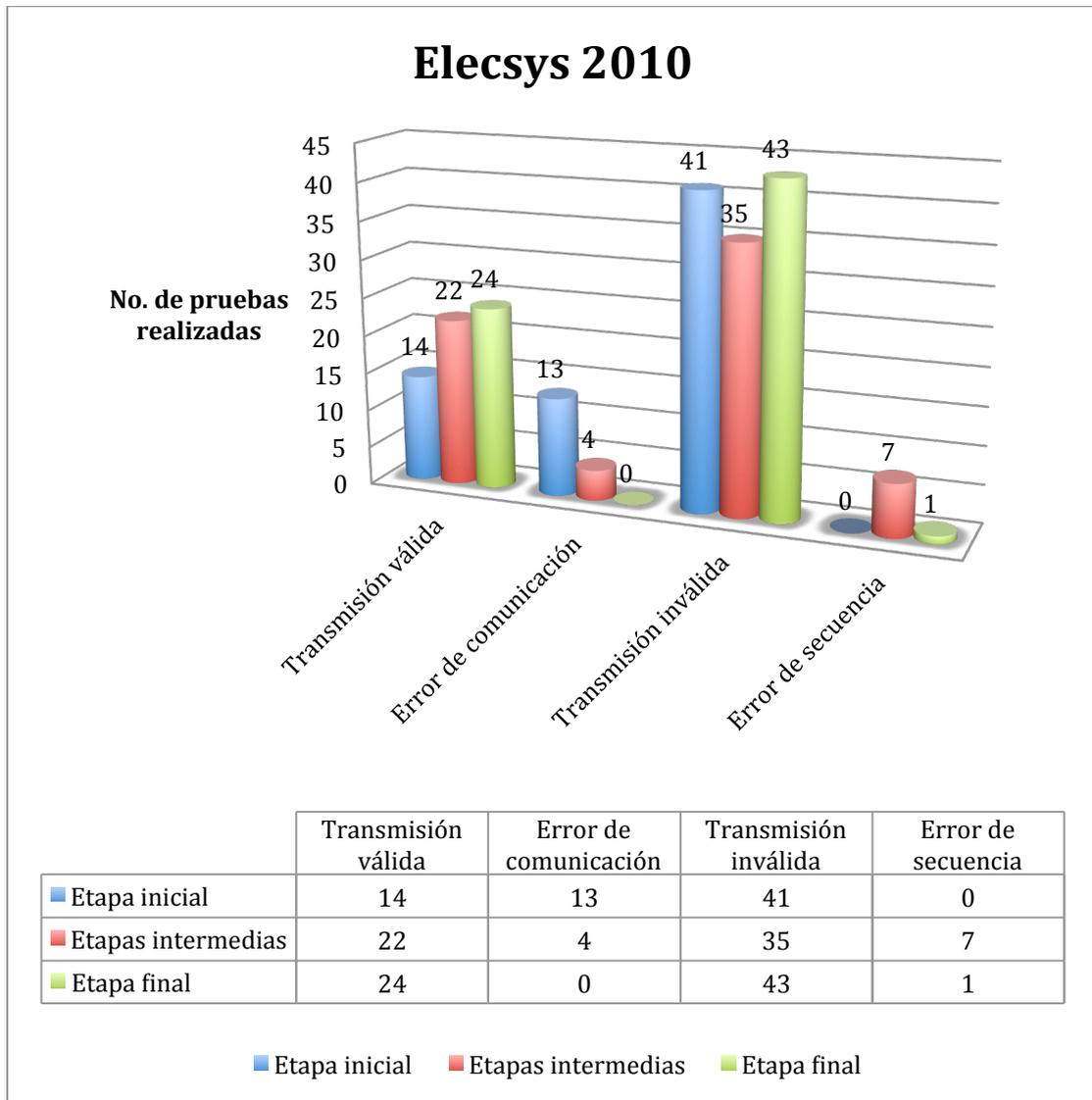


Ilustración 5-D Resultados Elecsys 2010

En las pruebas realizadas con equipos clínicos reales, notamos inmediatamente un claro incremento en las transmisiones inválidas, por lo tanto, existe un mayor número de fracasos del sistema de aprendizaje en el equipo real que con homologo simulado. Las transmisiones inválidas se encuentran esparcidas de manera generalizada en las tres etapas de comunicación.

Los errores de comunicación y de secuencia de igual manera son ligeramente mayores en el equipo clínico real.

5.2.3.2 Urisys

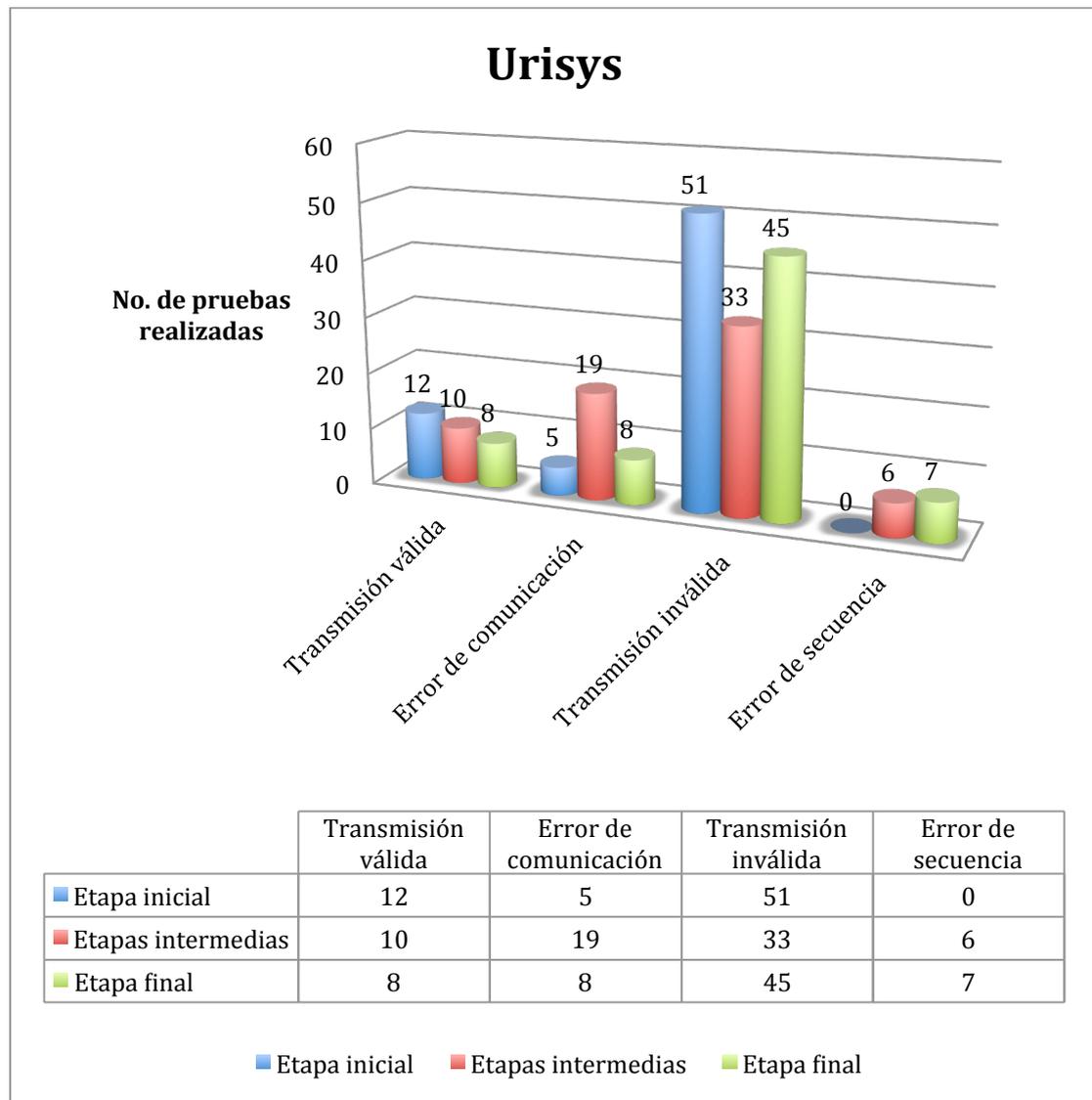


Ilustración 5-E Resultados Urisys

La gráfica de nueva cuenta tiene una tendencia mayor en las transmisiones inválidas, esto da una idea que en el entorno de trabajo real existe una amplia diferencia de aprendizaje

Todas las estadísticas anteriores muestran la forma en que el sistema de aprendizaje encontró errores en cada etapa de comunicación.

5.2.4 Medidas de éxito

Son necesarias realizar dos medidas de éxito para analizar el rendimiento del sistema de aprendizaje.

La primera de ellas es generar un gráfica con respecto al porcentaje total de aprendizaje de la herramienta con 25 pruebas realizadas. El sistema de aprendizaje tiene su punto central en la base de datos, por lo tanto, para cada prueba realizada se utilizaron diferentes protocolos de comunicación y por lo tanto diferentes rasgos descriptivos. Esto es, en cada prueba se cambiaron por completo los protocolos que contenía la base de datos (ninguna prueba contenía la misma información de aprendizaje), donde algunas veces no se tenían los rasgos adecuados para completar la comunicación exitosamente.

Esto nos permite dar una diversidad de rasgos y poder medir el porcentaje de éxito, sin ello los resultados serán siempre los mismos.

Existe una diferencia de interpretación entre las gráficas anteriores y la gráfica mostrada en la Ilustración 5-F; mientras las primeras nos muestran las estadísticas de transmisión por segmentos, la segunda gráfica muestra el aprendizaje por transmisión completa (no por segmentos) siempre con rasgos distintos en la base de datos.

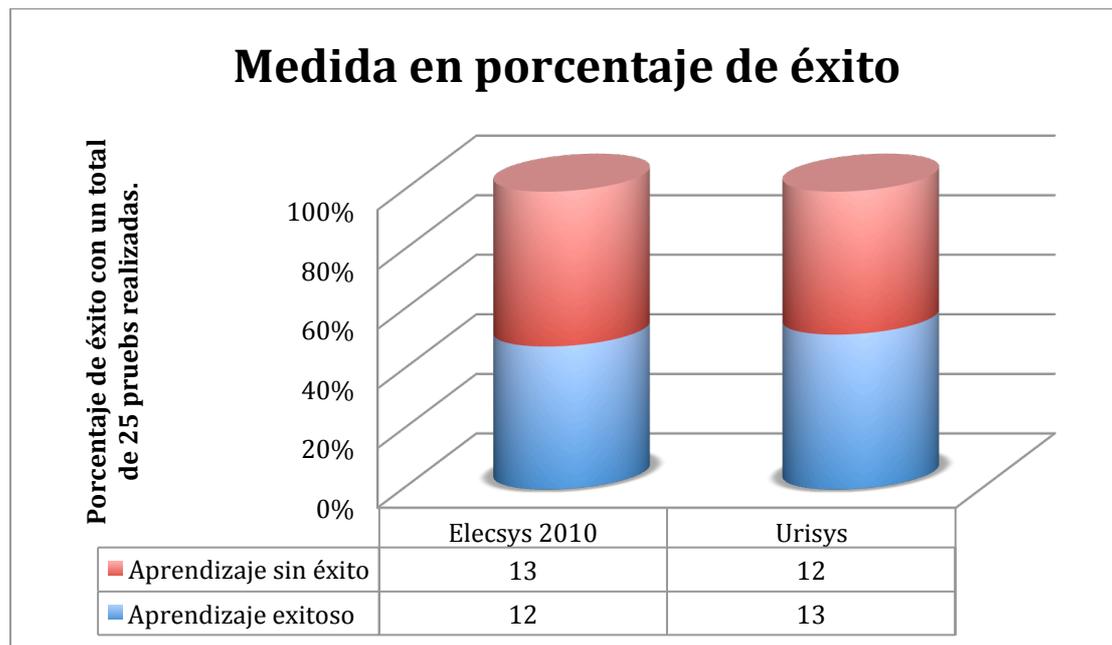


Ilustración 5-F Porcentaje de éxito

Por lo tanto, en un total de 25 pruebas (100%) y cada prueba con diferentes rasgos, el sistema de aprendizaje logro su objetivo en un total de 48% con el equipo Elecsys 2010 y un 52% con el equipo Urisys. Todo esto se traduce a la capacidad de aprender con las mutaciones realizadas al protocolo sin tener todos los rasgos completos del equipo.

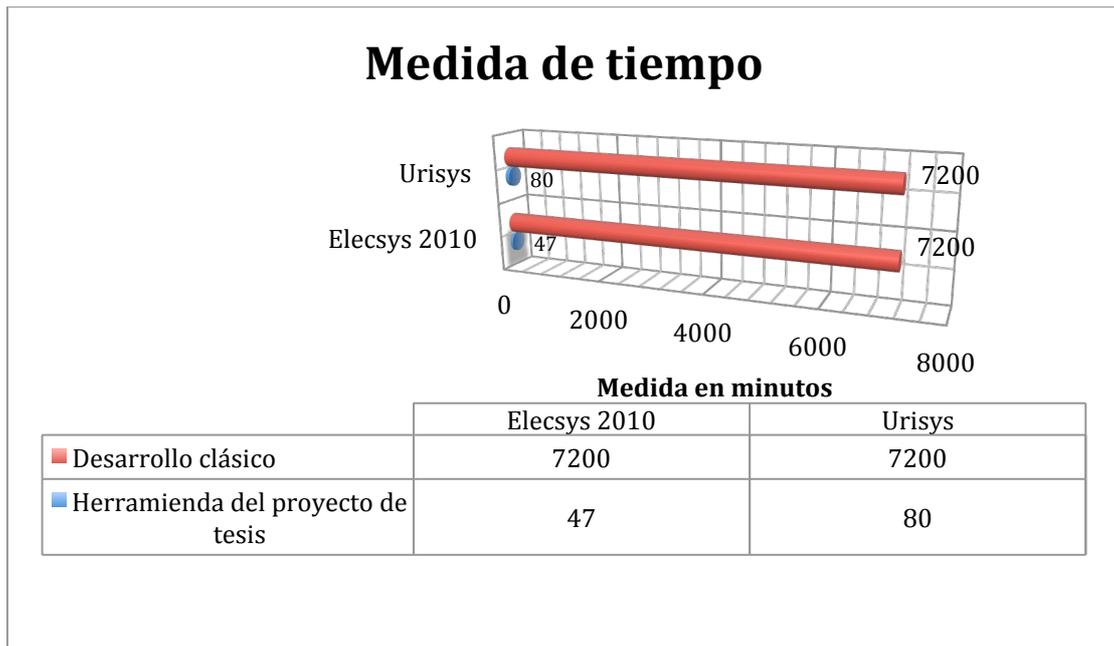


Ilustración 5-G Medida de tiempo

Para la segunda medida realizada, la línea de tiempo se encuentra dividida en minutos, para el caso de la herramienta de aprendizaje se tomo el tiempo promedio de las pruebas realizadas de manera exitosa en la medida anterior, en el caso del desarrollo humano, se encuentra basado en experiencias profesionales de diferentes desarrolladores.

5.3 Conclusiones sobre los resultados

La herramienta sobre el entorno de simulación ofrece mayor rango de fases de comunicación validadas, sin embargo, una vez que el sistema es utilizado y probado en el entorno de trabajo real, la situación cambia notablemente. No obstante, el sistema logró su cometido al realizar una comunicación exitosa entre el equipo clínico y el sistema de cómputo de manera automática.

Se logro una rango de aprendizaje alrededor del 50% para ambos equipos (no confundir con una comunicación a la mitad de exitosa), esto se traduce en que el sistema llegará a una comunicación correcta en la mitad de los casos donde no todos los rasgos necesarios para ese equipo se encuentran en la base de datos.

Las medidas realizadas indican que la herramienta tiene de manera general una buena tasa de aprendizaje y el tiempo de solución es extremadamente rápido.

5.3.1 Objetivos específicos y resultados

Objetivo:

“Proponer una herramienta única de software que permita reducir el tiempo de investigación y desarrollo de una interfaz de laboratorio de análisis clínico”

Resultado:

A diferencia de utilizar el simulador para reducir el tiempo de desarrollo, la herramienta presentada aquí demuestra que no solo se dedica a reducir tiempo, sino a realizar el trabajo de comunicación de manera automática, siendo así la primera herramienta en su género.

Objetivo:

“Generar un motor de aprendizaje de protocolos de comunicación apegado al estándar ASTM E1394-97 para el intercambio de datos de manera automática”.

Resultado:

El sistema demuestra que realiza el intercambio de datos de manera correcta y de manera independiente.

Objetivo:

“Crear una comunicación válida y exitosa independientemente de la versión, modelo, marca y/o fabricante de equipos clínicos”.

Resultado:

Los dos equipos clínicos tiene diferentes versiones del estándar de comunicación ASTM y modelos completamente diferentes logrando un intercambio de datos exitoso.

Objetivo:

“Eliminar en la medida de lo posible la necesidad de manuales técnicos para desarrollar la comunicación de los equipos clínicos”.

Resultado:

El sistema logró su cometido y sin la necesidad de utilizar manual técnicos en las pruebas y resultados.

CAPÍTULO 6. CONCLUSIONES Y TRABAJO A FUTURO

Aunque se ha logrado el objetivo general y los objetivos específicos del trabajo de investigación, se han aprendido diferentes lecciones durante el desarrollo y las pruebas, se pueden mencionar en general a continuación:

- Algunos de los equipos clínicos cuentan con rasgos definidos por el fabricante y que no pertenecen a ningún estándar en particular.
- La herramienta de simulación puede a ser poco fiable para realizar pruebas bastantes detalladas.

6.1 Sobre el sistema de aprendizaje

El sistema de aprendizaje en definitiva es un buen camino para ser utilizado en fases de producción empresarial y así programar de manera más rápida y eficiente las interfaces de laboratorio de análisis clínico. Generando en la todos de los casos un sistema de comunicación automática.

Sin embargo, el porcentaje de éxito y error podemos notar una tendencia de un promedio de 50% para cada uno (**cuando no existen todos los rasgos del equipo**), este resultado es algo alentador pero aún está lejos de las expectativas esperadas. Esto no es tan malo, ya que aproximadamente el sistema encuentra el protocolo correcto en la mitad de las veces. Cuando lo encuentra, el tiempo ahorrado es considerable, y vale la pena usarlo. Cuando no lo encuentra, en pocos minutos dice “no lo encontré”, y no se perdió casi nada de tiempo probando a ver si lo halla. Además, la persona que está desarrollando el protocolo puede empezar no desde cero, sino desde el protocolo incompleto.

El software fue diseñado con la idea de poder mejorar cada módulo del sistema de aprendizaje, esto hace que las mejoras y actualizaciones se hagan de manera más rápida y eficiente, notemos también que cualquier

modificación en cada una de ellas es independiente de las demás partes, creando un sistema escalable y fácil de ampliar.

6.2 Sobre el estándar ASTM y sus reglas

El manual técnico del estándar ASTM E1394-97 utilizado en el presente trabajo de investigación se encuentra en desuso, y fue reemplazado por una versión más reciente y con modificaciones sustanciales de los rasgos de comunicación, esto puede ser un factor importante en el porcentaje de éxito y fracaso del sistema de aprendizaje que puede inclinar la balanza a favor.

6.3 Limitaciones

Existe una gran limitación en los resultados del presente trabajo de investigación, puesto que los experimentos fueron realizados solamente en dos equipos clínicos auténticos, ya que por políticas de seguridad y administrativas del laboratorio, el acceso y uso de los equipos clínicos es negado al público en general, por lo tanto, los resultados pueden variar positivamente.

Existe también una seria limitación en cuanto la información proporcionada inicialmente al sistema de aprendizaje, si la base de datos al inicio cuenta con información y rasgos deficientes, el aprendizaje será pobre o completamente nulo.

El uso de un convertidor USB a RS-232 es un inconveniente que puede afectar considerablemente los resultados, los controladores del convertidor son inestables y el tipo de calidad del convertidor deja mucho que desear.

6.4 Trabajo a futuro

Aunque el sistema de aprendizaje tiene un buen promedio de éxito, hace falta llegar a la expectativa deseada para hacer un producto más viable y visible para las empresas.

Se plantea realizar mejoras en los motores y el sistema de aprendizaje en general.

Generar un simulador con las necesidades adecuadas para el correcto manejo de errores y errores más generados en los equipos clínicos. Por lo cual tengo el comportamiento exacto de un equipo en específico.

Proporcionar al público en general un herramienta de software con interfaz gráfica adecuada para el uso y administración del trabajo de tesis.

6.4.1 Limitaciones

Una de los más importantes puntos que no se realizaron el presente trabajo de investigación, es el de proporcionar una herramienta inmediatamente aplicable en la línea de producción de las empresas, solo se proporcionaron los métodos y técnicas de aprendizaje de protocolos de

comunicación para desarrollar una herramienta a gran escala y de uso inmediato.

Además, no se incluyeron algunos estándares de comunicación compatibles con ASTM. Ni de ninguna manera se creó una interfaz gráfica para el uso y administración de la herramienta de aprendizaje.

REFERENCIAS

- [1] American Section of the International Association for Testing Materials, 1997, Standard Specification for Transferring Information Between Clinical Instruments and Computer Systems.
- [2] Jeff Atwood, 2009, XML: The Angle Bracket Tax.
- [3] http://www.roche.com/about_roche/business_fields/about-diagnostics.htm
- [4] <http://www.hedonics.ch/EN/Default.aspx>
- [5] Sam Schultz, 1987, HL7 International.
- [6] Dennis M. Ritchie, 1993, The Development of the C Language.
- [7] Nicklaus Wirth, 1996, Compiler Construction.
- [8] Alfred V. Aho, Monica S. Lam, Ravi SEIT Jeffrey D. Ullman, 2008, Compiladores: Principios, técnicas y herramientas segunda edición Ed. Pearson Adison Wesley.
- [9] Roman Ivantsov, 2007, Irony .Net Language implementation kit.
- [10] Richard Stallman, 2010, Various Licenses and Comments about Them # Expat License
- [11] Manuel Alfonseca Moreno, Marina de la Cruz Echandía, Alfonso Ortega de la Puente, Estrella Pulido Cañabate, 2006. Compiladores e Intérpretes, Ed. Pearson- Prentice Hall.
- [12] John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman - Introduction to Automata Theory, Languages, and Computation (2nd Edition).
- [13] American Section of the International Association for Testing Materials, 1997, Standard Specification for Transferring Clinicial Observations Between Independent Computer System.
- [14] International Organization for Standardization, 1998, ISO/IEC 8859-1 8-bit single-byte coded graphic character sets, Part 1: Latin alphabet No. 1
- [15] International Organization for Standardization, 1986, Standard ECMA-94 8-Bit Single Byte Coded Graphic Character Sets - Latin Alphabets No. 1 to No. 4 (2nd Edition).

[16] American National Standard, 1997, Standard X3.30 Representation of Date for Information Interchange.

[17] American National Standard, 1986, Standard X3.43 Representation of Local Time of Day for Information Interchange.

[18] International Organization for Standardization, 1976, Standard ISO 2855:1976 Radioactive materials.

[19] ARC Electronics, 2010, RS232 Tutorial on Data Interfaces and Cables.

[20] Addison Wesley, las Sommerville, 2005, PrenticeHall (Sexta Edición).

ANEXO A.

EJEMPLO DE CÓDIGO FUENTE

```
static IdentifierTerminal IdentifierLiteral = new IdentifierTerminal("IdentifierLiteral",
"0123456789", "0123456789");

static IdentifierTerminal FreeLiteral = new IdentifierTerminal("FreeLiteral",
"^\s|!|\"@.#$%&'()*=?;:[]_-.<>+ 0123456789^", "^\s|!|\"@.#$%&'()*=?;:[]_-.<>+
0123456789/");

static NumberLiteral NumberLiteral = new NumberLiteral("NumberLiteral");

    static NumberLiteral IntLiteral = new NumberLiteral("IntLiteral",
NumberOptions.IntOnly);
    static CharList CharLiteral = new CharList();
    static IdentifierTerminal DateTimeLiteral = new IdentifierTerminal("DateTimeLiteral",
"0123456789", "0123456789");
    static IdentifierTerminal SenderNameLiteral = new
IdentifierTerminal("SenderNameLiteral", "0123456789.- /", "0123456789.- /");
    static NonTerminal Expr = new NonTerminal("Expr");
    static NonTerminal ConstantTransmition = new NonTerminal("ConstantTransmition");
    static NonTerminal FieldDelimiter = new NonTerminal("FieldDelimiter");
    static NonTerminal RepeatDelimiter = new NonTerminal("RepeatDelimiter");
    static NonTerminal ComponentDelimiter = new NonTerminal("ComponentDelimiter");
    static NonTerminal EscapeDelimiter = new NonTerminal("EscapeDelimiter");
    static NonTerminal Literal = new NonTerminal("Literal");
    static NonTerminal Identifier = new NonTerminal("Identifier");
    static NonTerminal Date = new NonTerminal("Date");
    static NonTerminal Free = new NonTerminal("Free");
    static NonTerminal SenderName = new NonTerminal("SenderName");
    static NonTerminal SenderNameList = new NonTerminal("SenderNameList");
    static NonTerminal Address = new NonTerminal("Address");
    static NonTerminal AddressFree = new NonTerminal("AddressFree");
    static NonTerminal AddressList = new NonTerminal("AddressList");
    static NonTerminal Processing = new NonTerminal("Processing");
    static NonTerminal Statement = new NonTerminal("Statement");
    static NonTerminal Program = new NonTerminal("Program",
typeof(StatementListNode));
    public HeaderGrammar(): base(false)
    {
        DateTimeLiteral.ValidateToken += new
EventHandler<ParsingEventArgs>(DateTimeLiteral_ValidateToken);
        #region reglas header
        Expr.Rule = "H" + FieldDelimiter + RepeatDelimiter + ComponentDelimiter +
EscapeDelimiter +
```

```

FieldDelimiter +
Identifier + //Control
FieldDelimiter +
Identifier + //Password
FieldDelimiter +
SenderNameList | //Sender Name
"H" + FieldDelimiter + RepeatDelimiter + ComponentDelimiter +
EscapeDelimiter +
FieldDelimiter +
Identifier + //Control
FieldDelimiter +
Identifier + //Password
FieldDelimiter +
SenderNameList + //Sender Name
FieldDelimiter +
AddressList + //Address
FieldDelimiter +
Empty + //Unused
FieldDelimiter +
Free + //Telephone
FieldDelimiter +
Free + //Characteristics
FieldDelimiter +
Free + //Receiver
FieldDelimiter +
Free + //Comment
FieldDelimiter +
Processing | //Processing
"H" + FieldDelimiter + RepeatDelimiter + ComponentDelimiter +
EscapeDelimiter +
FieldDelimiter +
Identifier + //Control
FieldDelimiter +
Identifier + //Password
FieldDelimiter +
SenderNameList + //Sender Name
FieldDelimiter +
AddressList + //Address
FieldDelimiter +
Empty + //Unused
FieldDelimiter +
Free + //Telephone
FieldDelimiter +
Free + //Characteristics
FieldDelimiter +
Free + //Receiver
FieldDelimiter +
Free + //Comment
FieldDelimiter +
Processing + //Processing
FieldDelimiter +
Free | //Version
"H" + FieldDelimiter + RepeatDelimiter + ComponentDelimiter +
EscapeDelimiter + //Header
FieldDelimiter +
Identifier + //Control
FieldDelimiter +
Identifier + //Password
FieldDelimiter +
SenderNameList + //Sender Name

```

```

FieldDelimiter +
AddressList + //Address
FieldDelimiter +
Empty + //Unused
FieldDelimiter +
Free + //Telephone
FieldDelimiter +
Free + //Characteristics
FieldDelimiter +
Free + //Receiver
FieldDelimiter +
Free + //Comment
FieldDelimite //Processing+FieldDelimiter +
Free + //Version+FieldDelimiter +Date;
#endregion
FieldDelimiter.Rule = "|";
RepeatDelimiter.Rule = "\\";
ComponentDelimiter.Rule = "^";
EscapeDelimiter.Rule = "&";
Identifier.Rule = IdentifierLiteral | Empty;
Literal.Rule = IdentifierLiteral | Empty | NumberLiteral;
Free.Rule = FreeLiteral | Empty;
Date.Rule = DateTimeLiteral | Empty;
SenderName.Rule = Empty |
    SenderNameLiteral |
    ComponentDelimiter + SenderNameLiteral |
    RepeatDelimiter + SenderNameLiteral |
    " ";
SenderNameList.Rule = MakePlusRule(SenderNameList, SenderName);
Processing.Rule = ToTerm("P") | "T" | "D" | "Q";
Address.Rule = Empty |
    FreeLiteral |
    " ";
AddressFree.Rule = MakePlusRule(AddressList, Address);
AddressList.Rule = Empty | AddressFree |
    AddressFree + ComponentDelimiter + AddressFree + ComponentDelimiter +
AddressFree + ComponentDelimiter + AddressFree + ComponentDelimiter + AddressFree |
    " ";
Statement.Rule = Expr | Empty;
Program.Rule = MakeStarRule(Program, Statement);
this.Root = Program;
this.LanguageFlags = LanguageFlags.CreateAst | LanguageFlags.CanRunSample;

```

Anexo A Ejemplo de código fuente del análisis léxico y sintáctico

ANEXO B. ORGANIZACIÓN DEL ESTÁNDAR ASTM E1394-97

```
<?xml version="1.0" encoding="UTF-8"?>
<E1394-97>
<HeaderRecord>
  <RecordTypeID>H</RecordTypeID>
  <DelimiterDefinition>^\&amp;</DelimiterDefinition>
  <MessageControlID>INTEGER</MessageControlID>
  <AccessPassword>STRING</AccessPassword>
  <SenderNameorID>
    <item1>STRING</item1>
    <item2>STRING^STRING</item2>
    <item3>STRING\STRING</item3>
  </SenderNameorID>
  <SenderStreetAddress>
    <item1>STRING</item1>
    <item2>STRING^STRING</item2>
  </SenderStreetAddress>
  <ReservedField></ReservedField>
  <SenderTelephoneNumber>STRING</SenderTelephoneNumber>
  <CharacteristicsOfSender>STRING</CharacteristicsOfSender>
  <ReceiverID>STRING</ReceiverID>
  <CommentOrSpecialInstructions>STRING</CommentOrSpecialInstructions>
  <ProcessingID>
    <item1>P</item1>
    <item2>T</item2>
    <item3>D</item3>
    <item4>Q</item4>
  </ProcessingID>
  <VersionNo>STRING</VersionNo>
  <DateAndTimeOfMessage>
    <item1>YYYYMMDD</item1>
    <item2>YYYYMMDDHHMMSS</item2>
  </DateAndTimeOfMessage>
</HeaderRecord>
<PatientInformationRecord>
  <RecordType>P</RecordType>
  <SequenceNumber>INTEGER</SequenceNumber>
  <PracticeAssignedPatientID>STRING</PracticeAssignedPatientID>
  <LaboratoryAssignedPatientID>STRING</LaboratoryAssignedPatientID>
```

```
<PatientIDNo3>STRING</PatientIDNo3>
<PatientName>
  <item1>STRING</item1>
  <item2>STRING^STRING</item2>
  <item3>STRING\STRING</item3>
</PatientName>
<MothersMaidenName>STRING</MothersMaidenName>
<Birthdate>
  <item1>YYYYMMDD</item1>
  <item2>YYYYMMDDHHMMSS</item2>
</Birthdate>
<PatientSex>
  <item1>M</item1>
  <item2>F</item2>
  <item3>U</item3>
</PatientSex>
<PatientRaceEthnicOrigin>
  <item1>W</item1>
  <item2>B</item2>
  <item3>O</item3>
  <item4>NA</item4>
  <item5>H</item5>
</PatientRaceEthnicOrigin>
<PatientAddress>
  <item1>STRING</item1>
  <item2>STRING^STRING</item2>
</PatientAddress>
<ReservedField></ReservedField>
<PatientTelephoneNumber>STRING</PatientTelephoneNumber>
<AttendingPhysician>
  <item1>STRING</item1>
  <item2>STRING^STRING</item2>
  <item3>STRING\STRING</item3>
</AttendingPhysician>
<SpecialField1>STRING</SpecialField1>
<SpecialField2>STRING</SpecialField2>
<PatientHeight>
  <item1>DOUBLE</item1>
  <item2>DOUBLE^STRING</item2>
</PatientHeight>
<PatientWeight>
  <item1>DOUBLE</item1>
  <item2>DOUBLE^STRING</item2>
</PatientWeight>
<PatientsKnowOrSuspectedDiagnosis>
  <item1>STRING</item1>
  <item2>STRING\STRING</item2>
</PatientsKnowOrSuspectedDiagnosis>
<PatientActiveMedications>STRING</PatientActiveMedications>
<PatientsDiet>STRING</PatientsDiet>
<PracticeFieldNo1>STRING</PracticeFieldNo1>
<PracticeFieldNo2>STRING</PracticeFieldNo2>
<AdmissionAndDischargeDates>
  <item1>YYYYMMDD</item1>
  <item2>YYYYMMDDHHMMSS</item2>
</AdmissionAndDischargeDates>
<AdmissionStatus>
  <item1>OP</item1>
  <item2>PA</item2>
  <item3>IP</item3>
```

```

    <item4>ER</item4>
  </AdmissionStatus>
  <Location>STRING</Location>

  <NatureOfAlternativeDiagnosticsCodeAndClassifiers>STRING</NatureOfAlternativeDiagnosticsCodeAndClassifiers>
  <AlternativeDiagnosticCodeAndClassification>
    <item1>STRING</item1>
    <item2>STRING^STRING</item2>
    <item3>STRING\STRING</item3>
  </AlternativeDiagnosticCodeAndClassification>
  <PatientReligion>STRING</PatientReligion>
  <MaritalStatus>
    <item1>M</item1>
    <item2>S</item2>
    <item3>D</item3>
    <item4>W</item4>
    <item5>A</item5>
  </MaritalStatus>
  <IsolationStatus>STRING</IsolationStatus>
  <Language>STRING</Language>
  <HospitalService>
    <item1>STRING</item1>
    <item2>STRING^STRING</item2>
    <item3>STRING\STRING</item3>
  </HospitalService>
  <HospitalInstitution>
    <item1>STRING</item1>
    <item2>STRING^STRING</item2>
    <item3>STRING\STRING</item3>
  </HospitalInstitution>
  <DosageCategory>STRING</DosageCategory>
</PatientInformationRecord>
<TestOrderRecord>
  <RecordTypeID>O</RecordTypeID>
  <SequenceNumber>INTE</SequenceNumber>
  <SpecimenID>
    <item1>STRING</item1>
    <item2>STRING^STRING</item2>
  </SpecimenID>
  <InstrumentSpecimenID>
    <item1>STRING</item1>
    <item2>STRING^STRING</item2>
  </InstrumentSpecimenID>
  <UniversalTestID></UniversalTestID>
  <Priority>
    <item1>CHAR</item1>
    <item2>CHAR\CHAR</item2>
  </Priority>
  <RequestedOrderedDateAndTime>
    <item1>YYYYMMDD</item1>
    <item2>YYYYMMDDHHMMSS</item2>
  </RequestedOrderedDateAndTime>
  <SpecimenCollectionDateAndTime>
    <item1>YYYYMMDD</item1>
    <item2>YYYYMMDDHHMMSS</item2>
  </SpecimenCollectionDateAndTime>
  <CollectionEndTime>
    <item1>YYYYMMDD</item1>
    <item2>YYYYMMDDHHMMSS</item2>

```

```

</CollectionEndTime>
<CollectionVolume>DOUBLE^STRING</CollectionVolume>
<CollectorID>STRING</CollectorID>
<ActionCode>STRING</ActionCode>
<DangerCode>STRING</DangerCode>
<RelevantClinicalInformation>STRING</RelevantClinicalInformation>
<DateTimeSpecimenReceived>
  <item1>YYYYMMDD</item1>
  <item2>YYYYMMDDHHMMSS</item2>
</DateTimeSpecimenReceived>
<SpecimenDescriptor>
  <item1>STRING</item1>
  <item2>STRING^STRING</item2>
</SpecimenDescriptor>
<OrderingPhysician>STRING</OrderingPhysician>
<PhysiciansTelephoneNumber>STRING</PhysiciansTelephoneNumber>
<UserFieldNo1>STRING</UserFieldNo1>
<UserFieldNo2>STRING</UserFieldNo2>
<LaboratoryFieldNo1>STRING</LaboratoryFieldNo1>
<LaboratoryFieldNo2>STRING</LaboratoryFieldNo2>
<DateTimeResultsReportedOrLastModified>
  <item1>YYYYMMDD</item1>
  <item2>YYYYMMDDHHMMSS</item2>
</DateTimeResultsReportedOrLastModified>

<InstrumentChargetoComputerSystem>STRING</InstrumentChargetoComputerSystem>
<InstrumentSectionID>STRING</InstrumentSectionID>
<ReportTypes>
  <item1>O</item1>
  <item2>C</item2>
  <item3>P</item3>
  <item4>F</item4>
  <item5>X</item5>
  <item6>I</item6>
  <item7>Y</item7>
  <item8>Z</item8>
  <item8>Q</item8>
</ReportTypes>
<ReservedField>STRING</ReservedField>

<LocationorWardofSpecimenCollection>STRING</LocationorWardofSpecimenCollection>
<NosocomialInfectionFlag>STRING</NosocomialInfectionFlag>
<SpecimenService>STRING</SpecimenService>
<SpecimenInstitution>STRING</SpecimenInstitution>
</TestOrderRecord>
<ResultRecord>
  <RecordTypeID>R</RecordTypeID>
  <SequenceNumber>INTEGER</SequenceNumber>

</ResultRecord>
<CommentRecord>
  <RecordType>C</RecordType>
  <SequenceNumber>INTEGER</SequenceNumber>
  <CommentSource>
    <item1>STRING</item1>
    <item2>STRING^STRING</item2>
  </CommentSource>
  <CommentText>
    <item1>G</item1>
    <item2>T</item2>

```

```
<item3>P</item3>
<item4>I</item4>
<item5>N</item5>
</CommentText>
</CommentRecord>
<MessageTerminatorRecord>
  <RecordTypeID>L</RecordTypeID>
  <SequenceNumber>INTEGER</SequenceNumber>
  <TerminationCode>
    <item1>N</item1>
    <item2>Nil</item2>
    <item3>T</item3>
    <item4>R</item4>
    <item5>E</item5>
    <item6>Q</item6>
    <item7>I</item7>
    <item8>F</item8>
  </TerminationCode>
</MessageTerminatorRecord>
<ScientificRecord>
  <RecordType>S</RecordType>
  <SequenceNumber>INTEGER</SequenceNumber>
  <AnalyticalMethod>STRING</AnalyticalMethod>
  <Instrumentation>STRING/STRING</Instrumentation>
  <Reagents>STRING</Reagents>
  <UnitsOfMeasure>STRING</UnitsOfMeasure>
  <QualityControl>STRING</QualityControl>
  <SpecimenDescriptor>STRING^STRING</SpecimenDescriptor>
  <ReservedField></ReservedField>
  <Container>STRING</Container>
  <SpecimenID>STRING</SpecimenID>
  <Analyte>DOUBLE</Analyte>
  <ResultUnits>STRING</ResultUnits>
  <CollectionDateAndTime>
    <item1>YYYYMMDD</item1>
    <item2>YYYYMMDDHHMMSS</item2>
  </CollectionDateAndTime>
  <ResultDateAndTime>
    <item1>YYYYMMDD</item1>
    <item2>YYYYMMDDHHMMSS</item2>
  </ResultDateAndTime>
  <AnalyticalPreprocessingSteps>STRING</AnalyticalPreprocessingSteps>
  <PatientDiagnosis>STRING</PatientDiagnosis>
  <PatientBirthdate>
    <item1>YYYYMMDD</item1>
    <item2>YYYYMMDDHHMMSS</item2>
  </PatientBirthdate>
  <PatientSex>
    <item1>M</item1>
    <item2>F</item2>
    <item3>U</item3>
  </PatientSex>
  <PatientRace>
    <item1>W</item1>
    <item2>B</item2>
    <item3>O</item3>
    <item4>NA</item4>
    <item5>H</item5>
  </PatientRace>
</ScientificRecord>
```

```
<ManufacturerInformationRecord>  
  <RecordTypeID>M</RecordTypeID>  
  <SequenceNumber></SequenceNumber>  
</ManufacturerInformationRecord>  
</E1394-97>
```

[Anexo B Organización del estándar ASTM E139497](#)